Special Features: Telegames And Nonviolent Games

COMPUTE!

$2.50
October
1983
Issue 41
Vol. 5, No. 10
£1.85 UK $3.25 Canada
63379
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

Two Gripping Games With Excellent Graphics For The Commodore 64, VIC-20, And Atari: *Moving Maze* And *Dragonmaster*

***Spiralizer*: A High-Res Drawing Utility For Atari, Apple, VIC-20, Radio Shack Color Computer And Commodore 64**

Merging Programs From Commodore Disks

Atari Multicolor Character Editor

A Sound Tutorial For The Commodore 64

10
0
74470 63379

BUY A BANANA™. SAVE A BUNCH.

Meet the Banana,™ the very tough, versatile, portable, and reliable dot-matrix printer from Gorilla.™

At $249.95 retail it's an ideal and inexpensive companion for personal computers like Apple® (or Apple "look alikes" such as Franklin® or Albert®), TI,® Commodore,® TRS-80,® Kaypro,® Timex,® Osborne,® etc.

After that, it's merely comparable to other printers that can cost up to three times as much: 10" carriages (to handle standard 9½" paper), 80 columns, graphics capability, 10 characters per inch (expandable to 5 cpi) draft-quality print (for perfectly acceptable form letters, data processing, business reports, etc.) tractor feed (for precise alignment and quick loading), parallel or serial interface (take your pick), self-inking ribbon cassette (for long life and easy installation), 10 portable pounds in weight, and compatibility with so many of the most popular personal computers on the market.

Plus its printhead features a unique single-hammer design that eliminates a lot of moving parts, to eliminate a lot of annoying repairs.

That's the Banana: silly name, serious service. It's everything the expensive dot-matrix printers are . . .

Except expensive.

Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021
Toll-free: 1-800-343-6833; In Massachusetts (617) 828-8150. Telex 951-624.

FOR YOUR COMMODORE WORDPROCESSING NEEDS
INVEST IN THE BEST

WORDPRO PLUS. IN A CLASS BY ITSELF.

When choosing a Word Processor for your Commodore™ computer, there's no reason to settle for anything but the best — in a word . . . WordPro™.

With over 30,000 happy clients churning out letters and documents all over the world, the WordPro Plus™ Series is unquestionably the #1 selling software package on Commodore computers! So when you choose WordPro, you know you're investing in a trial-tested program that's a real winner. And WordPro is NOW available for your Commodore 64™ computer — at prices starting as low as $89.95.

Designed for the user who has no computer or word processing experience whatsoever, WordPro Plus brings a new dimension to the term "user-friendly." More than just easy to use, WordPro will turn your Commodore computer into a sophisticated time saving word processing tool — loaded with the same inventory of features found in systems costing much, much more.

Our nationwide team of over 600 Professional Software/Commodore computer dealers will help you choose the WordPro Plus system that is best for your needs. Our full-service dealers have been set up to provide strong customer support. In addition to helping you choose the right system, many Professional Software dealers also offer WordPro Plus training and system installation.

Professional Software offers a complete spectrum of WordPro word processing software for Commodore computers ranging from the Commodore 64 to the more business oriented 8000/9000 series computers. And WordPro 4 Plus and 5 Plus also interact with our database management systems including InfoPro and The Administrator. So whatever your Word Processing needs, there's a WordPro system that's right for you.

WordPro™ and WordPro Plus™ are trademarks of Professional Software Inc.

The WordPro Plus Series was designed and written by Steve Punter of Pro Micro Software Ltd.

Commodore™ and the Commodore 64™ are trademarks of Commodore Electronics, Inc.

Dealer inquiries invited.

Invest in the best . . .
WordPro Plus. In a class by itself.

Call us today for the name of the WordPro Plus dealer nearest you.

Professional Software Inc.

51 Fremont Street
Needham, MA 02194

(617) 444-5224
Telex: 951579

SPINNAKER'S LINE OF EARLY LEARNING GAMES IS GROWING AS FAST AS YOUR CHILD'S MIND.

Watching your kids grow up is a lot of fun. But making sure their minds grow as fast as their bodies is even more rewarding. That's where we can help. With a growing line of Early Learning Programs that are not only lots of fun to play, but also educational.

Some of the games you see on these two pages help exercise your child's creativity. Others help improve vocabulary and spelling skills. While others improve your child's writing and reading abilities. And all of them help your child understand how to use the computer.

So if you're looking for computer programs that do more than just "babysit" for your kids, read on. You'll find that our Early Learning Programs are not only compatible with **Apple**,® **Atari**,® **IBM**® and **Commodore 64**™ computers, but also with kids who like to have fun.

FRACTION FEVER™ brings fractions into play. Ages 7 to Adult.

FRACTION FEVER is a fast-paced arcade game that challenges a child's understanding of fractions. As kids race across the screen in search of the assigned fraction, they're actually developing a basic understanding of what a fraction is and of relationships between fractions. They're even discovering that the same fraction may be written in a number of different ways.

All in all, FRACTION FEVER encourages kids to learn as much as they can about fractions – just for the fun of it!

MOSAIC™

RAMMASTER™ 32

EXPANDING
YOUR
EXPERIENCE

FOR VIC 20®

The Rammaster™32 introduces you to the world of more powerful computing. Now VIC 20® owners have one of the most advanced memory devices available. *It's what you'd expect from* **MOSAIC MEMORY**. *This is a full service memory device. Not only will the Rammaster™give you 7 times the computing RAM power (as much usable memory as the Commodore 64®) but also many advanced features.*

The ***expansion port*** *allows you to use your cartridges without unplugging the memory board, saving wear and tear on your computer. It's much more effective. With the flexibility the* ***Master Control switches*** *give you, you'll have no need for a mother board. The* ***Write Protect*** *switch protects your programs' subroutines and data from accidental interference from basic language and the automatic erase built into some cartridges. There's a* ***pause switch*** *built in to stop in the middle of a program, plus an* ***external port*** *is built in for a convenient optional cable pause/reset switch. You even have a switch to overlay a 3K block of RAM for expansion to 40K. The* ***Master Control*** *gives you complete control over the configuration of your VIC 20® Memory Map so it will never become obsolete.*

The Rammaster™ 32 is completely compatible with all the VIC 20® products and programs, and will be the most useful peripheral you'll buy for your VIC 20® computer. With **MOSAIC MEMORY** *expand your experience into more powerful computing.*

$129.95

- ***32K RAM***
- ***Built-in expansion port***
- ***Pause switch***
- ***Write protect***
- ***Master control for super flexible memory***
- ***Cartridge dissabler***
- ***Low cost***
- ***Gold edge connectors***
- ***Picture guided instructions***
- ***Compatible with VIC 20® software and cartridges***
- ***And more features***

Call your nearest **MOSAIC MEMORY** *dealer, 1-800-547-2807 (653-7885 in Oregon).*

RAMMASTER 16 *This is a lower cost version of the Rammaster 32™. It's a full service memory device just like the Rammaster 32™. With 16K less RAM. Expand up to 37K using the Mosaic Memory Standard 16 cartridge or other VIC 20 16K RAM cartridge. That will give you 37K RAM usable for programming or cassette/disk based programs and 21K for cartridge based programs.*

Standard 16 *This is a limited frills RAM board designed to give you a low cost way to expand with Mosaic Memory. This has no expansion port and limited memory flexibility. Use it with the RAMMASTER 16™ to expand to 37K. The Standard 16 offers Mosaic Memory's exceptional reliability and performance at a very low cost.*

MOSAIC™
ELECTRONICS, INC.
P.O. Box 708 • Oregon City, Oregon 97045

October 1983 Vol. 5, No. 10

NOTE: See page 50 before typing in programs

AP Apple **AT** Atari, **P** PET/CBM, **V** VIC-20, **C** Radio Shack Color Computer, **64** Commodore 64, **TS** Timex/Sinclair, **TI** Texas Instruments. *All or several of the above.

COMPUTE! Publications, Inc. abc
A Subsidiary Of American Broadcasting Companies, Inc.

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published 12 times each year by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 505 Edwardia Drive, Greensboro, NC 27409. Domestic Subscriptions: 12 issues, $20.00. Send subscription orders or change of address (P.O. form 3579) to Circulation Dept., **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X.

EDITOR'S NOTES

Our thanks to those of you who recently participated in our subscriber survey. Initial results are being reported now, and we'll share some of those with you. One of the number sets of most interest to me was spouse readership. This industry has quite typically been reflected in readership studies as a predominantly male audience. While this is accurate, we have maintained that, as a "family" oriented magazine, we have a high degree of family/spouse participation in computing and COMPUTE!. We suspected that, in addition to the seven or eight percent female readership that would show up in response to a normal question, analysis of family utilization would show far more general involvement. We're pleased to report that this is, in fact, the case. In COMPUTE! families, we discovered that 29 percent of spouses regularly read COMPUTE!. Further evidence that our industry is turning into a truly broad-based consumer computing marketplace.

Here are some of the other results of the survey that we think you'll find of interest:

Age	%
under 28	25.7
29-35	26.9
36-43	26.6
over 44	20.8

How do you use your personal computer? (Check all that apply)

You	Spouse	Children	
85.6%	61.6%	75.4%	Recreation and Games
42.2	22.7	3.6	Education for Adults
16.0	8.5	59.7	Education for Children
60.6	22.3	1.5	Personal Business
60.8	24.7	5.4	Other Home Applications
27.4	7.4	0.5	Company Business

And finally, 97.5 percent of you own one or more personal computers. Our survey showed the true depth and breadth of our readership. For example, note the variety of uses in the chart. It's obvious that home computing is far more than simply playing games.

Random Bits:
Given the recent acceleration of rumors, we'll be quite surprised if IBM hasn't announced their new home computer by the time you're reading this .... While avoiding the various issues of rights/wrongs, Kathy Yakal's article on nonviolent gaming in this issue presents a variety of interesting thoughts on the subject of violence and nonviolence in gaming. It's a good article and shows a bit of the depth of thought that people in our industry are putting into one aspect of our collective future. As always, we're interested in your feedback on the topic.

Next month, among other things, we'll have an article sharing in more detail the results of the subscriber survey.

Robert Lock

Robert Lock, Editor In Chief

Important Notice

We anticipate that by early spring we may begin to make our subscriber list available to carefully selected marketeers. If you wish to have your name *withheld* from any non-COMPUTE! mailings by vendors, please send a note, with your mailing label attached, to:

COMPUTE! Customer Service
P.O. Box 5406
Greensboro, NC 27403
Attn: Mailing List

Please remember to indicate in your note that you *do not* wish your name included in any non-COMPUTE! mailings. We will be happy to keep your name off the list, and will continue to respect your privacy.

T&F SOFTWARE

Two Hot New Action Games!

FOR COMMODORE 64

Speed Racer

$29.95

Feel like a devil today? Better adjust your halo. Start the ignition. It's up to you to decide... Okay, hit the accelerator. You can be the ultimate speed demon or put wings on your wheels! There's a pedestrian now...will you go for speed, no matter what pedestrians are in the way (for the devilish points)— or be an angel in a challenging obstacle course? The heavenly points may get you to heaven, but the devilish points may get you to...well, better make your decision— the time clock is running!

Candy Bandit

$29.95

So your sweet tooth has gotten out of hand again? Well, this time the sheriff is after you and he's no sweetheart. There he is now! Quick! Grab all the candy in sight and dive for the doorway! Don't look now, the doors are rotating...better be good at getting out of sticky situations, 'cause if you hit the wall you're stuck with it! There's plenty of candy here and many more levels to cross, so get on the move, sugar!

Available at finer Software Stores everywhere.

OR CALL (213) 501-5845 FOR THE NAME OF YOUR LOCAL DEALER OR DISTRIBUTOR.

T&F SOFTWARE

T&F Software Company 10902 Riverside Drive / North Hollywood, California 91602. (213) **501-5845**

Coming In November

Special Telecommunications And Peripherals Issue Including:

How To Get Started In Telecommunications

Buyer's Guide To Modems

Programs For Your Commodore Modems

Alphablast

COMPUTE! Publications, Inc. publishes:

COMPUTE! The Journal for Progressive Computing

COMPUTE! Books

Telephone: 919-275-9809

Corporate office:
505 Edwardia Drive,
Greensboro, NC 27409 USA

Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA

Subscription Information

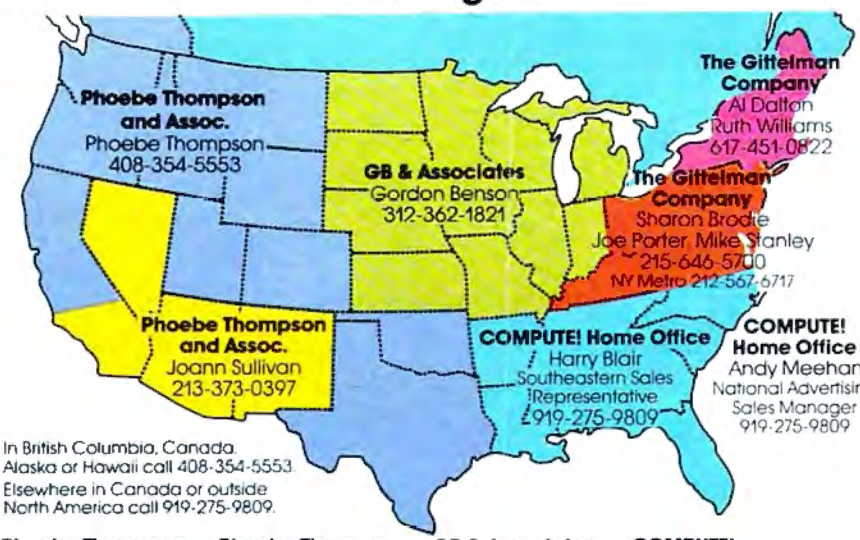
COMPUTE! Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403

TOLL FREE Subscription Order Line
800-334-0868
In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.) $20	Air	
	(two yrs.) $36	Europe, Australia	$38
	(three yrs.) $54	Middle East, Central America and North Africa	$48
Canada and Foreign Surface Mail	$25	South America, South Africa, Far East	$68

Advertising Sales



In British Columbia, Canada, Alaska or Hawaii call 408-354-5553.
Elsewhere in Canada or outside North America call 919-275-9809.

Phoebe Thompson and Associates
101 Church Street
Suite 13
Los Gatos, CA 95030
PHOEBE THOMPSON

Phoebe Thompson and Associates
2556 Via Tejon
Palos Verdes Estates, CA 90274
JOANN SULLIVAN

GB & Associates
P.O. Box 335
Libertyville, IL 60048
GORDON BENSON

COMPUTE! Home Office
505 Edwardia Drive
Greensboro, NC 27409
HARRY BLAIR
Southeastern Sales Representative

The Gittelman Company
Statler Office Building
Suite 518
20 Providence Street
Boston, MA 02116
AL DALTON
RUTH WILLIAMS

The Gittelman Company
Summit Office Center
7266 Summit Avenue
Fort Washington, PA 19034
SHARON BRODIE
JOE PORTER
MIKE STANLEY

Address all advertising materials to:
Patti Williams
Advertising Production Coordinator
COMPUTE! Magazine
505 Edwardia Drive
Greensboro, NC 27409

WordPro 3 Plus/64™

The Best Word Processor for your Commodore 64™

When choosing a Word Processor for your Commodore 64 computer, there's no reason to settle for anything but the best—in a word. . . .WordPro™.

With over 30,000 happy WordPro clients churning out letters and documents all over the world, the WordPro Plus™ Series is unquestionably the #1 selling software package on Commodore computers! So when you choose WordPro, you know you're investing in a trial-tested program that's a real winner. And WordPro 3 Plus/64 is NOW available for your Commodore 64 computer at prices as low as $89.95.

Designed for the novice user with no computer or word processing experience whatsoever, WordPro 3 Plus/64 brings a whole new dimension to the term "user-friendly." More than just easy to use, WordPro 3 Plus/64 is a commercial level word processor loaded with powerful features including: Auto Page Numbering, Headers and Footers, Math Functions, Global Search and Replace, the Ability to Create Multiple Personalized Letters and Documents, Underlining, Boldface, Super and Subscripts and much, much, more.

Best of all, WordPro 3 Plus/64's powerful arsenal of features can be put to use almost immediately—by even the novice user. So whether you're a writer, in business, education, or just a hobbyist, you'll quickly become a WordPro Pro!

Invest in the best. . . WordPro Plus. In a class by itself.

Call us today for the name of the WordPro 3 Plus/64 dealer nearest you.

Professional Software Inc.

51 Fremont Street
Needham, MA 02194

(617) 444-5224
Telex: 951579

WordPro 3 Plus/64™ is a trademark of Professional Software Inc.
The WordPro Plus Series was designed and written by Steve Punter of Pro-Micro Software Ltd.
Some printers may not support certain WordPro 3 Plus/64 functions and/or require an interface. Please check with your dealer.
Commodore 64™ is a trademark of Commodore Electronics Ltd.
Dealer and Distributor inquiries are invited.

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

Using An Electric Typewriter As A Printer

After reading an article in the June COMPUTE!, *"How to Buy the Right Printer," several readers were intrigued by the possibility of using an electric typewriter as a printer for their computer. Although we cautioned that an electric typewriter may not be fast or durable enough to tolerate continuous typing, there are several interfaces on the market which let you connect your computer to an electric typewriter.*

Some modern daisywheel typewriters have an RS-232 serial port, so (assuming your computer can talk to RS-232 devices) if you have the proper cable and software you can attach your computer directly. Other companies manufacture overlays that "press" the keys for you, or have you open up the typewriter to install solenoids which activate levers inside the typewriter. Still another company claims to have a device that converts "computer language" to "typewriter language," but only for certain models. With the price of even letter-quality printers dropping below $400, however, you will have to consider whether or not you want to risk modification of your typewriter which could void its warranty or service contract.

Storing Scriptor And Video 80 On The Atari

In your April issue, you published two interesting Atari programs, "Scriptor" and "Video 80." Here are a few questions. How many pages can you store in a 48K Atari 400 when using Scriptor with 8K BASIC? What is the memory required for Video 80? Can Scriptor and Video 80 be merged, and, if so, what changes would have to be made?

On another subject, how would I "hook up" an Epson MX-80 series printer to my 48K Atari 400 with or without the Atari 850 interface?

Ed Hallinan

Scriptor adapts itself to either 24K, 32K, or 48K and will display the number of lines free when you first run it. Each line is 38 characters. Since a printed page (double-spaced) takes about 30 75-column lines, just divide the "lines free" by 15 for a rough estimate.

Video 80 requires about 2K for the driver routine and another 8K for the high-resolution GRAPHICS 8 screen. Due to this, there is not enough memory left over in a 40K or 48K to let you store the programs and text.

You can attach almost any Centronics parallel or RS-232C serial printer to the Atari via the Atari 850 Interface Device. The new Atari 1025 80-column printer does not require the 850, however.

In our July 1983 Issue, we inadvertently used the name "Castle Quest." The publication of this article is in no way intended to cause confusion with the commercially available product called "Castle Quest" by Michael S. Holtzman and Timothy Baldwin. In the future, we will refer to it as "Castle Search."

Fine Tuning The VIC's Audio And Video

For those who might be having trouble with the picture and sound on their VIC-20, I have a solution.

It is essential that the sound and picture be adjusted properly in the RF modulator so that they are synchronized on either Channel 3 or 4. The computer is sold with the audio and video signals adjusted for one particular TV model. This may not work well on yours.

To make adjustments for your TV, carefully follow these steps:

1. Turn the computer off and the TV on.

2. Turn off the AFT (Automatic Fine Tuning) switch on the TV (if your set has one).

3. Adjust the fine tuning knob on the TV to the middle range of that channel. Don't worry if the picture is partly fading or the sound is distorted at this point.

4. Open the RF modulator very slowly and carefully. *Be sure your power is off.* Remove the casing screw and slowly lift the cover.

5. Turn on your power for the computer.

6. There are two holes which have screws in the metal casing. The one closest to the channel selector switch is the video signal. The other is for audio (see figure). Using *only* a small nonmetallic screwdriver, adjust the video signal until the TV

Presenting the class of 64.

The Commodore 64™ is one of the most exciting home computers in memory.

But memory isn't the only thing that's exciting about the 64.

Because Tronix is here.

Class act.

The people who have been bringing out the best in the VIC 20™ (and Atari,® too) have graduated to the Commodore 64.

Which means that now you can enjoy fast action, complex strategies, interesting characters, superior sound effects and challenging, play patterns.

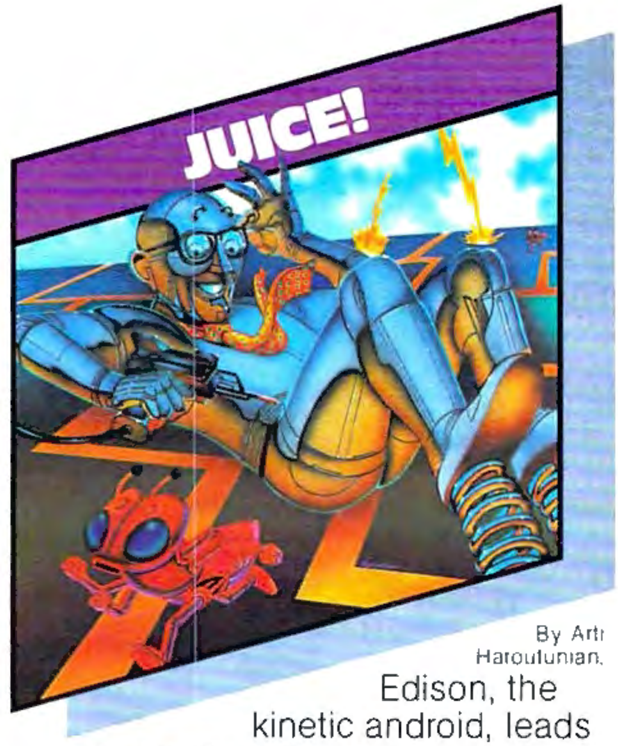
Just like VIC 20 and Atari owners. Only faster, more complex, and more challenging, too.

More memorable, in other words.

In a class by ourselves.

Of course, if you'd rather not take our word for it, you don't have to. The experts at Electronic Games have called *Kid Grid* for Atari "one of the most compulsive, utterly addictive contests in the world of computer gaming."

They haven't seen anything yet.



By Arti Haroutunian.

Edison, the kinetic android, leads a frustrating life.

All he wants to do is build his circuit boards and go with the flow. But things keep getting in the way.

Nohms—a negative influence—bug him constantly. Flash, the lightning dolt, disconnects everything in his path.

And the cunning Killerwatt is out to fry poor Edison's brains.

You'll get a charge out of this one. And a few jolts, too!

(Suggested retail $34.95)

By Arti Haroutunian.

Connecting the dots on our colorful grid should be easy, right?

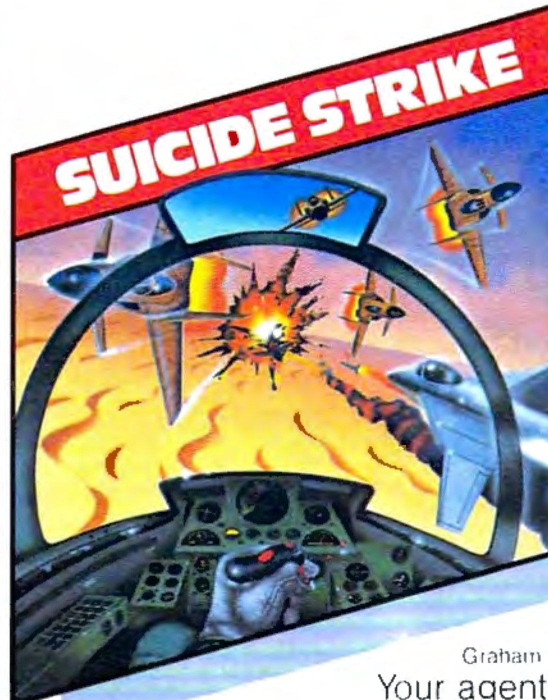
Wrong. Because the bullies are in hot pursuit!

Squashface, Thuggy, Muggy and Moose are their names. And *you* are their game. And what's more, they're faster than you are.

But you're smarter. And you control the stun button.

So keep your eyes peeled for the mysterious question mark and don't slow down at corners!

(Suggested retail: $34.95)



By Graham McKay.

Your agents risked their lives to find the enemy's secret headquarters.

Now you're risking yours to destroy it.

And they know you're coming.

As you fly over water and across hundreds of miles of unfriendly territory, the action is thick, fast and three-dimensional.

Fighter aircraft. Surface-to-air missiles. Helicopter gunships. The attacks come from every direction.

Even from behind.

(Suggested retail: $34.95)

TRONIX™

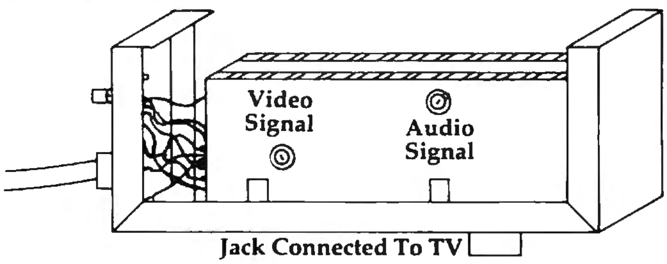
8295 South La Cienega Blvd., Inglewood, CA 90301 picture is crisp and sharp. This acts like another fine tuning knob. Check the other channel and make sure it is in the middle of the RF signal range by adjusting the fine tuning knob on your TV.

7. Adjust your TV volume to about one-half. Now adjust the audio signal level in the RF modulator until a quiet, clear sound is heard on the TV.

8. Make sure that, on the other channel, both audio and video signals are perfectly adjusted.

9. Turn the computer off and replace the cover on the RF modulator.

RF Modulator



Mark Kormendy

We tried your suggestions and they work very well indeed. However, we do not advocate hardware modifications except when performed by qualified, experienced technicians. Readers should be extremely careful when making internal adjustments on any electrical component. Also, tampering with the RF modulator will void your warranty. (Later versions of the VIC RF modulator may not have the adjustment screws.)

We tested this also on the internal RF modulator in the 64 but found no appreciable difference in screen or audio quality. If we hear of anything significant on this, we'll publish it in a future issue.

TRS-80 Equipment Needed

I represent a nonprofit, charitable organization which uses TRS-80 Model I equipment in virtually every aspect of its affairs. We would be grateful if your readers would consider contributing additional Model I equipment: keyboards, expansion interfaces, drives, and printers would be welcome. All contributions would be fully tax-deductible, since we hold "public charity" status with the IRS.

If you are in a position to make such a contribution or would like more information, please drop me a note or call me (collect, if you like) at (617) 495-9020.

Dr. Robert Epstein
Cambridge Center for Behavioral Studies
11 Ware Street
Cambridge, MA 02138

PET To VIC Or 64 Transfers

In connection with your article on merges in the June 1982 issue of COMPUTE!, I stumbled onto something which I have never seen reported, although something so simple must have been noticed by someone else. I tried loading programs recorded on a cassette on my PET (Original ROM from 1977) into my new Commodore 64. I found that the program appeared to load and would apparently list OK. However, it would not RUN, and one could not list individual lines or groups of lines. Also, on closer examination of the listing, I found that the first line of the original program was missing, and in its place was a 4 or 5 digit number.

I found by experimenting that I could convert the program as loaded into a normal program by a very simple procedure. I started a LIST, but stopped it (with the STOP key) while the first line was still on the screen. I then positioned the cursor first over the number which had replaced the first program line, then executed a RETURN, and then a second RETURN when the cursor was over the second line (which now appeared to be the first). After that, the program could be edited and RUN normally (after replacing the first line, if necessary). I have now used this procedure to transfer a number of programs from my 1977 PET to the 64, but put a line like:

```
10 REM JUNK LINE
```

at the beginning of each program before recording it on the PET. You and others are probably familiar with this, but I have not seen it anywhere. It is a very simple way to transfer programs from one system to the other. I presume this would also work on the VIC-20, and I intend to try this as I also do some programming on a VIC-20 at a school for handicapped children.

Myron T. Kelley

Jim Butterfield replies:

Your letter draws attention to something that is not well known. The very first Commodore computers – the "Original ROM" PETs – saved programs in a slightly different way from all subsequent Commodore machines.

It worked this way: just before the beginning of your BASIC program there is always a value of zero in memory. This is still true, even in VICs and Commodore 64s. But the first model of PET saved programs including the zero. All subsequent Commodore machines saved only the program – not the zero.

This meant that early PETs saved programs with a Load address of 1024; and that all subsequent PET/CBMs saved with a Load address of 1025. On the PET/CBM computers, this made little difference: programs did not relocate, and were loaded back to the right place. With the arrival of the VIC and the Commodore 64, however, programs became relocatable; and these new machines did not expect the extra zero. Result: a confused first line.

THE BEST REASON FOR HAVING A HOME COMPUTER.

Your children. That's why we created the Early Games series for them. We're educators as well as computer specialists. We create games that teach children important skills.

There are five programs in the Early Games series. **Early Games for Young Children** is a set of nine entertaining activities for children 2½ to 6. They can work with numbers and letters and create colorful pictures. **Matchmaker** uses shapes, sizes, directions and colors to help children develop reading readiness skills. Children ages 5 to 12 can learn to play melodies with Early Games **Music.** Our **Piece of Cake** turns math problems into, well, a piece of cake. And **Fraction Factory** takes the work out of fractions.

Early Games feature multiple activities, easy to use picture menus, and colorful graphics. The games are fun, children love to play them! That's why they learn from them.

And they're the best reason for having a home computer.

CSI

counterpoint software inc.

The easiest way to fix up this problem is to LOAD them to any newer PET/CBM and then SAVE them once again. The zero will be dropped and everything will become compatible.

Timex/Sinclair Compatibility Problem

I own a Sinclair ZX81 computer and a Memopak 32K RAM. I recently purchased a Timex/Sinclair 2040 printer, and I am unable to get the computer to work with the Memopak and the printer attached.

The Memopak works fine with the computer alone or with the Sinclair 16K RAM, and the printer works with the computer and with the Sinclair 16K RAM. However, the computer will not display its cursor when the Memopak 32K RAM and printer are both attached.

Can you tell me what's wrong?

Paul R. Harrison

The Timex/Sinclair 2040 printer is not compatible with Memopak for some unknown reason. Memotech, manufacturer of Memopak, is investigating. Contact them directly for further information. As you have noted, the Memopak works very well with the ZX81 – and with the proper interface, works well with a wide range of printers.

Apple Programs For Commodore

I would like to know if Apple programs can be loaded into the Commodore 64 from disk or tape and vice versa?

Anthony Tutter

We have published two articles on loading Apple program tapes into the PET and vice versa. The same procedures which work on the PET should also work on the Commodore 64. These articles are "Feed Your PET Some Applesoft" in COMPUTE!'s First Book of PET/CBM *and "Load Commodore BASIC Program Tapes Into The Apple II" in* COMPUTE! *(April 1983).*

Some commercial software is available for loading 64 programs from disk into the Apple and vice versa.

PET Monitor Printout

I am an avid user of the PET machine language monitor, but I have never found a way to direct a hexadecimal dump to a printer. Is it possible to do so or am I wasting my time? Please help!

Michael Silano

For 4.0 BASIC PETs, you can print the output by typing X to exit to BASIC, then type:

```
OPEN 4,4: CMD 4: SYS 54386
```

then type the monitor command M. For example, if you wanted a printout of the hex numbers between 0400 and 0420, you would type:

```
.M 0400 0420
```

For Upgrade PETs, you can use the above, but just SYS to 4 (the way you'd ordinarily enter the monitor).

However, the easiest way to control the printer (and many other aspects of machine language programming on the PET) is a "monitor extension" program, "Micromon," published in COMPUTE! *(January 1982). It's an excellent tool for machine language programming. It's available for $3 from* COMPUTE!*'s Back Issues Department.*

Conserving Your Computer's Power

I have a VIC-20, and I want to know which is better for the computer – to leave it on constantly or shut it off when you are done using it?

Bob Weber

This is debatable. Some would say that the initial power surge when turning on a computer is actually more damaging to the electronic components than leaving it on continuously. On the other hand, most electronic parts have a definite life span, and leaving the computer on 24 hours a day could shorten the overall useful life of the computer.

Commodore has recently changed the power supplies of both the VIC-20 and the Commodore 64. Some of the new power supplies (especially those with the 64) seem to run somewhat hotter than previous production models. This tips the balance, at least in the case of these machines, so we recommend that you turn off a VIC or 64 and even unplug the power supply after each use.

A New Atari Graphics Mode?

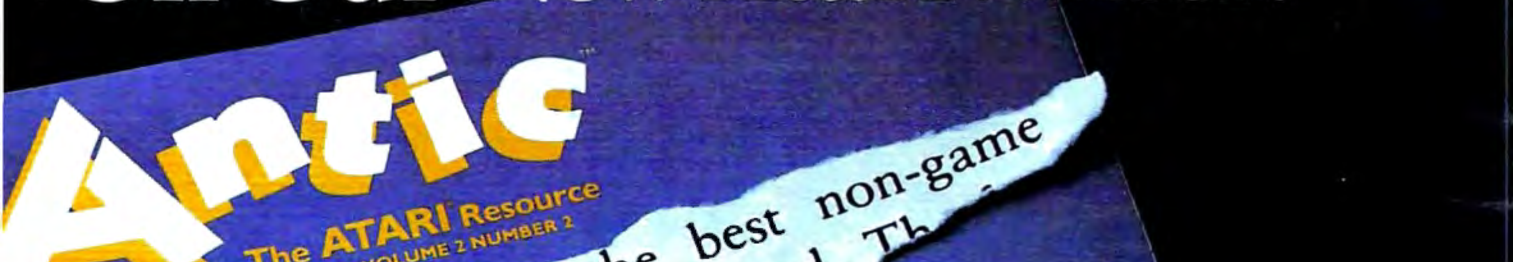
I have been programming with my Atari 400 for almost a year now and have had a lot of fun with it. And recently, I have noticed advertisements in magazines about programs written in graphics mode 7.5. Is this a new graphics mode? If so, what does it and the other ".5" modes have over the others or GTIA?

David Brundage

There are no ".5" modes. GRAPHICS 7 is a 4-color mode with a horizontal resolution of 160 and a vertical resolution of 96. GRAPHICS 8 is the high-resolution, one-color (except through artifacting) mode with a resolution of 320h x 192v.

Built into the ANTIC chip is another mode that is not supported by the 400/800 operating system. It is a four-color mode with a resolution of 159h x 192v, the same vertical resolution as GRAPHICS 8. Since it is "halfway" between modes 7 and 8, it has been called GRAPHICS 7½ or GRAPHICS 7.5. You can turn a GRAPHICS 8 screen into this new mode by replacing

Have You Read The Reviews On Our New AtariWriter™?



Atari's New Word Processor.

Try it, be your own critic. Use new AtariWriter for all your family correspondence, school papers, committee reports, mailings.

Our advanced ROM-based cartridge technology means you can use it on any ATARI Home Computer (even 16K). It also lets you choose between cassette and disk storage systems.

Easy to learn and to use. "Yet," says ANALOG Magazine, "as your uses for word processing grow, (you'll employ) its more advanced features."

Menu driven, with helpful prompts.

One special feature: you can correct as you write, instantly, without switching back and forth between Create and Edit modes.

You can move entire paragraphs. Search and replace. Even restore text you've just deleted, with our memory buffer.

You can change format instantly, and preview on-screen at full printer width. Print extra copies for your records.

As ANTIC Magazine concluded, new AtariWriter is "clearly superior on price and performance to Text Wizard and the ATARI Word Processor."

At Atari dealers.

ATARI® HOME COMPUTERS

A Warner Communications Company.

all the 15's and 79's in the display list with 14's and 78's. (The hexadecimal code for the number 14, which is this mode's ANTIC number, is E, so GRAPHICS 7½ is sometimes called GRAPHICS E.) This will do the trick:

```
10 GRAPHICS 8+16:DLIST=PEEK(560)+256
   *PEEK(561)+4
20 FOR I=-1TO200:A=PEEK(DLIST+I):IF
   A=15 OR A=79 THEN POKE DLIST+I,A-
   1
30 NEXT I:POKE 87,7
```

If you want a text window, just remove the "+16" from line 10. The POKE on line 30 fools the operating system into thinking that you are in GRAPHICS 7. This lets you use only the top half of the screen with BASIC PLOTs and DRAWTOs, since the OS considers vertical numbers greater than 95 to be in error for GRAPHICS 7. The Atari 1200XL and all the new 600XL, 800XL, 1400XL, and 1450XL computers fully support this mode in the operating system and BASIC.

Sprites, SYS, And Storage On The 64

I have several questions about the Commodore 64:

First of all, I know you can check if a sprite has hit a background character by PEEKing (V+31) when V=53248. Is there any way to find out what kind of character it hit?

Second, I have seen many programs which read: 10 SYS(X). How do they do this? Using Commodore's assembler package, you must load in the machine language program off the disk.

Last, where is there room to insert machine language programs in memory other than C000 - CFFF (49152 - 53247)?

Matthew Price

The sprite-to-background collision detection byte (53279) is set up to be a "toggle" switch; in other words, it registers an off/on condition (collision or no collision). It does not offer the option of telling you what character the sprite collided with. However, you can program this option for yourself. For instance, after a collision has been detected you could branch to a subroutine that would convert the affected sprite's position to its screen position, and then perform a simple PEEK to see what character it hit.

The SYS command is used to start a machine language program running. When, in a BASIC program, you see a statement such as 10 SYS XXXX, the program is simply branching to a machine language routine much the same way that a pure BASIC program would branch to a BASIC subroutine via the GOTO or GOSUB command.

Commodore's machine language assembler is stored on disk. Like BASIC, machine language programs may be stored on either tape or disk and, once LOADed (by the LOAD command, or via a BASIC loader), it may be called or branched to at any time by the SYS command.

There are many places for you to safely place your machine language programs. Besides the obvious BASIC programming area (2048 to 40959), there are other good locations available: $02A7 to $02FF (decimal 679 to 767, 89 bytes), $033C to $03FB (decimal 828 to 1019 – the tape cassette buffer – 192 bytes), and, as you mentioned, $C000 to $CFFF (decimal 49152 to 53247, 4096 bytes).

Can An Atari VCS Run 400/800 Programs?

I just heard that a keyboard will be coming out for the Atari 2600. Are programs from Atari 400/800 compatible with this sytem? Do they have the same programming language, *not* software?

Frank Martone

Atari's new add-on computer keyboard, previously called "My First Computer," has been rechristened (with a few keyboard changes) as "The Graduate." The add-on module comes with a version of BASIC that is more like Microsoft BASIC than Atari BASIC. The Graduate was designed independently of the home computer line, so although some of the graphics are similar, The Graduate has no more in common with the 400/800 than it has with a Commodore VIC-20.

Atari Machine Language Graphics

I own an Atari 400 and would like to know how to change graphics modes in machine language. Also, I'd like to know how to change the background colors in the GTIA modes.

Bim Feysteryga

All the graphics routines you find in BASIC (GRAPHICS, PLOT, COLOR, DRAWTO, etc.) can be accessed from the operating system. Refer to Bill Wilkinson's "Insight: Atari" in the February 1982 issue. For complete information on the GTIA, you will find a series of illuminating articles in COMPUTE!'s First Book of Atari Graphics.

*The background color in GTIA modes 9 and 11 is, as usual, in color register four (SETCOLOR 4, POKE 712). In mode 9 you can only control the color, not the luminance, and the reverse is true for mode 11. The background color in GTIA mode GRAPHICS 10 is controlled by memory location 704. POKE it with the color multiplied times 16 and then add the luminance. Instead of SETCOLOR n,4,6 use POKE 704,4*16+6.*

COMPUTE! welcomes questions, comments, or solutions to issues raised in this column. Write to: Readers' Feedback, COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403. COMPUTE! reserves the right to edit or abridge published letters.

Computers And Society

David D Thornburg, Associate Editor

That's Not A Program, That's A Language

In the wild and woolly early days of personal computing (the late 1970s, to be exact), one of my favorite sayings was "That's not a game, that's a simulation!" Implicit in this saying was the idea that games were somehow not worth spending time on, but that the same piece of software took on extra value when viewed as a simulation.

A program to emulate a lemonade stand (in which the player experimented with advertising, pricing, inventory, etc.) could be viewed as a game (see how much money you can make in the shortest time) as well as a simulation (examine the effect of advertising on sales). But as time passed, the game/simulation controversy seemed to die away.

I have seen some recent developments that point the way to new types of software. I am speaking of application programs that are actually computer languages. This "language aspect" is sometimes so carefully hidden that it is likely the programmers themselves may not realize all the implications of their efforts.

This and next month's columns are devoted to a description of a few such "languages." Those of you who think computer languages are limited to BASIC, PROLOG, Pascal, PILOT, Logo, and the like are in for a surprise. I am presenting only a very small sampling of these languages, and by the time we are done you will be able to identify many others.

Before giving examples, it might be a good idea to list the essential elements of a language. A computer language must allow the user to create computer-based activities that are custom-tailored to the user's needs. The language must have a vocabulary and a grammar. The user should be able to edit his or her program and to "run" it. All the languages I will describe interpret the user's programs and are highly interactive both in their programming style and in the types of programs that are created.

VisiCalc As A Language

VisiCalc (a product of VisiCorp) is an example of what is called an "electronic spread sheet." This program allows the user to create tables of data and relationships between certain pieces of data. As the data is entered into the spread sheet, calculations are made automatically to fill in the areas of the sheet containing the "results." The ability to play "What if?" games by quickly seeing the effect of changes in the data has made VisiCalc and its kin invaluable tools for today's computer-using executive.

Electronic spread sheet programs are computer languages.

If you find this surprising, look at how they are used. The user creates a "mask" for the spread sheet (the program). This mask contains the labels for the rows and columns of the matrix, the relationships between data in the various entry cells, and any fixed data that the program might need. This mask can then be saved for later use, or for use by someone other than the programmer. A VisiCalc mask for tax forms, for example, can be used by millions of taxpayers, each of whom would enter his or her own data and let the spread sheet calculate the results.

Interestingly, "programming in VisiCalc" doesn't require a knowledge of anything like a "normal" computer language. It doesn't matter if you program the labels first, or if you create some of the row and column relationships (e.g., making the contents of each entry in column C the product of the contents of each entry in column A and column B) and then fill in some data. The programmer has access to the whole program at once and can try out bits and pieces as he or she moves along. If writing programs in BASIC can be considered a "serial" process, programming in VisiCalc is a "parallel" process.

It is this radical departure from traditional programming styles that makes VisiCalc interesting to me as a language. Philosophers have long maintained that what we think is influenced by the choice of language in which we express our thoughts. That many hundreds of thousands of business people seem to be better "thinkers" with the aid of programs like VisiCalc is testimony to the power of this language.

Rocky's Boots As A Language

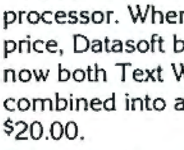
The Learning Company is well known for its high-quality educational software and for its special emphasis on the development of logical thinking skills. Hidden among its application software packages is a computer language called Rocky's

WORD PROCESSING THE DECISION IS EASY!!!

Datasoft eliminates the frustrations of choosing a word processor. When it comes to ease of use, versatility, and price, Datasoft beats the competition hands down, and now both Text Wizard™ and Spell Wizard™ have been combined into a specially marked package to save you $20.00.

Text Wizard, one of few programs to receive four (4) excellents on the Infoworld Report Card Review, let's you create letters, documents, reports and much, much more. It's as simple as a few keystrokes to edit, erase, move lines or paragraphs.

With Spell Wizard and its 33,000 + Word Dictionary, you can eliminate embarrassing typos or spelling errors.



Spell Wizard automatically indicates errors, and allows you to immediately view your dictionary for easy correction. You can also create or add unique words to your own custom dictionary.

You might expect to pay $79.00 to $99.00 for only one of these programs, however, Text Wizard normally $49.95 and Spell Wizard normally $49.95 suggested retail, can both be purchased for a limited time in specially marked packages for only $79.95 total price, a savings of $20.00.

Check with your local dealer or send check or money order with $3.00 postage and handling (California residents add 6½% sales tax) to:

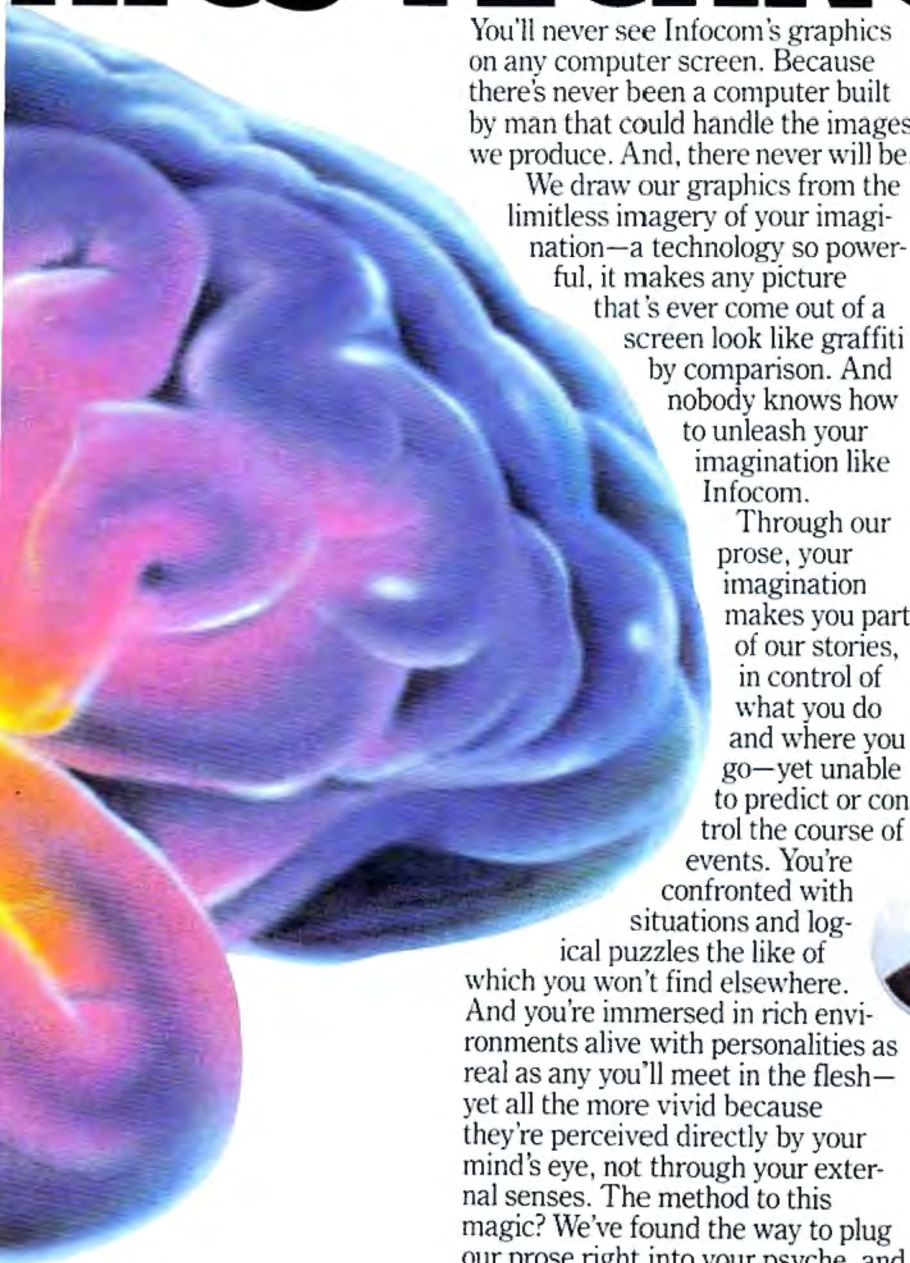
Text Wizard
Atari 400/800 (32K min.)

Spell Wizard
Atari 400/800/1200 (32K min.)

DATASOFT, INC.
9421 Winnetka Avenue, Chatsworth, CA 91311 (213) 701-5161

Datasoft®

E WORLD'S MOST HICS TECHNOLOGY.



You'll never see Infocom's graphics on any computer screen. Because there's never been a computer built by man that could handle the images we produce. And, there never will be.

We draw our graphics from the limitless imagery of your imagination—a technology so powerful, it makes any picture that's ever come out of a screen look like graffiti by comparison. And nobody knows how to unleash your imagination like Infocom.

Through our prose, your imagination makes you part of our stories, in control of what you do and where you go—yet unable to predict or control the course of events. You're confronted with situations and logical puzzles the like of which you won't find elsewhere. And you're immersed in rich environments alive with personalities as real as any you'll meet in the flesh—yet all the more vivid because they're perceived directly by your mind's eye, not through your external senses. The method to this magic? We've found the way to plug our prose right into your psyche, and catapult you into a whole new dimension.

Take some tough critics' words about our words. SOFTALK, for example, called ZORK® III's prose "far more graphic than any depiction yet achieved by an adventure with graphics." And the NEW YORK TIMES saw fit to print that our DEADLINE™ is "an amazing feat of programming." Even a journal as video-oriented as ELECTRONIC GAMES found Infocom prose to be such an eye-opener, they named one of our games their Best Adventure of 1983.

Better still, bring an Infocom game home with you. Discover firsthand why thousands upon thousands of discriminating game players keep turning everything we write into instantaneous bestsellers.

Step up to Infocom. All words. No graffiti. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

INFOCOM™

The next dimension.

Infocom, Inc., 55 Wheeler St., Cambridge, MA 02138

For your: Apple II, Atari, Commodore 64, CP/M 8", DEC Rainbow, DEC RT-11, IBM, NEC APC, NEC PC-8000, Osborne, TI Professional TRS-80 Model I, TRS-80 Model III

Figure 1: OR Gate Connected To Clapper

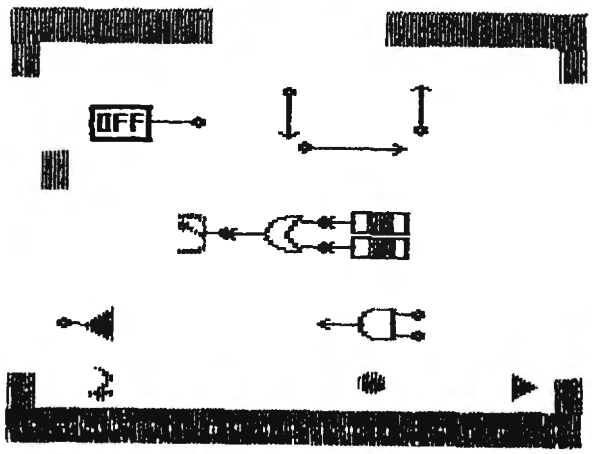


Figure 2: Knife Cuts Clapper From OR Gate

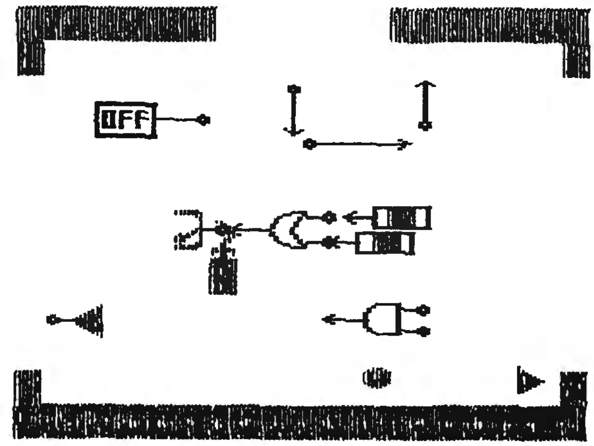


Figure 3: Clapper Replaced By OFF/ON Box

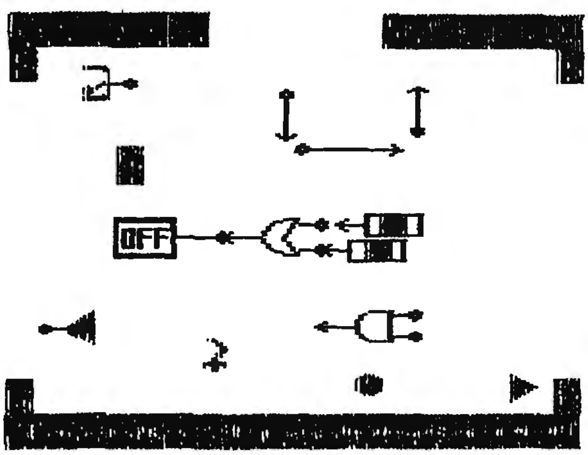


Figure 4: AND Gate Connected To Machine Shown In Figure 3

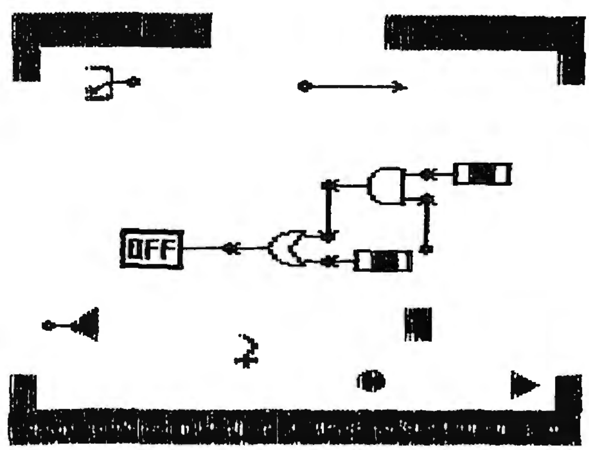


Figure 5: Completed Machine In Use

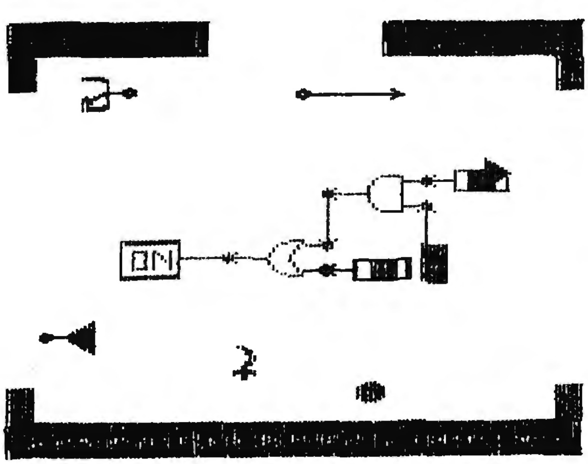
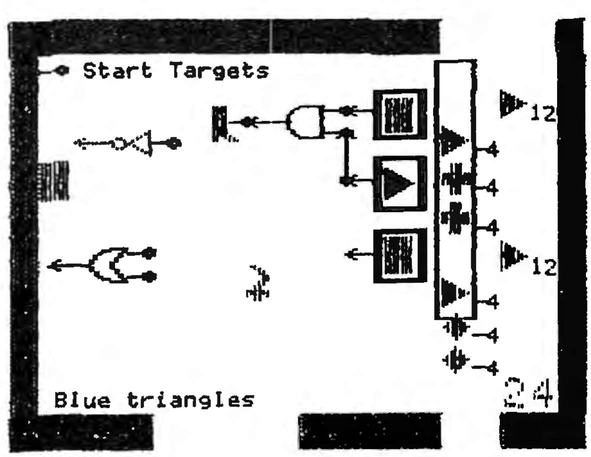


Figure 6:
The Machine Used To Identify Blue Triangles (Solving A Problem In Logic)



Boots. This language assumes the guise of a program designed to let the user solve logic problems by building "logic machines." These logic machines are, in fact, schematic diagrams constructed from AND, OR, and NOT gates, flip-flops, wires, input sensors, and output devices. Through a series of carefully staged exercises, the user is taught how to use the program (or, as I claim, how to write programs in the Rocky's Boots language).

Interestingly, the construction of logic machines (this is the programming task) can be done without access to the keyboard by using a joystick to pick up objects (program subroutines) and interconnect them into a complete machine (program). Incorrect connections can be "cut" with a special knife (the editor). The only frustration I have encountered with this language is that no provision was made for the saving of finished "machines."

The figures show how a program can be edited in Rocky's Boots (remember that a machine is in reality the computer program you have created). Figure 1 shows a machine consisting of an OR gate connected to a clapper. One of the OR gate inputs is a blue sensor, and the other is a green sensor. The remaining items in the picture are "spare parts" (i.e., subroutines available for use). Suppose you want to change this machine into a new one that will turn the word OFF to ON if the green sensor is activated OR if both the blue sensor is activated AND the cursor is touching an input.

To modify the machine, the cursor (shown as the large solid rectangle) is used to pick up the knife shown at the bottom-left side of Figure 1. In Figure 2 the blue sensor has been cut, and the knife is shown cutting the clapper from the OR gate. In Figure 3 the clapper has been replaced by the OFF/ON box, and in Figure 4 the AND gate and some wires have been connected to complete the new machine. Figure 5 shows what happens when the machine is used and the proper conditions are satisfied (in this case, both the blue sensor and the remaining input of the AND gate are activated).

Rocky's Boots is similar in many ways to VisiCalc. The user has access to the whole program at once (it is a parallel environment), and the program (the machine) can be tested as it is being built. Of course, the application areas for these programs are quite different from each other.

The machines created by Rocky's Boots are designed to be used to solve logic problems. Figure 6 shows one such problem in which a machine has been built to identify blue triangles.

One can argue that Rocky's Boots is a game, a simulation, and a language.

No matter what it is called, it is a very powerful piece of software.

Next month we will conclude this series with the description of yet another powerful program/language, Dancing Bear from Koala Technologies. We will also discuss the potential impact of these types of languages. ©

Richard Mansfield, Senior Editor

Your First Useful Program

It doesn't take long. Soon after you buy a computer, someone will ask what it's doing for you that's *useful*. You've been learning to program in BASIC, but what practical results have you got to show for your efforts? The questioner might be a friend or even someone in your immediate family, someone you usually love.

Here's a program which can be used in dozens of practical ways. It makes it easy to store and analyze information. We all keep lists and records. If you have a collection of something – recipes, books, stamps, albums, whatever – you can enter all sorts of information into this program and then look things up later in a variety of ways. You can ask for everything starting with the letter A, everything on a particular topic, from a particular country, or whatever specifications you want. This type of program is often called a *data base manager*. It's one of the most effective, impressive applications for a personal computer.

How To Modify The Program

A similar program was published in this column two years ago and proved popular. I received this letter the other day:

> *I would like to ask a favor. I have been using your program "Searching Files" (November 1981) for some time, and very much appreciate it. But I've tried without success to add categories like 3. Publication, 4. Date. If possible, could you explain how to expand the program to include additional categories?*
>
> *Mel Leiserowitz*

That a programmer can make modifications to a program is one of the most subtle, but powerful, aspects of computing. Unlike other kinds of tools, a given computer program can often be radically transformed so that it can accomplish a great variety of tasks. Let's take this program apart, looking at each aspect of it, and then explore how to modify it to include extra categories.

Data base management is frequently divided into two phases: the manager program and the actual base of data. A *data base* would be a list of pieces of information, perhaps in alphabetic or some other order, like the phone book. The data base is often stored on a tape or disk, and the manager program looks up information by opening and closing files on the tape or disk. The second phase, the manager program, can make all kinds of lists for you; it can analyze the data base. For example, if the yellow pages were on a disk, you could write a program which would allow you to ask highly specific questions. You might want to know the phone number of all pizza shops within a five-mile radius of your house. If the data base included map information, the manager program could give you the answer.

A Faster And Easier Way

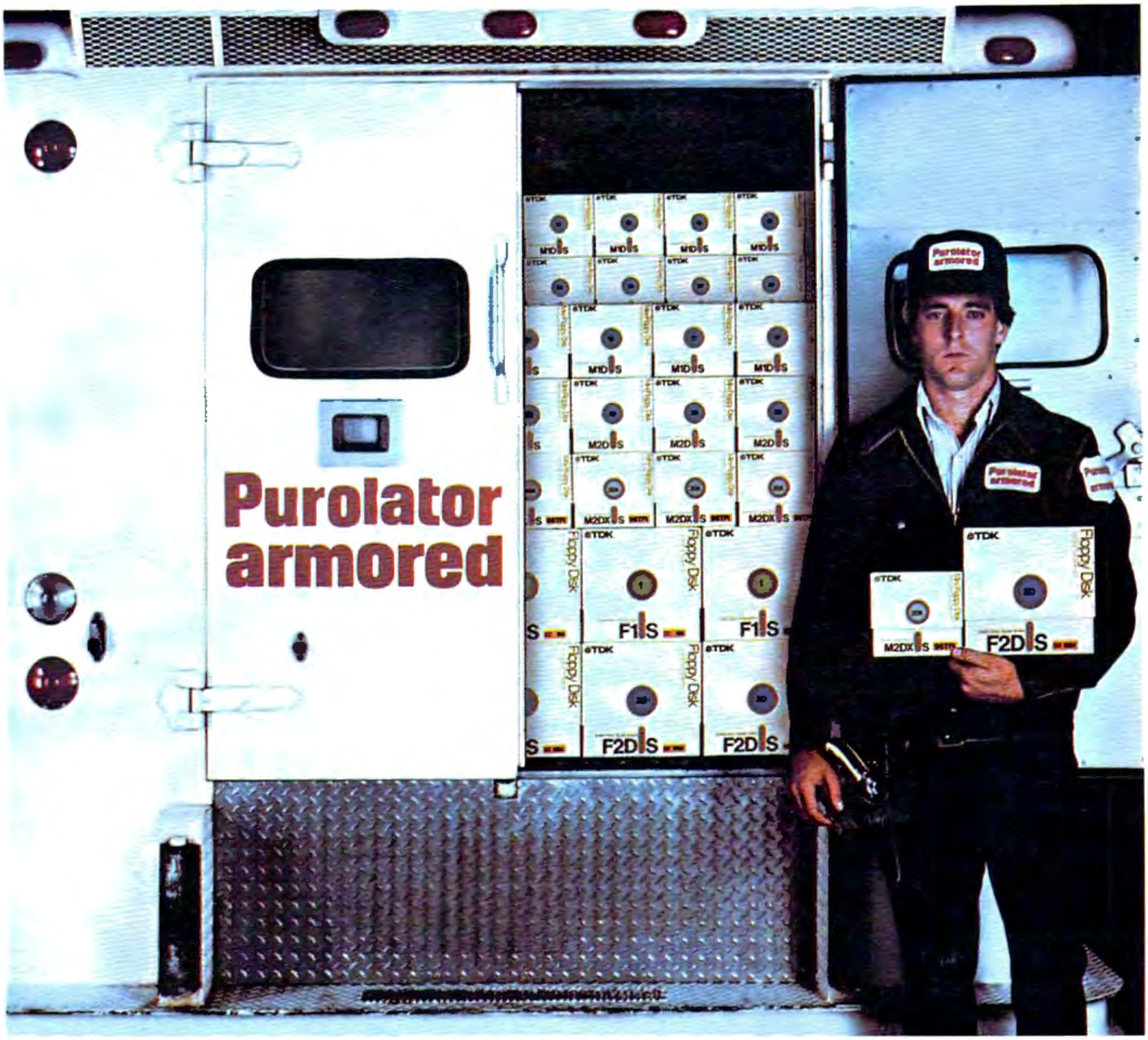
This program, "The Filer," combines the data base into the manager program in the form of DATA statements, each holding an individual record. The advantage of this is that you can add to and modify the data easily, on screen. It's also faster: the computer doesn't need to bring data in from tape or disk. The program contains the data already. The disadvantage is that your data base cannot be larger than the amount of memory in your computer. You should be able to find many uses for this program, however. If your computer has 32K of RAM memory, you can store detailed information about all the articles in COMPUTE! for any given year. If you want to start a data base for a new year, you can simply use the program again, with a new set of DATA statements. In our example use of The Filer we'll start a cross-indexed reference file of all COMPUTE! articles.

The program is designed to work on all computers using Microsoft BASIC (Commodore, Radio Shack, TI, Apple, etc.). If you have an Atari, there's a version for your machine in COMPUTE!, November 1981.

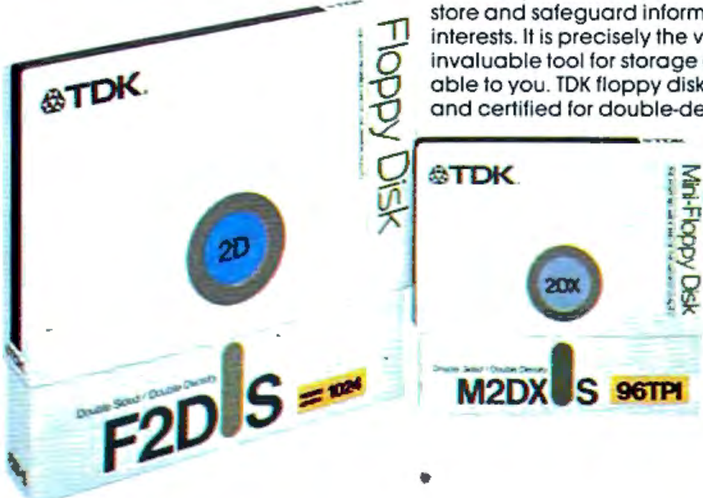
Let's go through the program to see how it works:

Line Number

1 This lets the computer know the total number of items in our data base. T = 10 because we've got ten DATA lines in this program. The data base starts at line 502 and continues to the



TDK Floppy Disks.
Invaluable security for irreplaceable information.



Today, more and more companies are relying on convenient floppy disks to record, store and safeguard information. Irreplaceable information which is vital to their business interests. It is precisely the value placed on this information that makes the floppy disk an invaluable tool for storage and security. And this is where TDK floppy disks become invaluable to you. TDK floppy disks are guaranteed 100% error-free at the time of manufacture and certified for double-density encoding. Furthermore, each track of every TDK floppy disk is tested to exceed industry standards... including those of IBM, Shugart, ANSI, ECMA, ISO and JIS. Once you insert a quality TDK floppy disk into your computer system, you're guaranteed highly reliable, ultra smooth performance. This is due to TDK's proprietary disk-burnishing technique that provides optimum head-to-disk contact.

TDK floppy disks are available in 5¼ and 8-inch sizes in the most popular formats. Each disk comes in its own protective Tyvek-type envelope. For a copy of our brochure, "Some Straight Talk About Floppy Disks," write to: TDK Electronics Corp., Computer Products Marketing Dept., 12 Harbor Park Drive, Port Washington, NY 11050, or call 516-625-0100.

TDK®
The heart of your system.

We just made owning an Atari computer a lot more logical.

Introducing the Rana 1000 disk drive. It's a whole new game for Atari computers.

This two digit LED readout displays a code that tells you everything you need to know.

This switch tells you your write protect feature is keeping your information safe.

The remaining switches provide readouts on density storage, error status, and drive number.

This switch, when you touch it, tells you what track you're on.

When Rana Systems introduced the Elite Series of Apple® compatible disk drives, we didn't know what a tremendous impact they would make. It turned out to be a line so outstanding in performance, styling, capacity, and price, that it instantaneously made us a major force in the market. Well, needless to say, the response was so great that we were forced to create the same highly advanced disk drive for Atari® A disk drive that when coupled with Atari's computer, could perform everything from accounting, financial planning, and stock charting, to word processing, business management, and letting you write your own programs. Plus, we made it simple enough for a child to use, for learning anything from the alphabet to a foreign language.

Working with a diskette versus playing with a cassette.

Let's face it. The only reason Atari made a cassette option to their computer was to make it affordable. But now you don't have to settle for less. Because now you can get a diskette for your Atari computer which outperforms their cassette. With Atari's cassette you only get half the functions of a computer compared to what our floppy disk can give you. Their cassette is not only limited in the software available, but it also takes 20 times longer to get the information you need. And Rana's disk drive offers twice the storage capacity of either their cassette or disk drive.

Why even stylewise our new low profile design not only looks 100 times more spectacular, but it occupies 3 times less space. And our new Rana 1000 also gives you a piece of its mind every time you use it, because our disk drive gives you information as well as takes it. And we think that says a lot.

The disk drive that has all the answers.

Rana offers you a myriad of features Atari couldn't even conceive of. Like five electronic functions on the front panel that give you a LED readout when touched. Our disk drive tells you what track you're on, and what density and how much information you're storing. And, we have a write protect feature which protects your diskette from being erased. In fact, no other disk drive can offer you that.

As you can see, it was easy to build a disk drive superior to Atari's. Because for every reason you buy a disk drive, Rana has superior technology.

The Rana 1000 disk drive. It brings your Atari computer to a higher level of sophistication for a price one third lower than Atari's. So your choice shouldn't even be a matter of logic.

Just common sense.

Always a step ahead.

21300 Superior Street, Chatsworth, CA 91311 213-709-5484. For dealer information call toll free 1-800-421-2207. In California only call 1-800-262-1221. Source Number: TCT-654 end of the program. If you add 400 more DATA lines, you should change line 1 to read: T=410.

2 Since reading a large data base might take some time, this statement appears on screen to let the user know that the computer is busy and will return control to him or her shortly.

3 Here we DIMension the three variables which will be holding our data. These tens, too, would need to be changed to 410s if you added 400 more DATA lines.

10 The computer assigns a special variable name to *each* item of data by READing through the entire list. We've got three categories per record. A$(? – whatever I= during the READing) will be topic identification for the COMPUTE! articles in our data base. B$(?) will contain the issue number and the page number. C$(?) holds the author name.

15-45 Here the computer gives us a choice. We can look things up either by topic or by author.

50 We now make our request. If X=1 (see line 35), then we're after the author so we're sent down to line 70, which searches through C$( ). If not, we proceed to line 55 for a search of A$( ), topics. Lines 55-65 and 70-80 are identical searches, except one looks at C$( ), the other at A$( ). Since they're the same, we'll just examine the interesting pattern-matching technique where it appears the first time, in line 60.

60 This is the heart of the program. It's the trick that lets you look things up without knowing their exact names. It also makes possible varying *depths* of specificity. If you add a data line: 522 DATABASIC MEMORY SAVING, you can then request anything from B to BASIC MEMORY SAVING and this item will show up on the list. If you request matches to BASIC, you'll get this one and line 510. If you request BASIC MEMORY, line 510 will be ignored.

How does it work? N$ is the word or words you entered (line 50) and to which you want all matches. A$(I) will scan through the entire data base "topics column." For a match to take place, only the leftmost part of A$(I) needs to match N$. Z does this for us; it's the length of N$ (see line 50). That is, we're looking for matches from pieces of A$(I) only as big as N$. So, you type in your topic DATA with this in mind. Enter each record so that the first word is the most general, the second more specific, etc.

85 This subroutine prints out any matches which are found. It contains descriptions of the categories. There are a number of ways to format such things. You might prefer, for example, to list the category titles only once, at the top of the screen, and then list everything in columns underneath them.

Expanding The Categories

As you can see by following the changes suggested in Programs 2-4, some minor structural modifications are necessary to make Program 1 handle a fourth category, Computer brand. However, it would be simple to add a fifth or more categories to this new version.

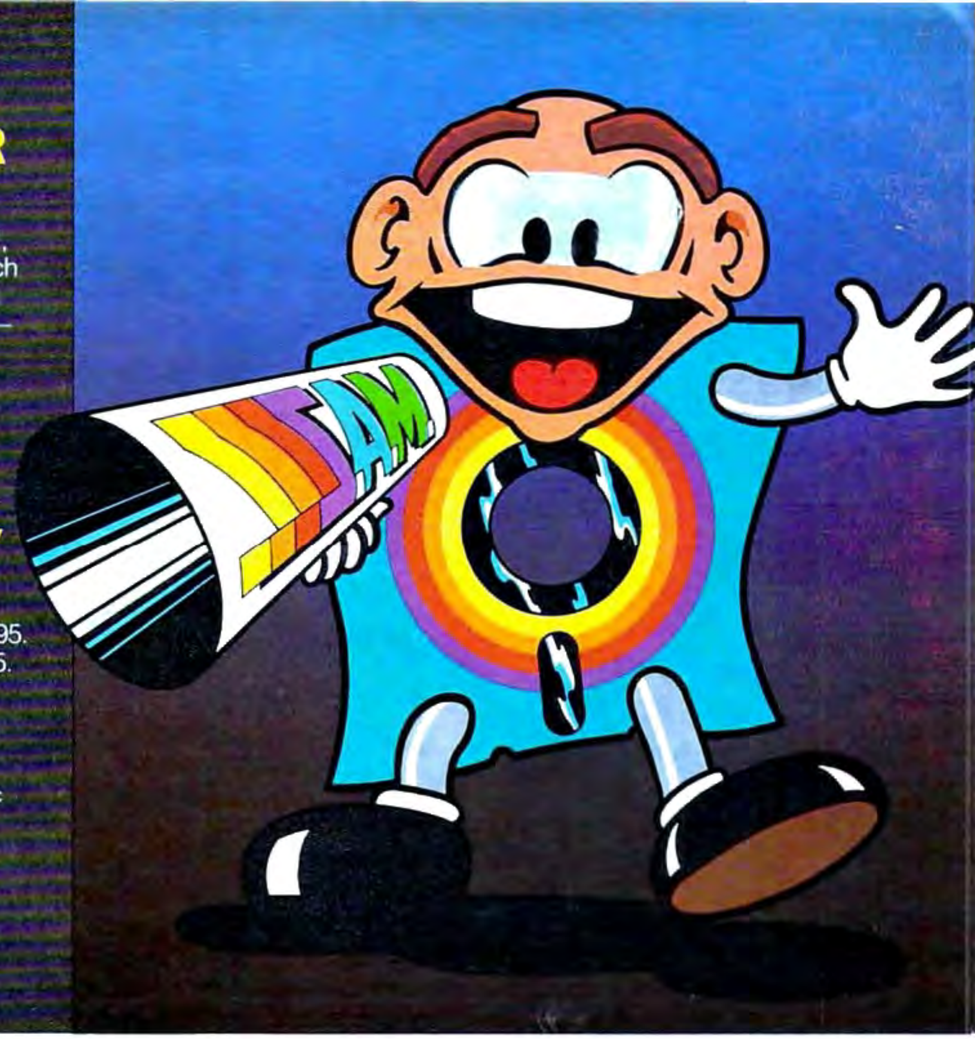
One thing to watch out for: the READ statement doesn't care what data is on a given DATA line. It reads things very literally and checks for commas (or the end of a line) to tell it that a particular item has ended. So, if you get odd responses such as EDUCATION when you're asking for author names starting with E, or an OUT OF DATA ERROR – you've probably left out a comma somewhere in the DATA lines. Also, your DATA lines will be longer with this new, four-category, version of The Filer. They'll now look something like this:

```
520 DATALANGUAGES PILOT,1/40,THORNBURG,VIC
```

If you have any questions or topics you'd like to see covered in this column, write to "The Beginner's Page," COMPUTE! Magazine, P.O. Box 5406, Greensboro, NC 27403.

Program 1: The Filer

```
1 T=10:REM{8 SPACES}  TOTAL NUMBER OF
  ITEMS OF DATA
2 PRINT"{3 SPACES}READING{5 SPACES}COMPU
  TE!{5 SPACES}DATABASE
3 DIMA$(10),B$(10),C$(10)
10 FORI=1TOT:READA$(I),B$(I),C$(I):NEXT
15 PRINT:PRINT"SELECT A CATEGORY":T$="TO
   PIC"
20 PRINT"{4 SPACES}1. AUTHOR
25 PRINT"{4 SPACES}2. SUBJECT
30 K$="":GETK$:IFK$=""THEN30
35 X=VAL(K$):IFX>2THEN30
40 IFX=1THENT$="AUTHOR'S NAME"
45 PRINT"{3 SPACES}PLEASE ENTER "T$
50 INPUTN$:Z=LEN(N$):IFX=1THEN70
55 FORI=1TOT
60 IFN$=LEFT$(A$(I),Z)THENGOSUB85
65 NEXTI:GOTO15
70 FORI=1TOT
75 IFN$=LEFT$(C$(I),Z)THENGOSUB85
80 NEXTI:GOTO15
85 PRINTA$(I);"...IN ";B$(I);" (ISSUE#/P
   AGE), WRITTEN BY ";C$(I):RETURN
498 REM
499 REM
500 REM *** TOPIC OF ARTICLE -- ISSUE/PG
     -- AUTHOR ***
502 DATABUSINESS,1/4,SAWYER
504 DATASORTING,1/7,HULON
506 DATAWORDPROCESSORS,1/13,LINDSAY
508 DATASCIENTIFIC INSTRUMENTATION,1/24,
    BYRD
510 DATABASIC TOKENS,1/29,HERMAN
512 DATAMODEM,1/30,TULLOCH
514 DATAA-D CONVERTER,1/31,HERMAN
516 DATAML MEMORY TEST,1/32,MOSER
518 DATAEDUCATION,1/34,BARRETTE
520 DATALANGUAGES PILOT,1/40,THORNBURG
```



Program 2: Change These Lines

```
3 DIMA$(10),B$(10),C$(10),D$(10)
10 FORI=1TOT:READA$(I),B$(I),C$(I),D$(I)
   :NEXT
35 X=VAL(K$):IFX>3THEN30
50 INPUTN$:Z=LEN(N$)
55 FORI=1TOT:ONXGOSUB60,65,70:NEXTI:GOTO
   15
60 IFN$=LEFT$(C$(I),Z)THENGOSUB85
65 IFN$=LEFT$(A$(I),Z)THENGOSUB85
70 IFN$=LEFT$(D$(I),Z)THENGOSUB85
```

Program 3: Add These Lines

```
26 PRINT"{4 SPACES}3. COMPUTER
42 IFX=3THENT$="COMPUTER"
61 RETURN
66 RETURN
71 RETURN
```

Program 4: Drop These Lines

```
75 IFN$=LEFT$(C$(I),Z)THENGOSUB85
80 NEXTI:GOTO15
```

©

TELEGAMES

Computer Games By Phone

John Blackford

A new species of game is suddenly gaining in popularity. It's the telegame – played over the phone lines. In some versions, you play against the computer; in others, many players can join in a single game. When one player makes a move, the others see it almost instantly. Such games allow team efforts and that opens up an intriguing new set of possibilities.

In the movie *WarGames*, a high school student accidentally taps into NORAD's war-game computer. The computer, which is equipped with artificial intelligence programs, is prepared to play such favorites as chess, tick-tack-toe, and global thermonuclear war. After the student chooses the latter, the computer won't quit, seizing control of launch codes and missile silos in preparation for a real nuclear strike.

How did the student bring about this near disaster? Practically the same way that people around the country now call up computerized information services and use them to play games: he put his telephone receiver in a modem – a device connecting the computer to the phone lines – readying his computer to call another computer.

Many information services contain – among other things – a library of games, which people can call up and play. In some, users play against the computer, as in *WarGames*. In others – the multiplayer, interactive games – the computer acts as a referee, doing the housekeeping chores, accounting for players' moves, and generally running the game. At present, interactive gaming is available only from CompuServe Information Service, though The Source and Delphi are working to catch up.

Only At Lunch Hour

CompuServe got a head start in interactive games by chance. The company began as a data base for business users, offering stock market quotes, sugar futures quotes, and the like during the day. After a few years of setting up such services, Russ Ranshaw, one of the company's programmers, decided to create a simple space-war game called Space Wars (SPCWARS) for the recreational use of other employees.

"It seemed like a logical thing to do," he says. "But it got so darned popular that people were playing all the time." After looking the other way for some time, company officials eventually limited game access to lunch hour. Even that didn't do the trick, so finally, in 1976, the game was completely banned from CompuServe – and it wasn't to be found in the memory banks for several years after that.

As personal computers began reaching homes in increasing numbers, planners at CompuServe (and other information utilities) decided to make their services available to home users at reduced rates during off-hours. To make the service attractive to non-business users, some new features were added. One of the first was SPCWARS, and it proved just as popular as it had been during lunch at CompuServe. In fact, it and two other interactive space games added later are now among the most widely used parts of the system.

SPCWARS is fairly simple to play. The commands aren't hard to follow, and a *help* function permits you to learn the rules as you play. It's an interactive game in which everyone is gunning for any player who signs onto the system. You can hide in clouds, duck around stars, and even display the section of the galaxy your ship is probing. Since true graphics aren't possible, the display consists of various symbols and letters to designate the location and direction of your ship and the other objects in the game. As more people

sign onto the game, the size of the playfield expands, and if the number of players gets unwieldy, the host computer starts a new game for the newcomers.

No Help In Sight

Two other currently available interactive games are exceedingly complex. Forget trying to learn them as you go. Before you even figure out how to move your ship, you'll see a long string of messages race across your screen. You are under attack. The *help* command no longer works. You may notice the coordinates of the attacking ship and attempt to direct some phaser fire his way. But suddenly it's over. As you try to figure out what happened, you'll get a message like, "Sorry, Cadet, you're dead. You didn't cut it in MegaWars."

Save yourself some embarrassment – and wasted time – and order the instruction book before trying to play this one. You can order on-line (through your computer) or by writing CompuServe direct (5000 Arlington Centre Boulevard, P.O. Box 20212, Columbus, OH 43220).

DECWARS was the first really complex interactive game. It's actually a revision of a space game that had been residing for some years on a mainframe at the University of Texas. Ranshaw got it and worked obsessively to get rid of all the bugs. The task proved more than he had bargained for, and he now thinks he could have done the whole thing from scratch in less time. Still, reaction was favorable, right from the start. Players signed on again and again to play. A special interest group (SIG) was even formed for DECWARS fans. Users would use a special area of CompuServe to exchange comments and ideas about the game.

Some of these players began suggesting improvements, and Ranshaw got in touch with them to refine the concept for an improved game. In both games, players can form teams, but the regulars thought MegaWars – the upgraded version – should also assign ranks based on past performance. This would reward ability, yet allow beginners to fly more durable ships, increasing their survival time immensely.

After a long development – marked by enthusiastic suggestions from nearly everyone – the game went on-line. Immediately, CompuServe was flooded with suggestions for improvement. People would sign on the DECWARS SIG and fill the screen with criticisms. It became a significant problem.

After that experience, Ranshaw says they all realized that while suggestions are great, there comes a time when the programmer simply must do what seems best, letting others decide whether the game's fun to play. In spite of criticism by DECWARS fans, MegaWars went on to become highly successful. At present, it is CompuServe's second most profitable offering – behind a simulation of CB radio that permits people around the nation to sign on and chat.

Old Favorites

Although CompuServe has the only interactive games right now, traditional favorites played against the computer are popular on all systems. Other information utilities, such as The Source, Dialog, and Delphi, also offer single-player games on-line. There are adventure-type classics such as *Wumpus*, *Star Trek*, and *Zork* as well as computer versions of such popular games as *Othello*, backgammon, and even chess. Most of the companies also offer card games, roulette, and dozens of other brain teasers.

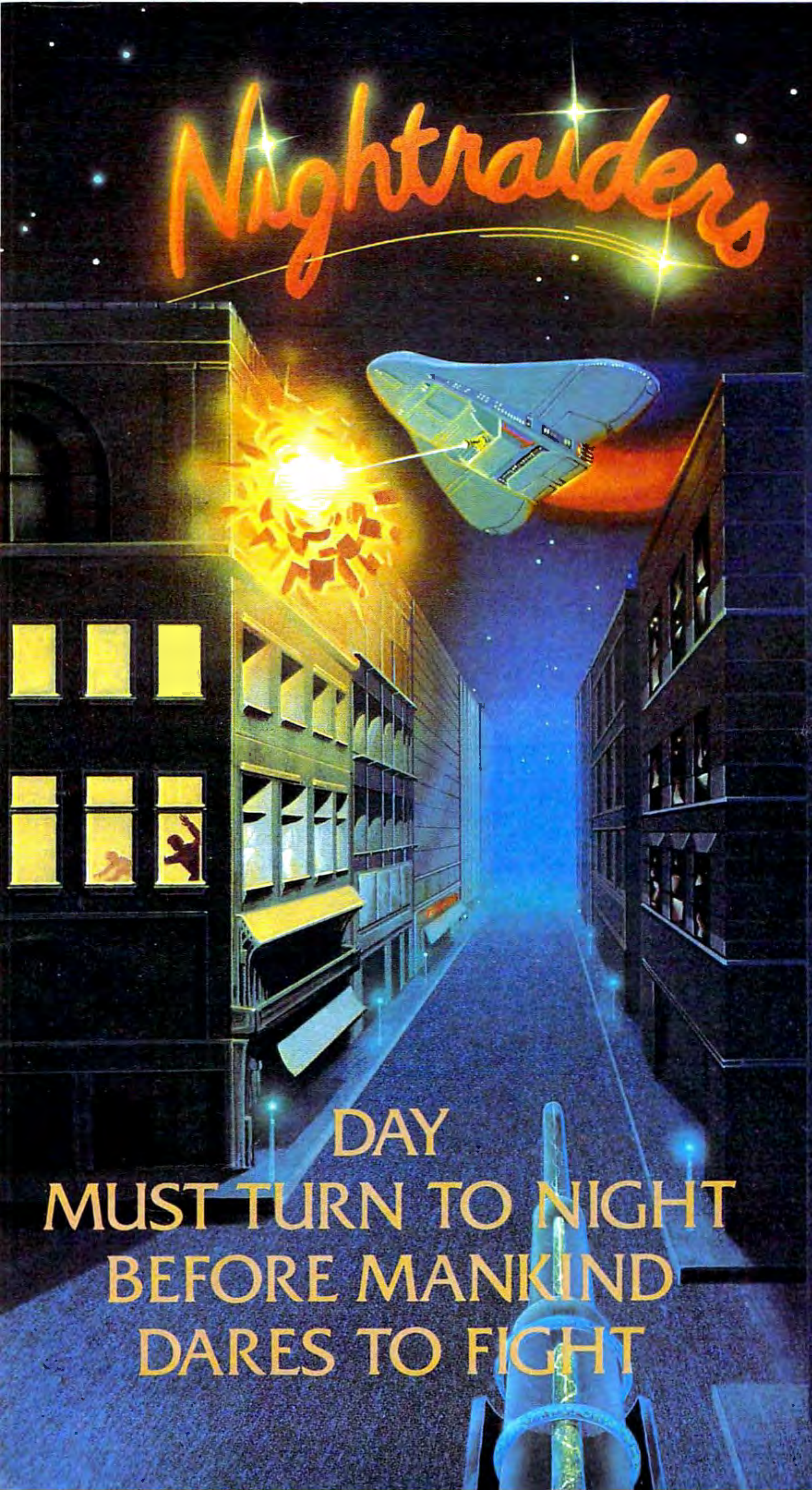
To supplement such traditional games, The Source and Delphi have their programmers working hard to complete several multiplayer games. The Source isn't saying what titles it's creating, but Delphi is finishing up two space games, *Conquest* and *Parsec*, and will introduce more fantasy-oriented fare, too: *Scales of the Gods*, a medieval adventure, and *Timelords*, a game which involves exploration of "the fourth dimension."

As Delphi tried to hammer these into shape, its own users beat it to the punch, creating several multiplayer games right on the system. One player, known as the "Dragon," served as dungeon master for a couple of adventure games. The players took advantage of the bulletin board and electronic mail services to create the games. Now Delphi has made special space available for the game devotees. Though these adventures have generated excitement, they aren't *programmed* games. Instead, it's the players themselves who make things happen. Users create their own story, using the dungeon master to communicate with others. That's actually part of the idea at Delphi: the service should change to reflect the needs of users. Says president Wes Kussmaul, "Delphi is almost a creation of the users – they are the ones who bring it to life."

Interactive Games

What is it that makes interactive games hard to create? According to Ranshaw – and he's built every one that's commercially available – they require tricky programming with built-in safeguards to protect the game when someone drops out. Says Ranshaw, "What if you are playing a four-card game, and one player's cat knocks his modem off the table, disconnecting the phone? Suddenly the player is gone. Do you step in with the computer and have it play for the missing person, or bomb the whole game?"

The ideal, according to Ranshaw, is to minimize computer involvement. But in a case like the example above, the program would have to take the missing hand. Delphi's Kussmaul has a different philosophy, noting that if you are playing a real game of bridge, and someone walks


Nightraiders
DAY
MUST TURN TO NIGHT
BEFORE MANKIND
DARES TO FIGHT

off, the game is over. The same should be true of interactive computer games, he thinks.

A tougher problem arises with multiplayer games such as MegaWars. If the program isn't properly done, the game can crash when one player pulls out, destroying what may be hours of effort for some of the players. To prevent this, each player is considered to be a separate "case" by the host computer. The game program controls each player's input individually. Thus, if someone drops out, his or her specific case is closed, but the game data remains intact.

Graphics To Come

All the games currently available through information utilities are done in alphanumeric characters. Whatever you see on the screen could be typed on paper with a standard computer printer. In the early days of telecommunications, a set of standard characters, called the ASCII code, was developed to improve the service beyond that available on teletype machines. ASCII permits upper- and lowercase letters plus punctuation marks and a few control codes. The graphics characters on home computers aren't standard ASCII. Since information services must be able to communicate with many different computer brands, they are limited to the ASCII character set.

That could change, however. Developers at CompuServe are trying to create telegames with movable graphics. Besides the differences between computers, the limitations of the phone lines themselves hinder graphics transmission. Although baud (bits per second) rates of 1200 or higher are possible, modems capable of such speeds are still fairly expensive. Most home users communicate with information utilities at a leisurely 300 baud, far too slow to permit movement of graphics on your screen. But the people at CompuServe are working on a way to get around this bottleneck.

They are attacking the problem in two ways: first, by creating software for each computer type, and second, by devising ways to transmit graphics information without actually having to transmit the entire picture. Most videogames consist of a background design (the playfield) and objects that move within it. CompuServe's game designers hope to define the playfields and graphics shapes and download the definitions to the user. Then, the host computer only needs to transmit enough information to move the predefined shape. Software purchased by each user will make the graphics information compatible with his or her particular computer.

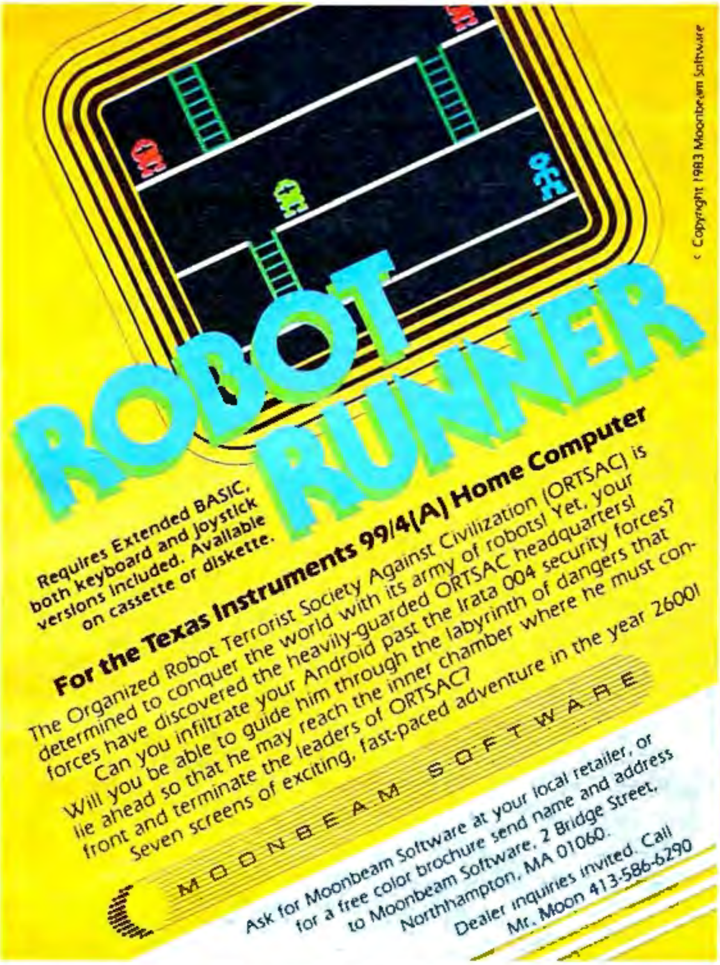
New Directions

Such efforts may be only the beginning. With inexpensive modems and software now available, more and more home computerists are reaching information sources. And such organizations have found to their surprise that there's a healthy profit in home-oriented services. A company by the name of Gameline even has a plan to sell plug-in cartridges to allow Atari VCS game machines (11 million are presently in use) to download game software. And parents will have a special code enabling them to limit their children's use of the device.

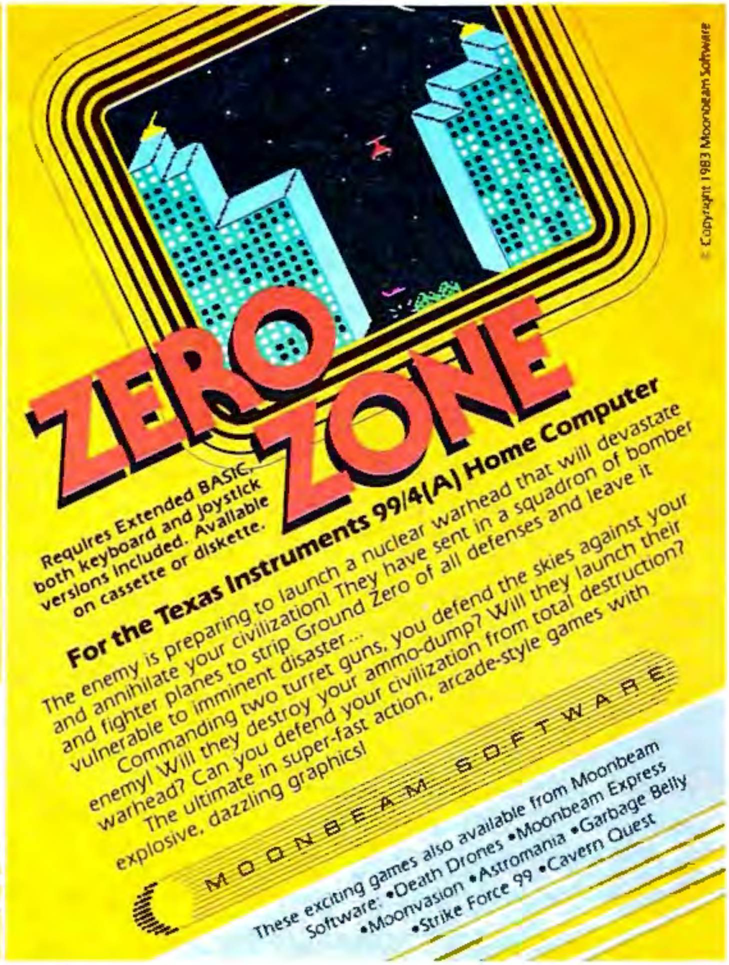
In fact, activity could become so widespread that phone lines might be filled up with people using personal computers plugged into various data bases, including teletex, on-line bulletin boards, and even users chatting directly to one another. Bell telephone researchers are reportedly concerned that the entire phone network could become overloaded if market penetration of telecomputing services reaches as little as 3½ percent.

To avoid these problems, alternatives such as cable TV and local communications networks may be used. The Games Network is planning to offer a cable channel dedicated to videogames. Sytek, in cooperation with General Instruments Corporation, plans to introduce a series of local, high-speed communications networks using cable TV lines. These would be cheaper than comparable ones offered by American Bell, and would be compatible with personal computers. Such a system could support extensive graphics, because of the large transmission capacity of cable lines.

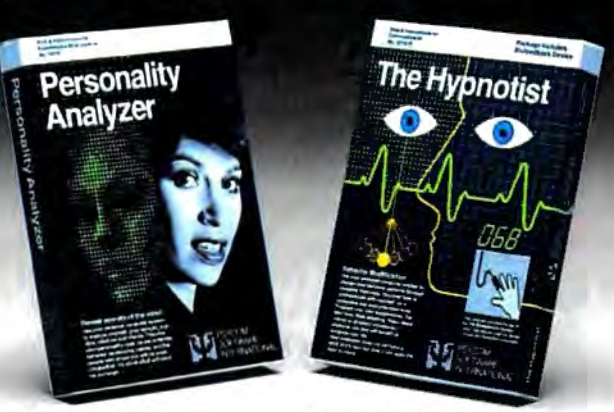
Whichever specific projects eventually succeed, the prospect is for more and better computer games played over the communication lines. Just as home computers changed in only a few years from hobbies to mass-market items, telegaming is now poised to be the vanguard of a massive upsurge in computing by phone. ©



ROBOT RUNNER
Requires Extended BASIC, both keyboard and joystick versions included. Available on cassette or diskette.
For the Texas Instruments 99/4(A) Home Computer
The Organized Robot Terrorist Society Against Civilization (ORTSAC) is determined to conquer the world with its army of robots! Yet, your forces have discovered the heavily-guarded ORTSAC headquarters! Can you infiltrate your Android past the Irata 004 security forces? Will you be able to guide him through the labyrinth of dangers that lie ahead so that he may reach the inner chamber where he must confront and terminate the leaders of ORTSAC?
Seven screens of exciting, fast-paced adventure in the year 2600!
MOONBEAM SOFTWARE
Ask for Moonbeam Software at your local retailer, or for a free color brochure send name and address to Moonbeam Software, 2 Bridge Street, Northhampton, MA 01060.
Dealer inquiries invited. Call Mr. Moon 413-586-6290
© Copyright 1983 Moonbeam Software



ZERO ZONE
Requires Extended BASIC, both keyboard and joystick versions included. Available on cassette or diskette.
For the Texas Instruments 99/4(A) Home Computer
The enemy is preparing to launch a nuclear warhead that will devastate and annihilate your civilization! They have sent in a squadron of bomber and fighter planes to strip Ground Zero of all defenses and leave it vulnerable to imminent disaster...
Commanding two turret guns, you defend the skies against your enemy! Will they destroy your ammo-dump? Will they launch their warhead? Can you defend your civilization from total destruction?
The ultimate in super-fast action, arcade-style games with explosive, dazzling graphics!
MOONBEAM SOFTWARE
These exciting games also available from Moonbeam Software: •Death Drones •Moonbeam Express •Moonvasion •Astromania •Garbage Belly •Strike Force 99 •Cavern Quest
© Copyright 1983 Moonbeam Software



Personality Analyzer
The Hypnotist

NONVIOLENT GAMES

Kathy Yakal, Editorial Assistant

The violence that is inherent in many of today's video games is disturbing to some people. Others don't see it as a negative influence; they stress the positive aspects of playing and programming video games. In this article, we explore both sides of this controversial issue, and look at some software designers who are providing alternatives to typical arcade games.

VIDEO GAMES (see Murder)

This entry can be found in *The New York Times Index* for January 1-16, 1983. The article alluded to is a small item in the January 9 *Times* about a high school senior in Dallas who was "shot to death in the parking lot of an arcade after a quarrel over 75 cents worth of video display games."

It's not so unusual anymore to hear about someone being killed over something rather trivial. But what might make this act of violence significant to some people is its relationship to video games.

Video games embody competition. In order to win (and it's a temporary victory), you have to shoot down spaceships or gobble up something or rescue creatures in peril. Meanwhile, someone or something is always after you, trying to destroy you.

Does this mean that a long afternoon at the *Asteroids* machine will make you want to inflict bodily harm on the first person who gives you a funny look? Some studies have shown that a person's blood pressure will rise and pulse quicken after playing video games. But can't the same thing happen when you're up to bat in the big softball game or trying to meet an impossible deadline at work or even watching a frightening movie?

Game As Villain

The 1969 rock opera *Tommy*, by The Who, is the story of a young deaf, dumb, and blind boy who is a champion at the pinball machines. He becomes a cult hero as a result of that and, after he regains his senses later in the story, is worshipped by devoted followers who try to emulate his pinball wizardry.

If *Tommy* were written today, we might be humming along to "Pac-Man Wizard," instead of "Pinball Wizard." Ever since the introduction of Atari's *Pong* game in 1972 and the ensuing evolution of the video arcade game, these high-tech pinball machines have been showing up in cameo roles in movies and television. And they're usually the bad guys.

In this year's *The Star Chamber*, lawyer Michael Douglas can't even get a "Hi, Dad" from the kids because they won't turn away from their home video game. A fight over an arcade game that causes television interference in a restaurant gets a young woman involved with a young boy who does nasty things to people he doesn't like in *Twilight Zone: The Movie*. And *WarGames* follows the activities of a teenager who almost instigates World War III by tapping into the national defense system with a home computer, a modem, and some big floppy disks. Worse than that, he's flunking biology.

It's not just the computers themselves that are shown in a less-than-positive light. The player's involvement with the computer or arcade game, as portrayed by movie makers, usually points out some kind of character flaw that is intensified by his obsession with these high-tech villains.

Movies may not be the best way to gauge a society's attitudes, but they often reflect sources of conflict which are easily identifiable. And video games certainly seem to be that right now. You might be hard pressed to find a young person who doesn't have an opinion about *Donkey Kong*, or who couldn't at least hum the theme song.

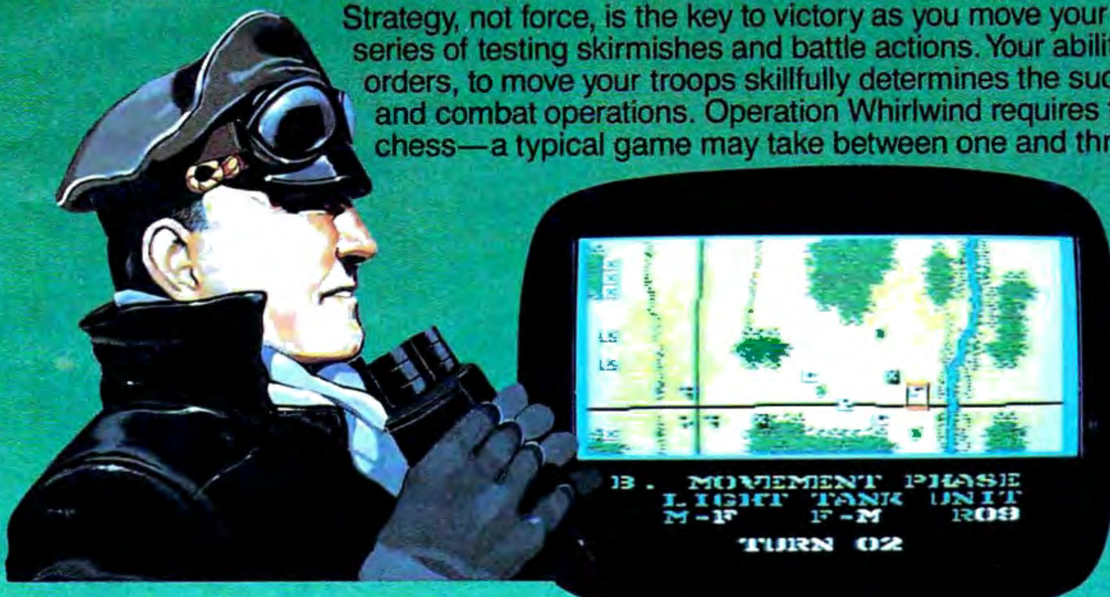
Teaching Disassociation

There does seem to be a degree of backlash against video games. Joseph Weizenbaum, author of *Computer Power and Human Reason* and Professor of

The Challenge:
Match Wits with the Mind Games from Brøderbund!

OPERATION WHIRLWIND™

Your mind is the ultimate weapon in this unique war game. Strategy, not force, is the key to victory as you move your battalion through a series of testing skirmishes and battle actions. Your ability to command, to give orders, to move your troops skillfully determines the success of your assault and combat operations. Operation Whirlwind requires the concentration of chess—a typical game may take between one and three hours—and the ability to master the many tactics of waging a full scale war. How well you can focus your strategic energies will decide whether your victory is Questionable, Marginal, Tactical, Strategic or a Breakthrough. A great game for the thinking game player. Available on disk.



MATCHBOXES™

Pit your concentration against another player or the computer. Here's a puzzle game that will keep the whole family bemused, befuddled and playing happily for hours. A grid of 36 numbered boxes conceals an animated menagerie of colorful characters, creatures and objects. The object of the game is to match identical squares and then solve a hidden word puzzle. There are many game variations with puzzles that are frontward, backward, or scrambled. Matchboxes. It's the most memorable memory game you've ever played! Available on disk and cassette.

OPERATION WHIRLWIND AND MATCHBOXES FOR THE ATARI® HOME COMPUTERS. ATARI® is a trademark of Atari, Inc.

Make sure to look for The Arcade Machine™ and Brøderbund's new game phenomenon, Lode Runner™—also for Atari.

Brøderbund Software
Discover the Difference
17 Paul Drive San Rafael, CA 94903

Introducing Snooper Troops™ detective series. Educational games that turn ordinary homes into Sherlock homes.

Where can you find educational computer games that your kids will really enjoy playing?

Elementary, my dear Watson. From Spinnaker.

Our SNOOPER TROOPS™ detective games are fun, exciting and challenging. And best of all, they have real educational value. So while your kids are having fun, they're learning.

As a Snooper Trooper your child will have a great time solving the mysteries. But it will take some daring detective work. They'll have to question suspects, talk to mysterious agents, and even search dark houses to uncover clues.

Luckily, the program provides your kids with everything they need: a SnoopMobile, a wrist radio, a SnoopNet computer, a camera for taking Snoopshots and even a notebook for keeping track of information.

SNOOPER TROOPS detective games help your children learn to take notes, draw maps, organize and classify information and they help develop vocabulary and reasoning skills. All while your kids are having a good time.

So if you want to find educational games that are really fun, here's a clue: Ask your local retailer for SNOOPER TROOPS computer games.*

Available in disks for IBM, Atari,* Apple,* Commodore 64.**

SPINNAKER™
We make learning fun.

Computer Science at the Massachusetts Institute of Technology, has an explanation for why the backlash exists. "The video arcade is the modern version of the pool hall. Some people are opposed to them for the same reasons they opposed pool halls. This reasoning is not relevant, and it masks other things that are much more important.

"It's just as Marshall McLuhan predicted: the next medium takes aspects of the previous medium. In this case, video games have taken the worst of television: its mindless violence, which is expressed in all the shoot-em-ups." Weizenbaum cites the television show "Knight Rider" as an example. "It's not that that one is exceptionally violent. It just exaggerates the cartoon-type violence."

Then why don't parents get as upset over cartoons as they do video games? Weizenbaum doesn't know. "It's the same thing you see during the week on regular TV shows. Only the television acts as babysitter on Saturday mornings," he says.

Some people claim that, even though video games may be as violent as television, they are more interactive. "The advertising claim for video games is that you can actually participate. But what is it that you're actually participating in? Killing. You can't win — all you can do is survive longer than anyone else."

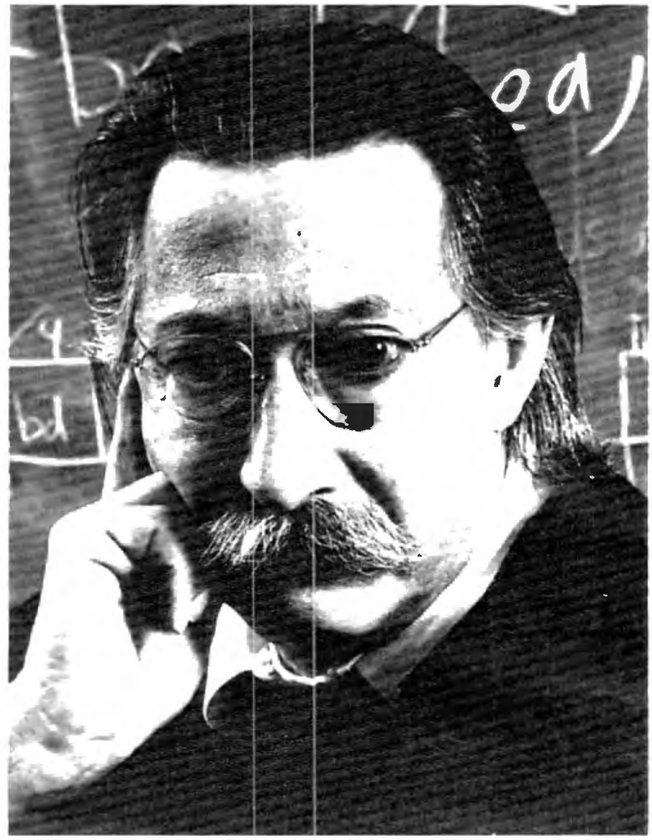
Weizenbaum's chief criticism is that what's being practiced in video games is disassociation. "Video games encourage you to believe that there is no relationship between what you are doing and the ultimate victim of that action. The crucial thing is that these are lessons in what it is necessary to do in order to survive in this society. In some sense, that's really the social purpose.

"It's like women working in a bomb factory. If they couldn't disassociate themselves from what they were doing, if they were really aware of what they were actually doing, they couldn't do it," says Weizenbaum. "The same thing applies to students and teachers who believe that artificial intelligence is possible. It's very necessary in this society to render a great many things abstract, to take them out of context."

Because of this, he believes, the video arcade is a "necessary and useful training ground. The video game is *not* the cause of this societal trait; it is a reflection of what our society is. It would be a mistake to yell and storm at the reflection."

Lack Of Creativity

Christopher Cerf has been entertaining children for a long time. He founded the nonbroadcast division of Sesame Street in 1970, and has written music and lyrics for the television show. Since the introduction of microcomputers, he has been developing ways of educating and entertaining kids with them; Cerf and Jim Henson of Muppet fame created the video game version of *The Dark Crystal* for Sierra On-Line.



Joseph Weizenbaum, author of Computer Power and Human Reason.

Cerf also developed the original concept of Sesame Place, parks near Dallas and Philadelphia which house computer centers where children can learn to use micros.

And he doesn't believe that kids are being deeply affected by the violence in video games. "I'm certainly not pro-violence," Cerf says. "I don't want to put it in games that I work on. But I think people greatly overestimate the horrible dangers of video games. Unless a child is greatly disturbed in some other way, I don't think he's going to go out and kill someone after playing a game of *Space Invaders*.

"I'm not denying that we don't all have some sort of aggressive instinct. Look at the way dogs will have mock fights — not really hurt each other, but just play. If the violence in a game is silly, it's just as good to play.

"Any medium that comes along has a reaction like this. Look at how horrified parents used to be that their children were wasting their quarters in movie theaters. And in the fifties, it was comic books. Doing anything in excess is a problem. You need to try to see it in perspective."

Cerf believes that resorting to extreme violence in a video game indicates a lack of creativity on the programmer's part. "I was appalled by the violence in *Death Race 2000*. In the last year or two, program-

mers have been designing games that are less violent and more creative. *Pac-Man* and *Frogger* are good examples. So are the new interactive fiction games."

"Software companies need to be innovative about the uses of computers for women. And that means producing something that appeals to what women traditionally have valued and needed. Not violence."

Mary Rowe, assistant to the president, M.I.T.

A Generation Of Loners?

Violence aside, some people argue that video games promote antisocial behavior. Maybe *Galaxia* won't make you want to shoot everything in sight, but how is a child or young adult going to learn how to interact with other people if he or she spends a great deal of time in an arcade or the house playing games on the home computer?

Christopher Cerf believes that computers foster, rather than hinder, communication. "Computers as a medium are one of the most exciting," he says. "They use elements of many other media.

"In schools, kids get excited about computing. They stay after school and compare notes and try to work out programming problems. And services like CompuServe and The Source also tend to bring people together. Kids who spend a lot of time alone with their computers or in arcades would probably be doing something else alone anyway.

"What's really interesting about this whole computer business is that, for the first time, the kids generally know more than the adults. My father was in publishing and he read everything — except science fiction. I loved science fiction and could recommend books to him. In that way, I think computers tend to bring families together."

Nothing For Women

There is little question that men are generally more interested in video games than women are at this point. *Pac-Man* was a breakthrough game in that sense; lots of women liked it, perhaps because of its apparent lack of violence.

Still, women are not leaping into the computer age with the same fervor as men seem to be. Mary Rowe, Assistant to the President at M.I.T., thinks that this is due to a lack of sensitivity on the part of many software producers. And to the fact that there is a lot of violence and sexism in video games.

"As a feminist, I'm concerned about the male slant of these things," Rowe says. "Why have computer companies made so few attempts to produce games that are not violent and sexist?

"Software companies need to be innovative about the uses of computers for women. And that means producing something that appeals to what women traditionally have valued and needed. Not violence."

Rowe does believe that some software companies are taking risks and developing programs that meet these needs. "I became computer-literate on Infocom's games. We need more games like that that require the player to actually think, not just hit the fire button at the right time."

Subtle Software

Nonviolent games fare very well on lists of best-selling software these days. Brøderbund's successful *Choplifter* is a good example. It's not an absolutely nonviolent game — there are terrorists and enemy tanks and guns going off. But the player does not get points for destroying things, only for rescuing people from the terrorists.

However, software companies which are producing nonviolent games are not necessarily trying

Christopher Cerf is currently working on a nonviolent video game, Pigs In Space.

to counteract any backlash against video games. Pat Marriot, of Electronic Arts, believes that people's opposition to video games is "an emotional thing. Parents wondering if their kids should be hanging out in arcades. *Donkey Kong* and *Pac-Man* are not really violent. It's just the environment of a video arcade that is disturbing.

"We look for quality and uniqueness in our programs," says Marriot. "We're not reacting against anything, we're going for quality. We look for authors whose values are consistent with those of the company. Each of our designers has a story to tell, and that story becomes the product.

"We don't really consciously try to make our games nonviolent, but because of our authors' basic philosophies, they usually do not involve violence," says Marriot. She points to *Hardhat Mack* as an example: "The character is very appealing. There's lots of humor in it. It seems to appeal to younger girls and to people who don't necessarily like games."

The Adventure Alternative

A video game doesn't have to have blasting guns and anguished screams to be violent. Even the pacifist Pac-Man has his own sublimated violence. He's a cute, nonthreatening little guy, but there are four potential killers on his trail. To avoid being destroyed, he must turn around and try to destroy them first.

It may be impossible to create a video game that does not incorporate some amount of violence, however unobtrusive it may be. Games involve competition. Even if you're just playing against yourself, you're always trying to overcome someone or something.

But in some games, you can actually benefit by resisting the urge to commit a violent act. In the text adventure *Witness*, by Infocom, you play a detective trying to solve a murder case. While you're trying to find the murderer, you have ample opportunity to rough up some of the suspects if you like. The game was designed to anticipate a variety of responses, even violent ones.

A violent response, though, is counterproductive, says Marc Blank, Vice-President of Product Development at Infocom. If a player reacts that way, the result is not good, and may lead to someone else getting killed.

Yet the designers at Infocom did not set out to produce games with pacifistic messages. "I don't think violence plays any part in our choices," says Blank. "We're not making a conscious effort to be nonviolent. We're just trying to produce programs of more literary quality."

This may be a contributing factor to Infocom's popularity with women, a market that software producers are sometimes finding difficult to please. "There is very little software for young women," says Blank. "Women generally read more than men, so our adventures are more appealing to them."

Better Technology?

Maybe the arcade is the monster, not the video game. According to an article in *Newsweek* (August 8, 1983), video games peaked with an average weekly earning of $140 per machine in 1981, but last year it was down to $109. Is this because people are playing games at home on their personal computers and don't need arcades anymore? Or is it a result of the backlash against video games?

It may be neither. *Dragon's Lair*, an arcade game recently released by Bluth Animation, has people lined up around the block in some cities, waiting for their turn to play. *Newsweek* says single machines featuring this game are taking in up to $1400 per week. Even at 50 cents a crack, that's about a 500% increase over the current average earnings of arcade games.

Dragon's Lair is anything but nonviolent. Its hero, Dirk the Daring, must battle countless foes in 38 different scenes in order to rescue the game's heroine, Daphne.

But what's attracting people to it is a new technology that combines the use of laser disks and computers. Unlike other arcade games, this one projects a movie-quality image. It's like stepping into a cartoon and controlling the characters yourself.

Dragon's Lair, *a popular new arcade game, combines laser disk technology and computers to create a movie-like image.* ©

dilithium Press is cooking again. With more than 32 practical, educational, and fun programs specifically for your microcomputer.

Here's what's on the menu:

Annual	*Hamcode*	*Roadrace*
Argo	*Integrate*	*Simeqn*
Arithmetic	*Jot*	*Sortlist*
Biorhythm	*Kaleido*	*Sparkle*
Birthday	*Loan*	*Squares*
Checkbook	*Metric*	*Stats*
Curve	*Mileage*	*Stopwatch*
Decide	*Numbers*	*Tachist*
Decode	*Obstacle*	*Tune*
Diffeqn	*Pi*	*Vocab*
Flashcard	*Powers*	*Walloons*
Graph	*Pythag*	*Wari*
Groan	*Quest/Exam*	

Written in easy-to-understand language, each book contains complete program listings plus suggested projects. For only $19.95, we serve up everything from math, to business, to games with color graphics. Plus, you'll receive generous portions of screen display examples to help guide you through each program.

And we have a blue plate special! Save your fingers from typing in the program by buying the diskette or cassette for only $15 more ($34.95 for the book/software package.) Fully-tested and machine-specific, our software is backed with a **forever** replacement guarantee. For questions or problems, call our toll-free customer service number.

You'll find dilithium Software at your nearest bookstore or computer store. Use our toll-free number for their locations or ask for the latest copy of our catalog, "Brainfood", absolutely free.

If your microcomputer is starving for new projects, call today for the smorgasbord from dilithium Software.

dilithium Software
P.O. Box E
Beaverton, Oregon 97075
800-547-1842 or
646-2713 inside Oregon

The Commodore 64, VIC 20 and IBM P.C. software smorgasbord.

How To Type COMPUTE!'s Programs

Many of the programs which are listed in **COMPUTE!** contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to tell exactly what to type when entering one of these programs into your computer, we have established the following listing conventions. There is a separate key for each computer. Refer to the appropriate tables when you come across an unusual symbol in a program listing. If you are unsure how to actually enter a control character, consult your computer's manuals.

Atari 400/800

Characters in inverse video will appear like: INVERSE VIDEO
Enter these characters with the Atari logo key, {⅄}.

When you see	Type	See	
(CLEAR)	ESC SHIFT <	↖	Clear Screen
(UP)	ESC CTRL -	↑	Cursor Up
(DOWN)	ESC CTRL =	↓	Cursor Down
(LEFT)	ESC CTRL +	←	Cursor Left
(RIGHT)	ESC CTRL *	→	Cursor Right
(BACK S)	ESC DELETE	◀	Backspace
(DELETE)	ESC CTRL DELETE		Delete character
(INSERT)	ESC CTRL INSERT		Insert character
(DEL LINE)	ESC SHIFT DELETE		Delete line
(INS LINE)	ESC SHIFT INSERT		Insert line
(TAB)	ESC TAB	▶	TAB key
(CLR TAB)	ESC CTRL TAB		Clear tab
(SET TAB)	ESC SHIFT TAB		Set tab stop
(BELL)	ESC CTRL 2		Ring buzzer
(ESC)	ESC ESC		ESCape key

Graphics characters, such as CTRL-T, the ball character • will appear as the "normal" letter enclosed in braces, e.g. {T}.

A series of identical control characters, such as 10 spaces, three cursor-lefts, or 20 CTRL-R's, will appear as {10 SPACES}, {3 LEFT}, {20 R}, etc. If the character in braces is in inverse video, that character or characters should be entered with the Atari logo key. For example, {♥} means to enter a reverse-field heart with CTRL-comma, {5U} means to enter five inverse-video CTRL-U's.

Commodore PET/CBM/VIC/64

Generally, any PET/CBM/VIC/64 program listings will contain words within braces which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S̲ would mean to type the S key while holding the shift key. If you find an underlined key enclosed in braces (e.g., {10 N̲}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's). Some graphics characters are inaccessible from the keyboard on CBM Business models (32N, 8032).

For the VIC and 64, if a key is enclosed in special brackets, ⟦ ⟧, you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as indicated.

Rarely, you'll see in a Commodore 64 program a solitary letter of the alphabet enclosed in braces. These characters can be entered by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A.

About the *quote mode*: you know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INSerT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following tables when entering special characters:

When You Read:	Press:	See:	When You Read:	Press:	See:
{BLK}	CTRL 1		⟦5⟧	C= 5	
{WHT}	CTRL 2		⟦6⟧	C= 6	
{RED}	CTRL 3		⟦7⟧	C= 7	
{CYN}	CTRL 4		⟦8⟧	C= 8	
{PUR}	CTRL 5		{F1}	f1	
{GRN}	CTRL 6		{F2}	f2	
{BLU}	CTRL 7		{F3}	f3	
{YEL}	CTRL 8		{F4}	f4	
⟦1⟧	C= 1		{F5}	f5	
⟦2⟧	C= 2		{F6}	f6	
⟦3⟧	C= 3		{F7}	f7	
⟦4⟧	C= 4		{F8}	f8	

All Commodore Machines

Clear Screen	{CLR}	Cursor Left	{LEFT}
Home Cursor	{HOME}	Insert Character	{INST}
Cursor Up	{UP}	Delete Character	{DEL}
Cursor Down	{DOWN}	Reverse Field On	{RVS}
Cursor Right	{RIGHT}	Reverse Field Off	{OFF}

Apple II / Apple II Plus

All programs are in Applesoft BASIC, unless otherwise stated. Control characters are printed as the "normal" character enclosed in brackets, such as {D} for CTRL-D. Hold down CTRL while pressing the control key. You will not see the special character on the screen.

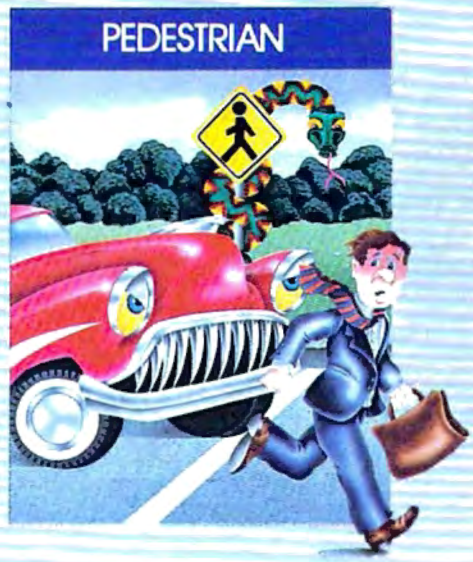
Texas Instruments 99/4

The only special characters used are in PRINT statements to indicate where two or more spaces should be left between words. For example, ENERGY {10 SPACES} MANAGEMENT means that ten spaces should be left between the words ENERGY and MANAGEMENT. Do not type in the braces or the words 10 SPACES. Enter all programs with the ALPHA LOCK on (in the down position). Release the ALPHA LOCK to enter lowercase text.

Newest VIC 20 and Commodore 64 Arcade Games

Will definitely be on top 10 chart!

Luna Software presents three new games for the Commodore '64 and VIC 20 computers: CARGO RUN, PEDESTRIAN, and HYPER-HEN. High resolution, full color graphics combined with state-of-the-art concepts and programming make Luna's complete library of arcade-style software the games of choice for the '80s.



An arcade-style, maze configured game, HYPER-HEN challenges even the most proficient joystick jockeys. Five levels of fast-paced play assure that your adrenalin level stays at its peak while starving coyotes, dive-bombing chicken-hawks, and deadly ZOMPIES attack from every side.

	LUNA	PROTECTO
Cassette VIC 20	$19.95	$14.95
Cassette CM64	$21.95	$16.95
Diskette CM64	$24.95	$19.95

With your Commodore '64 computer plotting the precise coordinates for this dangerous mission, prepare for the most engaging video battle of your life. Three separate and highly detailed scenarios mean 48K of memory is required to present CARGO RUN, the most sophisticated game now available for the Commodore '64.

	LUNA	PROTECTO
Diskette CM64	$31.95	$26.95

Keep your eyes wide open and your wits about you. As the PEDESTRIAN you'll have to cross the rush-hour freeway and race through the park full of muggers and monsters. If you're lucky you'll be able to keep your balance while jumping from log to log across the river to your home.

	LUNA	PROTECTO
Cassette VIC 20	$19.95	$14.95
Cassette CM 64	$21.95	$16.95
Diskette CM 64	$24.95	$19.95

Other LUNA products

		LUNA Suggested Price	PROTECTO'S Price
Galactic Crossfire	Vic 20 Cassette	$19.95	$16.95
Alien Soccer	Vic 20 Cassette	$19.95	$16.95
Warp Runner	Vic 20 Cassette	$19.95	$16.95
Cosmic Snake	Vic 20 Cassette	$19.95	$16.95
Starduster	Vic 20 Cassette	$19.95	$16.95

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COUPON FILE

Ken D. McCann and Dale McBane,
Technical Assistant

Not only will this program create easily accessible files for disk or tape storage, but also it will run on any Commodore machine, Atari, TI, Apple, and Color Computer. And with minor adjustments, you can file nearly anything.

This program allows you to file and search for coupons. Coupons may be located by brand name, product, or expiration date, and you may scan the contents of all the files. "Coupon File" could also help you file a great variety of things.

The use of DATA statements as file structures is one of the things that makes this program so versatile. Because the DATA statements are saved with the program, file retrieval is not a problem, even if you don't have disk capability.

The DATA statements were placed before the main program loop to simplify file insertions and deletions. Because the file number and line number for the DATA statements are the same, those of you unfamiliar with programming will find it easy to create files.

Using The Program On Your Computer

There is room for 499 files, assuming you have enough memory to store 499 files. Because of the search routine, each file must have the same format. This is especially true with the date search. 6/30/83, 6-30-83, and 30JUN83 are all different representations of the same date, but for the computer to locate that date, you must choose one format and be consistent.

Coupon File was written to run on any machine which supports BASIC, with one exception. Lines 501, 1000, 1550, 2000, 4000, 6000, and 7100 consist of the statement PRINT "{CLR}". This is COMPUTE!'s listing convention for clear screen on the Commodore 64 and VIC-20. You should substitute the statement to clear the screen on your particular machine (ESC SHIFT < for Atari, CALL CLEAR for TI, etc.).

To make more room for files, you can leave out the instructions. To do this, delete lines 550-555, 630, and 6000-7000, and change line 590 to:

```
590 IF (K$<>"L")*(K$<>"B")*(K$<>"P")*(K$<>"D")*
    (K$<>"C") THEN GOTO 570
```

This versatile program is very easy to use, and it's easy to adapt for other purposes. With a few changes, you can create a program to file nearly anything.

Special Note To Timex/Sinclair Users:

Because your machine's version of BASIC does not contain READ or DATA statements, this program will not run on your machine. You may be able to adapt it to your machine using strings for file storage.

Program Explanation

Lines	
1-499	DATA
500-650	main menu
1000-1700	list all routine
2000-3600	brand search routine
4000-5700	date search routine
6000-7000	instructions
7100-8610	product search routine
9000-9600	display routine
10000-10300	file input routine

Coupon File

```
1 DATA 1,OXYDOL,DETERGENT,FREE,NONE
2 DATA 2,PLANTERS,MIXED NUTS,25C,31MAR84
3 DATA 3,FREE N' SOFT,FABRIC SOFTNER,20C,
  1JAN84
4 DATA 4,JELLO,PUDDING,15C,15MAY84
5 DATA 5,JENOS,PIZZA,1$ R,4JUL84
6 DATA 6,CHINET,PLATES,20C,21DEC83
7 DATA 7,PEPPIS,PIZZA,60C,15MAY84
8 DATA 8,CHINET,CUPS,20C,20JUL83
9 DATA 9,NABISCO,SHREDDED WHEAT,20C,30JUN
  84
10 DATA 10,HEFTY,TRASH BAGS,25C,31OCT83
11 DATA 11,WHEATSWORTH,CRACKERS,12C,30JUN
   84
12 DATA 12,KRAFT,JELLY,10C,NONE
```

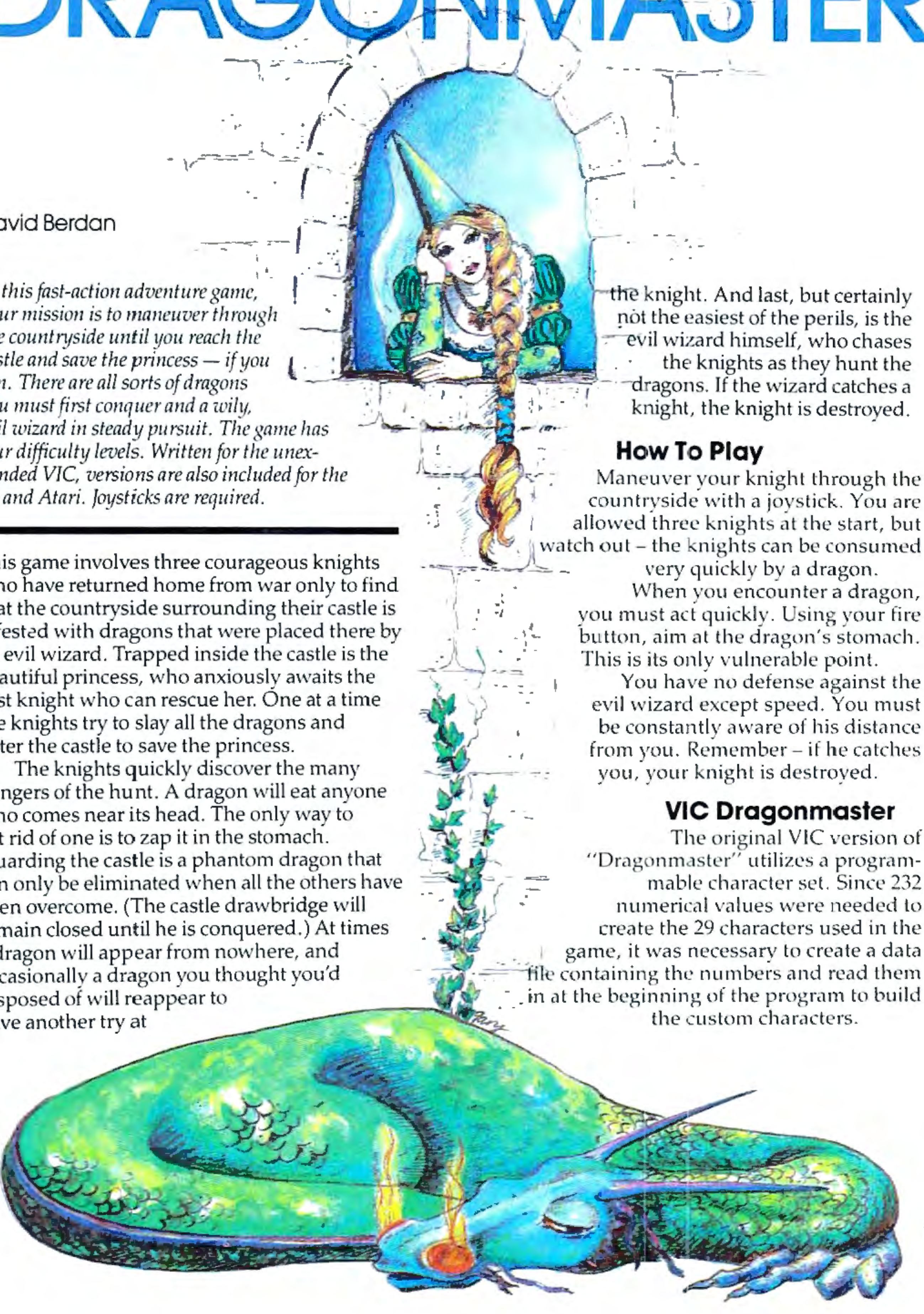
THIS PRINTER COST JUST $119.95*
Introducing the new Alphacom 42.
High quality dot matrix printer for
Commodore and Atari home computers.
Print program listings, charts, graphics, letters - most anything you create with your computer.
Full 40-column width.
Ultra quiet operation.
Fast -- two lines per second.
No maintenance -- 90 day limited warranty.
Complete with paper and instructions.
For the name of the dealer nearest you, call toll free, 800-538-7047
(in California 800-325-4395).
*Intelligent interface cable for Commodore or Atari sells separately.
Alphacom™

```
13 DATA 13,PHILADELPHIA,CREAM CHEESE,10C,
   NONE
14 DATA 14,PREGO,SPAGETTI SAUCE,20C,30JUN
   84
500 DATA END
501 PRINT "{CLR}":REM CLEAR SCREEN
502 PRINT "COUPON FILE"
503 PRINT
504 LET T=0
520 PRINT "<L>{2 SPACES}LIST ALL ENTRIES"
525 PRINT
530 PRINT "<B>{2 SPACES}BRAND NAME "
533 PRINT
535 PRINT "<P>{2 SPACES}PRODUCT"
537 PRINT
540 PRINT "<D>{2 SPACES}EXPIRATION DATE"
545 PRINT
550 PRINT "<H>{2 SPACES}INSTRUCTIONS"
555 PRINT
560 PRINT "<C>{2 SPACES}COMMAND MODE"
565 PRINT
570 PRINT "CHOICE ";
580 INPUT K$
590 IF (K$<>"L")*(K$<>"B")*(K$<>"P")*(K$<
    >"D")*(K$<>"H")*(K$<>"C")THEN GOTO 57
    0
600 IF K$="L" THEN GOSUB 1000
610 IF K$="B" THEN GOSUB 2000
615 IF K$="P" THEN GOSUB 7100
620 IF K$="D" THEN GOSUB 4000
630 IF K$="H" THEN GOSUB 6000
640 IF K$="C" THEN END
650 GOTO 501
1000 PRINT "{CLR}":REM CLEAR SCREEN
1010 PRINT "COUPON LIST"
1015 PRINT
1100 GOSUB 10000
1200 IF A$="END" THEN RETURN
1550 PRINT "{CLR}":REM CLEAR SCREEN
1600 GOSUB 9000
1700 GOTO 1000
2000 PRINT "{CLR}":REM CLEAR SCREEN
2200 PRINT "ENTER BRAND"
2300 INPUT J$
2400 PRINT
2410 PRINT "BRAND: ";J$
2420 PRINT
2430 PRINT
2500 GOSUB 10000
2600 IF A$="END" THEN GOTO 3300
3000 IF B$<>J$ THEN GOTO 2500
3100 GOSUB 9000
3150 LET T=1
3200 GOTO 2500
3300 IF T=1 THEN GOTO 3600
3350 PRINT J$;" NOT ON FILE"
3360 PRINT
3400 PRINT "HIT RETURN TO CONTINUE"
3450 PRINT
3500 INPUT K$
3600 RETURN
4000 PRINT "{CLR}":REM CLEAR SCREEN
4200 PRINT "ENTER EXPIRATION DATE"
4300 INPUT J$
4400 PRINT
4410 PRINT "DATE: ";J$
4420 PRINT
4430 PRINT
4500 GOSUB 10000
4600 IF A$="END" THEN GOTO 5300
5000 IF E$<>J$ THEN GOTO 4500
5100 GOSUB 9000
5150 LET T=1
5200 GOTO 4500
5300 IF T=1 THEN GOTO 5600
5350 PRINT "NONE EXPIRE ";J$
5360 PRINT
5400 PRINT "HIT RETURN TO CONTINUE"
5410 PRINT
5500 INPUT K$
5600 RESTORE
5700 RETURN
6000 PRINT "{CLR}":REM CLEAR SCREEN
6200 PRINT "TO ENTER A FILE, PLACE"
6350 PRINT "THE COMPUTER IN COM-"
6400 PRINT "MAND MODE (THE MODE"
6450 PRINT "BEFORE YOU TYPE RUN)."
6500 PRINT "TYPE IN THE LINE NUM-"
6550 PRINT "BER,' DATA ',THE FILE"
6600 PRINT "NUMBER, THE BRAND NAME"
6650 PRINT "THE PRODUCT, THE VALUE"
6700 PRINT "AND EXPIRATION DATE."
6710 PRINT
6720 PRINT
6800 PRINT "HIT RETURN TO CONTINUE"
6900 INPUT K$
7000 RETURN
7100 PRINT "{CLR}":REM CLEAR SCREEN
7200 PRINT "ENTER PRODUCT"
7300 INPUT J$
7400 PRINT
7410 PRINT "PRODUCT: ";J$
7420 PRINT
7430 PRINT
7500 GOSUB 10000
7600 IF A$="END" THEN GOTO 8300
8000 IF C$<>J$ THEN GOTO 7500
8100 GOSUB 9000
8150 LET T=1
8200 GOTO 7500
8300 IF T=1 THEN GOTO 8600
8350 PRINT J$;" NOT ON FILE"
8360 PRINT
8400 PRINT "HIT RETURN TO CONTINUE"
8410 PRINT
8500 INPUT K$
8600 RESTORE
8610 RETURN
9000 PRINT "FILE NUMBER>";A$
9010 PRINT
9100 PRINT "BRAND{6 SPACES}>";B$
9110 PRINT
9150 PRINT "PRODUCT{4 SPACES}>";C$
9160 PRINT
9200 PRINT "VALUE{6 SPACES}>";D$
9210 PRINT
9300 PRINT "EXP. DATE{2 SPACES}>";E$
9310 PRINT
9400 PRINT
9410 PRINT "HIT X TO ABORT TO MENU"
9420 PRINT "HIT RETURN TO CONTINUE"
9500 INPUT K$
9540 IF K$="X" THEN RESTORE
9550 IF K$="X" THEN GOTO 501
9600 RETURN
10000 READ A$
10100 IF A$="END" THEN RESTORE
10110 IF A$="END" THEN RETURN
10200 READ B$,C$,D$,E$
10300 RETURN
```
©

DRAGONMASTER

David Berdan

In this fast-action adventure game, your mission is to maneuver through the countryside until you reach the castle and save the princess — if you can. There are all sorts of dragons you must first conquer and a wily, evil wizard in steady pursuit. The game has four difficulty levels. Written for the unexpanded VIC, versions are also included for the 64 and Atari. Joysticks are required.



This game involves three courageous knights who have returned home from war only to find that the countryside surrounding their castle is infested with dragons that were placed there by an evil wizard. Trapped inside the castle is the beautiful princess, who anxiously awaits the first knight who can rescue her. One at a time the knights try to slay all the dragons and enter the castle to save the princess.

The knights quickly discover the many dangers of the hunt. A dragon will eat anyone who comes near its head. The only way to get rid of one is to zap it in the stomach. Guarding the castle is a phantom dragon that can only be eliminated when all the others have been overcome. (The castle drawbridge will remain closed until he is conquered.) At times a dragon will appear from nowhere, and occasionally a dragon you thought you'd disposed of will reappear to have another try at the knight. And last, but certainly not the easiest of the perils, is the evil wizard himself, who chases the knights as they hunt the dragons. If the wizard catches a knight, the knight is destroyed.

How To Play

Maneuver your knight through the countryside with a joystick. You are allowed three knights at the start, but watch out – the knights can be consumed very quickly by a dragon.

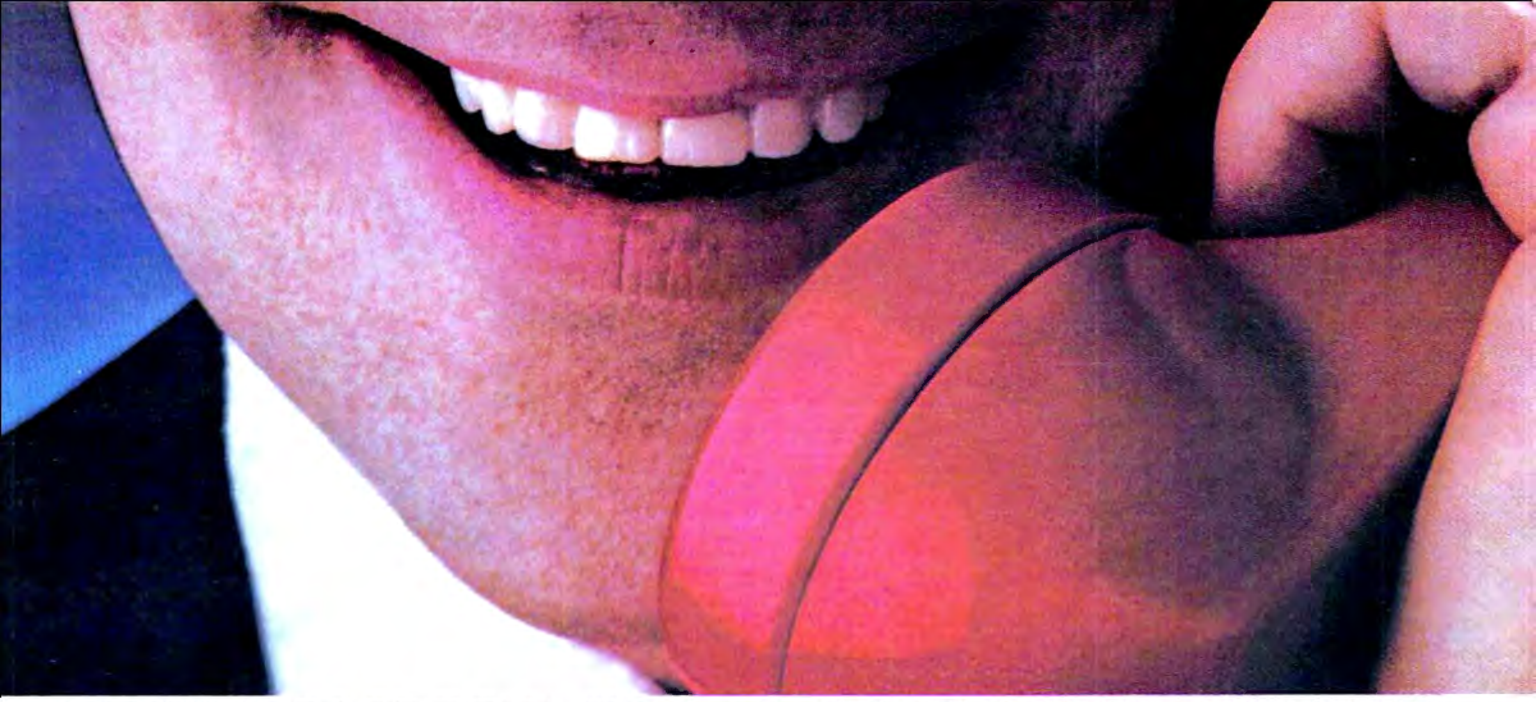
When you encounter a dragon, you must act quickly. Using your fire button, aim at the dragon's stomach. This is its only vulnerable point.

You have no defense against the evil wizard except speed. You must be constantly aware of his distance from you. Remember – if he catches you, your knight is destroyed.

VIC Dragonmaster

The original VIC version of "Dragonmaster" utilizes a programmable character set. Since 232 numerical values were needed to create the 29 characters used in the game, it was necessary to create a data file containing the numbers and read them in at the beginning of the program to build the custom characters.

Janice Fary



WHEN YOU BUY AN ATARI® COMPUTER, WE PUT OUR EXPERTISE ON THE LINE.

There's no limit to what you can do with a home computer...and no way any mere instruction manual can help you discover all the possibilities.

So ATARI gives you the extra help you need: an ATARI computer expert to answer your questions. Free. He'll help you write your own programs, learn how to do new things, and diagnose problems when the things you're trying to do just don't work out.

It's the ATARI Help Line. A toll-free help-and-information service to help you get more out of your ATARI Computer. Just call 1-800-538-8543.*

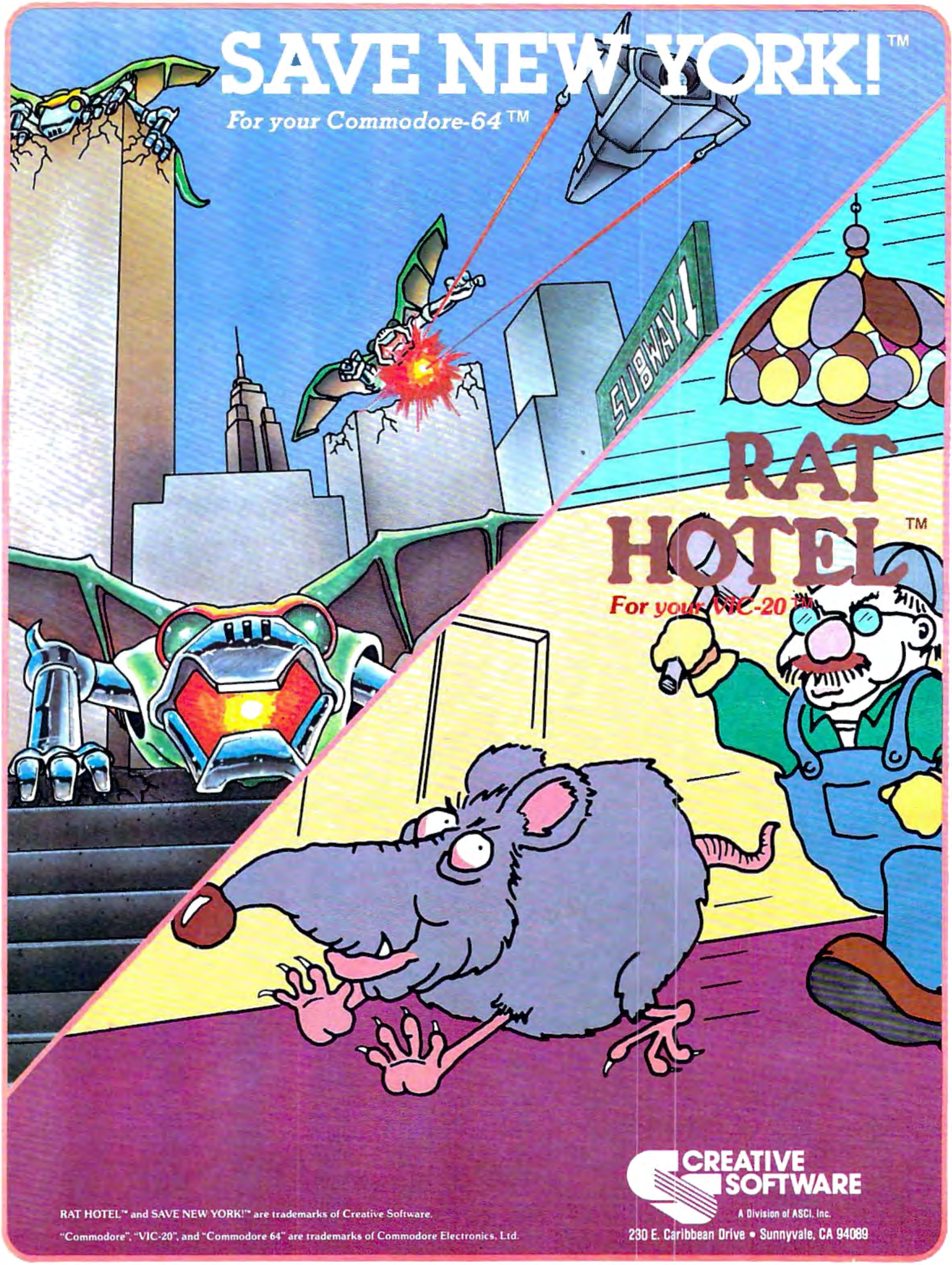
And if you ever need anything fixed, ATARI has over 1,600 ATARI SERVICE℠ Centers nationwide. You'll find the nearest one listed under "Computers" in your Yellow Pages.

ATARI SERVICE isn't the only good reason to buy an ATARI system. But it's an awfully good reason not to choose any other kind.

ATARI SERVICE
FACTORY AUTHORIZED NETWORK

1-800-538-8543*
THE ATARI SERVICE℠ "HELP LINE."

*California: 1-800-672-1404


SAVE NEW YORK!™
For your Commodore-64™
SUBWAY
RAT HOTEL™
For your VIC-20™
CREATIVE SOFTWARE
A Division of ASCI, Inc.
230 E. Caribbean Drive • Sunnyvale, CA 94089
RAT HOTEL™ and SAVE NEW YORK!™ are trademarks of Creative Software.
"Commodore", "VIC-20", and "Commodore 64" are trademarks of Commodore Electronics, Ltd.

Program 1 is the main game program. Program 2 is the data file needed to establish the character set used in the game. (Please note that due to the need to protect a block of memory for the character set, the VIC version will run only on the unexpanded VIC. Please remove any RAM expansion cartridges.)

Carefully type in Program 1, check it for accuracy, and SAVE it on tape. If you try to RUN Program 1 at this time, it will want to input the data from the data file which has not yet been made. Next, type NEW and then type in Program 2. Be careful while typing in the numbers here, as any slip up will result in ill-formed characters. When you are certain that everything is right, RUN Program 2. Your VIC will ask you to press Record and Play before it starts writing the data on the tape immediately following the copy of Program 1. After a minute or so the screen will display DATATAPE CREATED and the tape will stop. It would be wise to save a copy of Program 2 a little farther down on the tape so that if you need it again you won't have to retype all those numbers.

Rewind the tape to the beginning of Program 1 and type LOAD. When the program has been read in, type RUN.

Difficulty Levels

The first thing to appear on the screen is the instructions. You are asked to choose any of four skill levels by pressing the appropriate function key. The degree of difficulty is determined by how fast the wizard pursues the knight. The EASY level (F1) presents a real challenge to the newcomer. You can expect to be defeated quite often while you are learning the finer points of the game. The next levels, HARD (F3) and VERY HARD (F5), are progressively difficult. The IMPOSSIBLE level (F7) is only for the expert. You must be exceptionally alert and skillful with the joystick to get into the castle at this level.

If you'd rather not type in this program, I'll make copies (VIC version only). Just send a blank cassette, self-addressed stamped mailer, and $3 to:

David Berdan
31920 N.E. Big Rock Road
Duvall, WA 98019

BEFORE TYPING...

If you're new to computing, please read "How To Type COMPUTE!'s Programs" and "A Beginner's Guide To Typing In Programs."

Program 1: Dragonmaster – Main Program (unexpanded VIC)

```
100 PRINT"{CLR}":IFPEEK(7448)=60ANDPEEK(7
    679)=160THEN125
105 PRINT"DATA BEING READ IN"
110 OPEN1,1,0,"DATATAPE"
115 FORD=7448TO7679:INPUT#1,A:POKED,A:NEX
    T
120 CLOSE1:POKE56,29
125 CLR:PRINT"{CLR}{BLK}":U=36878:DR=15:K
    =2:JS=37154:P=7712:SW=0:S1=U-2:GOSUB2
    00:GOSUB280
130 ZZ=7912:POKEZZ,35
135 POKEJS,127:V=PEEK(JS-2)AND128
140 E=-(V=0):POKEJS-2,255:V=PEEK(JS-3)
145 S=-((VAND8)=0):W=-((VAND16)=0):N=-((V
    AND4)=0):SW=-((VAND32)=0):Q=P
150 IFSANDPEEK(P+22)=32THENP=P+22
155 IFWANDPEEK(P-1)=32THENP=P-1
160 IFNANDPEEK(P-22)=32THENP=P-22
165 IFEANDPEEK(P+1)=32THENP=P+1
170 POKEP+30720,0
175 GOSUB420:IFQ<>PTHENPOKEQ,32:POKEP,53
180 IFSWTHENGOSUB375
185 IFPEEK(P+1)=37ORPEEK(P+1)=60ORPEEK(P-
    1)=42THENGOSUB330
190 IFPEEK(P+22)=37ORPEEK(P+22)=60ORPEEK(
    P+22)=42THENGOSUB330
195 GOSUB470:GOTO135
200 PRINT"{DOWN}{5 SPACES}DRAGONMASTER"
205 PRINT"{DOWN}OBJECT: BLAST ALL":PRINT"
    {RIGHT}DRAGONS AND ENTER{5 RIGHT}CAST
    LE TO MARRY
210 PRINT"PRINCESS.
215 PRINT"{DOWN}ZAP DRAGONS FROM THE
    {2 SPACES}FRONT WITH FIRE BUTTON
220 PRINT"{DOWN}MUST DEFEAT PHANTOM":PRIN
    T"DRAGON LAST
225 PRINT:PRINT"PRESS: F1-EASY":PRINTTAB(
    7)"F3-HARD"
230 PRINTTAB(7)"F5-VERY HARD":PRINTTAB(7)
    "F7-IMPOSSIBLE"
235 SL=15:GETA$:IFA$=""THEN235
240 IFA$<CHR$(134)ORA$>CHR$(136)THENPRINT
    "{CLR}":RETURN
245 IFA$=CHR$(134)THENSL=10:PRINT"{CLR}":
    RETURN
250 IFA$=CHR$(135)THENSL=6:PRINT"{CLR}":R
    ETURN
255 SL=3:PRINT"{CLR}":RETURN
260 CL=INT(RND(1)*2)*2
265 CS=L+30720:POKECS,CL:POKECS+1,CL:POKE
    CS+22,CL:POKECS+23,CL
270 POKEL,M:POKEL+1,M+1:POKEL+22,M+2:POKE
    L+23,M+3:RETURN
275 POKEL,32:POKEL+1,32:POKEL+22,32:POKEL
    +23,32:RETURN
280 FORI=1TO126:PRINT"{4 SPACES}";:NEXT:P
    RINT" {HOME}":POKE38905,0
285 POKEU+1,30:FORC=7424TO7431:POKEC,0:NE
    XT:POKEU-9,255
290 L=7888:M=45:CL=7:GOSUB265:POKE38629,C
    L:POKE7909,49:CL=4:L=L-3:M=60:GOSUB26
    5:F=36
295 FORX=7680TO7701:POKEX,F:NEXT:FORX=770
    2TO8142STEP22:POKEX,F:POKEX+21,F:NEXT
300 FORX=8164TO8185:POKEX,F:NEXT:FORX=1TO
    15:Z=INT(RND(1)*2)
305 READA:M=41:L=7726+A:IFZTHENM=37
310 GOSUB260:NEXT:POKEU,15:POKE7712,53:PO
    KE7693,53:POKE7694,53
315 DATA 1,15,31,69,101,111,199,206,212,2
    68,275,316,353,362,367
320 FORX=49TO52:POKE7909,X:POKES1,(X-44)*
    28:FORT=1TO1000:NEXT:NEXT:GOSUB325:RE
    TURN
```


WE INTERRUPT THIS AD TO BRING YOU THIS IMPORTANT PROGRAM...
Now you could add **your** program to the rapidly expanding list of MMG Software with MMG's **Write Your Own Program Contest!** Each month a program will be chosen and developed from the entries and suggestions sent in by **you** and best of all, the winners will receive royalties from the sale of their entry. You don't have to be a computer whiz to enter, so see your local retailer who carries ATARI products for details. FINAL FLIGHT is only one of the many exciting and useful programs for the ATARI system now available from MMG Micro Software that are designed with you in mind. At MMG, we believe that people are as important as the programs they use.
FINAL FLIGHT!
MMG
MMG
micro software
P.O. Box 131 Marlboro, NJ 07746
(201) 431-3472
©ZADCO 1983
ATARI is a registered trademark of ATARI, Inc.

```
325 POKEU,0:POKES1,0:RETURN
330 POKEU,15:FORF=1TO2:FORX=130TO200:POKES1,X:NEXT
335 FORX=200TO130STEP-1:POKES1,X:NEXT:POKEP,F+53:NEXT:GOSUB325
340 POKE7695-K,36:K=K-1:IFK=-1THEN455
345 POKEP,32:P=7712:POKEP,53:RETURN
350 L=P-21:M=56:CL=0:GOSUB265:GOSUB400:L=P-21:GOSUB275:POKEU,15
355 FORX=52TO49STEP-1:POKE7909,X:POKES1,(X-44)*28:FORT=1TO1000:NEXT:NEXT:GOSUB325
360 FORX=7907TO7909:POKEX-1,32:POKEX+30720,0:POKEX,53:FORT=1TO1000:NEXT:NEXT:POKEX,32
365 POKEU+1,59:POKEU-9,240:PRINT"{CLR}":POKE214,22:PRINT:PRINT"{4 RIGHT}AND THEY LIVED"
370 PRINT"{DOWN}{2 RIGHT}HAPPILY EVER AFTER":GOTO460
375 CL=0:X=PEEK(P+1):Y=PEEK(P-1):IFX=62ANDDR=0THEN350
380 IFX<>39ANDY<>44THENRETURN
385 IFX=39THENL=P-21:M=56:GOSUB265
390 IFY=44THENL=P-24:M=56:GOSUB265
395 DR=DR-1
400 POKEU,15:FORX=250TO128STEP-1:POKES1,X:NEXT:GOSUB325
405 IFPEEK(P+1)=58THENL=P-21:GOSUB275
410 IFPEEK(P-1)=59THENL=P-24:GOSUB275
415 RETURN
420 H=INT(RND(1)*414)+7726:BB=INT(RND(1)*2):X=PEEK(H)
425 IFX=37THENL=H:M=41:GOSUB260:RETURN
430 IFX=41THENL=H:M=37:GOSUB260:RETURN
435 IFH<>7731ANDH<>7822ANDH<>7874ANDH<>8005ANDH<>8106THENRETURN
440 IFX<>32ORPEEK(H+1)<>32ORPEEK(H+22)<>32ORPEEK(H+23)<>32THENRETURN
445 IFBB=1THENL=H:M=37:DR=DR+1:GOSUB265:RETURN
450 L=H:M=41:DR=DR+1:GOSUB265:RETURN
455 POKEU-9,240:POKEU+1,59:PRINT"{CLR}":POKE214,22:PRINT:PRINT"{3 RIGHT}THE DRAGONS WIN"
460 FORX=1TO23:PRINT:FORT=1TO150:NEXT:NEXT
465 POKEU+1,27:GOTO125
470 CW=(CW+1)AND7:IFCW=1THENCW=2
475 POKEZZ+30720,CW:CC=CC+1:IFCC=SLTHEN485
480 RETURN
485 IZ=INT((ZZ-7680)/22):IP=INT((P-7680)/22):NZ=INT((IZ+IP)/2)*22
490 NZ=NZ+(P-IP*22+ZZ-IZ*22)/2:CC=1
495 IFPEEK(NZ)=32THENPOKEZZ,32:POKENZ,35:GOTO510
500 IFPEEK(NZ)=53THENGOSUB330
505 RETURN
510 ZZ=NZ:IFPEEK(ZZ+1)=53ORPEEK(ZZ-1)=53ORPEEK(ZZ+22)=53ORPEEK(ZZ-22)=53THENGOSUB330
515 RETURN
```

Program 2: Dragonmaster – Data File (for VIC)

```
1 OPEN1,1,2,"DATATAPE"
2 READX:PRINT#1,X:IFX=-1THEN4
3 GOTO2
4 CLOSE1:PRINT"DATATAPE CREATED"
5 END
10 DATA60,66,165,129,153,165,66,60,170,85,170,85,170,85,170,85,0,56,108,254,15
20 DATA15,31,31,0,0,0,0,1,6,12,152,255,63,63,31,31,15,31,61,176,176,152,216,220
30 DATA252,248,224,0,0,0,0,128,96,48,25,0,28,54,127,240,240,248,248,13,13,25,27
40 DATA59,63,31,7,255,252,252,248,248,240,248,188,0,0,84,124,108,56,56,189
50 DATA21,31,27,31,14,14,14,94,255,255,255,255,170,255,255,0,254,254,254,254,170
60 DATA254,254,0,0,0,0,0,0,255,0,0,0,0,32,24,6,1,0,0,0,16,8,4,2,1,0,0,8
70 DATA4,4,2,2,1,0,24,126,90,219,24,36,66,195,0,24,60,90,24,36,0,0,0,0,0
80 DATA24,24,0,0,0,16,0,2,72,2,21,75,21,130,16,64,162,72,160,90,224,7,42
90 DATA133,18,5,32,4,0,168,210,168,64,18,64,8,32,0,40,68,82,5,10,21,10,0,0,0,0
100 DATA1,2,4,16,85,42,21,10,21,10,21,40,128,144,0,144,72,164,80,160,-1
```

A dragon turns to dust in "Dragonmaster," VIC version.

Program 3: Dragonmaster – 64 Version

Translation by Chris Metcalf, Programming Assistant

```
1000 PRINT"{CLR}{7}":POKE53280,14:POKE53281,6
1010 IFPEEK(14846)=80ANDPEEK(14847)=160THEN1070
1020 PRINTTAB(14)"{8 DOWN}PLEASE WAIT":PRINTTAB(11)"DATA BEING READ IN"
1030 FORI=1TO24:READA:NEXT
1040 FORD=14616TO14847:READA:POKED,A:NEXT:POKE52,57:POKE56,57
1050 :
1060 :
1070 REM RESTART PROGRAM
1080 CLR:PRINT"{CLR}{7}":DR=24:K=2:P=1084:S1=54276:S2=54283:GOSUB1260:GOSUB1490
1090 POKES2-1,3:POKES2+1,8:POKES2+2,0
1100 ZZ=1446:POKEZZ,35:GR(0)=1:GR(1)=12:GR(2)=11:GR(3)=0:GR(4)=11:GR(5)=12
1110 :
1120 V=31-PEEK(56320)AND31:Q=P
1130 IFVAND2ANDPEEK(P+40)=32THENP=P+40
1140 IFVAND4ANDPEEK(P-1)=32THENP=P-1
1150 IFVAND1ANDPEEK(P-40)=32THENP=P-40
1160 IFVAND8ANDPEEK(P+1)=32THENP=P+1
```

TIME RUNNER™

WARNING !!

This game requires strategy as well as fast reflexes.

AVAILABLE FOR

ATARI 400/600/800/
1200/1400/1450
16K - CASSETTE
16K - DISK
$29.95
BY: Y. LEMPEREUR

COMMODORE 64
CASSETTE
DISK
$29.95
BY: S. MAXWELL and
T. LYNDON

TRS-80 MOD I/III/4
16K - CASSETTE
32K - DISK
$24.95
BY: Y. LEMPEREUR

Games are 100 % machine language for super-fast action.

All versions are joystick compatible. Atari version requires a joystick.

Join the 1st intergalactic SPACE RUSH. Newly discovered territory is yours for the taking! All that you have to do is to stake it out. Sounds easy? Well, there are the defenders who will block your moves and team up to get you. Also, staking out hyperspace is not so easy. Its value decreases with each addititonal moment you take to capture it. It's a battle requiring fast action as well as winning strategy! Take the challenge. Rush for it!

"An excellent and innovative game." SOFTSIDE MAGAZINE

If unavailable from your local dealer, add $2.00 to price for direct order

FUNSOFT

Dealer inquiries invited.

28611 Canwood St., Agoura, CA 91301
(213) 991-6540

"Atari", "Commodore", and "TRS-80" are trademarks of Atari Inc., Commodore Inc., and Tandy Corp. respectively.

```
1170 POKES2,8:POKEP+54272,GR(GR):GR=GR+1:
     IFGR=6THENGR=0
1180 GOSUB1930:IFQ<>PTHENPOKEQ,32:POKEP,5
     3:POKES2,65:POKES2-3,3
1190 IFVAND16THENGOSUB1800
1200 IFPEEK(P+1)=37ORPEEK(P+1)=60ORPEEK(P
     -1)=42THENGOSUB1640
1210 IFPEEK(P+40)=37ORPEEK(P+40)=60ORPEEK
     (P+40)=42THENGOSUB1640
1220 GOSUB2090:GOTO1120
1230 :
1240 :
1250 REM INSTRUCTIONS AND SKILL LEVEL
1260 PRINTTAB(8)"*** 64 DRAGONMASTER ***"
1270 PRINT"{2 DOWN} OBJECT: BLAST ALL DRA
     GONS AND ENTER"
1280 PRINT"{2 SPACES}THE CASTLE TO MARRY
     {SPACE}THE PRINCESS."
1290 PRINT"{2 DOWN} ZAP THE DRAGONS FROM
     {SPACE}THE FRONT WITH"
1300 PRINT "{2 SPACES}THE FIRE BUTTON, BU
     T STAY AWAY":PRINT"{2 SPACES}FROM TH
     EIR HEADS!"
1310 PRINT"{2 DOWN} YOU MUST ELIMINATE PH
     ANTOM DRAGON LAST"
1320 PRINT"{2 DOWN}{2 SPACES}PRESS: F1 --
      EASY":PRINTTAB(9)"{DOWN}F3 -- STAND
     ARD"
1330 PRINTTAB(9)"{DOWN}F5 -- VERY HARD":P
     RINTTAB(9)"{DOWN}F7 -- EXTREMELY HAR
     D"
1340 SL=15:GETA$:IFA$=""THEN1340
1350 IFA$<CHR$(134)ORA$>CHR$(136)THENPRIN
     T"{CLR}":RETURN
1360 IFA$="{F3}"THENSL=10:PRINT"{CLR}":RE
     TURN
1370 IFA$="{F5}"THENSL=6:PRINT"{CLR}":RET
     URN
1380 SL=3:PRINT"{CLR}":RETURN
1390 :
1400 :
1410 REM DRAW AND ERASE DRAGONS, ETC.
1420 CL=INT(RND(1)*2)*2
1430 CS=L+54272:POKECS,CL:POKECS+1,CL:POK
     ECS+40,CL:POKECS+41,CL
1440 POKEL,M:POKEL+1,M+1:POKEL+40,M+2:POK
     EL+41,M+3:RETURN
1450 POKEL,32:POKEL+1,32:POKEL+40,32:POKE
     L+41,32:RETURN
1460 :
1470 :
1480 REM INITIALIZE PLAYFIELD, DRAGONS
1490 POKE53265,0:POKE53280,6
1500 POKE53281,0:PRINT"{CLR}":POKE53281,1
     5:FORC=14592TO14599:POKEC,0:NEXT:CL=
     11
1510 L=1404:M=45:GOSUB1430:POKE1443,49:L=
     L-3:M=60:GOSUB1430:F=36
1520 FORX=1024TO1063:POKEX,F:NEXT:FORX=10
     64TO1944STEP40:POKEX,F:POKEX+39,F:NE
     XT
1530 FORX=1984TO2023:POKEX,F:NEXT:FORX=1T
     O24:Z=INT(RND(1)*2)
1540 READA:M=41:L=1106+A:IFZTHENM=37
1550 GOSUB1420:NEXT:POKE53272,31:POKE1084
     ,53:POKE1047,53:POKE1048,53
1560 DATA 1,30,65,82,91,138,203,222,234,3
     01,329,345,401,412,424,456,536
1570 DATA 550,567,632,690,706,724,734
1580 POKE54296,15:POKE54277,138:POKE54278
     ,0:POKE53265,27
1590 FORX=49TO52:POKE1443,X:POKE54273,(X-
```

Dragonmaster For The 64 And Atari

Chris Metcalf, Programming Assistant

The Commodore 64 version of "Dragonmaster" runs much the same as the VIC version. However, the colors are somewhat different. You will find that you are a flashing figure in shades of gray, a feature which could not be implemented on the VIC due to the memory shortage. The pursuing wizard flashes all the colors of the rainbow (magic!) and, by his mystic powers, pursues you by always halving the distance between you. If at any point his leap brings him to within one square of you, you're finished.

The dragons are the main obstacle in your chivalrous adventure. To eliminate a dragon, you have to move directly in front of his stomach and press the fire button. The dragon will disappear. The phantom dragon guarding the gates must be dispatched last. You will find that the dragons do move about and change color now and then, but this is only restlessness. In addition, a dragon will occasionally appear from nowhere to test your mettle.

The programming techniques involved in this version, as in the VIC version, are fairly straightforward. Programmable characters are used for dragons, castle, walls, knights, wizard, etc. The data for these characters is stored from 14336 to 16384, although the program uses only a part of this. The characters are put on the screen by POKEs, as are their colors (the color screen on the 64 is from 55296 to 56295). Simple sounds are created using two voices of the SID chip. Voice one controls dragon and the knights elimination as well as drawbridge noises; voice two maintains the movement noise.

The Atari version also employs programmable characters (the same data, in fact), located in the block of 512 bytes below the top of your available memory. The colors used are somewhat different. The castle and outside walls, the dragons, the wizard, and the knight each use different color registers. Graphics mode 1 is used to simulate the VIC's 22-column display as closely as possible. The sound used is simple Atari sound. However, the program is basically the same as the VIC version in terms of play.

```
     48)*1Ø:POKES1,8:POKES1,17
1600 FORT=1TO800:NEXT:NEXT:GOSUB1680:RETU
     RN
1610 :
1620 :
1630 REM PHANTOM DRAGON DIES, VICTORY
1640 FORF=1TO2:POKES1,8:POKES1,33:FORX=ØT
     O40STEP.8:POKE54273,X:NEXT
1650 POKES1,8:POKES1,33:FORX=40TO0STEP-.8
     :POKE54273,X:NEXT
1660 POKEP,F+53:NEXT:POKES1,8
1670 POKE1049-K,36:K=K-1:IFK<ØTHEN2030
1680 POKEP,32:P=1084:POKEP,53:RETURN
1690 L=P-39:CL=1:M=56:GOSUB1430:GOSUB1860
     :L=P-39:GOSUB1450:POKE53272,31
1700 POKE54277,138:POKE54278,Ø
1710 FORX=52TO49STEP-1:POKE1443,X:POKES1,
     8:POKES1,33:POKE54273,(X-48)*1Ø
1720 FORT=1TO800:NEXT:NEXT:POKES1,8
1730 FORX=1441TO1443:POKEX-1,32:POKEX+542
     72,1:POKEX,53:FORT=1TO800:NEXT:NEXT
1740 PRINT"{CLR}":POKE53280,7:POKE53281,7
1750 POKE53272,21:PRINT"{CLR}{2}"TAB(13
     )"{22 DOWN}AND THEY LIVED"
1760 PRINTTAB(11)"{DOWN}HAPPILY EVER AFTE
     R":GOTO2050
1770 :
1780 :
1790 REM ELIMINATE A DRAGON
1800 CL=Ø:X=PEEK(P+1):Y=PEEK(P-1):IFX=62A
     NDDR=ØTHEN1690
1810 IFX<>39ANDY<>44THENRETURN
1820 IFX=39THENL=P-39:M=56:GOSUB1430
1830 IFY=44THENL=P-42:M=56:GOSUB1430
1840 DR=DR-1
1850 :
1860 POKES1,8:POKES1,129:FORX=20TOØSTEP-.
     25:POKE54273,X:NEXT:POKES1,8
1870 IFPEEK(P+1)=58THENL=P-39:GOSUB1450
1880 IFPEEK(P-1)=59THENL=P-42:GOSUB1450
1890 RETURN
1900 :
1910 :
1920 REM TURN DRAGON / CREATE A NEW ONE
1930 H=INT(RND(1)*835)+1106:L=H:BB=INT(RN
     D(1)*2):X=PEEK(H)
1940 IFX=37THENM=41:GOSUB1420:RETURN
1950 IFX=41THENM=37:GOSUB1420:RETURN
1960 IFRND(1)>.Ø33THENRETURN
1970 FORI=-80TO120STEP40:FORJ=-2TO3:IFPEE
     K(H+I+J)<>32THENRETURN
1980 NEXT:NEXT:M=41:DR=DR+1:IFBB=1THENM=3
     7
1990 GOSUB1420:RETURN
2000 :
2010 :
2020 REM THE DRAGONS HAVE WON
2030 PRINT"{CLR}":POKE53280,2:POKE53281,2
2040 POKE53272,21:PRINT"{CLR}{3}"TAB(13
     )"{24 DOWN}THE DRAGONS WIN"
2050 FORX=1TO23:PRINT:FORT=1TO150:NEXT:NE
     XT:POKE53280,14:POKE53281,6:GOTO1080
2060 :
2070 :
2080 REM THE EVIL WIZARD MOVES
2090 CW=(CW+1)AND15:IFCW=15THENCW=Ø
2100 CC=CC+1:POKEZZ+54272,CW:IFCC<>SLTHEN
     RETURN
2110 IZ=INT((ZZ-1024)/40):IP=INT((P-1024)
     /40):NZ=INT((IZ+IP)/2)*40
2120 NZ=NZ+(P-IP*40+ZZ-IZ*40)/2:CC=1
2130 IFPEEK(NZ)<>32THENRETURN
```

A knight zaps a dragon in "Dragonmaster," 64 version.

```
2140 POKEZZ,32:POKENZ,35:ZZ=NZ:POKEZZ+542
     72,CW
2150 ZZ=NZ:IFPEEK(ZZ+1)=53ORPEEK(ZZ-1)=53
     ORPEEK(ZZ+40)=53THENGOSUB1640
2160 IFZZ>1103ANDPEEK(ZZ-40)=53THENGOSUB1
     640
2170 RETURN
2180 :
2190 :
2200 DATA60,66,165,129,153,165,66,60,170,
     85,170,85,170,85,170,85
2210 DATAØ,56,108,254,15,15,31,31,Ø,Ø,Ø,Ø
     ,1,6,12,152
2220 DATA255,63,63,31,31,15,31,61,176,176
     ,152,216,220,252,248,224
2230 DATAØ,Ø,Ø,Ø,128,96,48,25,Ø,28,54,127
     ,240,240,248,248
2240 DATA13,13,25,27,59,63,31,7,255,252,2
     52,248,248,240,248,188
2250 DATAØ,Ø,84,124,108,56,56,189,21,31,2
     7,31,14,14,14,94
2260 DATA255,255,255,255,170,255,255,Ø,25
     4,254,254,254,170,254,254,Ø
2270 DATAØ,Ø,Ø,Ø,Ø,Ø,255,Ø,Ø,Ø,Ø,32,24,6,
     1,Ø
2280 DATAØ,Ø,16,8,4,2,1,Ø,Ø,8,4,4,2,2,1,Ø
2290 DATA24,126,90,219,24,36,66,195,Ø,24,
     60,90,24,36,Ø,Ø
2300 DATAØ,Ø,Ø,24,24,Ø,Ø,Ø,16,Ø,2,72,2,21
     ,75,21
2310 DATA130,16,64,162,72,160,90,224,7,42
     ,133,18,5,32,4,Ø
2320 DATA168,210,168,64,18,64,8,32,Ø,40,6
     8,82,5,10,21,10
2330 DATAØ,Ø,Ø,Ø,1,2,4,16,85,42,21,10,21,
     10,21,40
2340 DATA128,144,Ø,144,72,164,80,160
```

Program 4: Dragonmaster – Atari Version

Translation by Chris Metcalf, Programming Assistant

```
1000 GOSUB 2270:POKE 77,Ø:POKE 752,1
     :SC=PEEK(88)+PEEK(89)*256
1010 CH=PEEK(742)*256-1024
1020 IF PEEK(CH+24)=6Ø AND PEEK(CH+2
     55)=16Ø THEN 1Ø6Ø
1030 POSITION 4,9:PRINT #6;"PLEASE W
     AIT":PRINT #6:PRINT #6;" DATA B
     EING READ IN"
```

"Dragonmaster," Atari version.

```
1040 FOR I=1 TO 15:READ A:NEXT I:FOR D=CH+24 TO CH+255:READ A:POKE D,A:NEXT D
1050 REM
1060 REM RESTART PROGRAM
1070 RESTORE :DR=15:K=2:P=SC+30:GR=0:CW=0
1080 GOSUB 1240:GOSUB 1450
1090 ZZ=SC+212:POKE ZZ,35+192
1100 REM
1110 V=15-STICK(0):Q=P:V=V/2
1120 IF V<>INT(V) AND PEEK(P-20)=0 THEN P=P-20
1130 V=INT(V)/2:IF V<>INT(V) AND PEEK(P+20)=0 THEN P=P+20
1140 V=INT(V)/2:IF V<>INT(V) AND PEEK(P-1)=0 THEN P=P-1
1150 V=INT(V)/2:IF V<>0 AND PEEK(P+1)=0 THEN P=P+1
1160 IF Q<>P THEN POKE Q,0:POKE P,53+128:SOUND 1,100,6,15
1170 GOSUB 1880:SOUND 1,0,0,0
1180 IF STRIG(0)=0 THEN GOSUB 1760
1190 IF PEEK(P+1)=37 OR PEEK(P+1)=60+64 OR PEEK(P-1)=42 THEN GOSUB 1590
1200 IF PEEK(P+20)=37 OR PEEK(P+20)=60+64 OR PEEK(P+20)=42 THEN GOSUB 1590
1210 GOSUB 2020:GOTO 1110
1220 REM
1230 REM INSTRUCTIONS AND SKILL LEVEL
1240 GOSUB 2270:POSITION 4,1:PRINT #6;"dragonmaster":PRINT #6:PRINT #6
1250 PRINT #6;"  OBJECT: BLAST ALL":PRINT #6;" DRAGONS AND ENTER"
1260 PRINT #6;" CASTLE TO RESCUE":PRINT #6;" PRINCESS.":PRINT #6:PRINT #6
1270 PRINT #6;"  ZAP DRAGONS FROM":PRINT #6;" FRONT WITH BUTTON.":PRINT #6:PRINT #6
1280 PRINT #6;"  ELIMINATE PHANTOM":PRINT #6;" DRAGON LAST."
1290 PRINT #6:PRINT #6
1300 PRINT #6;"{3 SPACES}1 = easy":PRINT #6;"{3 SPACES}2 = standard"
1310 PRINT #6;"{3 SPACES}3 = very hard":PRINT #6;"{3 SPACES}4 = impossible":SL=15:GR=0
1320 A=PEEK(764):IF PEEK(764)=255 THEN GR=GR+0.4:GR=GR-(GR>=256)*256+(INT(GR)=198):POKE 711,INT(GR):GOTO 1320
1330 POKE 764,255:GOSUB 2270:POKE 756,CH/256:IF A=30 THEN SL=10:RETURN
1340 IF A=26 THEN SL=6:RETURN
1350 IF A=24 THEN SL=3:RETURN
1360 RETURN
1370 REM
1380 REM DRAW AND ERASE DRAGONS, ETC.
1390 CL=0
1400 COL=CL*64
1410 POKE L,M+COL:POKE L+1,M+1+COL:POKE L+20,M+2+COL:POKE L+21,M+3+COL:RETURN
1420 POKE L,0:POKE L+1,0:POKE L+20,0:POKE L+21,0:RETURN
1430 REM
1440 REM INITIALIZE PLAYFIELD DRAGONS
1450 GOSUB 2270:POKE 756,CH/256
1460 FOR C=CH TO CH+7:POKE C,0:NEXT C:CL=1
1470 L=SC+190:M=45:GOSUB 1400:POKE SC+209,49+64:L=L-3:M=60:GOSUB 1400:F=36+128
1480 FOR X=SC TO SC+19:POKE X,F:NEXT X:FOR X=SC TO SC+460 STEP 20:POKE X,F:POKE X+19,F:NEXT X
1490 FOR X=SC+460 TO SC+479:POKE X,F:NEXT X:FOR X=1 TO 15:Z=INT(RND(1)*2)
1500 READ A:M=41:L=SC+42+A:IF Z THEN M=37
1510 GOSUB 1390:NEXT X:POKE 756,CH/256:POKE SC+30,53+128:POKE SC+13,53+128:POKE SC+14,53+128
1520 DATA 1,14,31,69,101,111,200,206,212,268,274,314,353,362,367
1530 FOR X=49 TO 52:POKE SC+209,X+64
1540 FOR T=1 TO 75:SOUND 0,(54-X)*40,10,15-T/5:NEXT T:NEXT X
1550 GOSUB 1630
1560 RETURN
1570 REM
1580 REM WE DIE AND ARE (?) REBORN
1590 FOR F=1 TO 2:FOR X=150 TO 80 STEP -1:SOUND 0,X,10,15:NEXT X
1600 FOR X=80 TO 150:SOUND 0,X,10,15:NEXT X:SOUND 0,0,0,0
1610 POKE P,F+53+128:NEXT F
1620 POKE SC+15-K,36+128:K=K-1:IF K<0 THEN 1970
1630 POKE P,0:P=SC+30:POKE P,53+128:RETURN
1640 REM
1650 REM PHANTOM DRAGON DIES; VICTORY
1660 L=P-19:CL=0:M=56:GOSUB 1400
1670 GOSUB 1820:L=P-19:GOSUB 1420
1680 FOR X=52 TO 49 STEP -1:POKE SC+209,X+64:FOR T=1 TO 75
1690 SOUND 0,(54-X)*40,10,15-T/5:NEXT T:NEXT X
```


NIGHTSTRIKE
When You're Ready for Action...
TG Software has the Game Plan
Try NIGHTSTRIKE™. Test your skill as unit commander of the city's defense forces. Demolish the attacking bombers, but watch out, don't get hit yourself. With eight levels of difficulty, the battle can rage on for hours. Or escape into outer space with DROIDS™. It takes speed and precision to outmaneuver the berserk droid crew. Can you lock them up before they destroy you and your spaceship? Good luck! Both games are available in versions compatible with the Atari* 400, 800 and XL series computers and the VIC 20**. You can find NIGHTSTRIKE, DROIDS and all the other action-packed TG games at your favorite software store or by calling TG Products' toll free number 1-800-531-5555. So the next time you're ready for action, TG Software is ready for you!
NIGHTSTRIKE
(Atari Version)
Eight levels of play
One or two players
16K Required
DROIDS
(Atari Version)
Seven levels of play
One or two players
16K Required
ATARI
400, 800, 1200
ROM CARTRIDGE
TG SOFTWARE
NIGHTSTRIKE and DROIDS are trademarks of TG Products.
*Atari 400, 800 and XL series are trademarks of Atari, Inc.
**VIC 20 is a trademark of Commodore Electronics.
DROIDS
TG SOFTWARE
1104 Summit Ave.
Suite 110
Plano, Texas 75074

```
1700 FOR X=SC+207 TO SC+209:POKE X-1,0:POKE X,53+128:FOR T=1 TO 75:NEXT T:NEXT X
1710 GRAPHICS 0:SETCOLOR 4,2,6:SETCOLOR 2,2,6:SETCOLOR 1,1,2
1720 POKE 756,224:POSITION 13,23:PRINT "AND THEY LIVED"
1730 PRINT :PRINT "{9 SPACES}HAPPILY EVER AFTER":GOTO 1990
1740 REM
1750 REM ELIMINATE A DRAGON
1760 CL=0:X=PEEK(P+1):Y=PEEK(P-1):IF X=62+64 AND DR=0 THEN 1660
1770 IF X<>39 AND Y<>44 THEN RETURN
1780 IF X=39 THEN L=P-19:M=56:GOSUB 1400
1790 IF Y=44 THEN L=P-22:M=56:GOSUB 1400
1800 DR=DR-1
1810 REM
1820 FOR X=200 TO 255:SOUND 0,X,10,15:NEXT X:SOUND 0,0,0,0
1830 IF PEEK(P+1)=58 THEN L=P-19:GOSUB 1420
1840 IF PEEK(P-1)=59 THEN L=P-22:GOSUB 1420
1850 RETURN
1860 REM
1870 REM TURN DRAGON, CREATE NEW ONE
1880 H=INT(RND(1)*396)+SC+42:L=H:BB=INT(RND(1)*2):X=PEEK(H)
1890 IF X=37 THEN M=41:GOSUB 1390:RETURN
1900 IF X=41 THEN M=37:GOSUB 1390:RETURN
1910 IF RND(1)>0.033 THEN RETURN
1920 FOR I=-40 TO 60 STEP 20:FOR J=-2 TO 3:IF PEEK(H+I+J) THEN RETURN
1930 NEXT J:NEXT I:M=41:DR=DR+1:IF BB=1 THEN M=37
1940 GOSUB 1390:RETURN
1950 REM
1960 REM THE DRAGONS HAVE WON
1970 GRAPHICS 0:SETCOLOR 4,4,4:SETCOLOR 2,4,4:SETCOLOR 1,4,8
1980 POKE 756,224:POSITION 13,23:PRINT "THE DRAGONS WIN"
1990 POKE 752,1:FOR X=1 TO 23:PRINT :FOR T=1 TO 35:NEXT T:NEXT X:GOTO 1060
2000 REM
2010 REM THE EVIL WIZARD MOVES
2020 CW=CW+7:IF CW>255 THEN CW=CW-256
2030 POKE 711,CW:CC=CC+1:IF CC<>SL THEN RETURN
2040 IZ=INT((ZZ-SC)/20):IP=INT((P-SC)/20):NZ=INT((IZ+IP)/2)*20
2050 NZ=NZ+(P-IP*20+ZZ-IZ*20)/2:CC=1:IF PEEK(NZ)=53+128 THEN GOSUB 1590
2060 IF PEEK(NZ) THEN RETURN
2070 POKE ZZ,0:POKE NZ,35+192:ZZ=NZ
2080 IF PEEK(ZZ+1)=53+128 OR PEEK(ZZ+20)=53+128 OR PEEK(ZZ-1)=53+128 THEN GOSUB 1590
2090 IF ZZ>SC+39 AND PEEK(ZZ-20)=53+128 THEN GOSUB 1590
2100 RETURN
2110 REM CHARACTER DATA
2120 DATA 60,66,165,129,153,165,66,60,170,85,170,85,170,85,170,85
2130 DATA 0,56,108,254,15,15,31,31,0,0,0,0,1,6,12,152
2140 DATA 255,63,63,31,31,15,31,61,176,176,152,216,220,252,248,224
2150 DATA 0,0,0,0,128,96,48,25,0,28,54,127,240,240,248,248
2160 DATA 13,13,25,27,59,63,31,7,255,252,252,248,248,240,248,188
2170 DATA 0,0,84,124,108,56,56,189,21,31,27,31,14,14,14,94
2180 DATA 255,255,255,255,170,255,255,0,254,254,254,254,170,254,254,0
2190 DATA 0,0,0,0,0,0,255,0,0,0,0,32,24,6,1,0
2200 DATA 0,0,16,8,4,2,1,0,0,8,4,4,2,2,1,0
2210 DATA 24,126,90,219,24,36,66,195,0,24,60,90,24,36,0,0
2220 DATA 0,0,0,24,24,0,0,0,16,0,2,72,2,21,75,21
2230 DATA 130,16,64,162,72,160,90,224,7,42,133,18,5,32,4,0
2240 DATA 168,210,168,64,18,64,8,32,0,40,68,82,5,10,21,10
2250 DATA 0,0,0,0,1,2,4,16,85,42,21,10,21,10,21,40
2260 DATA 128,144,0,144,72,164,80,160
2270 GRAPHICS 17:SETCOLOR 4,12,6:SETCOLOR 0,0,0:SETCOLOR 1,0,10:SETCOLOR 2,4,2:RETURN
```

©

Let your Atari experience the MPP CONNECTION!

MPP-1150 Printer Interface

New

- Replaces Atari 850™ Interface Module.
- Compatable with all software [including Visicalc™, Text Wizard™, Filemanager 800™, etc.].
- 5 foot cable with Centronics plug [compatible with Epson, NEC, Prowriter, etc.].
- 2 year warranty.
- Connects to serial bus on computer.
- Supports serial printer with additional cable.
- Works on **ALL** Atari Computers.

only $99.95

MICROBITS PERIPHERAL PRODUCTS

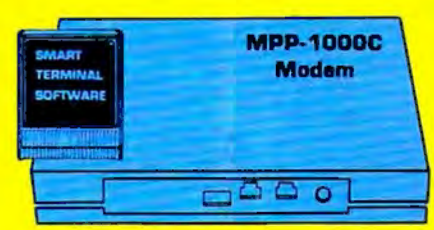
MPP-1000C Modem

Improved

- No Atari 850™ Interface Module needed.
- Smart Terminal Software on Cartridge.
- Direct Connect to Phone Line.
- Connects to Joystick Port.
- Works on **ALL** Atari Computers.

Smart Terminal Features:

- Multiple Buffers
- Off-Line Editing
- Upload/Download of Text and Programs
- Full/Half Duplex
- Supports XMODEM Protocol
- ASCII/ATASCII Translation
- Allows Transfer of Files Larger than Memory
- Variable Baud Rate
- Parity Options
- 100% Machine Language



~~$199.00~~
new low price $149.95

MICROBITS PERIPHERAL PRODUCTS

225 W. Third Street • Albany, Oregon 97321 • **(503) 967-9075**

Moving Maze

Matt Giwer

Can you maneuver through "Moving Maze" while fighting time and avoiding the relentless pursuit of the Rover? Impossible? Try it and see. Written for the Atari with joystick, versions are included for the unexpanded VIC (joystick optional) and the 64 with joystick.

During a stop for refueling on an out-of-the-way planet, you meet a mysterious old man who offers you the key to a fantastic treasure buried deep within a mountain. Many have tried to get to it, but all have been stopped by the unspeakable Rover of the Maze. The old man has the master control to a moving maze which guards the treasure. With his dying gasp he puts it in your hand. It looks like an antique Atari joystick. "Push the button," he says, "and you will control the maze." Never being one to pass up an adventure, you start off toward the mountain.

When you get there you find yourself on one side of a slowly moving maze. Every once in a while the dark red Rover passes your way. If you push the button on the joystick, the maze speeds up; if you release the button, the maze slows down. Armed with that secret you enter the maze. To get the treasure you must move as far as you can to the right side of the screen.

A ship is about to enter the maze in the VIC version of "Moving Maze."

Game Movement

"Moving Maze" uses Player/Missile Graphics with string manipulation. The heart of the game is the technique in lines 2210 and 2217. This routine scrolls the dimensioned PM$ RAM that is devoted to Players 1, 2, and 3. In this game the vertical yellow lines that form the maze are these players. The speed is controlled by lines 2400 and 2410, where the scrolling is increased or decreased with each pass through the loop.

You move vertically and horizontally with a joystick. The difference in the two motions is that you can move horizontally only one increment at a time. This is controlled by setting and unsetting the flag (F1) in lines 2300, 2301, and 2305, and is necessary due to the speed with which the program executes and the lack of sensitivity of the Atari joystick. Without this flag, it is too easy to move two steps at once and crash into a wall.

Hitting a maze wall costs you one life and sends you back to the start. If the Rover runs over you, it costs you two lives (but sometimes you will be lucky and lose only one and not be sent back to the beginning). You have five lives to lose. Your final score will be 300 points, less one point per second it takes you to get through, plus 100 points for each life you have left when you finish. The maximum score is 800 points, but it is impossible to achieve. A good score is 750.

Obstacles

The first difficulty is maneuvering through the walls of the maze, which are set up in the subroutine at the 5000 lines. Note that in line 55 the PM$ was not cleared to all blanks but was set to 146. This turned on bits 1, 4, and 7, which are those corresponding to 2, 16, and 128, respectively. Thus their sum, 146, is put into

In the flash of one second, Delta-10 can print the alphabet six times.

At 160 cps, 100% duty cycle, Delta-10 is built for speed. It's the dot matrix printer that quickly strikes up an intense working relationship with your computer.

There's no nonsense. Delta-10 performs with serial and parallel interfaces as standard. The 8K buffer gives Delta-10 the power to instantaneously store and print data as it frees your computer to continue its job. Turn Delta on and it can print three crisp copies simultaneously at the continuous speed of 160 cps.

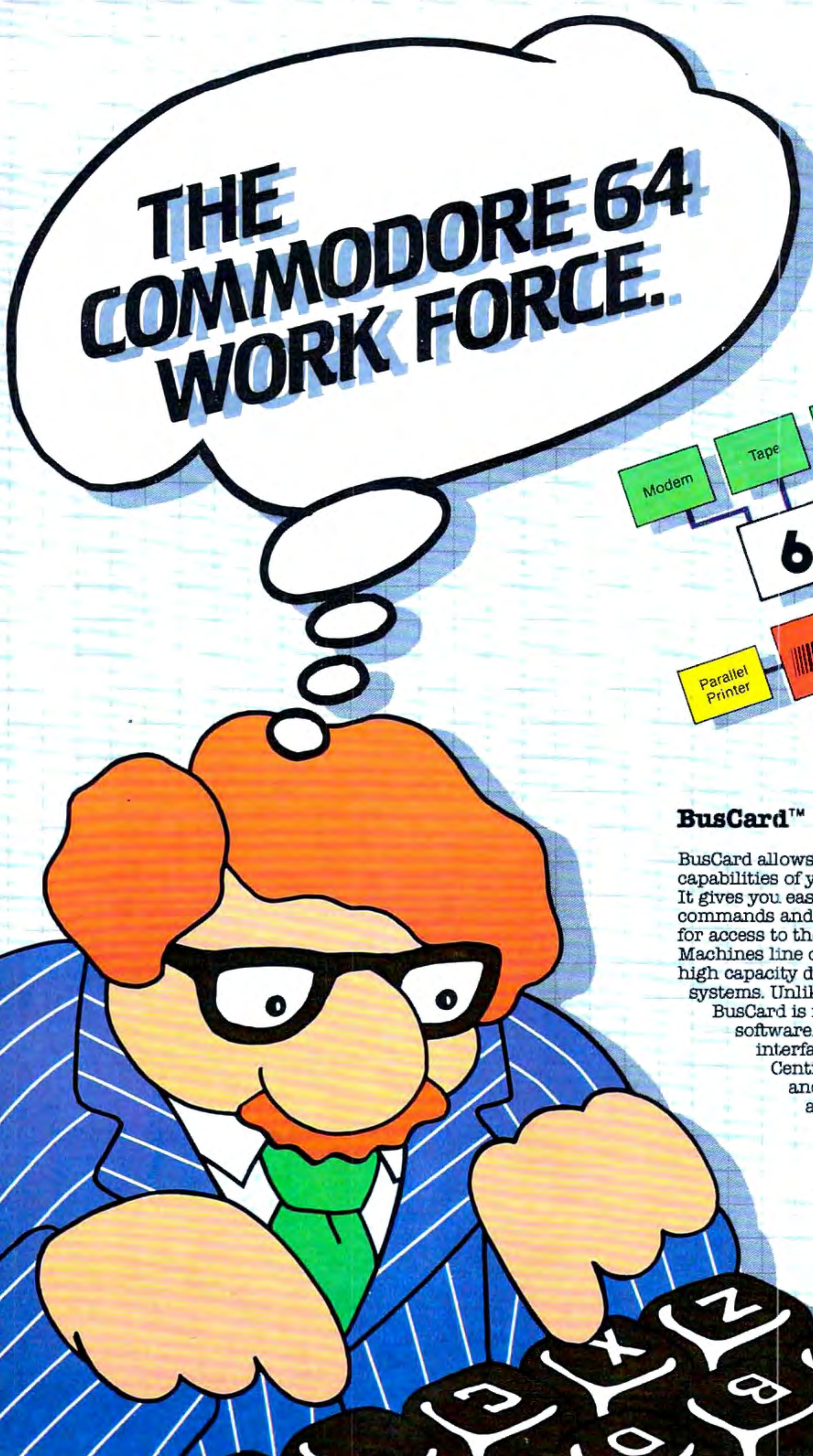
And Delta-10's performance isn't only measured in swiftness. There's the diversity of its character fonts, its true descender matrix, and its underlining ability.

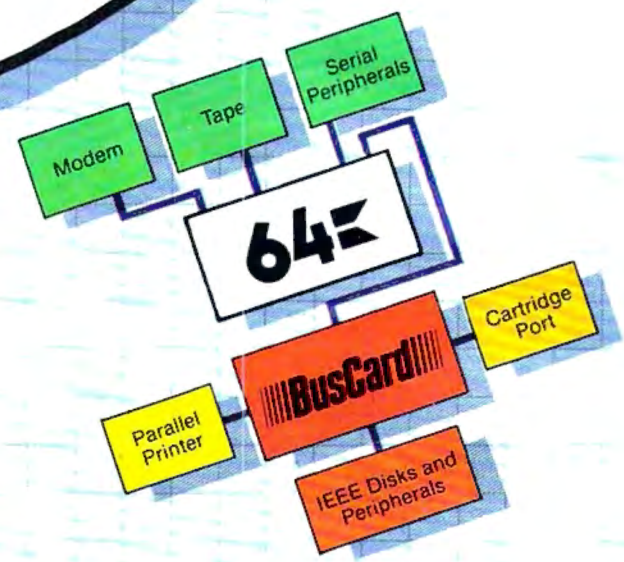
Star's new Delta-10. Designed for red-hot efficiency. And isn't that what you need?

star™
micronics • inc

THE POWER BEHIND THE PRINTED WORD.

Computer Peripherals Division
P.O. Box 612186, Dallas/Ft. Worth Airport, TX 75261





BusCard™ NEW!

BusCard allows you to expand the capabilities of your Commodore 64. It gives you easy to use "BASIC 4" commands and the fast IEEE interface for access to the Commodore Business Machines line of peripherals, including high capacity dual drives and hard disk systems. Unlike other interfaces, BusCard is fully compatible with software. No need to buy a printer interface — BusCard includes a Centronics printer port. Serial and user ports remain available for modem and serial peripherals. Plus: machine language monitor, complete documentation and a one year warranty.

$199

all of the string characters. In lines 5000 through 5040 I set some of these lines to other combinations of 2, 16, and 128, specifically, 18, 130, and 144. This sets up a random series of openings in the lines. However, since this is random, there may never be an opening in some walls, so lines 5050, 5060, and 5070 create such openings. Line 5073 puts blanks into the Missile and Player 0 pages, and lines at 5080 read in the shapes for Player 0 and the Rover which are the combination of the four missiles. Player 0 has two shapes, the normal playing shape and the explosion shape.

The next difficulty is avoiding the Rover, who sweeps the corridors of the maze. Its movement is not totally random, however. In the routine at 5200, the range of the random variable for the Rover depends upon your location in the maze. If you are outside, all corridors are equally likely. However, the farther you get into the maze, the less "choice" the Rover has. When you are in the last corridor, the Rover roams only the last two corridors. Thus, the closer you get to exiting the maze, the more likely the Rover is to attack you in that corridor. Since the Rover (the computer) knows where you are, it can attack you more frequently when you are closest to winning. The Rover is not a typical adversary.

Colliding with the maze wall is handled in the 5100 lines. This routine writes in the explosion shape, provides an audio effect, decreases the lives, does some housekeeping, and then puts the player back to the beginning with the original shape. The 5400 lines handle being run over by the Rover.

Other Game Features

The routine at line 5700 handles the timing and scoring for the game. Line 5700 calculates the number of seconds since the internal clock in registers 18, 19, and 20 was reset. When the game first comes on, it has measured the time since the GRAPHICS 0 call in line 30018, which resets them automatically. These are reset within the game in line 5790. Register 20 is read first and reset last since it changes the fastest. This method also gives the most accurate time measurement.

Line 5710 determines the score, and the second statement in that line requires that you complete the maze to get any score at all. Therefore, winning requires completing the maze and moving your player to the right. The rest of the lines are resets for a new game.

Finally, the data for the player, the Rover, and the explosion are in lines 10000-10020.

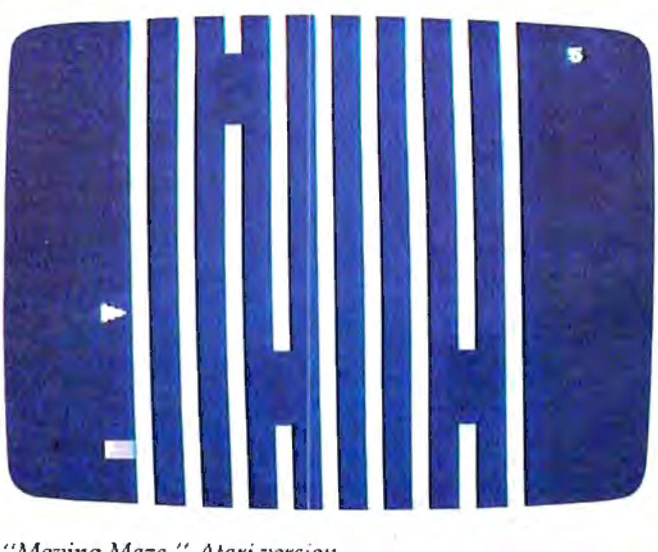
The string manipulation technique in this game uses scrolling to good advantage. It permits the user to control the speed of the motion and to employ it in the game as an additional control variable. This adds complexity and requires more than a little getting used to in order to play the game well.

The technique also permits a very compact game. The program requires a bit over 5K and, when running, well under 10K. Further, I made no use of the unused 2K in PM$ but rather DIMensioned another array B$ for convenience. As to the power of this technique, it would be difficult to find another game using full P/M graphics in single-line resolution with as much motion and challenge that uses so little RAM. This is why I urge you to examine and master the technique.

Program 1: Moving Maze – Atari Version

```
50 DIM PM$(4096),B$(250)
55 PM$(1)=CHR$(146):PM$(4096)=CHR$(1
   46):PM$(2)=PM$(1)
56 B$(1)=CHR$(0):B$(250)=CHR$(0):B$(
   2)=B$(1)
90 SP=0
1900 GOSUB 30000:GOSUB 5000:GOSUB 52
     00:GOSUB 5800
2100 FOR IJK=0 TO 1 STEP 0
2120 IF S1=0 THEN GOSUB 5700
2210 PM$(ST-768,ST-768+SP)=PM$(ST+10
     24+255-SP,ST+1024+255):PM$(ST-7
     67+SP,ST-1)=PM$(ST+512,ST+1280-
     SP)
2217 PM$(ST+2*256+1+SP,ST+5*256)=PM$
     (ST-767+SP,ST-1):PM$(ST+512,ST+
     512+SP)=PM$(ST-768,ST-768+SP)
2300 T=STICK(0):IF T=15 THEN F1=0:GO
     TO 2350
2301 IF F1=1 THEN 2320
2305 F1=1:IF T>12 THEN 2320
2310 PX=PX+6*(T=7)-6*(T=11)+6*(PX<55
     )-6*(PX>199):POKE 53248,PX
2314 PX=PX+6*(T=7)-6*(T=11)+6*(PX<56
     )-6*(PX>200):POKE 53248,PX:GOTO
      2350
2320 PY=PY+10*(T=13)-10*(T=14)+10*(P
     Y<35)-10*(PY>225):PM$(ST+236+PY
     ,ST+283+PY)=B$(1,57)
2350 PMY=PMY+10:IF PMY>506 THEN GOSU
     B 5200
2360 PM$(ST+486-PMY,ST+513-PMY)=B$(5
     1,78)
2365 IF PEEK(53260)<>0 THEN GOSUB 51
     00
2370 IF PEEK(53258)<>0 THEN GOSUB 54
     00
2400 IF STRIG(0)=0 THEN SP=SP+1:IF S
     P>200 THEN SP=200
2410 IF STRIG(0)=1 THEN SP=SP-2:IF S
     P<0 THEN SP=0
2500 IF PX>180 THEN S1=0:WIN=1
2501 IF LIVES<=0 THEN S1=0
2900 NEXT IJK
4999 REM SET UP MAZE & PLAYER
5000 FOR I=ST+512 TO ST+5*256-1 STEP
      32
5010 T=INT(30*RND(0))+1:IF T>25 THEN
      IF T<28 THEN FOR J=0 TO 31:PM$
     (I+J,I+J)=CHR$(18):NEXT J
5020 IF T>27 THEN IF T<29 THEN FOR J
     =0 TO 31:PM$(I+J,I+J)=CHR$(130)
     :NEXT J
5030 IF T>28 THEN FOR J=0 TO 31:PM$(
     I+J,I+J)=CHR$(144):NEXT J
```

```
5040 NEXT I
5048 REM ASSURES A PASSAGE THROUGH T
     HE MAZE
5050 FOR J=0 TO 31:PM$(768+ST+J,768+
     ST+J)=CHR$(18):NEXT J
5060 FOR J=0 TO 31:PM$(1024+ST+J,102
     4+ST+J)=CHR$(130):NEXT J
5070 FOR J=0 TO 31:PM$(1152+ST+J,115
     2+ST+J)=CHR$(144):NEXT J
5073 PM$(ST,ST+250)=B$:PM$(ST+256,ST
     +506)=B$
5079 PY=128:PX=67
5080 FOR I=1 TO 7:READ A:B$(20+I,20+
     I)=CHR$(A):NEXT I:POKE 53248,PX
     :POKE 704,78:PM$(ST+246+PY,ST+2
     96+PY)=B$(1,57)
5085 FOR I=1 TO 8:READ A:B$(60+I,60+
     I)=CHR$(A):NEXT I
5087 FOR I=1 TO 16:READ A:B$(90+I,90
     +I)=CHR$(A):NEXT I
5090 RETURN
5099 REM HIT WALL
5100 POKE 53278,0
5110 PM$(ST+246+PY,ST+293+PY)=B$(80,
     106)
5120 FOR J=15 TO 0 STEP -2:FOR I=250
      TO 50 STEP -50:SOUND 0,I,6,J:N
     EXT I:NEXT J:SOUND 0,0,0,0
5130 LIVES=LIVES-1
5131 POSITION 35,0:? LIVES;
5182 PM$(ST+246+PY,ST+276+PY)=B$(220
     ,250)
5184 PY=128:PM$(ST+236+PY,ST+283+PY)
     =B$(1,57):PX=67:POKE 53248,PX
5190 RETURN
5199 REM SETS RANDOM PATH FOR ROAMER
5200 J=8:IF PX>66 THEN IF PX<79 THEN
      J=9:GOTO 5220
5201 IF PX>78 THEN IF PX<91 THEN J=8
     :GOTO 5220
5202 IF PX>90 THEN IF PX<103 THEN J=
     7:GOTO 5220
5203 IF PX>102 THEN IF PX<115 THEN J
     =6:GOTO 5220
5204 IF PX>114 THEN IF PX<127 THEN J
     =5:GOTO 5220
5205 IF PX>126 THEN IF PX<139 THEN J
     =4:GOTO 5220
5206 IF PX>138 THEN IF PX<151 THEN J
     =3:GOTO 5220
5207 IF PX>150 THEN J=2
5220 T=INT(J*RND(0))+1:PM=178-T*12-3
     :PMY=230
5230 POKE 53255,PM:POKE 53254,PM+2:P
     OKE 53253,PM+4:POKE 53252,PM+6
5290 RETURN
5399 REM ROAMER HITS PLAYER
5400 POKE 53278,0
5410 FOR I=250 TO 190 STEP -30:FOR J
     =15 TO 8 STEP -1
5420 SOUND 0,I,10,J:SOUND 1,I,6,J-4:
     NEXT J:NEXT I:LIVES=LIVES-1
5480 SOUND 0,0,0,0:SOUND 1,0,0,0
5481 POSITION 35,0:? LIVES;
5490 RETURN
5700 SECS=INT((PEEK(18)*65536+PEEK(1
     9)*256+PEEK(20))/60)
5710 SCR=300-SECS+100*LIVES:SCR=SCR*
     (WIN=1):POKE 623,18
5711 IF SCR>HSCR THEN HSCR=SCR
5720 POSITION 22,0:? "{6 SPACES}LIVE
     S: ":LIVES;
5730 POSITION 22,1:? "  SECONDS: ";S
     ECS;"  ";
5731 POSITION 22,2:? "SCORE: ";SCR;"
     {3 SPACES}";
5732 POSITION 22,3:? "HIGH SCORE: ";
     HSCR
5733 POSITION 22,4:? "PUSH TRIG TO S
     TART"
5740 IF STRIG(0)=1 THEN 5740
5750 GOSUB 5800
5780 POKE 623,17:S1=1:LIVES=5:WIN=0:
     SECS=0:SP=0
5781 POSITION 35,0:? LIVES;
5783 PM$(ST+246+PY,ST+276+PY)=B$(220
     ,250)
5784 PY=128:PM$(ST+236+PY,ST+283+PY)
     =B$(1,57):PX=67:POKE 53248,PX
5790 POKE 18,0:POKE 19,0:POKE 20,0:R
     ETURN
5800 POKE 18,0:POKE 19,0:POKE 20,0
5810 POSITION 22,0:? "{12 SPACES}";
5815 POSITION 22,1:? "{16 SPACES}";
5820 POSITION 22,2:? "{16 SPACES}";
5825 POSITION 22,3:? "{16 SPACES}";
5830 POSITION 22,4:? "{18 SPACES}";
5890 RETURN
10000 DATA 128,224,248,252,248,224,1
      28
10010 DATA 255,255,255,255,255,255,2
      55,255
10020 DATA 24,60,126,255,255,126,60,
      24,24,24,24,24,24,24,90,126
30000 REM PM SETUP
30018 GRAPHICS 0
30022 POKE 559,62:POKE 53277,3:POKE
      623,17
30030 POKE 752,1:? "{CLEAR} ";:POKE
      710,96:POKE 709,12:POKE 712,96
31000 REM FIND PMBASE
31010 ADHI=INT(ADR(PM$)/256):REM ADL
      O=ADR(PM$)-256*ADHI
31020 P=INT((ADHI+8)/8)
31022 POKE 54279,P*8
31030 ST=8*P*256+1024-256-ADR(PM$)
31040 POKE 53249,75:POKE 53250,111:P
      OKE 53251,147
31041 FOR I=1 TO 3:POKE 53256+I,3:PO
      KE 704+I,218:NEXT I
31090 RETURN
32000 SAVE "D:MAZE"
```



"Moving Maze," Atari version.

$5 says you can't beat Gridrunner.™

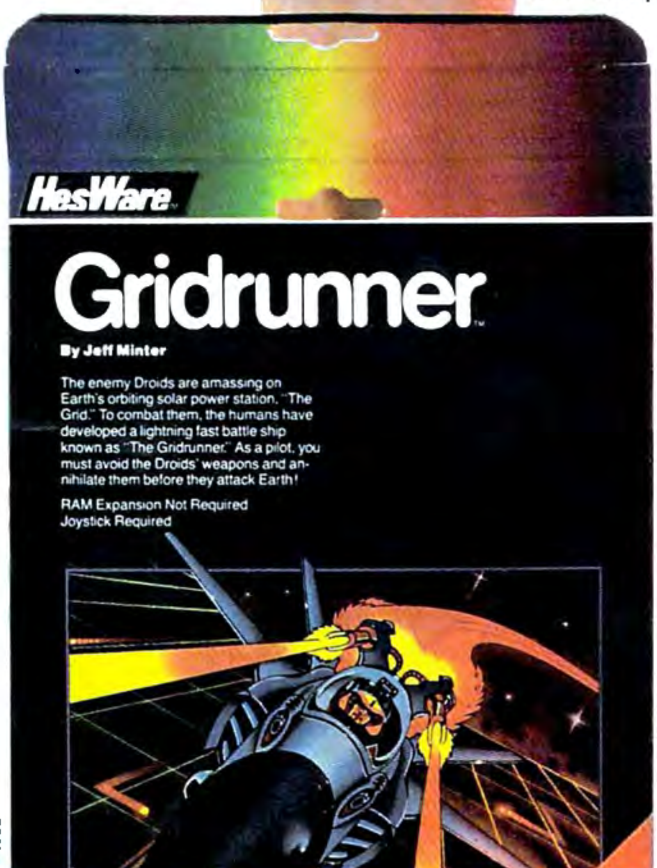
Gridrunner™ is the toughest, fastest, arcade quality game ever to challenge a Commodore™ or Atari® computer owner.

How tough is Gridrunner?

So tough that HesWare™ is offering $5 just for taking the Gridrunner challenge. And you don't even have to beat the game to get the greenbacks. There's a check right inside every Gridrunner box, good for $5 off your next HesWare purchase. You spend your check right in the store, on your choice of exciting HesWare action, education, and computer utility programs. Even HesWare peripherals.

No one, not even the author, has ever reached even the 14th level of Gridrunner. But maybe you've got what it takes to go all the way to the 31st level (20 levels on VIC 20™ version). Maybe you can master the patterns of the X/Y Zappers. And stop the Gridsearch Droids before they mutate into lethal Pods.

The Gridrunner challenge. Just one of the ways that HesWare is expanding the computer experience. Look for all the HesWare products at your favorite software retailer.



© 1983

Pleases the tough customer.

HesWare

Human Engineered Software
150 North Hill Drive
Brisbane, CA 94005
800-227-6703
(in California 800 632-7979)
Dept. C20

VIC Notes

Marc Sugiyama, Programming Assistant

The VIC version of "Moving Maze" (Program 2) uses the same game layout as the Atari version, but the rules of the game have been changed. Whenever you run into a wall or the Rover, you are sent back to the beginning of the maze, without exception. You maneuver through the maze using the I, J, K, and M keys or a joystick. Pressing the space bar or the fire button on the joystick causes the maze to speed up, but you lose one point every time the maze moves. You cannot move while the joystick button or the space bar is depressed. When you complete a maze, another will appear, but the openings in the walls will be smaller. The game ends when you lose all five lives.

Moving Maze uses up almost all of the memory of an unexpanded VIC, so do not enter any extra spaces or REMs. Also, the program will not handle VIC's floating memory, so remove any expansion memory cartridges.

Program 2: Moving Maze – VIC Version

by Marc Sugiyama, Programming Assistant

```
10 POKE55,94:POKE56,29:CLR
15 PRINT"{CLR}"CHR$(8);:SS=36879:CS=36865
   :FF=255:POKECS,FF:F8=484:Z4=4:Z5=5:RL=
   62:RR=63
20 DS=36869:SE=8164:T3=23:F4=44:H4=64:V1=
   36875:DEFFNA(J)=Z3+J*Z3+S+F8*(JANDZ1)
25 DIMT(5),D(4),K(4):Z3=3:SP=160:KS=197:Z
   Q=2:RB=7673:RC=7665:RM=1:BU=198:S2=.94
   :SN=1
30 H8=128:H3=32:S=7680:RS=57:Z0=0:Z1=1:SK
   =5:T2=22:T1=21:SH=57:ML=7518:ZB=176:LI
   =3
35 FORI=1TO4:READD(I),K(I):NEXT:JD=37154:
   J1=JD-3:J2=J1+1:POKEJD,127
40 FORI=1TO7:FORJ=0TO7:READA:POKE7616+I*8
   +J,A:NEXT:NEXT
45 FORI=7518TO7614:READA:POKEI,A:NEXT
50 POKESS-1,10:A$="{RVS}{WHT}{3 SPACES}
   {OFF}{RED}9{RVS}{2 SPACES}{OFF}{CYN}9
   {RVS}{2 SPACES}{OFF}{PUR}9{RVS}
   {2 SPACES}{OFF}{GRN}9{RVS}{2 SPACES}
   {OFF}{BLU}9{RVS}{2 SPACES}{OFF}{YEL}9
   {RVS} ":PRINT"{HOME}";:FORI=0TO21
55 PRINTA$"{RVS}"CHR$(160)" ";:NEXT:PRINT
   A$"{RVS}"CHR$(160)"{HOME}";:POKE8185,1
   60
60 FORI=0TOLI-1:POKES+T1+T2*I,61:NEXT:POK
   ESS,8:POKEDS,FF:POKECS,25:GOTO265
65 POKEV1,240:OE=OE-Z1
70 FORJ=0TO5:IFT(J)=Z0THEN85
75 T(J)=T(J)+Z1:IFT(J)>SKTHENPOKEFNA(J),R
   S:T(J)=Z0
80 GOTO90
85 IFRND(Z1)>S2THENT(J)=Z1:POKEFNA(J),SP
90 NEXT:POKEV1,Z0:POKESL,E1:SYSML:IFPEEK(
   SL)=E1ORPEEK(SL)=SPTHENPOKESL,OS
95 RETURN
100  IFROTHEN110
105 R1=RND(Z1)>.5:RD=T2+F4*R1:RT=INT(RND(
    Z1)*Z5)*Z3+Z4-F8*R1+S
110 POKEV1,200:POKERC+RM,FF:POKERB+RM,FF:
    IFR1THEN125
115 RM=RM+Z1:IFRM>Z4THENRM=Z0
120 GOTO130
125 RM=RM-Z1:IFRM<Z1THENRM=Z5
130 POKERC+RM,H8:POKERB+RM,Z1
135 POKERT,SP:POKERT+Z1,SP:RT=RT+RD:RO=RO
    +Z1:IFRO=T3THENRO=Z0:POKEV1,Z0:RETURN
140 POKERT,RL:POKERT+Z1,RR:POKEV1,Z0:RETU
    RN
145 IFPEEK(SL)<>OSTHENEM=PEEK(SL):GOTO240
150 IF(PEEK(KS)=H3)=Z0AND(PEEK(J1)ANDH3)T
    HEN165
155 ZW=(ZW+Z1)ANDZ3:IFZW=Z0THENZW=Z1
160 ONZWGOSUB65,100,65:GOTO145
165 DI=Z0:A=PEEK(KS):IFA<>H4THEN195
170 IF(PEEK(J2)ANDH8)=Z0THENDI=Z4:GOTO205
175 J=PEEK(J1):IF(JANDZ4)=Z0THENDI=Z1:GOT
    O205
180 IF(JAND8)=Z0THENDI=2:GOTO205
185 IF(JAND16)=Z0THENDI=Z3
190 GOTO205
195 FORI=1TO4:IFA=K(I)THENDI=I
200 NEXT
205 IFDI=Z0THEN235
210 IFDI=Z3THENOE=OE-SN:IFINT((SL-S)/T2)=
    (SL-S)/T2THEN235
215 IFDI=Z4THENOE=OE+SN
220 EL=SL+D(DI):IFEL<SOREL>SE+T1THEN235
225 POKEV1,220:E1=EM:EM=PEEK(EL):IFEM<>SP
    THEN240
230 POKESL,SP:OS=SH+DI:POKEEL,OS:SL=EL
235 ZQ=Z3-ZQ:POKEV1,Z0:ONZQGOSUB70,100:GO
    TO145
240 IFEM<>224THEN255
245 POKEV1,0:SN=SN+1:SK=SK-1:IFSK<2THENSK
    =2
250 FORI=1TO5:POKESS,25:A=TAN(I):POKESS,8
    :A=TAN(I):NEXT:GOTO265
255 POKESL,RS:POKEV1,0:POKEV1+2,129:FORI=
    15TO0STEP-1:POKESS-1,I:A=TAN(I):NEXT
260 POKEV1+2,0:POKESS-1,10:LI=LI-1:IFLI=-
    1THEN270
265 POKESL,E1:SL=7922:OS=61:EM=OS:E1=EM:P
    OKESL,OS:POKES+T1+T2*LI,SP:GOTO145
270 POKEV1,0:POKECS,FF:POKESS,42:POKEDS,2
    40:PRINT"{CLR}{OFF}{2 DOWN}{WHT} YOU
    {SPACE}HAVE USED ALL OF"
275 PRINTSPC(5)"YOUR LIVES...{DOWN}"
280 PRINTSPC(7)"SCORE:"OE:IFOE>HITHENHI=O
    E:PRINT"{RVS}";
285 PRINT"{2 RIGHT}HIGH SCORE{OFF}:"HI:OE
    =0
290 PRINT"{DOWN}{2 SPACES}PLAY AGAIN (Y/N
    )?":POKECS,25:POKEBU,0
295 GETA$:IFA$<>"Y"ANDA$<>"N"THEN295
300 IFA$="Y"THENPOKECS,255:LI=3:SK=5:GOTO
    50
305 PRINT"{CLR}{BLU}";:POKEJD,FF:POKESS,2
    7:END
310 DATA -22,12,22,36,-1,20,1,44
315 DATA 126,126,126,126,126,126,126,126
320 DATA 16,56,108,198,130,254,146,130
325 DATA 130,146,254,130,198,108,56,16
```

INTRODUCING A REAL-TIME WARGAME SO FAST YOU'LL CALL IT A STRATEGY ARCADE GAME!

At SSI, we're the first to admit that the last thing our wargames are famous for is their speed of play. But then, our strategy games are designed to challenge your mental — rather than manual — dexterity.

Now, don't get us wrong. We love real-time arcade action as much as the next person. So we've put a great deal of energy to develop a game that's the perfect union of these two worlds: heart-pounding arcade excitement in a thought-provoking, tactically challenging wargame.

Today, like proud parents, we are delighted to announce the successful culmination of our efforts: COMBAT LEADER™

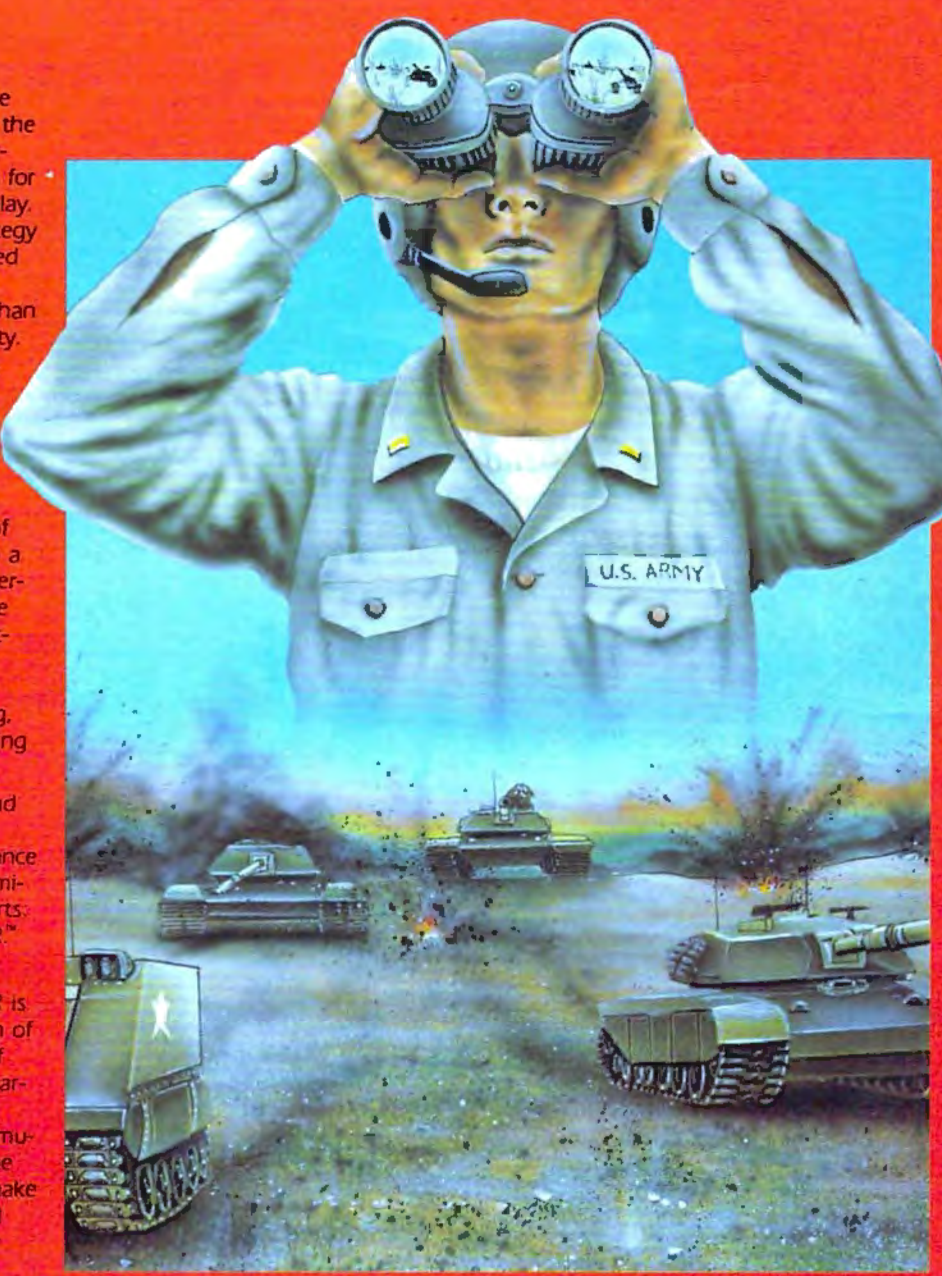
In every way, COMBAT LEADER is the ideal first-born of this new hybrid of strategy arcade wargames.

As a strategy simulation, it has all the detailed rules to make you feel like a real battle commander poised to engage enemy tanks and infantry on a scrolling map display of foliage, hills, open and rough terrain. For example, there are complete ratings of armor thickness and strength, speed and fire accuracy for over 70 historical and modern tanks. Accurate rules govern visibility and enemy sightings.

As an arcade game, COMBAT LEADER doesn't let any of these details slow down your play. The computer handles all "bookkeeping" chores while giving you **real-time** action: Your troops look, retreat, advance, patrol and fire the instant you order them to do so. No waiting, no pause. In fact, the pace is so fast we had to provide slower levels of play to give you a chance at this game.

At SSI, we make it a habit to keep advertising hype down to a minimum. So when we say this is one of the fastest and finest games around (strategy **or** arcade), we don't do so lightly. Our 14-day money-back guarantee promises you'll agree with us once you get your hands on it.

To do that, all you have to do is visit your local computer/software or game store today!



COMBAT LEADER™

On 48K disk or cassette for the ATARI® 400/800/1200; $39.95.

RapidFire

GAMES FROM SSI

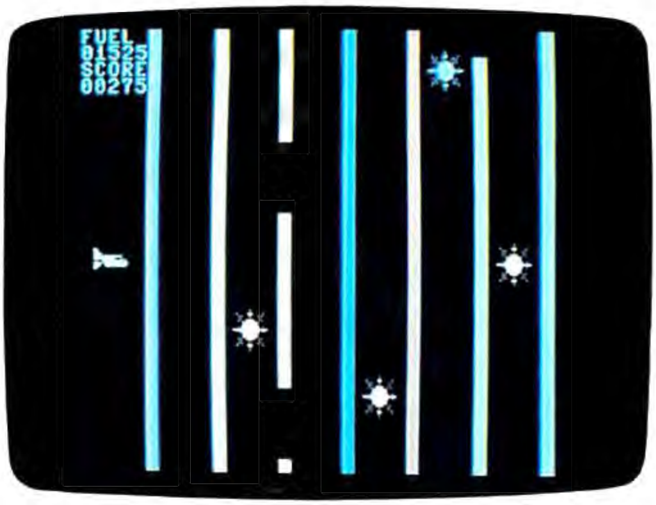
If there are no convenient stores near you, VISA and Mastercard holders can order direct by calling **800-227-1617, ext. 335 (toll free).** In California, call 800-772-3545, ext. 335.

To order by mail, send your check to: **Strategic Simulations Inc,** 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. Please specify disk or cassette. (California residents, add 6.5% sales tax.)

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

Atari is a registered trademark of Atari Inc.

```
330 DATA 31,52,100,198,100,52,31,0
335 DATA 248,44,38,99,38,44,248,0
340 DATA 127,255,255,255,255,255,255,127
345 DATA 254,255,255,255,255,255,255,254
350 DATA 162,3,134,0,160,3,162,22,169,30,
    133,2,169,0,133,1
355 DATA 177,1,133,251,24,165,1,105,22,13
    3,1,144,2,230,2,177
360 DATA 1,133,252,165,251,145,1,165,252,
    133,251,202,208,230,200,200
365 DATA 200,162,22,169,31,133,2,169,228,
    133,1,177,1,133,251,56
370 DATA 165,1,233,22,133,1,176,2,198,2,1
    77,1,133,252,165,251
375 DATA 145,1,165,252,133,251,202,208,23
    0,200,200,200,198,0,208,166,96
```



Guide your shuttle through the maze. 64 version.

Program 3: Shuttle Escape – 64 Version

by Eric Brandon, Programming Assistant

```
100 GOSUB3000
110 PRINTCHR$(142)
120 IF PEEK(49153)<>169 THEN GOSUB 10000:
    GOSUB 50000
130 V=13*4096:CO=50
140 POKE V+16,0
150 POKE V+39,1:POKEV+40,1
160 POKE V+0,CO:POKEV+4,CO-2:POKEV+5,221
170 POKE V+1,200
180 POKE V+2,CO
190 POKE V+3,179
200 POKE V+21,3
210 POKE V+39,1:POKEV+40,1:POKEV+41,8:POK
    EV+42,1
220 POKE 2040,245
230 POKE 2041,246:POKE2043,246
240 POKE 2042,247
250 FOR K=1 TO 500 : NEXT K:POKEV+21,7
260 GOSUB 2000
270 I=200
280 P=1
290 Q=Q+.01*P
300 P=P+.1:C=C+1
310 I=I-Q
320 IF PEEK(2042)=248 THEN POKE 2042,254:
    GOTO340
330 IF PEEK(2042)=254 THEN POKE 2042,248
340 POKE V+1,I:POKEV+3,I-21:POKEV+5,I+21
350 POKES+22,P2:POKES+23,1OR(16-P2/16)*16
360 P2=P2+P2/244
370 IF C=60THEN POKEV+23,4
380 IF C=20THEN POKEV+23,4
390 IF C=40 THEN POKEV+23,0:POKE2042,248
400 IF C<70 THEN 290
410 POKE 2040,249
420 POKE 2043,250:POKEV+6,CO:POKEV+7,I:PO
    KEV+21,15
430 Q=Q+.01*P
440 P=P+.1:C=C+1:C2=C2+.6
450 I=I-Q
460 IF PEEK(2042)=248 THEN POKE 2042,254:
    GOTO480
470 IF PEEK(2042)=254 THEN POKE 2042,248
480 POKE V+1,I:POKEV+5,I+21
490 NR=I+C2*C2:NC=CO+C2*3
500 POKE V+7,NR:POKEV+3,NR-21:POKEV+6,NC:
    POKEV+2,NC
510 IF C=83 THEN POKE 2043,251:POKE2041,253
520 IF C=86 THEN POKE 2043,252
530 IF C=89 THEN POKE V+21,5
540 POKES+22,P2:POKES+23,1OR(16-P2/16)*16
550 P2=P2+P2/244
560 IF I>25 THEN 430
570 POKE S+4,128
580 POKE V+5,I+21
590 I=I-2:IFI>0 THEN580
600 POKE V+21,1
610 FOR J=1 TO 2000:NEXT
620 PRINT"{HOME}{10 RIGHT}{WHT}{2 SPACES}
    ORBIT ACHIEVED...."
630 FOR I=1 TO 1000:NEXT
640 POKE 2040,244
650 POKE V,0:POKEV+1,117
660 FOR I=0 TO 348 STEP2
670 POKE V,I AND 255:POKEV+16,I/255
680 NEXT
690 FOR I=0 TO 1000:NEXT
700 GOTO 4000
710 POKE 53281,12:POKE53280,12:PRINT"{WHT}
720 END
2000 S=54272
2010 POKES+24,15+16+32:POKES+23,1+16*5
2020 POKES+5,0
2030 POKES+6,16*15+15
2040 POKES+4,129
2050 POKES+1,11
2060 P2=100:RETURN
3000 POKE 53281,0:POKE53280,0
3010 PRINT"{CLR}"
3020 PRINT"{5 DOWN}"
3040 T=12
3050 PRINTTAB(T)"{7}{RVS}£{2 SPACES}
     {RIGHT} {RIGHT} {RIGHT} {RIGHT}
     {RIGHT}{3 SPACES}{RIGHT}{3 SPACES}
     {RIGHT} {3 RIGHT}£{2 SPACES}"
3060 PRINTTAB(T)"{RVS} {3 RIGHT} {RIGHT}
     {SPACE}{RIGHT} {RIGHT} {2 RIGHT}
     {3 RIGHT} {2 RIGHT} {3 RIGHT} "
3070 PRINTTAB(T)"{*}{RVS} {*}{RIGHT}
     {3 SPACES}{RIGHT} {RIGHT} {2 RIGHT}
     {SPACE}{3 RIGHT} {2 RIGHT} {3 RIGHT}
     {2 SPACES}"
3080 PRINTTAB(T)"{RVS}{2 RIGHT} {RIGHT}
     {RIGHT} {RIGHT} {RIGHT} {2 RIGHT}
     {3 RIGHT} {2 RIGHT} {3 RIGHT} "
3090 PRINTTAB(T)"{RVS}{2 SPACES}{OFF}£
     {RVS}{RIGHT} {RIGHT} {RIGHT}{OFF}
     {*}{RVS} {OFF}£{RVS}{2 RIGHT}
     {3 RIGHT} {2 RIGHT}{OFF}{*}{RVS}
     {2 SPACES}{RIGHT}{OFF}{*}{RVS}
     {2 SPACES}"
```

DYNAMIC PRINTER INTERFACES
for the VIC 20™ and the COMMODORE 64™
UNLIKE ANY OTHERS THAT HAVE COME BEFORE

It's not quick or easy to do things right!! After 8 long months of research and development; RAK-Ware, TYMAC CONTROLS CORP. and MICRO-WARE D.I. have brought the world better parallel interfaces. Better because they both have the ability to provide TRUE EMULATION of the Commodore® printer. That's right!! Graphic Characters, tabbing, Dot Graphics, and the other features. A formidable task that was finally accomplished.

THE CONNECTION™ — The Ultimate Parallel interface for the VIC 20 or Commodore 64. This fully intelligent interface plugs into the disk (serial) socket just like the standard printer. It can easily be assigned any device number and it will provide virtually TOTAL EMULATION of the Commodore® printer. Using the latest technology, this interface will display the full GRAPHIC CHARACTERS or convert them to their equivalent representations in clear text. It supports all of the standard commands (OPEN, PRINT#, and CLOSE), Column tabbing, dot tabbing, graphic repeat, dot addressable graphics, and the other features of the Commodore® Printer. Software designed to operate with the Commodore® Printer will operate using "THE CONNECTION™." Beside this, a 2K buffer has been provided, a full printer self test, LED Status indicators, Printer Reset switch, skip over perf, margin set and programmable line length. This interface is printer specific to take advantage of the special features of your printer. In the standard mode (non-graphics), it is designed to interface virtually any parallel printer with a standard Centronics configuration and connector. Specify your printer when ordering. Additional ROM's may be purchased for other printer applications . . . All this for $119.00

BUFFERED PARALLEL CABLE & DRIVER — A parallel interface for the budget minded. This interface plugs into the USER-port and comes with an extensive manual with driver listings for the VIC 20™ and the Commodore 64™. It can be used with virtually any printer that has a standard Centronics type configuration and connection. Fully buffered for maximum protection of your computer ONLY $29.95 Add "CABLE BRAINS" cartridge for the VIC 20 (diskette for the 64) and get a full Graphic Emulation Driver. With this package you can print all of the Graphic Characters that your computer has plus EMULATE the Commodore® printer. This product is printer specific to take full advantage of your printer. Available for most graphic matrix printers . . . $29.95

micro WARE
DISTRIBUTING INC.
1342 B Rt. 23,
Butler, NJ 07405
201-838-9027

MasterCard VISA

Dealer and Distributor Inquiries Invited

NOTE: We solicit hardware and software items for the VIC 20® and CBM 64® Royalties, license fees, or outright purchases can be negotiated. CBM 64® & VIC 20® are Registered Trademarks of Commodore Business Machines Inc.

64 Notes

Eric Brandon, Programming Assistant

The Commodore 64 version of "Moving Maze" has been renamed "Shuttle Escape," since it has a space shuttle theme.

Shuttle Escape plays much like the other versions. The main difference is that a quantity called FUEL has been added to the game. You begin with 2000 fuel units which you lose at the rate of 60 units each second whether the shuttle is moving or not. If you touch a wall or one of the roving droids, you lose 100 units each 1/60 second. When you have run out of fuel, the game is over. Fortunately, you can refill your tanks by reaching the right-hand side of the screen.

If you want to stop the game for a moment, just hold down the SHIFT key. If you want to stop the game for a longer period of time, use SHIFT LOCK.

You can speed up the movement of the walls by holding down the fire button on the joystick. This won't make gaps appear any sooner, but it will speed up any gaps that are already there. The penalty is that while the fire button is down, your fuel disappears twice as fast.

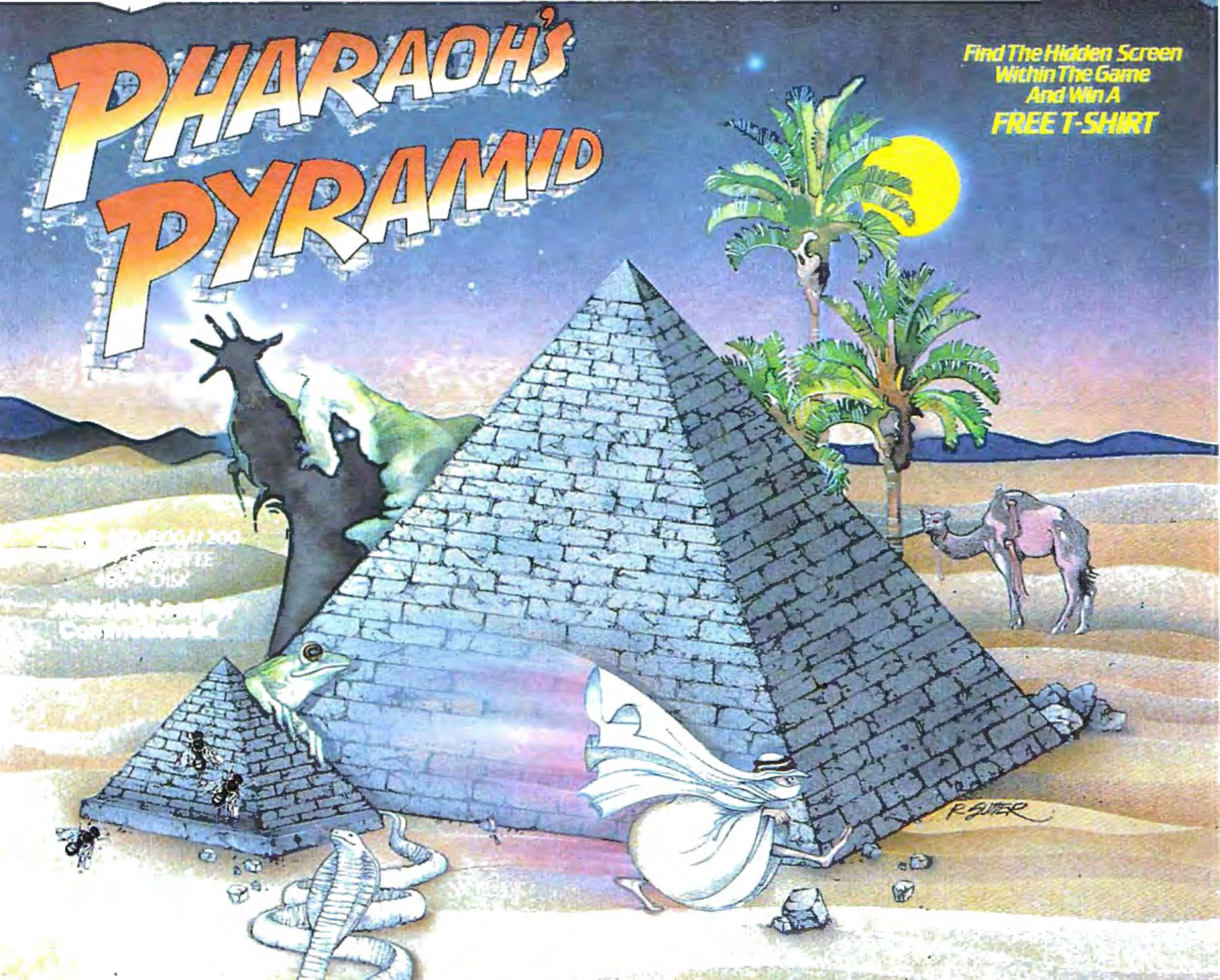
Programming Shuttle Escape revealed some interesting problems. The first is that *sparkle* – little specks of snow – appears on the screen. Usually this causes no difficulty, but when you try to use the VIC-II's sprite-background collision detection register, it turns out that sprites can collide with sparkle!

What this meant to Shuttle Escape was that occasionally, for no apparent reason, the shuttle would "collide" and you would lose 100 fuel units. Since moving the character set eliminates sparkle, it was relocated to $3000.

Another quirk of the 64 is that the VIC-II chip can look at only 16K of memory at a time. When you turn on your machine, it is looking at the first 16K block from $0000-$3FFF. It was decided to leave it there for simplicity. This meant that the sprite data, the relocated character set, and the entire BASIC program all had to be squeezed into 16K. Because of this memory limitation, when the machine language creates a character set at $3000, it destroys the DATA statements in the program. Fortunately, the DATA statements are no longer needed since they have already been POKEd into memory.

Because running the program will destroy it, be sure that when you type it in, you SAVE it before you try to RUN.

```
3100 PRINT
3110 PRINTTAB(T)"{CYN}{RVS}£{2 SPACES}
     {RIGHT}£{2 SPACES}{RIGHT}£
     {2 SPACES}{RIGHT}£ {*}{RIGHT}
     {2 SPACES}{*}{RIGHT}£{2 SPACES}"
3120 PRINTTAB(T)"{RVS} {3 RIGHT}
     {3 RIGHT} {3 RIGHT} {RIGHT} {RIGHT}
     {SPACE}{RIGHT} {RIGHT} "
3130 PRINTTAB(T)"{RVS}{2 SPACES}{2 RIGHT}
     {OFF}{*}{RVS} {*}{RIGHT}
     {3 RIGHT}{3 SPACES}{RIGHT}{2 SPACES}
     {OFF}£{RIGHT}{RVS}{2 SPACES}"
3140 PRINTTAB(T)"{RVS} {5 RIGHT} {RIGHT}
     {SPACE}{3 RIGHT} {RIGHT} {RIGHT}
     {3 RIGHT} "
3150 PRINTTAB(T)"{*}{RVS}{2 SPACES}
     {RIGHT}{2 SPACES}{OFF}£ {*}{RVS}
     {2 SPACES}{RIGHT} {RIGHT} {RIGHT}
     {3 RIGHT}{OFF}{*}{RVS}{2 SPACES}"
3999 RETURN
4000 V=13*4096:POKE53281,0:POKE53280,0
4010 POKE V+21,0:POKEV+23,0
4020 GOSUB 2000:POKE S+5,7*16:POKES+6,249
     :POKES+4,128
4030 POKE V+40,03:POKEV+41,07:POKEV+42,03
     :POKEV+43,07:POKEV+44,03:POKEV+45,07
4040 POKE V+46,03:FOR I=1 TO 6 : POKE V+2
     *I,(36+40*I)AND255:NEXT
4050 POKE V+16,64:POKE 2040,244:POKEV,30:
     POKEV+1,148:POKE V+21,255
4060 FOR I=2041 TO 2047:POKEI,255:NEXT
4070 PRINT"{CYN}{CLR}FUEL
4080 PRINT"02000"
4090 PRINT"SCORE:"
4100 PRINT"00000"
4110 P(0)=1029:P(4)=1994:P(1)=1039:P(5)=2
     004:P(2)=1049:P(6)=2014:P(3)=1059
4120 SYS 49152
4130 POKE P(0),227
4140 IF PEEK(2)=255 THEN 20000
4150 IF PEEK(653)=1 THEN 4150
4160 IF RND(1)>.05 THEN 4140
4170 IF RND(1)>.5 THEN 4200
4180 P=RND(1)*5:IF PEEK(P(P))<>160 THEN 4
     180
4190 POKE P(P),227:GOTO4140
4200 P=RND(1)*3+4:IF PEEK(P(P))<>160 THEN
      4200
4210 POKE P(P),228:GOTO4140
5000 FB=(J AND 16)
5010 IFFL=0AND FB=0 THEN POKE 2,0:POKE 49
     290,2:GOTO 5030
5020 IF FL=16 AND FB=16 THEN POKE 2,0:POK
     E 49290,3
5030 FL=FB
5040 IF (J AND 8)=0 AND DX<4 THEN DX=DX+1
     :GOTO5060
5050 IF (J AND 4)=0 AND DX>-4 THEN DX=DX-1
5060 IF (J AND 1)=0 AND DY>-4 THEN DY=DY-
     1:GOTO5080
5070 IF (J AND 2)=0 AND DY<4 THEN DY=DY+1
5080 X=PEEK(V)+PEEK(V+16)*256
5085 Y=PEEK(V+1)
```



It sits on the sands of Egypt–silent, foreboding; the blue waters of the Nile sliding silently by. Pharoah's Pyramid beckons to us across millennia; mysterious, alluring. Where did the stones come from? How were they laid one on top of another? How many lives were lost under hot suns and cool moons to provide a resting place for the pharoahs of Egypt? Now you can have a personal stake in its completion. Match wits with the ancient Egyptian God, Osiris. Take advantage of gifts given you by the beautiful Isis. Fight through the plagues of Moses as brick by brick you build one of the wonders of the world. There hasn't been a greater challenge since the original Pharoah's Pyramid!

HISTORY PROVES THAT ALL GREAT GAMES HAVE THREE THINGS IN COMMON.

FAST-PACED FUN THAT LASTS — EVER INCREASING CHALLENGE — GRAPHICS THAT LIVE

PHAROAH'S PYRAMID is a fast action game with that special kind of challenge that makes you say... "This time I go all the way!"

AVAILABLE NOW FROM... Master Control Software Inc.

"The company that loves great games as much as you do, see your local dealer or call toll free. 1•800•624•5596

P.O. BOX 26714 • SALT LAKE CITY • UTAH • 84126

```
5090 NX=X+DX:IF NX>21 AND NX<358 THEN POK
     E V,NXAND255:POKEV+16,NX/256
5100 NY=Y+DY:IF NY<20 THEN NY=210
5110 IF NY>210 THEN NY=20
5120 POKE V+1,NY
5130 RETURN
10000 I=15616:TI$="000000"
10005 PRINT"{HOME}{WHT}{12 RIGHT}READY IN
      "LEFT$(STR$(149-INT(TI/60)),4)" SEC
      ONDS "
10010 READ A:IF A=256 THEN 10025
10020 C1=C1+A:POKE I,A:I=I+1:GOTO 10005
10025 IF C1<>34430 THEN PRINT"CHECKSUM ER
      ROR IN LINE 10025":END
10026 RETURN
10030 DATA 0,0,0,0,0,0,0
10040 DATA 0,0,24,0,0,28,0
10050 DATA 0,31,0,0,31,255,240
10060 DATA 31,255,8,20,255,254,31
10070 DATA 127,255,30,63,254,24,0
10080 DATA 0,0,0,0,0,0,0
10090 DATA 0,0,0,0,0,0,0
10100 DATA 0,0,0,0,0,0,0
10110 DATA 0,0,0,0,0,0,0
10120 DATA 0,0,71,192,0,247,192
10130 DATA 0,247,192,1,255,192,2
10140 DATA 255,192,2,255,192,2,247
10150 DATA 192,2,247,192,3,247,192
10160 DATA 3,247,192,3,247,192,3
10170 DATA 247,192,3,247,192,3,247
10180 DATA 192,3,255,192,3,255,192
10190 DATA 7,103,192,7,103,192,15
10200 DATA 229,128,31,119,128,31,240
10210 DATA 0,0,0,0,0,0,0
10220 DATA 0,0,0,0,0,0,0
10230 DATA 0,0,0,0,0,0,0
10240 DATA 0,0,0,0,0,0,0
10250 DATA 0,0,0,0,0,0,0
10260 DATA 0,0,0,0,0,0,0
10270 DATA 0,0,0,0,0,0,0
10280 DATA 0,0,3,128,0,15,192
10290 DATA 0,15,192,0,15,192,0
10300 DATA 15,192,0,1,252,0,1
10310 DATA 116,0,1,212,0,0,88
10320 DATA 0,0,80,0,0,0,0
10330 DATA 0,0,0,0,0,0,0
10340 DATA 0,0,0,0,0,0,0
10350 DATA 0,0,0,0,0,0,0
10360 DATA 0,0,0,0,0,0,0
10370 DATA 0,0,0,0,0,0,0
10380 DATA 0,0,0,0,0,0,0
10390 DATA 0,0,0,0,1,252,0
10400 DATA 1,252,0,1,252,0,1
10410 DATA 254,0,7,248,0,6,249
10420 DATA 0,2,251,0,6,122,0
10430 DATA 3,242,0,0,248,0,0
10440 DATA 248,0,0,60,0,0,120
10450 DATA 0,0,56,0,0,56,0
10460 DATA 0,96,0,0,96,0,0
10470 DATA 8,0,0,32,0,0,0
10480 DATA 0,0,0,0,0,0,64
10490 DATA 0,0,240,0,0,240,0
10500 DATA 1,240,0,2,240,0,2
10510 DATA 240,0,2,240,0,2,240
10520 DATA 0,3,240,0,3,240,0
10530 DATA 3,240,0,3,240,0,3
10540 DATA 240,0,3,240,0,3,240
10550 DATA 0,3,240,0,7,96,0
10560 DATA 7,96,0,15,224,0,31
10570 DATA 112,0,31,240,0,0,0
10580 DATA 7,192,0,7,192,0,7
10590 DATA 192,0,7,192,0,7,192
10600 DATA 0,7,192,0,7,192,0
10610 DATA 7,192,0,7,192,0,7
10620 DATA 192,0,7,192,0,7,192
10630 DATA 0,7,192,0,7,192,0
10640 DATA 7,192,0,7,192,0,7
10650 DATA 192,0,7,192,0,7,192
10660 DATA 0,3,128,0,0,0,0
10670 DATA 0,2,0,0,7,192,0
10680 DATA 7,192,0,6,192,0,4
10690 DATA 192,0,3,64,0,6,192
10700 DATA 0,1,192,0,4,0,0
10710 DATA 7,192,0,7,128,0,7
10720 DATA 64,0,7,192,0,1,192
10730 DATA 0,5,192,0,6,64,0
10740 DATA 7,192,0,7,192,0,0
10750 DATA 128,0,3,128,0,0,0
10760 DATA 0,0,2,0,0,1,0
10770 DATA 0,6,64,0,0,64,0
10780 DATA 4,128,0,3,64,0,6
10790 DATA 0,0,1,0,0,0,0
10800 DATA 0,0,0,0,0,128,0
10810 DATA 1,64,0,6,0,0,1
10820 DATA 0,0,5,0,0,6,64
10830 DATA 0,0,0,0,4,0,0
10840 DATA 0,128,0,3,128,0,0
10850 DATA 0,0,0,0,0,0,0
10860 DATA 0,0,0,0,0,0,0
10870 DATA 0,0,0,0,0,0,0
10880 DATA 0,0,0,0,0,0,0
10890 DATA 0,0,0,0,0,0,0
10900 DATA 0,0,0,0,0,0,0
10910 DATA 0,0,0,0,0,0,0
10920 DATA 0,0,1,128,0,6,128
10930 DATA 0,2,64,0,5,192,0
10940 DATA 3,128,0,1,252,0,1
10950 DATA 252,0,1,236,0,1,126
10960 DATA 0,3,248,0,2,120,0
10970 DATA 0,248,0,0,120,0,0
10980 DATA 112,0,0,120,0,0,120
10990 DATA 0,0,48,0,0,48,0
11000 DATA 0,0,0,0,0,0,0
11010 DATA 0,0,0,0,0,0,0
11020 DATA 0,0,0,0,0,0,0
11030 DATA 0,0,0,0,0,0,0
11040 DATA 0,16,0,0,16,0,16
11050 DATA 56,16,10,16,160,4,16
11060 DATA 64,10,124,160,1,255,0
11070 DATA 1,255,0,11,255,144,127
11080 DATA 255,252,11,255,144,1,255
11090 DATA 0,1,255,0,10,124,160
11100 DATA 4,16,64,10,16,160,16
11110 DATA 56,16,0,16,0,0,16
11120 DATA 0,0,0,0,0,0,256
20000 SC=0:FOR I=0 TO 4:SC=SC+(PEEK(1148-
      I)-48)*10↑I:NEXT I
20010 IF H<SC THEN H=SC
20020 POKE S+4,128
20030 POKE 13*4096+21,0
20040 FOR I=1 TO 1000:NEXT I
20050 PRINT"{CLR}OUT OF FUEL...{DOWN}
20060 PRINT"YOU SCORED{WHT}"SC"{CYN}POINTS
20070 PRINT"HIGH SCORE{WHT}"H"{CYN}
20080 PRINT"{3 DOWN}{11 SPACES}AGAIN? (Y
      {SPACE}OR N)"
20090 PRINT"{DOWN} OR PRESS FIRE BUTTON T
      O START AGAIN"
20100 GETA$
20110 IF A$="N"THEN END
20120 IF (PEEK(56320) AND 16)=0 THEN GOTO
      4000
```

FROM THE ARCADES OF TOMORROW . . .

A MICROCOMPUTER EXPERIENCE FOR TODAY

Arcaders who've seen and played the ATARI and TRS-80 versions of STRATOS came to the same conclusions — these state-of-the-art games were ahead of their time. After all, any program that boasts crisp graphics, punchy sounds, joystick compatability and a full complement of extras, like high score saving and multi-player option has a definite touch of tomorrow.

THE GAME'S SCENARIO IS A REAL KNOCKOUT

The object is to successfully defend a futuristic city from waves of attacking alien ships.

ATARI SCREEN SHOT
The battle begins! Atari version features colors and sounds that YOU can change!

And these crafty alien critters are just part of a rapid-fire graphics bonanza that includes meteor swarms, multiple attack waves, and even a free-moving saucer that will repair your city's damaged force field on the ATARI version.

GOOD NEWS

You don't have to wait years for the spectacular — STRATOS is available now for the ATARI and TRS-80 systems.

STRATOS. Entertainment of the future — today.

TRS-80 SCREEN SHOT
The TRS-80 version blows you away with its Arcade Action Graphics(tm)!

© COPYRIGHT 1982

Adventure INTERNATIONAL

To order, see your local dealer. If he does not have the program, then call 1-800-327-7172 (orders only please) or write for our free catalog.

Published by ADVENTURE INTERNATIONAL
a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 830-8194

ARCADE GRAPHICS ACTION TM

AVAILABLE NOW!

ATARI 400/800/1200 16K TAPE	160-0161	$34.95
ATARI 400/800/1200 32K DISK	162-0161	$34.95
COMMODORE 64 TAPE	160-0161	$34.95
COMODORE 64 DISK	162-0161	$34.95
TRS-80 16K TAPE	010-0161	$24.95
TRS-80 32K DISK	012-0161	$24.95

Coming Soon For The Color Computer!

PRICES SUBJECT TO CHANGE

```
20130 IF A$<>"Y" THEN 20100
20140 GOTO4000
50000 I=49152:TI$="000000"
50010 PRINT"{HOME}{WHT}{12 RIGHT}READY IN
      "LEFT$(STR$(103-INT(TI/60)),4)" SEC
      ONDS "
50015 READ A:IF A=256 THEN PRINT"{HOME}
      {10 RIGHT}{21 SPACES}{SHIFT-SPACE}"
      :GOTO50045
50020 IF A=-1 THEN I=49920 : GOTO 50010
50030 IF A=-2 THEN I=50688 : GOTO 50010
50040 C2=C2+A:POKE I,A:I=I+1:GOTO 50010
50045 IF C2<>188431 THEN PRINT"CHECKSUM E
      RROR IN LINE 50045":END
50046 RETURN
50050 DATA 120,169,0,141,20,3,169
50060 DATA 195,141,21,3,88,173,14
50070 DATA 220,41,254,141,14,220,165
50080 DATA 1,41,251,133,1,160,0
50090 DATA 185,0,208,153,0,48,185
50100 DATA 0,50,153,0,50,185,0
50110 DATA 209,153,0,49,185,0,211
50120 DATA 153,0,51,185,0,212,153
50130 DATA 0,52,185,0,213,153,0
50140 DATA 53,185,0,214,153,0,54
50150 DATA 185,0,215,153,0,55,169
50160 DATA 15,141,156,200,200,208,200
50170 DATA 165,1,9,4,133,1,173
50180 DATA 14,220,9,1,141,14,220
50190 DATA 169,28,141,24,208,169,15
50200 DATA 141,156,200,169,255,141,15
50210 DATA 212,169,128,141,18,212,169
50220 DATA 0,133,2,141,224,207,141
50230 DATA 255,207,141,254,207,141,253
50240 DATA 207,141,252,207,141,249,207
50250 DATA 160,6,169,20,153,0,207
50260 DATA 169,0,153,16,207,136,208
50270 DATA 243,169,251,141,251,207,160
50280 DATA 0,169,4,133,252,132,251
50290 DATA 169,216,133,254,132,253,169
50300 DATA 160,160,5,145,251,160,10
50310 DATA 145,251,160,15,145,251,160
50320 DATA 20,145,251,160,25,145,251
50330 DATA 160,30,145,251,160,35,145
50340 DATA 251,165,251,24,105,40,133
50350 DATA 251,144,2,230,252,201,232
50360 DATA 208,211,169,1,160,10,145
50370 DATA 253,169,4,160,5,145,253
50380 DATA 169,7,160,15,145,253,169
50390 DATA 14,160,20,145,253,169,8
50400 DATA 160,25,145,253,169,13,160
50410 DATA 30,145,253,169,3,160,35
50420 DATA 145,253,165,253,24,105,40
50430 DATA 133,253,144,2,230,254,201
50440 DATA 232,208,199,96,-1
50450 DATA 173,141
50460 DATA 2,201,1,208,3,76,49
50470 DATA 234,230,2,165,2,201,2
50480 DATA 240,3,76,49,234,169,0
50490 DATA 133,2,169,3,133,252,169
50500 DATA 216,133,251,160,45,177,251
50510 DATA 32,79,195,160,55,177,251
50520 DATA 32,79,195,160,65,177,251
50530 DATA 32,79,195,160,75,177,251
50540 DATA 32,79,195,165,251,24,105
50550 DATA 40,133,251,144,2,230,252
50560 DATA 201,192,208,213,76,0,198
50570 DATA 201,160,240,19,201,32,240
50580 DATA 37,162,1,232,221,174,195
50590 DATA 208,250,202,189,174,195,145
50600 DATA 251,96,152,56,233,40,168
50610 DATA 177,251,201,32,240,1,96
50620 DATA 152,24,105,40,168,169,227
50630 DATA 145,251,96,165,252,201,3
50640 DATA 240,22,152,56,233,40,168
50650 DATA 177,251,201,160,240,1,96
50660 DATA 152,24,105,40,168,169,99
50670 DATA 145,251,96,152,24,105,120
50680 DATA 168,177,251,201,100,240,1
50690 DATA 96,152,56,233,120,168,169
50700 DATA 99,145,251,96,160,228,239
50710 DATA 249,226,120,119,99,32,32
50720 DATA 100,111,121,98,248,247,227
50730 DATA -2,169,7,133,252
50740 DATA 169,32,133,251,160,170,177
50750 DATA 251,32,47,198,160,180,177
50760 DATA 251,32,47,198,160,190,177
50770 DATA 251,32,47,198,165,251,56
50780 DATA 233,40,133,251,176,2,198
50790 DATA 252,201,56,208,220,76,160
50800 DATA 198,201,160,240,19,201,32
50810 DATA 240,37,162,1,232,221,142
50820 DATA 198,208,250,202,189,142,198
50830 DATA 145,251,96,152,24,105,40
50840 DATA 168,177,251,201,32,240,1
50850 DATA 96,152,56,233,40,168,169
50860 DATA 228,145,251,96,165,251,201
50870 DATA 32,240,22,152,24,105,40
50880 DATA 168,177,251,201,160,240,1
50890 DATA 96,152,56,233,40,168,169
50900 DATA 100,145,251,96,152,56,233
50910 DATA 120,168,177,251,201,99,240
50920 DATA 1,96,152,24,105,120,168
50930 DATA 169,100,145,251,96,32,99
50940 DATA 119,120,226,249,239,228,160
50950 DATA 160,227,247,248,98,121,111
50960 DATA 100,32,173,0,220,72,41
50970 DATA 15,201,15,240,8,169,129
50980 DATA 141,4,212,76,183,198,169
50990 DATA 128,141,4,212,104,41,16
51000 DATA 205,255,207,240,48,141,255
51010 DATA 207,201,16,208,24,169,2
51020 DATA 141,15,195,169,1,141,252
51030 DATA 198,141,229,200,169,0,141
51040 DATA 250,207,141,224,207,76,239
51050 DATA 198,169,1,141,15,195,169
51060 DATA 2,141,252,198,141,229,200
51070 DATA 169,0,133,2,32,245,198
51080 DATA 76,32,200,238,250,207,173
51090 DATA 250,207,201,1,240,1,96
51100 DATA 169,0,141,250,207,173,0
51110 DATA 220,141,254,207,41,1,208
51120 DATA 13,173,253,207,201,253,240
51130 DATA 23,206,253,207,76,45,199
51140 DATA 173,254,207,41,2,208,10
51150 DATA 173,253,207,201,3,240,3
51160 DATA 238,253,207,173,254,207,41
51170 DATA 8,208,13,173,252,207,201
51180 DATA 3,240,23,238,252,207,76
51190 DATA 82,199,173,254,207,41,4
51200 DATA 208,10,173,252,207,201,253
51210 DATA 240,3,206,252,207,173,254
51220 DATA 207,41,3,201,3,208,16
51230 DATA 173,253,207,240,11,16,6
51240 DATA 238,253,207,76,107,199,206
51250 DATA 253,207,173,254,207,41,12
51260 DATA 201,12,208,16,173,252,207
51270 DATA 240,11,16,6,238,252,207
51280 DATA 76,132,199,206,252,207,174
51290 DATA 249,207,208,32,174,240,207
51300 DATA 224,60,176,25,173,253,207
51310 DATA 24,109,1,208,201,80,176
```

```
51320 DATA 5,169,244,76,191,199,201
51330 DATA 244,144,27,169,80,76,191
51340 DATA 199,173,253,207,24,109,1
51350 DATA 208,201,41,176,5,169,244
51360 DATA 76,191,199,201,244,144,2
51370 DATA 169,41,141,1,208,173,252
51380 DATA 207,48,32,24,109,0,208
51390 DATA 141,240,207,173,249,207,105
51400 DATA 0,141,249,207,201,1,208
51410 DATA 42,173,240,207,201,55,144
51420 DATA 35,32,155,200,76,4,200
51430 DATA 24,109,0,208,141,240,207
51440 DATA 173,249,207,105,255,141,249
51450 DATA 207,208,12,173,240,207,201
51460 DATA 25,176,5,169,25,141,240
51470 DATA 207,173,240,207,141,0,208
51480 DATA 173,16,208,41,254,13,249
51490 DATA 207,141,16,208,173,31,208
51500 DATA 41,1,240,3,76,101,200
51510 DATA 96,162,5,189,119,4,201
51520 DATA 57,240,6,254,119,4,76
51530 DATA 58,200,169,48,157,119,4
51540 DATA 202,208,235,76,58,200,162
51550 DATA 5,189,39,4,201,48,240
51560 DATA 6,222,39,4,76,222,200
51570 DATA 169,57,157,39,4,202,208
51580 DATA 235,120,169,234,141,21,3
51590 DATA 169,49,141,20,3,88,169
51600 DATA 255,133,2,76,222,200,0
51610 DATA 162,0,160,240,238,32,208
51620 DATA 232,208,250,200,208,247,169
51630 DATA 0,141,32,208,162,3,189
51640 DATA 39,4,201,48,240,4,222
51650 DATA 39,4,96,169,57,157,39
51660 DATA 4,202,208,237,162,5,169
51670 DATA 48,157,39,4,202,208,250
51680 DATA 104,104,76,81,200,160,15
51690 DATA 162,3,189,39,4,201,57
51700 DATA 240,6,254,39,4,76,180
51710 DATA 200,169,48,157,39,4,202
51720 DATA 208,235,136,208,230,169,0
51730 DATA 141,249,207,169,25,141,240
51740 DATA 207,169,148,141,1,208,172
51750 DATA 156,200,192,9,240,4,136
51760 DATA 140,156,200,173,5,4,201
51770 DATA 160,208,5,169,227,141,5
51780 DATA 4,96,238,224,207,173,224
51790 DATA 207,201,1,240,3,76,124
51800 DATA 201,169,0,141,224,207,173
51810 DATA 27,212,201,7,176,25,168
51820 DATA 185,0,207,201,20,208,8
51830 DATA 169,1,153,16,207,76,16
51840 DATA 201,201,255,208,5,169,255
51850 DATA 153,16,207,160,6,185,0
51860 DATA 207,24,121,16,207,153,0
51870 DATA 207,72,152,10,170,104,157
51880 DATA 1,208,136,208,235,160,6
51890 DATA 185,0,207,201,20,240,10
51900 DATA 201,255,240,6,136,208,242
51910 DATA 76,66,201,169,0,153,16
51920 DATA 207,76,52,201,173,30,208
51930 DATA 41,1,240,51,162,0,160
51940 DATA 240,238,32,208,232,208,250
51950 DATA 200,208,247,169,0,141,32
51960 DATA 208,162,3,189,39,4,201
51970 DATA 48,240,6,222,39,4,76
51980 DATA 49,234,169,57,157,39,4
51990 DATA 202,208,235,162,5,169,48
52000 DATA 157,39,4,202,208,250,76
52010 DATA 49,234,256
```
©

Mosaic Puzzle

Bruce Jordan

Remember that once-popular sliding-squares game? With only one free space, you tried to move the colored plastic tiles around to get a particular sequence or color pattern. Although it was a challenge, this computer variation of the game can be a mind-boggling test of skill and dexterity. Versions for VIC, 64, Atari, and the TI-99/4A.

"Mosaic Puzzle" is a computer version of those sliding-squares puzzles that used to drive people nuts before the advent of Rubik's Cube. It can run on either an unexpanded or 3K expanded VIC. The object of the game is to arrange the 15 numbered squares (hexadecimal numbers 1-F in this version) into some predetermined order by sliding them around in their frame. The first few moves are easy, but as the game progresses, it gets a lot more complicated. You'll find yourself rearranging everything just to get the last few squares in place.

This version of the game has a timer for up to 23 hours, 59 minutes, 59 seconds, and a chicken switch. It also automatically checks for the winning order and allows you to go back to the puzzle the way you left it or reset it to the beginning arrangement.

When you start the game, you're asked if you wish to set a time limit. If you answer Y for yes, enter the time limit in one line with no spaces or punctuation between the values. For example, for a 1-hour, 23-minute limit, enter 012300.

Next, enter the goal order. This will be the order that you will try to match to win the game. When this is done, the upper half of the screen will clear, and the puzzle will appear. A moment later, the message !GO! will flash on the screen, along with a tone. The controls for moving the squares are as follows:

@ up
?/ down
= right
: left

A game is just underway in the TI version of "Mosaic Puzzle."

If you succeed in getting the squares in the goal order, the message YOU WIN! appears on the screen, accompanied by a short tune and the elapsed time. If the time runs out before you are finished, you'll hear an unpleasant sound. If you want to stop the game, press RETURN and the screen will display the elapsed time. You can then restart the game, either as you left it or reset, by hitting RETURN a second time.

Below is a brief description of the program as originally written on the VIC.

Line 1 sets the limit of memory at 7600. This gives a place to store the image of the puzzle.

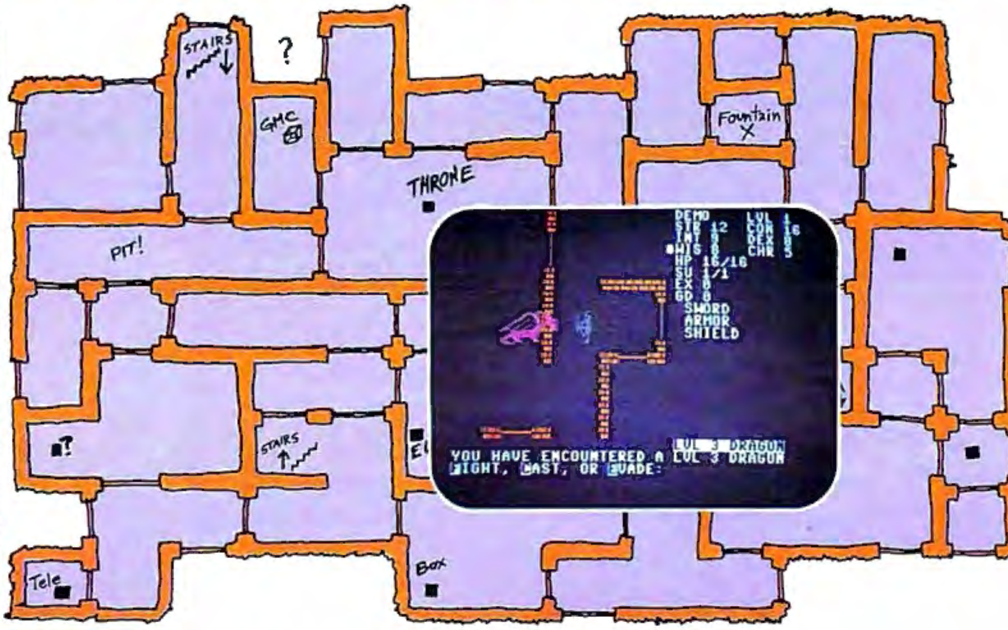
Lines 2-5 define variables. Note that S, SC, and SS are defined in two consecutive lines. This is done because there are two possible entry points when restarting the game.

Lines 10-66 GET the time limit.

Lines 70-290 GET the goal order and make sure that no number is entered more than once.

Lines 300-365 put the puzzle on the screen, either from the data table or from memory, then

Be Amazed!



Telengard: How low can you go?

We've created a subterranean monster. Fifty stories low.

That's the number of levels in the TELENGARD dungeon.

Each labyrinthine level holds hundreds of dark chambers and tomb-like corridors for the mighty adventurer to explore. It goes without saying that a shifting collection of hideous monsters with unpredictable behavior patterns can make life in the TELENGARD maze quite interesting—and frequently quite short!

Using wits, magic and true grit, your character delves deeper and deeper into the depths of TELENGARD in this realtime fantasy role-playing game. Available on cassette for Commodore® 64, Atari® Home Computers (40K), TRS-80® Mods. I/III (32K) and PET® 2001 (32K) for a ghoulish $23.00. 48K diskettes for Apple® II, Atari®, Commodore® 64 and TRS-80® available also, for $28.00.

AND FOR THOSE WHO DON'T DIG UNDERGROUND GAMES . . . There's **B-1 NUCLEAR BOMBER**, a nail-biting solitaire simulation of a manned B-1 on a mission over the Soviet Union. Your plane is equipped with six Phoenix Missiles, a one megaton warhead and orders to retaliate! Cassette for Commodore 64, Atari Home Computers (32K), TI99/4 & 4A (16K), VIC-20 (16K), Timex/Sinclair 1000 (16K), and TRS-80 Mods. I/III (16K) are available for an explosive $16.00. Diskette versions for Apple (48K), TRS-80 (32K), Atari (24K) and IBM (48K) just $21.00.

NUKEWAR: Defend your country by massive espionage efforts, or by building jet fighter bombers, missiles, submarines and ABM's. Your cold and calculating computer will choose its own strategy! Cassette for Commodore 64, VIC-20 (16K), TRS-80 Mods. I/III (16K) and Atari Home Computers just $16.00.

FOOTBALL STRATEGY: Animated action on a scrolling field. A head-to-head challenge or solitaire as you select the best offensive or defensive plays in response to your opponent. Commodore 64, Atari Home Computers (32K) and TRS-80 Models I/III/IV (16K) cassette for $16.00. Atari (32K), IBM (64K) and TRS-80 Models I/III/IV (32K) diskettes available at $21.00.

T.G.I.F.: Thank Goodness It's Friday! Avalon Hill's new party game for one to four players recreating an often-not-so-typical week in the lives of the working class. Half the fun is just making it from Monday to Sunday. Commodore 64, Atari Home Computers (40K) cassette for a meager $20.00. Atari diskette (48K) for $25.00.

AVAILABLE WHEREVER GOOD COMPUTER GAMES ARE SOLD or call Toll-Free: 1 (800) 638-9292 for the name of a store near you. Ask for Operator C.

microcomputer games®

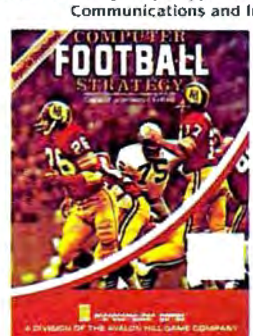
a Division of the

Avalon Hill Game Company

4517 Harford Road, Baltimore, MD 21214 U.S.A. Earth

* Trademarks of Tandy Corp., Apple Computers, Commodore Business Machines, Warner Communications and International Business Machines.

AH
QUALITY



print the go message and start the timer.

Lines 360-560 check the timer, move the number squares, and check for the winning order.

Lines 570-710 print the winning or losing message, display the elapsed time, and play the sound effects.

Lines 720-740 save the position of the blank square, then clear the variables and reset the game.

Lines 745-790 are the DATA statements for the puzzle's beginning order and the winning tune.

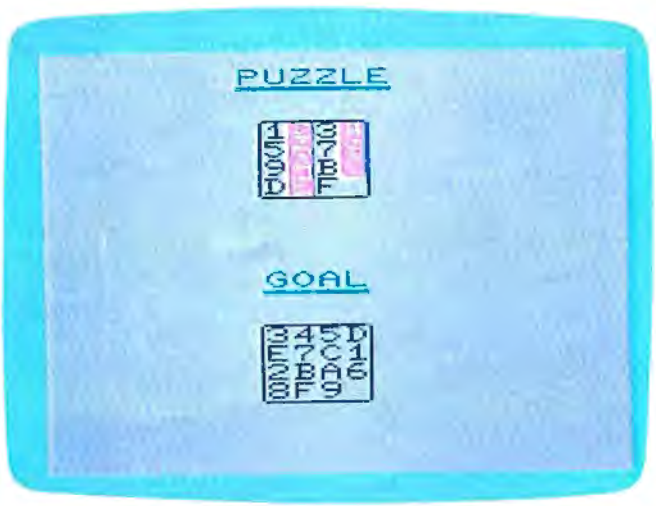
Program 1: Mosaic Puzzle – VIC Version

> BEGINNING PROGRAMMERS
> **If you're new to computing, please read "How To Type COMPUTE!'s Programs" and "A Beginner's Guide To Typing In Programs."**

```
1 POKE55,176:POKE56,29:CLR
2 S=7845:SC=38565:SS=7603:GOTO4
3 SS=7603:S=PEEK(SS)*256+PEEK(SS+1):SC=S+
  30720
4 DIMA$(16)
5 PRINT"{CLR}":G=8018:X=0:DX=1:P=38738:V=
  36878:S1=36876:S2=36877:POKEV,15
10 PRINT"TIME LIMIT? {RVS}{GRN}Y{OFF}
   {RVS}{PUR}N{OFF}{BLU}"
20 GETA$:IFA$=""THEN20
30 IFA$<"N"ORA$>"Y"THEN20
40 IFA$=>"O"ANDA$<="X"THEN20
50 IFA$="N"THEN70
60 PRINT"{CLR}":INPUT"HRS:MINS:SEC";T$:H=
   1:IFLEN(T$)<>6THEN60
62 IFLEFT$(T$,2)>"23"ORLEFT$(T$,2)<"0"THE
   N60
64 IFMID$(T$,3,2)>"59"ORMID$(T$,3,2)<"0"T
   HEN60
66 IFRIGHT$(T$,2)>"59"ORRIGHT$(T$,2)<"0"T
   HEN60
70 PRINT"{CLR}"TAB(24)"TYPE IN GOAL ORDER
   "
80 PRINT"{DOWN}{3 SPACES}1 2 3 4 5 6 7 8
   {SPACE}9"SPC(8)"A B C D E F {RVS}S
   {OFF}"TAB(49)"IN ANY ORDER"
90 PRINTTAB(118)"GOAL"TAB(30)"{4 Y}"
100 FORK=0TO3:POKE7996+K,100:POKE8106+K,9
    9:POKE38716+K,0:POKE38826+K,0:NEXTK
110 FORK=22TO88STEP22:POKE7995+K,103:POKE
    8000+K,101:POKE38715+K,0:POKE38720+K,
    0:NEXTK
130 FORI=1TO16STEP1
140 GETA$(I):IFA$(I)=""THEN140
150 FORL=I-1TO0STEP-1:IFA$(I)=A$(L)THEN14
    0
160 NEXTL
165 IFA$(I)="S"THEN190
170 IFA$(I)<="0"ORA$(I)=>"G"THEN140
180 IFA$(I)=>":"ANDA$(I)<="@"THEN140
190 B=VAL(A$(I)):C=B+48:IFB=0THEN220
200 POKE38467+2*B,7:POKEG+X,C:POKEP+X,0:X
    =X+DX:IFX=4THENG=G+22:P=P+22:X=0
210 NEXTI:IFI=17THEN300
220 IFA$(I)="A"THENB=1
230 IFA$(I)="B"THENB=2
240 IFA$(I)="C"THENB=3
250 IFA$(I)="D"THENB=4
260 IFA$(I)="E"THENB=5
270 IFA$(I)="F"THENB=6
272 B2=B
275 IFA$(I)="S"THENB=7:B2=32
280 POKE38492+2*B,7:POKEG+X,B2:POKEP+X,0:
    X=X+DX:IFX=4THENG=G+22:P=P+22:X=0
290 NEXTI
300 FORR=0TO132:POKE7702+R,32:POKE38422+R
    ,1:NEXTR
310 PRINT"{HOME}"TAB(29)"PUZZLE"TAB(29)"
    {6 Y}"
320 FORK=0TO3:POKE7754+K,100:POKE38474+K,
    0:POKE7864+K,99:POKE38584+K,0:NEXTK
330 FORK=22TO88STEP22:POKE7753+K,103:POKE
    38473+K,0:POKE7758+K,101:POKE38478+K,
    0:NEXTK
335 IFP1=0THEN340
336 FORK=0TO66STEP22:FORJ=0TO3:POKE7776+J
    +K,PEEK(7605+J+K)
337 POKE38496+J+K,PEEK(7609+J+K):NEXTJ:NE
    XTK:GOTO352
340 READA,B,C:IFA=-1THEN355
350 POKE7776+A,B:POKE38496+A,C:GOTO340
352 READA,B,C:IFA=-1THEN355
353 GOTO352
355 FORT=1TO1500:NEXT
360 POKES1,235:POKEV,15:PRINT"{HOME}"TAB(
    228)"{RVS}{RED}!GO!{OFF}{BLU}"
365 FORT=1TO500:NEXT:PRINT"{HOME}"TAB(228
    )"{4 SPACES}":POKES1,0:TI$="000000"
370 IFH<>1THEN380
375 IFT$<=TI$THENT$=TI$:GOTO600
380 GETB$:IFB$=""THEN370
390 D=ASC(B$):ON-(D=47)-2*(D=58)-3*(D=61)
    -4*(D=64)-5*(D=13)GOTO410,440,470,500
    ,620
400 GOTO370
410 IFPEEK(S-22)=100THEN370
420 POKES,PEEK(S-22):POKESC,PEEK(SC-22):P
    OKES-22,32:POKESC-22,1:S=S-22:SC=SC-2
    2
430 GOSUB530:GOTO370
440 IFPEEK(S+1)=101THEN370
450 POKES,PEEK(S+1):POKESC,PEEK(SC+1):POK
    ES+1,32:POKESC+1,1:S=S+1:SC=SC+1
460 GOSUB530:GOTO370
470 IFPEEK(S-1)=103THEN370
480 POKES,PEEK(S-1):POKESC,PEEK(SC-1):POK
    ES-1,32:POKESC-1,1:S=S-1:SC=SC-1
490 GOSUB530:GOTO370
500 IFPEEK(S+22)=99THEN370
510 POKES,PEEK(S+22):POKESC,PEEK(SC+22):P
    OKES+22,32:POKESC+22,1:S=S+22:SC=SC+2
    2
520 GOSUB530:GOTO370
530 FORM=0TO66STEP22:FORN=0TO3STEP1
540 W=PEEK(7776+M+N):IFW-128>0THENW=W-128
550 IFW<>PEEK(8018+M+N)THENRETURN
560 NEXTN:NEXTM
570 T$=TI$:PRINT"{HOME}"TAB(226)"{RVS}
    {YEL}YOU WIN!{OFF}{BLU}"
580 READN,D:IFN=-1THEN620
585 POKES1,N:FORT=1TOD:NEXT:GOTO580
600 PRINT"{HOME}"TAB(225)"{RVS}{GRN}!YOU
    {SPACE}LOSE!{OFF}{BLU}":POKES2,135:PO
    KES1,128:FORT=1TO500:NEXT
610 POKES2,0:POKES1,0
620 T$=TI$:FORK=0TO66STEP22:FORJ=0TO3:POK
    E7605+K+J,PEEK(7776+K+J)
630 POKE7609+K+J,PEEK(38496+K+J):NEXTJ:NE
```


NÜFEKOP
P.O. Box 156, 21255 Hwy. 62,
Shady Cove, OR 97539
1-800-525-2529
Software for the VIC 20™
and Commodore 64™
3-D MAN
THE CATCH
DEFENDER ON TRI
Exterminator
VIC 20 programming
NÜFEKOP

```
    XTK
700 PRINTTAB(199)"HIT RETURN TO PLAY"SPC(
    10)"AGAIN"
710 PRINT"{3 RIGHT}TIME="LEFT$(T$,2);":"M
    ID$(T$,3,2);":"RIGHT$(T$,2)
720 GETC$:IFC$=""THEN720
730 IFC$=CHR$(13)THENPOKESS,INT(S/256):PO
    KESS+1,SAND255:CLR:GOTO736
735 GOTO720
736 PRINT"{CLR}RESET.............{RVS}
    {GRN}1{OFF}{BLU}{HOME}{2 DOWN}OR AS Y
    OU LEFT IT.{RVS}{PUR}2{OFF}{BLU}"
737 GETV$:IFV$=""THEN737
738 IFV$<"1"ORV$>"2"THEN737
739 IFV$="1"THEN2
740 IFV$="2"THENP1=1:GOTO3
745 DATAØ,49,Ø,1,178,4,2,51,Ø,3,180,4
750 DATA22,53,Ø,23,182,4,24,55,Ø,25,184,4
760 DATA44,57,Ø,45,129,4,46,2,Ø,47,131,4
770 DATA66,4,Ø,67,133,4,68,6,Ø,69,32,1
780 DATA-1,-1,-1
790 DATA208,150,Ø,50,208,75,Ø,50,208,75,2
    18,175,208,115,218,175,224,250,Ø,Ø,-1
    -1
```



"Mosaic Puzzle," VIC version.

Program 2: Mosaic Puzzle – 64 Version

Translation by Chris Metcalf, Programming Assistant

```
1 POKE55,176:POKE56,29:CLR
2 S=7845:SC=38565:SS=7603:GOTO4
3 SS=7603:S=PEEK(SS)*256+PEEK(SS+1):SC=S+
  30720
4 DIMA$(16)
5 PRINT"{CLR}":G=8018:X=Ø:DX=1:P=38738:V=
  36878:S1=36876:S2=36877:POKEV,15
10 PRINT"TIME LIMIT? {RVS}{GRN}Y{OFF}
   {RVS}{PUR}N{OFF}{BLU}"
20 GETA$:IFA$=""THEN20
30 IFA$<"N"ORA$>"Y"THEN20
40 IFA$=>"O"ANDA$<="X"THEN20
50 IFA$="N"THEN70
60 PRINT"{CLR}":INPUT"HRS:MINS:SEC";T$:H=
   1:IFLEN(T$)<>6THEN60
62 IFLEFT$(T$,2)>"23"ORLEFT$(T$,2)<"Ø"THE
   N60
64 IFMID$(T$,3,2)>"59"ORMID$(T$,3,2)<"Ø"T
   HEN60
66 IFRIGHT$(T$,2)>"59"ORRIGHT$(T$,2)<"Ø"T
   HEN60
70 PRINT"{CLR}"TAB(24)"TYPE IN GOAL ORDER
   "
80 PRINT"{DOWN}{3 SPACES}1 2 3 4 5 6 7 8
   {SPACE}9"SPC(8)"A B C D E F {RVS}S
   {OFF}"TAB(49)"IN ANY ORDER"
90 PRINTTAB(118)"GOAL"TAB(30)"[4 Y]"
100 FORK=ØTO3:POKE7996+K,100:POKE8106+K,9
    9:POKE38716+K,Ø:POKE38826+K,Ø:NEXTK
110 FORK=22TO88STEP22:POKE7995+K,103:POKE
    8000+K,101:POKE38715+K,Ø:POKE38720+K,
    Ø:NEXTK
130 FORI=1TO16STEP1
140 GETA$(I):IFA$(I)=""THEN140
150 FORL=I-1TOØSTEP-1:IFA$(I)=A$(L)THEN14
    Ø
160 NEXTL
165 IFA$(I)="S"THEN190
170 IFA$(I)<="Ø"ORA$(I)=>"G"THEN140
180 IFA$(I)=>":"ANDA$(I)<="@"THEN140
190 B=VAL(A$(I)):C=B+48:IFB=ØTHEN220
200 POKE38467+2*B,7:POKEG+X,C:POKEP+X,Ø:X
    =X+DX:IFX=4THENG=G+22:P=P+22:X=Ø
210 NEXTI:IFI=17THEN300
220 IFA$(I)="A"THENB=1
230 IFA$(I)="B"THENB=2
240 IFA$(I)="C"THENB=3
250 IFA$(I)="D"THENB=4
260 IFA$(I)="E"THENB=5
270 IFA$(I)="F"THENB=6
272 B2=B
275 IFA$(I)="S"THENB=7:B2=32
280 POKE38492+2*B,7:POKEG+X,B2:POKEP+X,Ø:
    X=X+DX:IFX=4THENG=G+22:P=P+22:X=Ø
290 NEXTI
300 FORR=ØTO132:POKE7702+R,32:POKE38422+R
    ,1:NEXTR
310 PRINT"{HOME}"TAB(29)"PUZZLE"TAB(29)"
    [6 Y]"
320 FORK=ØTO3:POKE7754+K,100:POKE38474+K,
    Ø:POKE7864+K,99:POKE38584+K,Ø:NEXTK
330 FORK=22TO88STEP22:POKE7753+K,103:POKE
    38473+K,Ø:POKE7758+K,101:POKE38478+K,
    Ø:NEXTK
335 IFP1=ØTHEN340
336 FORK=ØTO66STEP22:FORJ=ØTO3:POKE7776+J
    +K,PEEK(7605+J+K)
337 POKE38496+J+K,PEEK(7609+J+K):NEXTJ:NE
    XTK:GOTO352
340 READA,B,C:IFA=-1THEN355
350 POKE7776+A,B:POKE38496+A,C:GOTO340
352 READA,B,C:IFA=-1THEN355
353 GOTO352
355 FORT=1TO1500:NEXT
360 POKES1,235:POKEV,15:PRINT"{HOME}"TAB(
    228)"{RVS}{RED}!GO!{OFF}{BLU}"
365 FORT=1TO500:NEXT:PRINT"{HOME}"TAB(228
    )"{4 SPACES}":POKES1,Ø:TI$="ØØØØØØ"
370 IFH<>1THEN380
375 IFT$<=TI$THENT$=TI$:GOTO600
380 GETB$:IFB$=""THEN370
390 D=ASC(B$):ON-(D=47)-2*(D=58)-3*(D=61)
    -4*(D=64)-5*(D=13)GOTO410,440,470,500
    ,620
400 GOTO370
410 IFPEEK(S-22)=100THEN370
420 POKES,PEEK(S-22):POKESC,PEEK(SC-22):P
    OKES-22,32:POKESC-22,1:S=S-22:SC=SC-2
    2
430 GOSUB530:GOTO370
440 IFPEEK(S+1)=101THEN370
```

Commodore 64 Notes

Chris Metcalf, Programming Assistant

The Commodore 64 version of "Mosaic Puzzle" is very much the same as the original VIC version. However, some minor changes have been made. Either the RETURN key or the fire button allows you to pause momentarily before resuming the game, restarting the program, or stopping play entirely. Breaking off and resuming has no effect on the time clock (displayed at the top of the screen along with the time limit).

As an aid to the user, various keys for up, down, right, and left can be selected at the beginning of the game. A joystick can also be used, as long as it is plugged into control port two. The time limit is an option in this version; if no time limit is selected, the screen will display elapsed time and TIME LIMIT:NONE.

```
450 POKES,PEEK(S+1):POKESC,PEEK(SC+1):POKES+1,32:POKESC+1,1:S=S+1:SC=SC+1
460 GOSUB530:GOTO370
470 IFPEEK(S-1)=103THEN370
480 POKES,PEEK(S-1):POKESC,PEEK(SC-1):POKES-1,32:POKESC-1,1:S=S-1:SC=SC-1
490 GOSUB530:GOTO370
500 IFPEEK(S+22)=99THEN370
510 POKES,PEEK(S+22):POKESC,PEEK(SC+22):POKES+22,32:POKESC+22,1:S=S+22:SC=SC+22
520 GOSUB530:GOTO370
530 FORM=ØTO66STEP22:FORN=ØTO3STEP1
540 W=PEEK(7776+M+N):IFW-128>ØTHENW=W-128
550 IFW<>PEEK(8018+M+N)THENRETURN
560 NEXTN:NEXTM
570 T$=TI$:PRINT"{HOME}"TAB(226)"{RVS}{YEL}YOU WIN!{OFF}{BLU}"
580 READN,D:IFN=-1THEN620
585 POKES1,N:FORT=1TOD:NEXT:GOTO580
600 PRINT"{HOME}"TAB(225)"{RVS}{GRN}!YOU{SPACE}LOSE!{OFF}{BLU}":POKES2,135:POKES1,128:FORT=1TO500:NEXT
610 POKES2,Ø:POKES1,Ø
620 T$=TI$:FORK=ØTO66STEP22:FORJ=ØTO3:POKE7605+K+J,PEEK(7776+K+J)
630 POKE7609+K+J,PEEK(38496+K+J):NEXTJ:NEXTK
700 PRINTTAB(199)"HIT RETURN TO PLAY"SPC(10)"AGAIN"
710 PRINT"{3 RIGHT}TIME="LEFT$(T$,2);":"MID$(T$,3,2);":"RIGHT$(T$,2)
720 GETC$:IFC$=""THEN720
730 IFC$=CHR$(13)THENPOKESS,INT(S/256):POKESS+1,SAND255:CLR:GOTO736
735 GOTO720
736 PRINT"{CLR}RESET.............{RVS}{GRN}1{OFF}{BLU}{HOME}{2 DOWN}OR AS YOU LEFT IT.{RVS}{PUR}2{OFF}{BLU}"
737 GETV$:IFV$=""THEN737
738 IFV$<"1"ORV$>"2"THEN737
739 IFV$="1"THEN2
740 IFV$="2"THENP1=1:GOTO3
745 DATAØ,49,Ø,1,178,4,2,51,Ø,3,18Ø,4
750 DATA22,53,Ø,23,182,4,24,55,Ø,25,184,4
760 DATA44,57,Ø,45,129,4,46,2,Ø,47,131,4
770 DATA66,4,Ø,67,133,4,68,6,Ø,69,32,1
780 DATA-1,-1,-1
790 DATA2Ø8,15Ø,Ø,5Ø,2Ø8,75,Ø,5Ø,2Ø8,75,218,175,2Ø8,115,218,175,224,25Ø,Ø,Ø,-1,-1
```

"Mosaic Puzzle," 64 version.

Program 3: Mosaic Puzzle – Atari Version

Translation by Marc Sugiyama, Programming Assistant

```
110 DIM ANS$(1),PUZZLE$(16),GOAL$(16),TEMP$(1)
120 PUZZLE$="123456789ABCDEF ":GOAL$="{16 SPACES}"
130 CHBASE=PEEK(106)*256:IF PEEK(CHBASE+17)=102 AND PEEK(CHBASE+470)=126 THEN 190
140 POKE 106,PEEK(106)-6:GRAPHICS 2+16:SETCOLOR Ø,Ø,Ø:POSITION 2,6:PRINT #6;"PLEASE STAND BY"
150 CHBASE=PEEK(106)*256
160 FOR I=Ø TO 479:POKE CHBASE+I,PEEK(57344+I):POKE 708,PEEK(5377Ø):NEXT I
170 FOR I=48Ø TO 487:POKE CHBASE+I,1:POKE CHBASE+8+I,128:NEXT I
180 FOR I=496 TO 511:POKE CHBASE+I,Ø:NEXT I:POKE CHBASE+496,255:POKE CHBASE+511,255
190 GRAPHICS 2:SETCOLOR 2,Ø,1Ø
200 SETCOLOR 4,Ø,Ø:SETCOLOR 2,Ø,Ø:SETCOLOR 1,Ø,8
210 TIME=Ø:PRINT "{CLEAR}DO YOU WANT A TIME LIMIT N{2 LEFT}";:INPUT ANS$:IF ANS$<>"Y" AND ANS$<>"N" THEN 21Ø
220 IF ANS$="N" THEN 29Ø
230 TRAP 23Ø:PRINT "{CLEAR}NUMBER OF MINUTES 9{2 LEFT}";:INPUT MIN
240 TRAP 4ØØØØ:IF MIN<Ø OR MIN<>INT(MIN) THEN 23Ø
250 TIME=36ØØ*MIN
260 TRAP 26Ø:PRINT "{CLEAR}NUMBER OF SECONDS Ø{2 LEFT}";:INPUT SEC
```

Stake a Claim on the most exciting New Game for your Atari Home Computer.
MINER 2049ER
STARRING BOUNTY BOB
BY BILL HOGUE
• From the programming team that brought you "Robot Attack", "Defense Command" and many other great Arcade games for your TRS-80™
• 100% machine language
• 16K ROM Cartridge, the largest available anywhere!
• Written specifically for the Atari® — not a converted Apple® game.
• Ten different rounds
• Difficulty adjustment
• High score table
• Demo mode
• Spectacular sound and graphics
• Runs on any 400/800 with at least 16K memory
• Only $49.95
Available also For the 5200
Here are just three of ten rounds in the game:
Round 1: The Mine Shaft.
Sharpen your miner's skill!
Round 3: The Transporters.
Beam yourself up!
Round 10: The Cannon.
Shoot yourself to the top!
BIG FIVE SOFTWARE
P.O. Box 9078-185 · Van Nuys, CA 91409 · (213) 782-6861
To order, see your local dealer. If they do not have "Miner 2049er" then send a check or money order to us for $49.95 plus $2.00 for shipping and handling for immediate shipment! (California residents add 6½% sales tax.)
"Atari," "Apple," and "TRS-80" are trademarks of Atari Inc., Apple Computer, and Tandy Corp., respectively.

Atari Notes

Marc Sugiyama, Programming Assistant

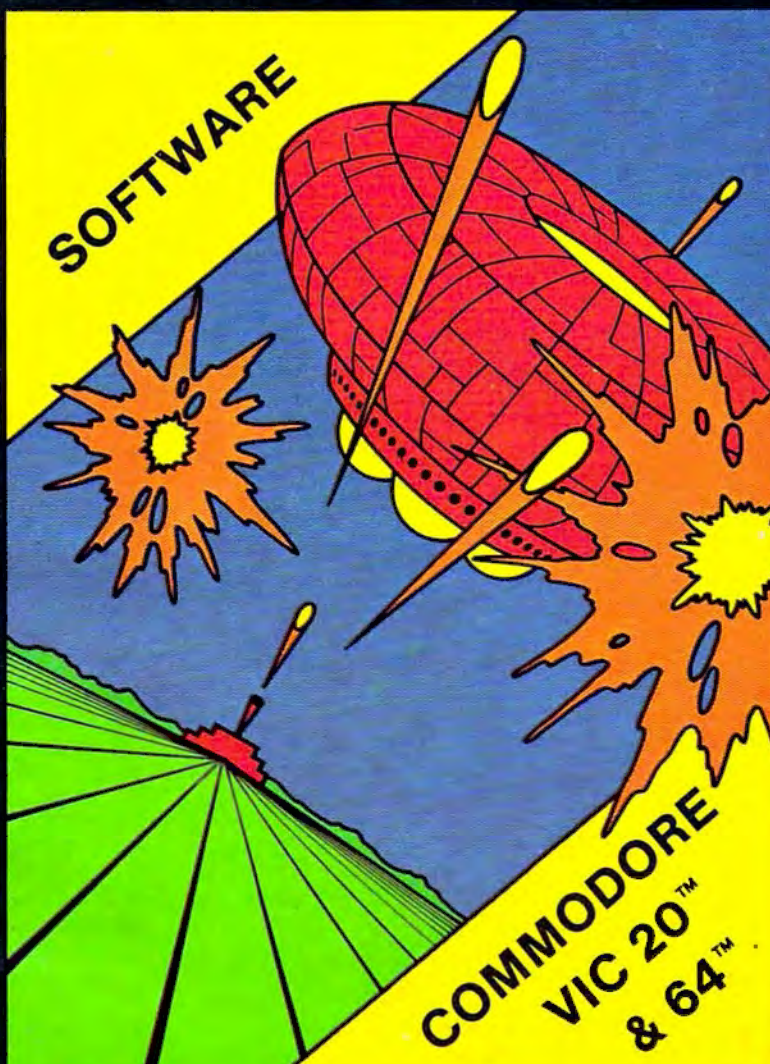
"Mosaic Puzzle" for the Atari is similar to the original VIC version; however, there are some differences. As you select the order for the goal, the letters will move from the puzzle box to the goal box. The game begins once you have selected all of the letters and numbers (do not forget about the space). If you have selected a time limit, the amount of time you have left is displayed at the bottom of the screen. If you did not select a time limit, the elapsed time is displayed.

You move the space (hole) around the puzzle board with a joystick plugged into the first port. Pressing the trigger activates the pause function. The timer is stopped, but the screen is cleared as well. Pressing the trigger again returns you to the game. Pressing Q while you are in the game allows you to quit the program.

```
270 TRAP 40000:IF SEC<0 OR SEC<>INT(
    SEC) OR SEC>59 THEN 260
280 TIME=TIME+SEC*60:IF TIME=0 THEN
    230
290 GRAPHICS 2+16:SETCOLOR 4,0,0:SET
    COLOR 2,8,8:SETCOLOR 0,1,4:SETCO
    LOR 1,5,10:POKE 756,CHBASE/256
300 XPOS=4:YPOS=3
310 GOSUB 880:GOSUB 860
320 POSITION 1,11:PRINT #6;"TYPE IN
    GOAL ORDER":SPACE=1:OPEN #1,4,0,
    "K:"
330 GET #1,AC:IF (AC<49 OR AC>70 OR
    (AC>57 AND AC<65)) AND AC<>32 TH
    EN 330
340 IF AC=32 THEN HERE=16:GOTO 360
350 HERE=AC-48:IF HERE>9 THEN HERE=H
    ERE-7
360 IF PUZZLE$(HERE,HERE)=" " THEN 3
    30
370 PUZZLE$(HERE,HERE)=" ":GOAL$(SPA
    CE,SPACE)=CHR$(AC+128):SPACE=SPA
    CE+1
380 GOSUB 860:GOSUB 870:IF SPACE<>17
     THEN 330
390 CLOSE #1:PUZZLE$="123456789ABCDE
    F■":GOSUB 860:POSITION 1,11:PRIN
    T #6;"{18 SPACES}"
400 POKE 18,0:POKE 19,0:POKE 20,0
410 CLOCK=PEEK(18)*65536+PEEK(19)*25
    6+PEEK(20)
420 IF TIME<>0 AND CLOCK>=TIME THEN
    580
430 POSITION 5,11:IF  NOT TIME THEN
    450
440 MIN=INT((TIME-CLOCK)/3600):SEC=I
    NT((TIME-CLOCK)/60)-60*MIN:GOTO
    460
450 MIN=INT(CLOCK/3600):SEC=INT(CLOC
    K/60)-60*MIN
460 PRINT #6;"TIME ";MIN;":";SEC;" "
    :IF STRIG(0)=0 THEN 780
470 IF PEEK(764)=47 THEN 670
480 JOY=STICK(0):IF JOY=15 THEN 410
490 HERE=XPOS+YPOS*4
500 IF JOY=13 AND YPOS<>0 THEN THIS=
    -4:GOSUB 560:YPOS=YPOS-1
510 IF JOY=14 AND YPOS<>3 THEN THIS=
    4:GOSUB 560:YPOS=YPOS+1
520 IF JOY=7 AND XPOS<>1 THEN THIS=-
    1:GOSUB 560:XPOS=XPOS-1
530 IF JOY=11 AND XPOS<>4 THEN THIS=
    1:GOSUB 560:XPOS=XPOS+1
540 POKE 77,0:GOSUB 860:IF PUZZLE$=G
    OAL$ THEN 720
550 GOTO 410
560 SOUND 0,100,10,10:TEMP$=PUZZLE$(
    HERE+THIS,HERE+THIS):PUZZLE$(HER
    E,HERE)=TEMP$
570 PUZZLE$(HERE+THIS,HERE+THIS)="■"
    :SOUND 0,0,0,0:RETURN
580 GRAPHICS 2+16:SETCOLOR 0,0,0:POS
    ITION 0,5:PRINT #6;"YOU RAN OUT
    OF TIME!":FOR I=0 TO 254 STEP 2
590 POKE 708,255-I:SOUND 0,I,10,10:N
    EXT I:SOUND 0,0,0,0
600 GRAPHICS 2+16:SETCOLOR 0,0,10
610 POSITION 7,1:PRINT #6;"PRESS":P
    OSITION 3,3:PRINT #6;"1)TO CONTI
    NUE"
620 POSITION 2,5:PRINT #6;"2)TO STAR
    T OVER"
630 OPEN #1,4,0,"K:"
640 GET #1,AC:IF AC<>49 AND AC<>50 T
    HEN 640
650 CLOSE #1:IF AC=49 THEN TIME=0:GO
    SUB 880:GOSUB 860:GOSUB 870:POKE
     18,0:POKE 19,0:POKE 20,0:GOTO 4
    10
660 RUN
670 T1=PEEK(18):T2=PEEK(19):T3=PEEK(
    20):GRAPHICS 2+16:SETCOLOR 0,0,1
    0
680 POSITION 0,6:PRINT #6;"DO YOU WA
    NT TO QUIT?":OPEN #1,4,0,"K:"
690 GET #1,AC:IF AC<>89 AND AC<>78 T
    HEN 690
700 CLOSE #1:IF AC=89 THEN GRAPHICS
    0:END
710 GOSUB 880:GOSUB 860:GOSUB 870:PO
    KE 20,T3:POKE 19,T2:POKE 18,T1:G
    OTO 410
720 FOR I=1 TO 100:POKE 712,PEEK(537
    70):NEXT I
730 GRAPHICS 2+16:SETCOLOR 0,0,0:POS
    ITION 4,5:PRINT #6;"YOU DID IT!!
    ":FOR I=0 TO 254 STEP 2:POKE 708
    ,I
740 SOUND 0,255-I,10,10:NEXT I
750 GRAPHICS 2+16:POSITION 3,5:SETCO
    LOR 0,1,10:PRINT #6;"PRESS  TRIG
    GER":POSITION 4,6:PRINT #6;"TO T
    RY AGAIN"
760 IF STRIG(0)=1 THEN 760
770 RUN
780 GRAPHICS 2+16:T1=PEEK(18):T2=PEE
    K(19):T3=PEEK(20):POSITION 4,10:
    PRINT #6;"PAUSE ACTIVE"
790 POSITION 3,11:PRINT #6;"PRESS  T
    RIGGER"
800 IF STRIG(0)=0 THEN 800
810 IF STRIG(0)=1 THEN 810
```

COMM * DATA
COMPUTER HOUSE, INC.

MILFORD, MICHIGAN



ARCADE STYLE GAMES

- Supercuda *
- Pegasus Odyssey *
- Ape Craze *
- Escape MCP *
- Maelstrom *
- Firing Line *
- Pakacuda *

OTHER GAMES

- Street Maze v
- Caves of Annod v
- Dragons & Treasure v
- Cribbage Partner †

EDUCATIONAL

- Toddler Tutor *
- Primary Math Tutor *
- Math Tutor *
- English Invaders Games *
- Gotcha Math Games *

POTPOURRI

- Basic Tools †
- Sketch & Paint *
- Multi-Level Marketing Manager (Direct Selling) †

On Shelves Everywhere - Ask For Comm*Data

Dealer Inquiries Welcome.

† Commodore 64
v VIC 20
* Both

MasterCard VISA

COMM * DATA
COMPUTER HOUSE, INC.

320 Summit Avenue
Milford, Michigan 48042
(313) 685-0113

Arcade Style Games are High Res Full Machine Code.
Commodore 64 and VIC 20 are Registered Trademarks of Commodore Business Machines, Inc.

```
820 IF STRIG(0)=0 THEN 820
830 GOSUB 880:GOSUB 860:GOSUB 870
840 POKE 18,T1:POKE 19,T2:POKE 20,T3
    :GOTO 410
850 END
860 FOR I=0 TO 3:POSITION 2,I+4:PRIN
    T #6;PUZZLE$(I*4+1,I*4+4):NEXT I
    :RETURN
870 FOR I=0 TO 3:POSITION 14,I+4:PRI
    NT #6;GOAL$(I*4+1,I*4+4):NEXT I:
    RETURN
880 GRAPHICS 2+16:SETCOLOR 0,1,4:SET
    COLOR 1,5,4:SETCOLOR 2,8,8:SETCO
    LOR 3,10,6:SETCOLOR 4,0,0
890 POKE 756,CHBASE/256:POSITION 5,2
    :PRINT #6;"hex puzzle"
900 POSITION 2,3:PRINT #6;"____":POS
    ITION 14,3:PRINT #6;"____"
910 FOR I=4 TO 7:POSITION 1,I:PRINT
    #6;"\{4 SPACES}]":POSITION 13,I:
    PRINT #6;"\{4 SPACES}]":NEXT I
920 POSITION 2,8:PRINT #6;"^^^^":POS
    ITION 14,8:PRINT #6;"^^^^"
930 POSITION 1,9:PRINT #6;"puzzle":P
    OSITION 14,9:PRINT #6;"goal":RET
    URN
```

"Mosaic Puzzle," Atari version.

Program 4: Mosaic Puzzle – TI Version

by Rick Rothstein

```
190 CALL MAGNIFY(4):: RANDOMIZE ::
    DIM TILE(16),TEMP(16):: TEMP(16
    )=16 :: FR=153
200 CALL CLEAR :: CALL SCREEN(11)::
     CALL CHARSET :: CALL CHAR(35,"
    0",71,RPT$("0",12)&"FF0000FF")
210 DISPLAY AT(1,9):RPT$("G",12)::
    DISPLAY AT(2,9):"OPTION##MENU"
220 DISPLAY AT(3,9):RPT$("H",12)::
    DISPLAY AT(10,4):"PRESS###FOR"
     :: DISPLAY AT(11,4):"HHHHH####
    HHH"
230 DISPLAY AT(13,6):"1######NUMBER
    #PUZZLE" :: DISPLAY AT(16,6)BEE
    P:"2######LETTER#PUZZLE"
240 WASTE=RND :: CALL KEY(0,K,ST)::
     IF ST=0 THEN 240
250 IF K=49 THEN I=0 ELSE IF K=50 T
    HEN I=1 ELSE IF K=7 THEN 700 EL
    SE CALL SOUND(150,110,0):: GOTO
    240
260 CALL CLEAR :: CALL SCREEN(4)::
    IF I=0 THEN RESTORE 710 ELSE RE
    STORE 790
270 FOR I=80 TO 136 STEP 4 :: READ
    A$,B$ :: CALL CHAR(I,"FFFF"&A$&
    "FFFFFFFF"&B$&"FFFF"):: NEXT I
    :: CALL CHAR(140,RPT$("0",64))
280 CALL CHAR(71,"0000000000000003
    03030303030300000FFFF")
290 CALL CHAR(74,"00003F3F30303030
    000FCFC0C0C0C0C303030303F3F0000
    0C0C0C0CFCFC",78,RPT$("0C",8)&"0
    0000000FFFF")
300 RESTORE 870 :: FOR I=38 TO 47 :
    : READ A$ :: CALL CHAR(I,A$)::
    NEXT I
310 CALL COLOR(5,15,2,6,15,2):: CAL
    L HCHAR(2,7,71,20):: DISPLAY AT
    (3,5):"GJ"&RPT$("I",16)&"KG"
320 FOR I=4 TO 16 STEP 4 :: DISPLAY
     AT(I,1):RPT$("####GH"&RPT$("G"
    ,16)&"NG####",4):: NEXT I

330 DISPLAY AT(20,5):"GL"&RPT$("O",
    16)&"MG" :: CALL HCHAR(21,7,71,
    20)
340 GOSUB 890 :: FOR I=1 TO 15 :: T
    ILE(I)=I :: NEXT I
350 FOR J=1 TO 15 :: R=1+INT(RND*(1
    6-J)):: TEMP(J)=TILE(R):: TILE(
    R)=TILE(16-J):: NEXT J
360 N=0 :: FOR I=1 TO 14 :: FOR J=1
     TO 15-I :: IF TEMP(I)>TEMP(I+J
    )THEN N=N+1
370 NEXT J :: NEXT I :: IF N/2<>INT
    (N/2)THEN TEMP(16)=TEMP(15):: T
    EMP(15)=TEMP(14):: TEMP(14)=TEMP
    (16):: TEMP(16)=16

380 FOR I=1 TO 16 :: TILE(I)=TEMP(I
    ):: NEXT I :: N=0 :: SP=16
390 FOR I=22 TO 124 STEP 34 :: FOR
    J=62 TO 164 STEP 34 :: N=N+1 ::
     CALL LOCATE(#TILE(N),I,J):: NE
    XT J :: NEXT I
400 MOVE$="" :: TOTAL=0 :: DISPLAY
    AT(24,1):CHR$(32+6*DIR)&"#####(
    )*+#,-(.+/:#0#######"&CHR$(39-7
    *NO):: CALL SOUND(150,666,0)::
    GOTO 430
410 CALL SOUND(150,110,0)
420 WASTE=RND :: IF K=73 OR K=78 OR
     K=105 OR K=110 THEN 450
430 CALL KEY(1,KK,ST):: IF KK=18 TH
    EN 670 ELSE CALL JOYST(1,X,Y)::
     IF ABS(X)+ABS(Y)=8 OR X+Y=0 TH
    EN 450
440 IF X/4=(-1)^(1-DIR)THEN 510 ELS
    E IF X/4=(-1)^(2-DIR)THEN 550 E
    LSE IF Y/4=(-1)^(2-DIR)THEN 630
     ELSE IF Y/4=(-1)^(1-DIR)THEN 5
    90
450 CALL KEY(0,K,ST):: IF ST=0 THEN
     430 ELSE IF K=45 THEN 670 ELSE
     IF K=83-HORZ OR K=115-HORZ THE
    N 510
460 IF K=68+HORZ OR K=100+HORZ THEN
     550 ELSE IF K=88-VERT OR K=120
```

TI-99/4A Notes

Rick Rothstein

The TI/99-4A version of "Mosaic Puzzle" requires Extended BASIC. In this version, you have the option of requesting either letters (A-O) or numbers (1-15) within a 4-by-4 frame. When you have entered your choice, the game board with its lettered or numbered blocks appears in a scrambled order. The object of the game is to slide the blocks about, one at a time, to bring them to one of several preselected patterns. Some patterns that you can try to duplicate are given in the table.

Move the lettered or numbered blocks around the game board with a joystick or the keyboard (E, S, D, and X keys). You actually have a choice of moving either the free space (hole) or the labeled blocks. The game is initially set to move the free space, but by pressing I (note the appearance of the left-right arrow symbol in the lower-left corner of the screen), you can move the labeled blocks.

Each move that you make is tallied, and the total number of moves is given at the bottom of the screen. Moves are normally accompanied by a sliding noise (notice the note symbol at the lower-right corner of the screen). If this noise becomes annoying, press N and the noise will cease (the note will also disappear).

At certain times during the game, you may wish to retrace your previous moves. Press - (minus sign) or hit the fire button to step back through each preceding move. With this option, a maximum of 250 moves can be recalled.

Once you've achieved the desired preselected pattern from its scrambled beginnings, you can restore the original game board pattern by pressing FCTN (REDO) and challenge yourself or others to beat your tally.

Other options available to you during the game are:

Keystroke	Description
FCTN (BACK)	Returns to the letter or number option menu
FCTN (BEGIN)	Starts a new game
FCTN (ERASE)	Ends the program

Possible Patterns For Puzzle, TI Version

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Horizontal

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	

Vertical

7	8	9	10
6	1	2	11
5	4	3	12
	15	14	13

Spiral

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

Peripheral

12	2	1	15
7	9	10	4
11	5	6	8
	14	13	3

Adds To Thirty

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	

Horizontal

A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	

Vertical

G	H	I	J
F	A	B	K
E	D	C	L
	O	N	M

Spiral

A	B	C	D
L	M	N	E
K		O	F
J	I	H	G

Peripheral

F	O	G	
B	I	N	D
H	E	L	M
J	A	C	K

Words(1)

E	L	F	
J	A	M	B
D	O	C	K
N	I	G	H

Words(2)

```
    -VERT THEN 590 ELSE IF K=69+VER
    T OR K=101+VERT THEN 630
470 IF K=6 THEN CALL HCHAR(24,4,32,
    26):: GOSUB 890 :: GOTO 380 ELS
    E IF K=7 THEN 700
480 IF K=15 THEN CALL DELSPRITE(ALL
    ):: GOTO 200 ELSE IF K=14 THEN
    CALL HCHAR(24,4,32,26):: GOTO 34
    0 ELSE IF ST=-1 THEN 420
490 IF K=78 OR K=110 THEN NO=1-NO :
    : FR=153+NO*30000 :: CALL HCHAR
    (24,30,39-7*NO):: GOTO 420
500 IF K=73 OR K=105 THEN HORZ=15-H
    ORZ :: VERT=19-VERT :: DIR=1-DI
    R :: CALL HCHAR(24,3,32+6*DIR):
    : GOTO 420 ELSE 410
510 IF SP=1 OR SP=5 OR SP=9 OR SP=1
    3 THEN 410 ELSE SP=SP-1 :: CALL
     POSITION(#TILE(SP),ROW,COL)::
    CALL SOUND(4000,FR,14*NO)
520 IF MINUS=0 THEN MOVE$="L"&MOVE$
     ELSE MINUS=0
530 FOR I=COL TO COL+34 STEP 2 :: C
    ALL LOCATE(#TILE(SP),ROW,I):: N
    EXT I
540 TILE(SP+1)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,30):: GO
    TO 690
550 IF SP=4 OR SP=8 OR SP=12 OR SP=
    16 THEN 410 ELSE SP=SP+1 :: CAL
    L POSITION(#TILE(SP),ROW,COL)::
     CALL SOUND(4000,FR,14*NO)
560 IF MINUS=0 THEN MOVE$="R"&MOVE$
     ELSE MINUS=0
570 FOR I=COL TO COL-34 STEP -2 ::
    CALL LOCATE(#TILE(SP),ROW,I)::
    NEXT I
580 TILE(SP-1)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,30):: GO
    TO 690
590 IF SP>12 THEN 410 ELSE SP=SP+4
    :: CALL POSITION(#TILE(SP),ROW,
    COL):: CALL SOUND(4000,FR,14*NO
    )
```

ANY WAY YOU PLAY IT... YOU'RE IN TROUBLE!

The word's out on Parallax. FAST. You better be fast, really fast to beat those evil Silurians. Watch out! Wave after screaming wave of gravity force fields can tear you apart. Quick! Aim your missiles. Fire! Fire again! You must destroy those oscillating gravitonic generators or else! Oh no... the Silurian orbital launchers are now slamming multiple warheads at you from all sides at once. Don't panic! You're fast... you have to be... to meet all the high-speed challenges of Parallax, the most kinetic game going!

10 DIFFERENT FAST-ACTION SCREENS + MULTI-BONUS SCREENS
- 2-PLAYER OPTION
- 100% MACHINE LANGUAGE
- 16K PLUS JOYSTICK
- DISK OR CASSETTE

From the makers of Trion (in 3-D) and Space Ace, #1-Rated Game of 1982. $34.95. See your dealer and play the game or order direct.*

London SOFTWARE

374 Wildwood Ave., Piedmont, CA 94611
PHONE ORDERS: (415) 893-1090 VISA/MC

Please add $1.50 postage and handling.
Calif. residents add 6.5% sales tax.

*COMPUTER DEALER, Jan., 1983

DESIGNED BY GREG YOUNG. FOR ALL ATARI COMPUTERS AND THE COMMODORE 64.

```
600 IF MINUS=Ø THEN MOVE$="D"&MOVE$
     ELSE MINUS=Ø
610 FOR I=ROW TO ROW-34 STEP -2 ::
    CALL LOCATE(#TILE(SP),I,COL)::
    NEXT I
620 TILE(SP-4)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,3Ø):: GO
    TO 69Ø
630 IF SP<5 THEN 41Ø ELSE SP=SP-4 :
    : CALL POSITION(#TILE(SP),ROW,C
    OL):: CALL SOUND(4ØØØ,FR,14*NO)
640 IF MINUS=Ø THEN MOVE$="U"&MOVE$
     ELSE MINUS=Ø
650 FOR I=ROW TO ROW+34 STEP 2 :: C
    ALL LOCATE(#TILE(SP),I,COL):: N
    EXT I
660 TILE(SP+4)=TILE(SP):: TILE(SP)=
    16 :: CALL SOUND(-1,FR,3Ø):: GO
    TO 69Ø
670 IF MOVE$="" THEN 41Ø ELSE MINUS
    =1 :: K=ASC(MOVE$):: MOVE$=SEG$
    (MOVE$,2,25Ø):: TOTAL=TOTAL-2
680 IF K=82 THEN 51Ø ELSE IF K=76 T
    HEN 55Ø ELSE IF K=85 THEN 59Ø E
    LSE IF K=68 THEN 63Ø
690 TOTAL=TOTAL+1 :: DISPLAY AT(24,
    19)SIZE(4)BEEP:TOTAL :: MOVE$=S
    EG$(MOVE$,1,25Ø):: GOTO 42Ø
700 CALL DELSPRITE(ALL):: CALL CLEA
    R :: END
710 DATA FEFEFEFEFEFEFEFEFEFEFEFE,7
    F7F7F7F7F7F7F7F7F7F7F7F,F8FØF3F
    FFFFFFEFCF8F1FØFØ,1FØFCFCF8F1F3
    F7FFFFFØFØF
720 DATA F8FØF3FFFFFEFEFFFFF3FØF8,1
    FØFCFCF8F1F1F8FCFCFØF1F,FFFFFEF
    CF8F1FØFØFFFFFFFFF,8FØFØF4FCFCFØ
    FØFCFCFCFCF
730 DATA FØFØF3F3FØFØFFFFFFF3FØF8,1
    F1FFFFF3F1F8FCFCF8F1F3F,FCF8F1F
    3FØFØF1F3F3F1F8FC,1F1FFFFF3F1F8F
    CFCF8F1F3F
740 DATA FØFØF3FFFFFFFFFEFEFEFEFE,Ø
    FØFCF8F9F1F3F3F7F7F7F7F,FCF8F9F
    9FCF8F1F3F3F1F
8FC,3F1F9F9F3F1F8FCFCF8F1F3F
750 DATA FCF8F1F3F3F1F8FCFFFFF8F8,3
    F1F8FCFCF8FØFØFCF8F1F3F,CFCECCC
    CCCCCCCCCCCCCECF,ØFØ763F3F3F3F
    3F3F363Ø7ØF
760 DATA F3F3F3F3F3F3F3F3F3F3F3F3,C
    FCFCFCFCFCFCFCFCFCFCFCF,CECCCCC
    FCFCFCFCFCECCCCCC,Ø7Ø3F3F3E3C78
    F1F3F7FØ3Ø3
770 DATA CECCCCCFCFCFCFCFCFCCCCCE,Ø
    7Ø3F3F3E38787E3F3F3Ø3Ø7,CFCFCFC
    FCECCCCCCCFCFCFCF,E3C383133373Ø
    3Ø3F3F3F3F3
780 DATA CCCCCCCCCCCCFCFCFCCCCCE,Ø
    7Ø7FFFFØFØ7E3F3F3E3Ø7ØF
790 DATA FEFCF8F1E3E7EØEØE7E7E7E7,7
    F3F1F8FC7E7Ø7Ø7E7E7E7E7,EØEØE7E
    7E7EØEØE7E7E7EØEØ,1FØFC7E7C7ØFØF
    C7E7C7ØF1F
800 DATA F8FØE3E7E7E7E7E7E7E3FØF8,Ø
    FØ7E7FFFFFFFFFFFFE7Ø7ØF,EØEØE7E
    7E7E7E7E7E7E7EØEØ,3F1F8FC7E7E7E
    7E7C78F1F3F
810 DATA EØEØE7E7E7EØEØE7E7E7EØEØ,Ø
    7Ø7FFFFFF1F1FFFFFFØ7Ø7,EØEØE7E
    7E7EØEØE7E7E7E7E7,Ø7Ø7FFFFFF1F1
    FFFFFFFFFF
820 DATA F8FØE3E7E7E7E7E7E3FØF8,Ø
    FØ7E7FFFFFF8787E7E7Ø7Ø7,E7E7E7E
    7E7EØEØE7E7E7E7E7,E7E7E7E7E7Ø7Ø7
    E7E7E7E7E7
830 DATA F8F8FEFEFEFEFEFEFEFEF8F8,1
    F1F7F7F7F7F7F7F7F7F1F1F,FFFFFFF
    FFFFFFFFFE7E3FØF8,E7E7E7E7E7E7E7
    E7E7C7ØF1F
840 DATA E7E7E7E6E4EØEØEØE6E7E7E7,C
    78F1F3F7FFFFFF7F3F1F8FC7,E7E7E7E
    7E7E7E7E7E7E7EØEØ,FFFFFFFFFFFFFF
    FFFFFFØ7Ø7
850 DATA E7E3E1EØE4E6E7E7E7E7E7E7,E
    7C787Ø72767E7E7E7E7E7E7,E7E3E1E
    ØE4E6E6E7E7E7E7E7,E7E7E7E7E76767
    27Ø787C7E7
860 DATA F8FØE3E7E7E7E7E7E7E3FØF8,1
    FØFC7E7E7E7E7E7E7C7ØF1F
870 DATA 3Ø6ØFFØØØØFFØ6ØC,ØEØ9Ø8Ø86
    8F8F86,ØØ446C54444444,ØØ7C44444
    4447C,ØØ4444444281,ØØ7C4Ø784Ø4Ø
    7C
880 DATA ØØ4464544C44,ØØ444444444
    438,ØØ784478444478,ØØ784444784Ø8
    44
890 FOR I=1 TO 16 STEP 2 :: CALL SP
    RITE(#I,76+4*I,16,193,1,#I+1,8Ø
    +4*I,11,193,1):: NEXT I :: RETU
    RN
```

©


APPLE • ATARI • IBM • COM-64
The Official
FROGGER™
by SEGA®
AND THE VIC-20!
Frogger hops to IBM! Sierra On-Line brought this popular coin-op game to the Apple, Atari and Com 64. Now we're bringing the helpless critter home to IBM PC owners!
Frogger is available for $34.95 from your local dealer or directly from Sierra On-Line, Inc., Sierra On-Line Building, Coarsegold, CA 93614 (209) 683-6858. VISA, Mastercard, Check, COD accepted. Add one dollar for shipping.
® and TM designate trademarks of Sega Enterprises, Inc.
© 1981 Sega Enterprises, Inc.
SIERRAVISION is a trademark of Sierra On-Line, Inc.
SIERRAVISION™
Sierra ON-LINE INC

Questions Beginners Ask

Tom R Halfhill, Features Editor

Are you thinking about buying a computer for the first time, but don't know anything about computers? Or maybe you just purchased a computer and are still a bit baffled. Each month in this column, COMPUTE! will answer some questions commonly asked by beginners.

Q I'm looking around for my first personal computer, and am wondering if I should get an 8-bit or 16-bit computer. I don't really understand the difference, except I was told by a salesman that 16-bit computers are faster, have more memory, and will eventually replace 8-bit computers. Is this true? Are 8-bit computers becoming obsolete? Would I be better off with a 16-bit computer? What is the real difference, anyway?

A These questions come up fairly often, so let's deal with them in some detail.

First, it's not true that 8-bit computers are becoming obsolete. Most of the computers being sold today are 8-bit computers, especially the under-$1000 home computers. Although it is true that 16-bit computers will likely become increasingly common, 8-bit machines (especially in the low price range) will be around for a good while.

Nor is it true that 16-bit computers are necessarily faster or contain more memory.

The difference between 8- and 16-bit computers lies in their *Central Processing Units*. The CPU is the central "brain" of a computer. It fetches instructions written by programmers, performs arithmetic to execute the instructions, and stores the results of its calculations in memory. The CPU controls or oversees all the operations performed by the computer. Without a CPU, a computer would be a brainless collection of memory chips, support chips, and wires.

In very large *mainframe* computers, the CPU might be a unit the size of a refrigerator, made up of dozens of circuit boards. Microcomputers (including personal computers) have a *microprocessor* CPU – a CPU that fits on a single silicon chip smaller than a penny. Microprocessors work strictly with binary numbers (1's and 0's). All instructions and numbers required for calculations must first be converted to groups of binary numbers before the microprocessor CPU can handle them.

Some microprocessors are designed to handle groups of binary numbers only four digits long. That is, all numbers and instructions must be broken up into groups of four 1's and 0's, such as 1101 or 1011. This would be called a *4-bit* microprocessor (a bit is a binary digit – a 1 or a 0). Portable calculators use these 4-bit chips.

Other microprocessors are more powerful and can handle numbers and instructions in groups of eight 1's and 0's. These are *8-bit* microprocessors. A 16-bit microprocessor handles 16 bits at a time, a 32-bit microprocessor handles 32 bits, and so on.

Generally speaking, the larger these groupings (called *word size*), the more powerful the computer. Computers which handle numbers internally in larger chunks of bits can work faster and more efficiently. Also, they generally have more memory because they are designed to run larger programs and therefore need more memory.

But keep in mind that these are general rules, and, as always, there are exceptions. Several other factors also determine the speed of a computer and the amount of memory it contains.

For instance, the only 16-bit home computer now on the market is the Texas Instruments TI-99/4A. Its maximum memory expansion is 48K, which is no more (and in some cases less) than the maximum memory available in such 8-bit computers as the Commodore 64, Apple II/IIe, Atari, Radio Shack TRS-80, Timex/Sinclair, etc. Also, most of these 8-bit computers can run BASIC programs somewhat faster than the TI-99/4A.

In one benchmark test we heard about, a small 8-bit Timex/Sinclair 1000 actually outran a larger and much more expensive 16-bit IBM Personal Computer. Yet, a business person who needs the powerful features of an IBM PC would not want to choose the T/S 1000 on the basis of speed alone.

That's why you shouldn't base a buying decision solely on the question of 8-bit versus 16-bit. Too many other factors are important. Instead, carefully evaluate your own needs, and then shop for a computer and software combination that serves them well. ©

On The Road With Fred D'Ignazio

There's A Robot In My Room

If It's Tuesday, I Must Be In Benton Harbor

Last month I asked you to lace up your racing shoes and sprint with me on my whirlwind tour of computer and robot centers around the United States. I visited the West Coast Computer Faire in San Francisco, then the Florida Instructional Computing Conference in Tampa. After that I flew up to New York to see a preview of the new educational software being created at the Children's Computer Workshop (CCW). I spent a day in Benton Harbor, Michigan, teaching the HERO 1 robot, and another day in Chicago at ROBOTS VII, the world's largest robotics conference.

This month I'd like you to come with me to England. The trip to England will be like a visit into the future when we will be surrounded by intelligent, friendly machines. We'll see the kind of effect it has on an average person of the present who is still used to dealing mostly with people.

A Scene Out Of Dr. Who

After returning from Chicago, I spent a couple of days at home in Roanoke, Virginia. Then I climbed aboard a TWA jet and flew across the Atlantic to London, England. I went to London to teach a three-week course on "Robotics Literacy" at the Organization Reconstruction Travail (ORT), an international technical-training institute whose

Harry Blair

world headquarters is in London. I also helped with the course materials by arranging to have a HERO robot flown over from the United States and I carried a Tasman Turtle robot with me on the airplane.

There were 15 students in the Robotics Literacy course. The students came from countries all over the world, including India, France, Israel, the United States, Italy, and the United Kingdom. The students were all adults. They were directors of university engineering schools, vocational schools, and teacher training schools. They had come to London to learn how to set up a robotics course in their own countries. They returned to their countries with a ten-pound notebook full of course materials, another 50 pounds of books, brochures, and manuals, a computer, a robot arm, and lots of "hands-on" experience.

How High The Tech

I anticipated that a course on robotics would be "high tech," but I did not dream how high the tech would be. The classroom was like a scene out of science fiction's *Dr. Who* program. We had twenty BBC (Model B) computers networked together on the Econet Network. The computers sat on polished mahogany tables lining the walls of the classroom. Next to each computer was a Smart Arm robot. The robots were plugged into the computers and could be programmed using the Arm Controller keypads.

At one point the course manager had all the arms going at the same time. The arms waved, swooped, picked up pencils and half-empty coffee cups.

Also next to the tables was a robotics interface kit. The kit came in a box that looked like a steel briefcase. When you opened the box, you saw an array of wires and metal parts. The kit was developed by the Moshinsky Institute in Israel. It contained sensors and motors that could be connected to the computers and to the Smart Arm robots. It was like a robotic Erector set. With the kit, you could take the Smart Arm, give it eyes and ears, and incorporate it into a miniature factory workstation. Two Smart Arms could work together, sensing each other's movements via infrared, electrical, and optical sensors.

The kit sounds sophisticated, but it was assembled by students at the institute out of spare parts scavenged from the institute's supply room.

At the front of the room was an enormous television set and a BBC computer as the network controller. The teacher could conduct an experiment or give a demonstration simultaneously on the master computer and on the 20 student computers and robots.

Also, whenever a student wanted to ask a question or show off something he was proud of, the instructor could copy the student's computer screen onto any other student's screen or onto the big TV screen at the front of the class.

At the back of the room was a huge multiprojector slide system mounted on a six-foot platform. During the course, we got to see several videotapes and slide shows on different robot applications.

Also at the back was a hulking, hydraulically powered robot arm. The robot was used in major demonstrations and became a star when TV and newspaper reporters arrived.

Running around the floor were lots of little robots, like robot gremlins. A robot "buggy" zipped across the floor, following a twisting, turning piece of white tape with its photoelectric sensor.

The robot turtle was on the floor, talking, beeping, blinking its little LED "eyes," and trying to find its way out of a maze made of cardboard walls.

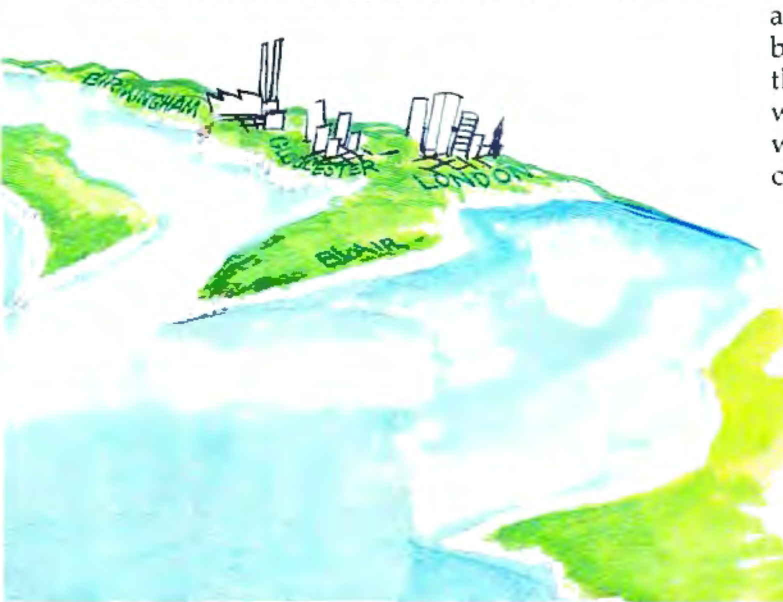
And HERO was there too, waving his arm, and rolling over the cables and wires that crisscrossed the floor.

An English HERO

When I first arrived in England, I hoped to receive a HERO robot shipped directly from Benton Harbor, the world headquarters of Heath Company, the robot's manufacturer. Unfortunately, Heath was swamped by orders for the HERO and was way behind meeting shipments. A spare robot couldn't be found.

Luckily for me and my students, we found a HERO robot in England. Zenith Data Systems, a Heath-affiliated company located in Gloucester, England, near the west coast, offered to loan us their machine in return for some training on the HERO.

I journeyed to Gloucester by train on my second day in England. I met the English HERO and found that he had been



We're always thinking about helping kids think.

fitted with an English power supply and a deeper voice than the HERO I'd met in Benton Harbor.

I returned to London. Two days later HERO arrived in a box big enough to encase a circus gorilla. His wrist was bent, his head was on crooked, and his photoelectric sensor that monitored the number of times his wheel turned around was disconnected. But these were minor problems. After a little sprucing up, he worked perfectly.

The Master Of Ceremonies

HERO got to be our course's Master of Ceremonies. It was his job to say a few words to inspire the students and get the course started.

The only hitch was that to turn HERO into a congenial Master of Ceremonies, I had to program him. And I had never programmed a HERO before. (The mucking around I did in Benton Harbor was definitely not programming. Take a look at last month's "On the Road" column to see what happened the first time HERO and I were alone together.)

Without proper programming, all HERO was capable of was a few robot calisthenics. Of course, HERO could also say "Ready" (it sounded more like "RED-DY!"). But that meant HERO was ready to be programmed, not ready to do tricks.

I spent the entire weekend before the opening ceremonies programming HERO. I had to enter my entire program as two-digit hexadecimal commands typed into the keyboard on HERO's head. To make HERO's motors move, I had to tell HERO's computer which of six motors to turn on, what motor position to start from, and how far the motor should turn.

In order to get HERO to talk, I had to think up HERO's speech then break it into hundreds of *phonemes* – the sounds that are the building blocks of spoken words. Then I had to code the phonemes into HERO's "Robot Language" and enter more two-digit codes into HERO's onboard memory.

After all the hours of work, I didn't want to lose anything, so I hitched HERO to a tape recorder, and I saved this program on eight tape cassettes.

Then, late Sunday night, I turned HERO's power off. As a result, he forgot everything I had taught him. This was okay, I thought, since I had copies of the program on the eight tapes.

HERO Sat There

On Monday morning, right before HERO made his grand entrance into the classroom, I popped a cassette into the recorder and tried to load the speech program back into his onboard memory.

HERO said "RED-DY!" to signal me that the program was finished loading. I tried to run the program, but HERO just sat there. I looked at the locations in HERO's memory to make sure the program was there.

They weren't there! I was horrified to learn that the front-end of the program had, overnight, turned into computer mush.

I tried a new tape. More mush.

Another tape. And another. And another.

All eight tapes had incomplete copies of my program. I consulted the time on HERO's clock. HERO was to make his grand entrance in just fifteen minutes.

Luckily I had copied down all my commands on a scrap of paper. I found the paper and retyped the missing commands into HERO's head.

"RED-DY!" HERO said. I grinned. This time he was really ready.

I picked HERO up and carried him to the classroom door. I peeked into the room. Everyone was assembled. The director of the course was on the speaker's platform at the front of the room.

The director nodded his head. It was time. I pressed HERO's A button, his DO button, and keyed in the four-digit starting address of the program.

HERO took off. He marched into the classroom, spun around three times, and waved the WELCOME!! sign he held in his gripper "hand."

HERO stopped spinning. "Attention," he said. ("Attention, please" in French.) "Shekket." ("Be quiet" in Hebrew.)

Everyone was amazed that an American robot could talk in French and Hebrew. There were gasps of surprise. The room quickly grew silent.

Calc Result™

The Commodore 64™ Spreadsheet that puts you a million miles ahead

CALC RESULT...The one spreadsheet guaranteed to turn your Commodore into a powerful financial tool. Offering you every feature found on other more expensive programs for much less the cost.

Flexible...you can view four different areas at once

Versatile...customize your own print formats

Distinctive...display beautiful color graphics

CALC RESULT *Advanced* is a three-dimensional spreadsheet with built-in HELP function and 32 pages of memory. For the Commodore 64 $149.95. For the CBM™ 8032 $199.00.

For first time users CALC RESULT *Easy* gives you a fast way to perform financial calculations—easily. For the Commodore 64 $79.95.

For a down to earth demonstration of either version visit your local dealer today.

handic software ab

-a company in the Datatronic group-

Distributed by

Computer Marketing Services, Inc.

300 W. Marlton Pike, Cherry Hill, NJ 08002

609 795-9480

"Welcome to the ORT Robotics Literacy Course," HERO said. "I am HERO, the robot from America. I hope you have fun. Ha! Ha! Ha! Bye, bye."

HERO spun around once more, waved his sign, then marched toward the door. The audience began clapping.

But HERO was not destined to make a triumphant exit from the classroom. In fact, he never even made it to the door. On his way, he crashed headfirst into a chair.

Poor HERO didn't know he hadn't made it through the door. He thought he was out in the hallway. His "Master of Ceremonies" program finished executing. "RED-DY!" he said.

"Listen," said one of the instructors. "The robot knows its master's name. It's saying *'Freddie.'* "

Everyone began laughing.

I picked HERO up. I felt like an embarrassed parent. "It wasn't his fault," I stammered. "It was my programming...."

I lurched out of the room with HERO in my arms.

"RED-DY!" said HERO.

A Briefcase And A Sewing Machine

I brought two computers from the United States to help me teach the Robotics Literacy course. I had a briefcase-sized Epson HX-20 computer with me to record business expenses and to do course and calendar planning and memos. I brought along a sewing machine-sized Compaq computer to write up my course lectures and to create several graphs, figures, tables, and small data bases from the research material I had gathered for the course.

I used the computers to create lectures on the history of robots, the future of robots, the automated factory, artificial intelligence, the Tasman Turtle, the HERO robot, robots in the home, exotic (outer space, undersea, and legged) robots, CAD/CAM (Computer-Aided Design/Computer-Aided Manufacturing), industrial robots, robot anatomy, and the impact of robots on jobs, work, and people.

I also brought two robots, as I mentioned – HERO the robot and the Tasman Turtle. The turtle ran on an Apple computer loaned to me by Apple Computer/U.K.

The turtle, HERO, and the Compaq computer had to be converted to British current and voltage (a stepdown from 220 volts to 110, and a change in the current from 60 to 50 Hz). The little Epson computer was okay since it ran on rechargeable batteries. If I'd had to recharge it, I would have had a problem, but during my entire three-week course, the batteries never ran down.

There's A Robot In My Bedroom

I was the only full-time guest instructor for the course so I was always extremely busy. Often I would teach and work 12 hours a day.

I had to work constantly, and I needed my computers and robots near me to do my work. It would have been a great inconvenience to lug them back and forth from ORT to a hotel room every day. But I would need them during the day at ORT and in the evening back at the hotel. I felt almost like a bionic man. I had to have my computers and robots around me, or I couldn't function.

The director of the Robotics Literacy course came up with a solution: I could live and work at ORT in one of the two upstairs suites, down the hall from the course classroom. My office would be my bedroom, and vice versa.

Hey! That's great! I thought. I'm always talking about how people should get intimate with their computers and robots. Now here was the chance for me to see how intimate my machines and I could get.

I was looking forward to this arrangement. I would be all alone at night in the ORT building, but my robots and computers would keep me company. When I went to bed at night, my computers and robots would surround me like tiny sentinels. In the morning I could leap out of bed, turn on the computers and robots, and get to work immediately. I wouldn't have to waste time on nonessentials like getting dressed, brushing my teeth, or taking a shower. (I couldn't take a shower anyway since the ORT shower was broken.)

But I still had one problem: who would wake me up each morning at dawn so I could get right to work? I didn't have a travel clock with me, and I couldn't get a wake-up call from the hotel desk. In fact, I had no phone at all. I was isolated from the world. To make outgoing calls I had to throw on some clothes and run across the street to the Finchley Road tube (subway) station. I made all my business and personal calls at the station in a doorless booth, with dozens of people streaming by and trains rumbling by underneath sounding like earthquakes.

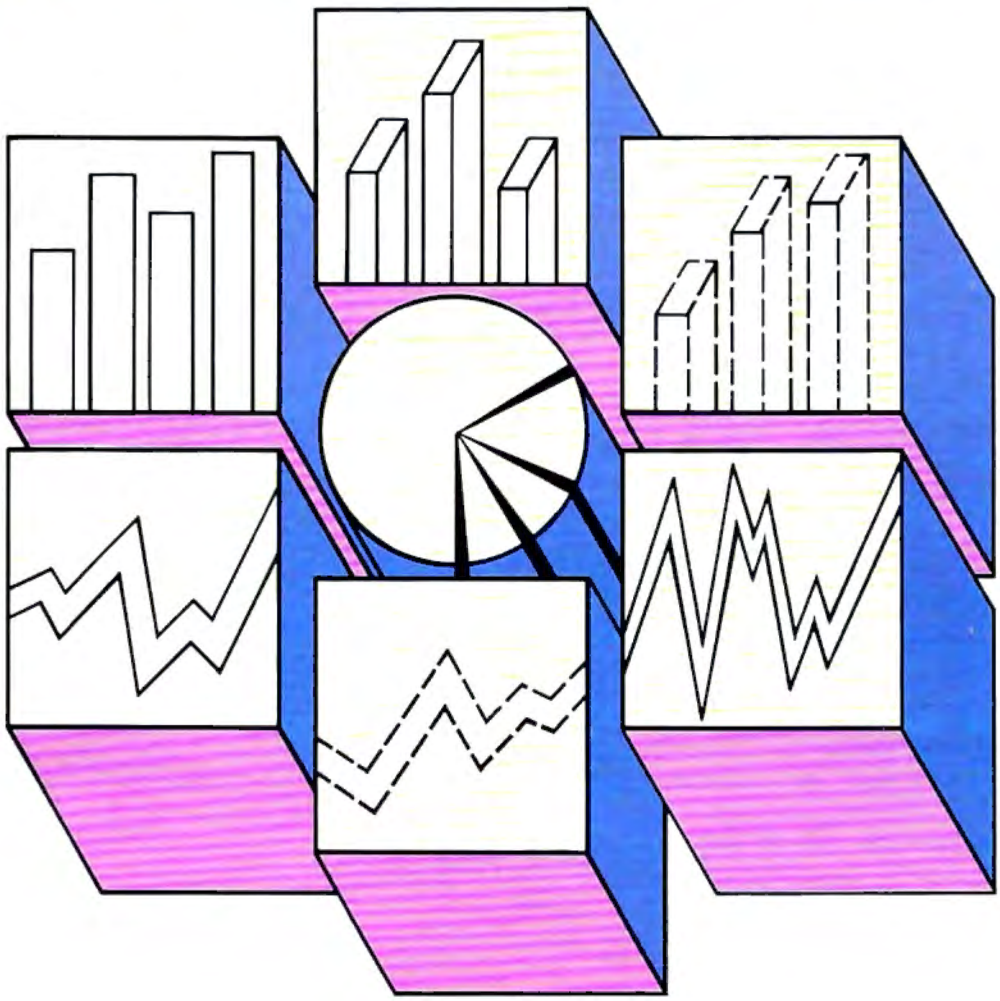
The Robot Alarm Clock

How was I to wake up each morning on time?

Then I remembered that HERO had a built-in realtime clock. I wrote a program using this clock and HERO's light sensor. The light sensor, a photoelectrical cell, can sense up to 256 levels of luminance, or brightness. I made a stab at how bright it would be at 5:30 in the morning, and I created a little wake-up speech for HERO to launch into.

That night I turned HERO on, plugged him into the wall current so his battery would not run

B/GRAPH T.M.



A PROFESSIONAL GRAPHICS-CHARTING AND STATISTICAL ANALYSIS PROGRAM FOR ATARI® PERSONAL COMPUTERS

B/Graph is for professionals in Marketing Sales, Administration, Forecasting, Accounting and General Management.
- Home and Small Business
- Educators, Students
- Hobbyists

B/Graph can create
- Bar charts
- 3-D Bar Charts
- Segmented Bar charts
- Pie charts
- Line Graphs
- Scatter Graphs
- Market Graphs (High-Low-Close)

Instant conversion between all graph types without re-entering data.

B/Graph can perform
- Statistical analysis
- Regression analysis
- Exponential smoothing
- Moving averages
- Graphing of 1-3 factors with up to 100 data points.

Save graphs as data files or high-res screens.
B/Graph Features
- Automatic labelling
- Re-labelling
- Custom labelling
- Full screen editor
- Multiple graph overlays
- Automatic or manual labelling
- Automatic and manual scaling
- Automated slide shows
- Grid overlays
- Multiple graph overlays
- Total screen color control

B/Graph works with graphics printers from
- Centronics
- Epson/Prowriter
- C. Itoh/Nec
- Seikosha
- Visicalc D.I.F. files
- User written programs

B/Graph comes with comprehensive professionally written documentation. A complete tutorial on Graphing/Charting and Statistics.

User registration with full program support.

INHOME SOFTWARE INCORPORATED
2485 Dunwin Drive, Unit 8, Mississauga, Ontario,
Canada, L5L 1T1
(416)828-0775

down during the night, and started his "wake-up" program running.

I tiptoed toward my bed.

Behind me, HERO came immediately to life and startled the heck out of me.

"Good morning, Fred!" he said cheerfully. "Time to wake up! Get out of bed, you sleepyhead. It's 11 p.m."

I spun around angrily. What was HERO doing delivering his wake-up message at eleven o'clock at night? Was he crazy? Was my program full of bugs?

Then I realized that HERO hadn't malfunctioned and that my program was working correctly. I had erred by starting the program running while the room was still lit. HERO had mistaken my bedside reading light for the early morning sun.

I turned out the bedside lamp. Phooey! I thought. If I want to read in bed, I'll have to do it under the covers with a flashlight.

When the lamp went out, the room became as black as the bottom of a well at midnight. I stumbled my way past two computers and a turtle robot to get to HERO on the far side of the room. I felt the keyboard carefully and pressed the keys to restart HERO's wake-up program. Then I made my way back to bed, trying carefully not to step on any of my mechanical friends in my bare feet. (Both computers were on the floor, along with the turtle. I like programming on the floor where I can spread out my work.)

I hopped into bed and fell immediately to sleep.

"Good morning, Fred!" HERO called. I sat up, shocked. It wasn't morning. The room was still pitch black.

"Time to wake up! Get out of bed, you sleepyhead," HERO continued. "It's 11:45 p.m."

What set him off this time? I wondered. Just then a car went by on the street beneath my window. The car headlights shone into my room. Then I realized what had happened. Headlights from a passing car had triggered HERO's wake-up message. I had to decrease his sensitivity to light.

I was beginning to get slightly paranoid (from lack of sleep). I was worried that HERO would wake me up if a tiny firefly flew into my bedroom. This kind of alarm clock I could do without.

I finished retuning HERO's program, turned out the lights, started the program running, and climbed back into bed. For a while I lay in bed gritting my teeth, just waiting for the starlight or some distant neon sign to set HERO off and give me that infernal, cheerful "Good Morning, Fred." But nothing happened, and I finally fell asleep.

Only A Supernova

The next morning I was awakened by someone hammering on my door. "Fred, are you in there? It's Steve. Are we going to breakfast, or not?"

It was my friend Steve Lubin.

"What time is it?" I asked.

"Eight thirty," Steve replied. "You ought to be up now."

Eight thirty! I had overslept!

I looked over at HERO. He hadn't been stolen. He was still there, looking peaceful, contented, and robotic. His red "power" light was on. I checked his display. The program was still working.

Then why hadn't he awakened me?

As I pulled on my pants, slipped into a shirt, and tied my shoes, it came to me. This time HERO's light sensor was set too *low*. The morning sun was not enough. HERO probably wouldn't wake me until the sun went supernova.

I finished getting dressed and went out to breakfast with Steve. As I ate a mushy egg on a bagel, I thought about my experiment to turn a robot into an alarm clock.

My experiment hadn't been too successful, but I resolved to keep trying.

The next night, HERO woke me up at 1 a.m. (a truck headlight) and at 4:30 a.m. (dawn came sooner than I realized).

But the next night after that, I finally got all the bugs out. HERO woke me up at 5:19. That was close enough to 5:30. I was satisfied. My robot alarm clock was working.

You Can't Tickle A Robot's Back

My experiment in turning a robot into an alarm clock was ultimately a success. But my experiment in robot and computer intimacy was a dismal failure.

Each night after my human colleagues at ORT abandoned me to return to their homes, I became desperately lonely. I was surrounded by friendly computers and robots, but I was still

lonely. I missed my wife and family, and I craved human companionship.

At first I tried to get the robots to simulate human companionship. I programmed the turtle to count to ten and say things like "Stop ... Go ... Left ... Right." And I taught HERO to say things like "That's a cute pair of pajamas you have on, Fred." But this wasn't the same as giving my son a piggyback ride to bed, or tickling my daughter's back, or reading the Sunday comics with my wife. Machines, even intelligent, friendly machines, could supplement human companionship, but they couldn't replace it. At least not in my life.

Pub Crawling, Crystal Balls, And Croissants

The Robotics Literacy course was a wild success. I learned more about robots than I had ever cared to learn. I learned about pneumatics, hydraulics, infrared sensors, flexible manufacturing systems, robot vision systems, stepper motors, servo motors, the whole bit.

And I taught my students all the things I knew. I told them about the giant industrial robots I had seen in Chicago, and the advanced thinking, feeling, sensing robots I had met at the Robotics Institute at Carnegie-Mellon University, in Pittsburgh. On the last day of the course, I gave my final lecture. I gazed into the crystal ball and talked about the future of robot-human relations. I talked about what sort of shape robots might take in the future, and the kind of impact they might have on our lives.

Then my wife came over to England, and I bailed out of the course. I had spent three weeks of almost nonstop work, surrounded by machines, and I'd had enough.

Before my wife arrived, in the evenings as a substitute for human companionship, I used to feast on chocolate éclairs and croissants at local patisseries (bakeries), then do a tour of the London bars and wash away my sorrows with warm, dark English beer. They call this sort of behavior "pub crawling," and it's a very apt phrase. By the end of an evening of videogames, whipped cream, and beer, I would totter back to my bedroom office, crawl into my bed, and dream strange dreams. In one dream, for example, I was pursued by my robots. HERO was chasing me, holding a buttery croissant in his gripper, and the turtle raced after me with a mug of beer sloshing around on his dome.

By the time Janet arrived in London, I'd had enough beer, enough sweets, and enough robots.

Robot In A Garbage Bag

But there was still one remaining chore. I had to return HERO to his home in Gloucester and teach the Zenith people in Gloucester some of the robot's finer points.

To get HERO back to Gloucester, Janet and I rented a little Ford Escort. We slipped a green garbage bag over HERO's head (so nobody would recognize him) and let him ride in the back seat with a seatbelt around his waist.

After only five minutes of practice to get used to driving on the right side of the car and on the left side of the road, we merged into mad, congested London traffic, and we were off.

Driving 60 miles an hour on the left-hand side of the road was scary. The trip to Gloucester left Janet and me shaken. But I didn't get a single complaint from HERO. In fact, he tolerated my driving beautifully. I banged his head against the car roof several times. And I jostled him against the front seat and bounced him off the back seat. He never complained. Instead, all I ever heard (from underneath the garbage bag) was an occasional, muffled "RED-DY."

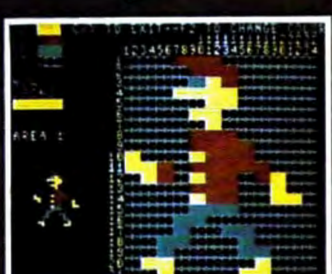
You can find out more about HERO by writing:

Douglas Bonham
Heath Company
Benton Harbor, Michigan 49022
(Or call: 616/982-3200)

On the road with Fred D'Ignazio and his friends at the ORT Robotics Literacy Course in London, England.

HERO the robot came from America. The Tasman Turtle came from Australia.

COMMODORE 64* SOFTWARE



SPRITEMASTER™ is not just another sprite editor. It's the finest utility available for multicolor sprite animation and game programming. It will have you making full color animated objects in just minutes. People running, birds flying or tanks rolling are a snap with Spritemaster. It will **automatically** append your sprites to other programs. It's easy to use and understand and comes with a full 21 page instruction manual and samples of animated sprites to get you started. (Suggested retail price... $35.95)

GENERAL QUARTERS! BATTLE STATIONS! As chief commander of land and sea forces in the Pacific, your mission is to obtain a quick naval victory, and invade enemy territory with land forces. **BEACH-HEAD**™ is a 100% machine language game and offers multi-screen action with high resolution, three dimensional graphics. (Suggested retail price... $34.95)

NEUTRAL ZONE™ takes you to the outer edges of the galaxy, to ALPHA IV, a long range early warning station whose mission is to detect alien intruders from other galaxies.

NEUTRAL ZONE™ is the ultimate in high resolution, fast action, arcade quality games. It is written in 100% machine language and features smooth scrolling of the 360 degree panorama. The realism is unbelievable. (Suggested retail price... $34.95)

SPRITEMASTER™, **NEUTRAL ZONE**™ AND **BEACH-HEAD**™
ARE AVAILABLE AT YOUR LOCAL COMMODORE DEALER
ON EITHER DISK OR TAPE.

ACCESS SOFTWARE INC

925 EAST 900 SOUTH, SALT LAKE CITY, UTAH 84105
TELEPHONE (801) 532-1134

*Commodore 64 is a registered trademark of Commodore Business Machines Inc.

You can find out more about the Tasman Turtle by writing:

Bill Glass
Tasman Turtle
260 Beacon Street
Somerville, MA 02143
(Or call: 617/492-0660)

You can find out more about the Robotics Literacy Course by writing me:

Fred D'Ignazio
2117 Carter Road, SW
Roanoke, VA 24015

In coming months "on the road," Fred will tell about:

- An educator who uses computer magic shows to teach children about computers.
- A London company that makes innovative educational software for children.
- The British government's effort to put computers and robots in all primary and secondary schools in the United Kingdom.
- The new British educational robots that will soon be "invading" our country.
- A look at the way computers, robots, and other high-technology subjects are integrated into the British school curriculum as "disciplines" children can learn to become *knowledge workers* of the future. ©

Spelling Quiz

Edward Perrin

Learning to spell is simplified by this program. It lets you create weekly lists of up to 20 words each, save the lists to tape or disk, and then retrieve the lists for practice sessions. For Atari and VIC.

"Spelling Quiz" allows you or your child to enter weekly spelling words into the computer and save them on tape or disk. All the words for an entire year can be saved at once or each week can be saved separately as the school year progresses.

The program allows you to enter up to 20 words at a time. Most weekly spelling assignments are no more than 20 words. The program prompts are self-explanatory, but it would be helpful to read through the following instructions.

Load in the program with the BASIC cartridge inserted. The loading time for tape is about four minutes.

Type in RUN, and after the title page you will be asked if you want to *C*reate or *R*etrieve a list of words to work on. You will also be asked if you are using a *D*isk or *T*ape.

Creating Word Lists

To create a list, simply type in up to 20 words, no more than 20 letters each and with no leading or trailing spaces, one at a time, and hit the RETURN key. Be sure each word is spelled correctly before hitting RETURN. If you enter fewer than 20 words, type in an * following the last input. After the last word or * you will be asked to type in some sort of identifier for that particular list. Use "Chapter 4" or "List 189," for example. You must use some type of identifier that your child will understand. The identifier is used to make sure your child has retrieved the correct list.

Disk users will be asked to enter a filename. Only the filename is necessary; the program will supply the "D1:". Be sure to make the name unique and meaningful.

Tape users will need a blank tape or a tape which has been used to save other word lists. Be sure to note the tape counter number on a sheet of paper and store the paper with the word tape.

If you already have words stored, just follow the prompts to LOAD the words.

Check the list and the identifier to be sure that this is the list you wanted to use. If not, you have the option to LOAD a new list or create a new one as needed.

Once the words are LOADed in with the create or retrieve option, your child is ready to use the program. You now have the option to either spell a *C*ertain number of words correctly, spell an unlimited number of words correctly, or *E*nd.

If you choose the C option, you will be graded, and the program will terminate when the number of words spelled correctly equals the number you entered at the prompt. If you choose the unlimited option (by pressing RETURN), you can spell only 10,000 words before the program terminates. It is easy to change the 10,000 to another upper limit. Change the number in the last line of the program to stop the program automatically at a preset number.

The Quiz Begins

When you have made all of the choices, the game is ready to play. The screen will show the number of the word being scrambled, the score (how many words you have spelled correctly), a scrambled word, and the attempt number. At the bottom of the screen is a GRAPHICS 0 window where you will type your answers. The word number on top will help if you cannot figure out the word. The program is supposed to check spelling competence rather than ability to unscramble words, so there is no penalty for not unscrambling correctly. Use this option as you wish.

Your child will then have three tries to spell the word correctly. If correct, the screen will respond with an encouraging CORRECT and a happy sound. After three tries, the program will give the correct spelling and set up a different screen to allow the child to practice the misspelled word.

Practice Screen

The practice screen will not allow misspellings. It does allow the child to exit when he or she wants to by pressing the *. In this mode, be sure to press the space bar after each word and not the RETURN. Pressing RETURN would cause the computer to register an error in the spelling.

After the number of correct spellings equals the number put in at the beginning, or if your child enters * instead of spelling a word during the main run, the quiz will end and the child will be graded on his or her performance. If you think the grading is too strict, change the limits in the grading subroutine in lines 9000-9400.

After the grading, the player can go back and retrieve or create and save a new file or use the same words that are already in the computer's memory. He or she also has the option to end at this time. If the player continues, the whole cycle repeats.

Spelling Quiz For The VIC

Marc Sugiyama, Programming Assistant

In Spelling Quiz for the VIC (Program 2), the words can be only 14 letters long; however, you are still allowed a maximum of 20 words. Remember, though, that memory is tight in an unexpanded VIC, so too many long words may cause an out-of-memory error. The percentage calculated at the end of the quiz reflects only the last 20 questions, not all of the questions asked during the session.

The VIC version of Spelling Quiz is easily converted to other computers; however, certain changes must be made. The cursor control characters (that is, those to clear the screen or to change the cursor's color) must be adapted to your computer. Three different POKE commands are used in this program. The first, to location 36879, controls the screen and border colors. The second POKE command, using location 214, changes the vertical position of the cursor. The last, involving location 204, turns the cursor on and off. When 204 contains a zero, the cursor is on; otherwise, the cursor is off.

Program 1: Spelling Quiz For Atari

```
2 REM SPELLING QUIZ
10 DIM A$(20),B$(20),C$(20),D$(20),E$(20),F$(20),G$(20),H$(20),I$(20),J$(20),K$(20),L$(20),M$(20),N$(20),O$(20)
20 DIM P$(20),Q$(20),R$(20),S$(20),T$(20),U$(20),ARRAY(20),Z$(20),STANDINGS(20),ANS$(1),WORD$(520),INWORD$(128)
30 DIM WEL$(38),DK$(15),ZZ$(1):TIME=0
40 GOSUB 3000
54 GOSUB 13000
55 PRINT "HOW MANY WORDS DO YOU WISH TO SPELL{3 SPACES}CORRECTLY BEFORE ENDING THIS DRILL?{3 SPACES}ENTER 0 TO END PROGRAM"
56 TRAP 56:INPUT RIGHT:IF RIGHT=0 THEN 4000
57 TRAP OFF:SCORE=0:ATT=0
58 GOSUB 5200:TRAP OFF
59 W1=0:W2=0:W3=0:W4=0:W5=0:W6=0:W7=0:W8=0:W9=0:W10=0:W11=0:W12=0:W13=0:W14=0:W15=0:W16=0:W17=0:W18=0:W19=0:W20=0
60 IF SCORE=RIGHT THEN 1000
65 NUM=1:W=INT(20*RND(1)+1)
70 GOSUB 7000:IF A$="{20 SPACES}" THEN NUM=-1:GOSUB 7000:GOTO 65
80 GRAPHICS 2:POKE 708,0:FOR AR=1 TO 20:ARRAY(AR)=-1:NEXT AR:PRINT "{BELL}";"INPUT '*' TO END QUIZ"
90 POSITION 11,3:? #6;"SCORE";" ";SCORE:POSITION 2,0:? #6;"#";W;" ON YOUR LIST"
110 FOR L=1 TO 20:IF A$(L,L)=" " THEN L=L-1:A$=A$(1,L):GOTO 115
112 NEXT L:IF L=21 THEN L=20
115 FOR LTR=1 TO L
120 ARR=INT(L*RND(0)+1):IF ARRAY(ARR)=1 THEN 120
125 P=ARR-1
130 POSITION P,7:? #6;A$(LTR,LTR):ARRAY(ARR)=1:NEXT LTR:POKE 708,200
135 TRY=0
137 TRY=TRY+1:ATT=ATT+1:POSITION 0,9:? #6;"ATTEMPT # ";ATT
140 INPUT Z$:IF Z$="*" THEN ATT=ATT-1:NUM=-1:GOSUB 7000:GOTO 1000
141 IF Z$=A$ THEN SCORE=SCORE+1:FOR N=100 TO 10 STEP -1:SOUND 0,N,10,10:NEXT N:SOUND 0,0,0,0
142 IF Z$=A$ THEN POSITION 12,6:? #6;"correct!":FOR N=1 TO 300:NEXT N:GOTO 60
143 IF TRY=3 THEN FOR N=1 TO 100:SOUND 0,20,4,10:NEXT N:SOUND 0,0,0,0:GOTO 2000
144 POSITION 0,2:? #6;"wrong...":POSITION 0,3:? #6;"try again.."
145 FOR N=1 TO 100:SOUND 0,11,4,10:NEXT N:SOUND 0,0,0,0
147 POSITION 0,2:? #6;"{8 SPACES}":POSITION 0,3:? #6;"{11 SPACES}"
150 GOTO 137
1000 GRAPHICS 18:PRINT #6;"  your score is ";SCORE
1010 PRINT #6:PRINT #6;"time to quit for now"
1020 ? #6:? #6:? #6;"{3 SPACES}CONGR
```

```
     ATULATIONS"
1030 ? #6;"on a job well done!!":? #
     6:? #6;" QUIZ WILL REPEAT!"
1040 GOSUB 5000:SOUND 0,0,0,0:SOUND
     1,0,0,0:GOTO 9000
2000 POSITION 0,5:? #6;A$:POSITION 0
     ,4:? #6;"answer:":NUM=-1:GOSUB
     7000
2010 FOR N=1 TO 1000
2020 NEXT N:GOSUB 8000
2030 GOTO 60
3000 GRAPHICS 18:POSITION 0,4:? #6;"
     {3 SPACES}SPELLING  quiz"
3005 GOSUB 5000:SOUND 1,0,0,0:SOUND
     0,0,0,0:RETURN
4000 GRAPHICS 18:POSITION 0,2:? #6;"
     VERY GOOD WORK..."
4010 POSITION 0,6:? #6;"see you agai
     n later"
4020 POSITION 4,10:? #6;"bye for now
     "
4030 GOSUB 5000:GOTO 7030
5000 FOR N=1 TO 200
5010 SOUND 0,RND(0)*200,10,2
5030 NEXT N
5040 RETURN
5100 FOR N=1 TO 100:SOUND 0,N,10,10:
     NEXT N:SOUND 0,0,0,0:RETURN
5200 FOR N=255 TO 200 STEP -1:SOUND
     0,N,10,10:NEXT N:FOR N=225 TO 1
     50 STEP -1:SOUND 0,N,10,10:NEXT
      N
5210 FOR N=175 TO 100 STEP -1:SOUND
     0,N,10,10:NEXT N:FOR N=150 TO 5
     0 STEP -1:SOUND 0,N,10,10:NEXT
     N:SOUND 0,0,0,0:RETURN
7000 IF W=1 THEN B$=WORD$(1,20):A$=B
     $:W1=W1+NUM
7001 IF W=2 THEN C$=WORD$(21,40):A$=
     C$:W2=W2+NUM
7002 IF W=3 THEN D$=WORD$(41,60):A$=
     D$:W3=W3+NUM
7003 IF W=4 THEN E$=WORD$(61,80):A$=
     E$:W4=W4+NUM
7004 IF W=5 THEN F$=WORD$(81,100):A$
     =F$:W5=W5+NUM
7005 IF W=6 THEN G$=WORD$(101,121):A
     $=G$:W6=W6+NUM
7006 IF W=7 THEN H$=WORD$(121,140):A
     $=H$:W7=W7+NUM
7007 IF W=8 THEN I$=WORD$(141,160):A
     $=I$:W8=W8+NUM
7008 IF W=9 THEN J$=WORD$(161,180):A
     $=J$:W9=W9+NUM
7009 IF W=10 THEN K$=WORD$(181,200):
     A$=K$:W10=W10+NUM
7010 IF W=11 THEN L$=WORD$(201,220):
     A$=L$:W11=W11+NUM
7011 IF W=12 THEN M$=WORD$(221,240):
     A$=M$:W12=W12+NUM
7012 IF W=13 THEN N$=WORD$(241,260):
     A$=N$:W13=W13+NUM
7013 IF W=14 THEN O$=WORD$(261,280):
     A$=O$:W14=W14+NUM
7014 IF W=15 THEN P$=WORD$(281,300):
     A$=P$:W15=W15+NUM
7015 IF W=16 THEN Q$=WORD$(301,320):
     A$=Q$:W16=W16+NUM
7016 IF W=17 THEN R$=WORD$(321,340):
     A$=R$:W17=W17+NUM
7017 IF W=18 THEN S$=WORD$(341,360):
     A$=S$:W18=W18+NUM
7018 IF W=19 THEN T$=WORD$(361,380):
     A$=T$:W19=W19+NUM
7019 IF W=20 THEN U$=WORD$(381,400):
     A$=U$:W20=W20+NUM
7020 RETURN
7030 END
8000 GRAPHICS 18:POKE 708,100:? #6;"
     please type the word":POSITION
     0,1:? #6;A$
8005 POSITION 0,2:? #6;"and hit spac
     ebar..."
8007 POSITION 0,3:? #6;"UNTIL SCREEN
      IS FULL"
8010 POSITION 0,4:? #6;"or type * to
      return":L=0:COUNTER=0
8100 OPEN #1,4,0,"K:"
8150 L=L+1
8200 GET #1,CHAR
8300 CLOSE #1
8350 IF CHR$(CHAR)="*" THEN GOTO 60
8355 IF CHR$(CHAR)=" " THEN L=0:GOSU
     B 8400
8360 IF CHR$(CHAR)<>A$(L,L) THEN GOS
     UB 12000:GOTO 8000
8370 IF L=LEN(A$) THEN L=0
8400 COUNTER=COUNTER+1:PRINT #6;CHR$
     (CHAR);:TRAP 40000
8450 IF COUNTER>139 THEN GOTO 60
8500 GOTO 8100
8600 RETURN
9000 POKE 752,1:PRINT "HERE IS A LIS
     T OF HOW MANY TIMES EACH WORD W
     AS SPELLED CORRECTLY THIS TIME.
     "
9010 NUM=0:FOR W=1 TO 20:GOSUB 7000:
     NEXT W
9050 FOR N=100 TO 240:SOUND 0,N,10,1
     0:NEXT N:SOUND 0,0,0,0
9100 ? W1;" ";B$:? W2;" ";C$:? W3;"
     ";D$:? W4;" ";E$:? W5;" ";F$:?
     W6;" ";G$:? W7;" ";H$:? W8;" ";
     I$:? W9;" ";J$:? W10;" ";K$
9150 ? W11;" ";L$:? W12;" ";M$:? W13
     ;" ";N$:? W14;" ";O$:? W15;" ";
     P$:? W16;" ";Q$:? W17;" ";R$:?
     W18;" ";S$:? W19;" ";T$
9200 ? W20;" ";U$
9250 POKE 752,1:POSITION 25,3:PRINT
     "ATTEMPTS":POSITION 28,5:PRINT
     ATT:FOR N=1 TO 200:SOUND 0,255,
     10,8:NEXT N
9260 POSITION 25,7:PRINT "CORRECT":P
     OSITION 28,9:PRINT SCORE:FOR N=
     1 TO 200:SOUND 0,200,10,8:NEXT
     N
9270 TRAP 9400:PER=INT((SCORE/ATT)*1
     00):POSITION 25,11:PRINT "PERCE
     NT":POSITION 28,13:PRINT PER;"%
     "
9280 FOR N=1 TO 200:SOUND 0,100,10,6
     :NEXT N
9300 POSITION 25,15:? "GRADE:"
9310 IF PER>=95 THEN POSITION 27,17:
     ? " A ":POSITION 25,21:? "EXCEL
     LENT!!!"
9320 IF PER>=88 AND PER<95 THEN POSI
     TION 27,17:? " B ":POSITION 25,
     21:? "VERY GOOD!!!"
9330 IF PER>=78 AND PER<88 THEN POSI
     TION 27,17:? " C ":POSITION 25,
     21:? "GOOD!!!"
9340 IF PER>=70 AND PER<78 THEN POSI
```

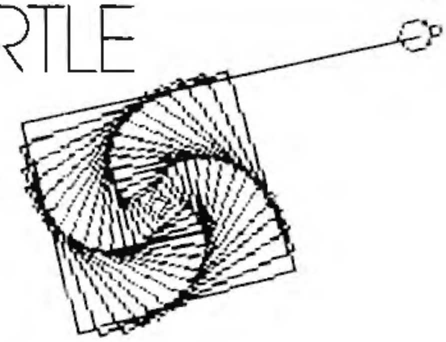
```
      TION 27,17:? "D":POSITION 25,
      21:? "HMMMM!!"
9350 IF PER<70 THEN POSITION 27,17:?
      "F":POSITION 25,21:? "STUDY!
      !!"
9360 SOUND 0,0,0,0:POKE 752,0:GOTO 5
      4
9400 PER=0:POSITION 28,13:PRINT PER:
      POSITION 25,11:PRINT "PERCENT":
      GOTO 9280
10000 WEL$="WELCOME TO THE WORLD OF
      SPELLING QUIZ":PRINT "{CLEAR}"
      :FOR N=1 TO 37:PRINT WEL$(N,N)
      ;:NEXT N:TIME=TIME+1
10010 OFF=40000:P=0:? :? "DO YOU WAN
      T TO CREATE OR RETRIEVE THE FI
      LE";
10011 O$="{20 SPACES}":FOR N=1 TO 520
       STEP 20:WORD$(N,N+19)=O$:NEXT
       N
10012 ? :? :? "ONCE YOU CREATE A FIL
      E IT WILL BE{5 SPACES}STORED O
      N TAPE OR DISK SO YOU CAN
      {5 SPACES}INPUT THE WORDS FROM
      "
10013 ? "THE TAPE OR DISK INSTEAD OF
        TYPING{3 SPACES}THE SAME WOR
      DS IN EVERY TIME YOU PLAY."
10014 ? :? :? "TYPE IN C OR R AND HI
      T RETURN NOW!"
10015 TRAP 10014:INPUT ANS$
10017 TRAP 10017:? "ARE YOU USING TA
      PE OR DISK":INPUT ZZ$:IF ZZ$(1
      ,1)<>"T" AND ZZ$(1,1)<>"D" THE
      N 10017
10020 IF ANS$<>"C" THEN GOTO 11000
10100 ? "TYPE IN WORDS NOW":N=1
10105 ? :? :? "BE SURE EACH WORD IS
      SPELLED CORRECTLY BEFORE YOU P
      RESS RETURN{14 SPACES}"
10110 FOR N=1 TO 400 STEP 20:INPUT I
      NWORD$
10120 IF N>399 THEN WORD$(401,520)="
       ":GOTO 10200
10125 IF INWORD$="*" THEN WORD$(N,52
      0)=" ":GOTO 10150
10130 WORD$(N,N+19)=INWORD$
10140 NEXT N
10150 ? "TYPE IN CHAPTER # OR LIST #
       ETC..."
10160 INPUT INWORD$:WORD$(401,420)=I
      NWORD$
10200 FOR N=1 TO 420 STEP 20:PRINT W
      ORD$(N,N+19):NEXT N
10202 IF ZZ$="D" THEN GOSUB 10500:TR
      AP 40000:OPEN #2,8,0,DK$:GOTO
      10209
10203 ? "POSITION THE TAPE AND TAKE
      NOTE OF{4 SPACES}THE COUNTER N
      UMBER.":? :? "PRESS THE PLAY A
      ND RECORD BUTTONS."
10204 ? :? "WHEN THE BUZZER SOUNDS,
      PRESS RETURN"
10205 N=1
10206 TRAP 10207:LPRINT
10207 OPEN #2,8,0,"C:"
10209 N=1:FOR X=1 TO 4
10210 PRINT #2;WORD$(N,N+119):N=N+12
      0
10220 NEXT X:CLOSE #2
10300 GOTO 13000
10500 PRINT "YOU MUST NOW ENTER THE
      FILENAME{7 SPACES}(WITHOUT 'D:
      ') OF THE FILE TO ";
10520 IF ANS$="R" THEN PRINT "LOAD":
      GOTO 10530
10525 PRINT "CREATE"
10530 TRAP 10500:INPUT DK$:DK$(4)=DK
      $:DK$(1,3)="D1:"
10540 RETURN
11000 IF ZZ$="D" THEN GOSUB 10500:N=
      1:TRAP 40000:OPEN #2,4,0,DK$:G
      OTO 11025
11005 ? "TO LOAD WORDS THAT ARE STOR
      ED ON TAPE BE SURE TO POSITION
       THE TAPE AT THE{3 SPACES}CORR
      ECT COUNTER # YOU NEED."
11010 ? "WHEN BUZZER SOUNDS, PRESS R
      ETURN AND WAIT FOR THE WORDS
      TO BE LOADED INTO THE COMPUTE
      R..."
11020 N=1:OPEN #2,4,0,"C:"
11025 FOR X=1 TO 4
11030 TRAP 11040:INPUT #2,INWORD$
11035 WORD$(N,N+119)=INWORD$:N=N+120
11040 NEXT X
11045 CLOSE #2
11060 FOR N=1 TO 400 STEP 20:PRINT I
      NT(N/20)+1;" ";WORD$(N,N+19)
11067 NEXT N
11068 PRINT "{9 SPACES}";WORD$(401,4
      20)
11070 ? "IS THIS THE GROUP OF WORDS
      THAT YOU{3 SPACES}WANTED (Y/N)
      ";:INPUT ANS$:IF ANS$="N" THEN
       GOTO 10000
11075 GOTO 13019
11080 END
12000 FOR N=1 TO 100:SOUND 0,20,4,10
      :NEXT N:SOUND 0,0,0,0
12005 GRAPHICS 18:? #6;"you typed ";
      CHR$(CHAR);" this is "
12010 POSITION 0,1:? #6;"wrong...TRY
       AGAIN"
12020 POSITION 4,3:? #6;"THE WORD IS
      "
12030 POSITION 0,4:? #6;A$
12040 POSITION 6,5:? #6;"READY???"
12050 L=0:FOR N=1 TO 400:NEXT N:RETU
      RN
13000 WEL$="NOW YOU MUST MAKE A BIG
      DECISION!!!":FOR N=1 TO 35:PRI
      NT WEL$(N,N);:NEXT N:NUM=1
13005 TRAP 13010:? :? "HIT THE RETUR
      N KEY WHEN READY";:INPUT A:IF
      A=0 THEN END
13010 PRINT "{CLEAR}{BELL}":TRAP OFF
13011 ? :? :? :? :? "DO YOU WANT TO
      USE THE LIST OF WORDS ALREADY
       IN THE COMPUTER OR DO YOU WAN
      TTO LOAD IN A NEW LIST"
13012 ? :? :? "TYPE IN N FOR A NEW L
      IST OR HIT RETURNTO USE THE O
      LD LIST.":? :? :? "TYPE IN E T
      O END"
13013 ? :? :? "OF COURSE, IF THIS IS
       THE FIRST TIME THROUGH THE P
      ROGRAM DURING THIS{7 SPACES}SE
      SSION YOU MUST HIT N!!!"
13014 INPUT ANS$:IF ANS$="N" THEN GO
      TO 10000
13015 IF ANS$="E" THEN GOTO 4000
13016 IF TIME=0 THEN ? "{3 BELL}":GO
```

```
       TO 13018
13017 GOTO 13019
13018 ? :? :? "THIS IS YOUR FIRST TI
       ME THROUGH THE{3 SPACES}PROGRA
       M. YOU MUST LOAD IN OR CREATE
       A NEW LIST NOW!":GOTO 13011
13019 PRINT "{CLEAR}"
13020 ? :? :? "IF YOU WANT TO PRACTI
       CE FOR A CERTAIN NUMBER OF TIM
       ES TYPE IN C AND HIT
       {5 SPACES}RETURN."
13025 ? :? :? "IF YOU WANT TO PRACTI
       CE UNTIL YOU GET TIRED JUST HI
       T RETURN."
13030 ? :? :? "IF YOU WANT TO QUIT,
       TYPE IN E."
13050 INPUT ANS$:IF ANS$="C" THEN GO
       TO 55
13060 IF ANS$="E" THEN GOTO 4000
13070 RIGHT=10000:GOTO 57
```

Program 2:
Spelling Quiz For VIC (Microsoft BASIC)

```
5 REM SPELLING QUIZ
10 DIMA%(14),W%(20),W$(20):TT=0
15 POKE36879,93:PRINT"{CLR}{2 DOWN}{RED}
   {4 SPACES}SPELLING QUIZ{2 DOWN}":GOTO3
   15
20 PRINT"{CLR}{2 DOWN}HOW MANY WORDS DO Y
   OU WISH TO SPELL":PRINT"CORRECTLY BEFO
   RE"
25 PRINT"ENDING?":PRINT"{DOWN}{WHT}RETURN
   {RED} TO END? ";:GOSUB380
30 RI=VAL(IN$):IFIN$=""THEN120
35 SC=0:AT=0:FORI=1TO20:W%(I)=0:NEXT
40 IFSC=RITHEN110
45 W=INT(20*RND(1)+1):A$=W$(W):IFA$=""THE
   N45
50 FORAR=1TO14:A%(AR)=0:NEXT:PRINT"{CLR}
   {4 DOWN}UNSCRAMBLE:{BLU}":L=LEN(A$):FO
   RLT=1TOL
55 AR=INT(L*RND(1)+1):IFA%(AR)THEN55
60 PRINTMID$(A$,AR,1);:A%(AR)=1:NEXT:TR=0
65 POKE214,20:PRINT:PRINT"{RED}PRESS
   {WHT}RETURN{RED} TO END"
70 PRINT"{HOME}CORRECT{WHT}"SCTAB(13)"
   {RED}WORD{WHT}"W
75 TR=TR+1:PRINT"ATTEMPT"AT+1"{4 DOWN}"
80 POKE214,3+TR*3:PRINT:PRINT"{DOWN}{WHT}
   TRY#"TR:PRINT"{RED}? ";:GOSUB380:IFIN$
   =""THEN110
85 AT=AT+1:IFIN$<>A$THEN95
90 CR$=LEFT$("R"+CR$,20):PRINT"{DOWN}CORR
   ECT!":FORN=1TO500:NEXT:W%(W)=W%(W)+1:S
   C=SC+1:GOTO40
95 IFTR=3THEN115
100 CR$=LEFT$("W"+CR$,20)
105 PRINT"{DOWN}WRONG, TRY AGAIN.{UP}":FO
    RN=1TO700:NEXT:PRINT"{17 SPACES}":GOT
    O70
110 PRINT"{CLR}{2 DOWN}{2 SPACES}YOUR SCO
    RE IS{WHT}"SC:PRINTSPC(6)"{RED}
    {2 DOWN}WELL DONE":GOSUB370:GOTO155
115 PRINT"{DOWN}ANSWER{WHT} ":PRINTA$:FOR
    N=1TO1000:NEXT:GOTO125
120 PRINT"{CLR}{DOWN}GOOD WORK":END
125 PRINT"{CLR}{DOWN}{RED}TRY TYPING THE
    {SPACE}WORD{WHT}":PRINTA$:PRINT"{RED}
    {DOWN}PRESS {WHT}RETURN{RED} TO END":
    L=1
130 PRINT" {LEFT}";:POKE204,0
135 GETC$:IFC$=CHR$(13)THENPOKE204,1:GOTO
    40
140 IFC$<>MID$(A$,L,1)THEN135
145 POKE204,1:IFL=LEN(A$)THENL=0:C$=C$+"
    {SPACE}"
150 L=L+1:PRINTC$;:GOTO130
155 PRINT"{CLR}{DOWN}HERE IS A LIST OF HO
    W MANY TIMES EACH WORD{2 SPACES}WAS S
    PELLED CORRECTLY:"
160 FORI=1TO20:PRINT"{WHT}"I"{RED}{LEFT}"
    W%(I)W$(I):IFI=9THENGOSUB370:PRINT"
    {CLR}{DOWN}"
165 IFI<20ANDW$(I+1)=""THENI=21
170 NEXT:GOSUB370
175 PRINT"{CLR}{DOWN}ATTEMPTS{WHT}"AT:PRI
    NT"{RED}{DOWN} CORRECT{WHT}"SC:L=LEN(
    CR$):IFL=0THEN315
180 C=0:FORI=1TOL:C=C-(MID$(CR$,I,1)="R")
    :NEXT:PE=INT(C/L*100):PRINT"{RED}
    {DOWN} PERCENT{WHT}"PE"{LEFT}%"
185 CR$="":FORG=1TO5:READDA,G$,C$:IFPE<DA
    THENNEXT
190 DATA90,A,EXCELLENT,80,B,VERY GOOD,70,
    C,GOOD,60,D,HMMM,,F,STUDY
195 RESTORE:PRINT"{RED}{DOWN}GRADE{WHT}
    {RVS} "G$" {OFF}{RED}"C$"!":GOTO315
200 TT=1:P=0:PRINT"{CLR}{3 DOWN}{PUR}PRES
    S:{DOWN}":PRINT"{WHT}C{RED} TO CREATE
    {DOWN}":PRINT"{PUR}OR{DOWN}
205 PRINT"{WHT}R{RED} TO RETRIEVE A FILE
    {DOWN}":FORI=1TO20:W$(I)="":NEXT:GOSU
    B375
210 IFAN$<>"C"THEN280
215 PRINT"{CLR}{DOWN}ENTER UP TO 20 WORDS
    {2 SPACES}ONE BY ONE AND PRESS
    {2 SPACES}{WHT}RETURN{RED}"
220 PRINT"{DOWN}LIMIT EACH WORD TO 14 LET
    TERS.
225 PRINT"{DOWN}PRESS {WHT}RETURN{RED} WH
    EN YOU ARE FINISHED.
230 PRINT"{2 DOWN}BE SURE YOU SPELL THE W
    ORDS CORRECTLY"
235 PRINT"{DOWN}{WHT}ENTER YOUR WORDS
    {DOWN}{RED}":FORN=1TO20
240 PRINT"{WHT}"N"{RED}? ";:GOSUB380:IFIN
    $=""THEN250
245 W$(N)=IN$:NEXT
250 PRINT"{DOWN}ENTER AN IDENTIFIER":PRIN
    T"{DOWN}? ";:GOSUB380:IFIN$=""THENIN$
    =" "
255 W$(0)=IN$:PRINT"{CLR}{2 DOWN}":FORI=1
    TO20:IFW$(I)=""THEN270
260 IFPOS(0)+LEN(W$(I))>20THENPRINT
265 PRINTW$(I)",";:NEXT
270 PRINT:PRINT"{WHT}"W$(0):PRINT"{DOWN}
    {WHT}POSITION THE TAPE{RED}":GOSUB370
275 PRINT"{CLR}":OPEN2,1,1,W$(0):FORI=0TO
    20:PRINT#2,W$(I):NEXT:PRINT#2,W$(0):C
    LOSE2:GOTO315
280 PRINT"{DOWN}POSITION THE TAPE":GOSUB3
    70:PRINT"{CLR}{2 DOWN}"
285 OPEN2,1,0:INPUT#2,W$(0):PRINT"{DOWN}N
    AME {BLU}"W$(0)"{RED}{DOWN}"
290 FORI=1TO20:INPUT#2,W$(I):IFPOS(0)+LEN
    (W$(I))>20THENPRINT
295 PRINTW$(I)" ";:NEXT:CLOSE2
300 PRINT:PRINT"{DOWN}{RED}IS THIS THE GR
    OUP OF":PRINT"WORDS YOU WANTED";:GOSU
    B375
305 IFAN$="N"THEN200
310 GOTO345
```

```
315 GOSUB370:IFTT=0THEN200
320 PRINT"{CLR}{2 DOWN}{PUR}PRESS:{DOWN}"
325 PRINT"{WHT}RETURN{RED} TO USE THE":PR
    INT" WORDS ALREADY IN THE{2 SPACES}CO
    MPUTER{DOWN}"
330 PRINT"{PUR}OR{DOWN}":PRINT"{WHT}N
    {RED} TO LOAD A LIST{2 DOWN}":PRINT"
    {PUR}OR{DOWN}":PRINT"{WHT}E{RED} TO E
    ND{DOWN}"
335 GOSUB375:IFAN$="N"THEN200
340 IFAN$="E"THEN120
345 PRINT"{CLR}{2 DOWN}{PUR}PRESS:{DOWN}"
    :PRINT"{WHT}RETURN{RED} TO PRACTICE":
    PRINT" UNTIL YOU GET TIRED{DOWN}"
350 PRINT"{PUR}OR{DOWN}":PRINT"{WHT}C
    {RED} TO PRACTICE A":PRINT" CERTAIN N
    UMBER OF":PRINT" TIMES{DOWN}"
355 PRINT"{PUR}OR{DOWN}":PRINT"{WHT}E
    {RED} TO QUIT{DOWN}":GOSUB375:IFAN$="
    C"THEN20
360 IFAN$="E"THEN120
365 RI=10000:GOTO35
370 PRINT"{DOWN}{2 SPACES}PRESS {WHT}RETU
    RN{RED}";
375 PRINT"? ";:GOSUB380:AN$=LEFT$(IN$,1):
    RETURN
380 IN$="":ZL=0
385 POKE204,0:PRINT"*{LEFT}";
390 GETZ$:IFZ$=""THEN390
395 POKE204,1:PRINT" {LEFT}";:IFZ$=CHR$(1
    3)THENPRINT:RETURN
400 IFZ$=CHR$(20)ANDZLTHENZL=ZL-1:IN$=LEF
    T$(IN$,ZL):PRINTZ$;
405 IF((Z$>"/"ANDZ$<":")OR(Z$>"@"ANDZ$<"[
    "))ANDZL<14THENPRINTZ$;:IN$=IN$+Z$:ZL
    =ZL+1
410 GOTO385
```
©



Ed Emberley's Drawing Procedures

Part of the appeal of turtle graphics is that it allows complex pictures to be built from simple building blocks. This feature arises from the fact that each shape description or procedure describes the shape itself, independently of its starting point or orientation. For example, once a square is defined with the procedure:

```
TO SQUARE :SIZE
    REPEAT 4 [FORWARD :SIZE RIGHT 90]
END
```

the computer can use this procedure to create a square of any size at any starting location and orientation.

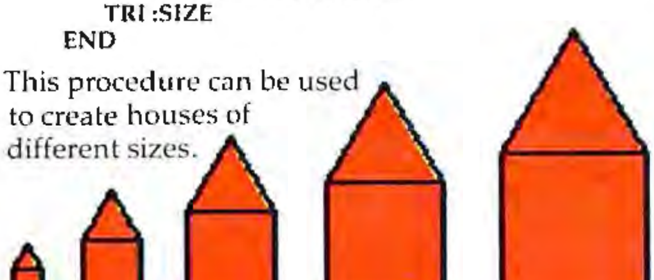
If the user has built up a set of useful geometric procedures, these can be combined to create more complex figures. If one also has a procedure for drawing triangles:

```
TO TRI :SIZE
    REPEAT 3 [FORWARD :SIZE RIGHT 120]
END
```

then a procedure for drawing a house can be created from a combination of a square and a triangle:

```
TO HOUSE :SIZE
    SQUARE :SIZE
    FORWARD :SIZE RIGHT 30
    TRI :SIZE
END
```

This procedure can be used to create houses of different sizes.



Many turtle graphics enthusiasts create libraries of basic figures from which quite interesting pictures can be created.

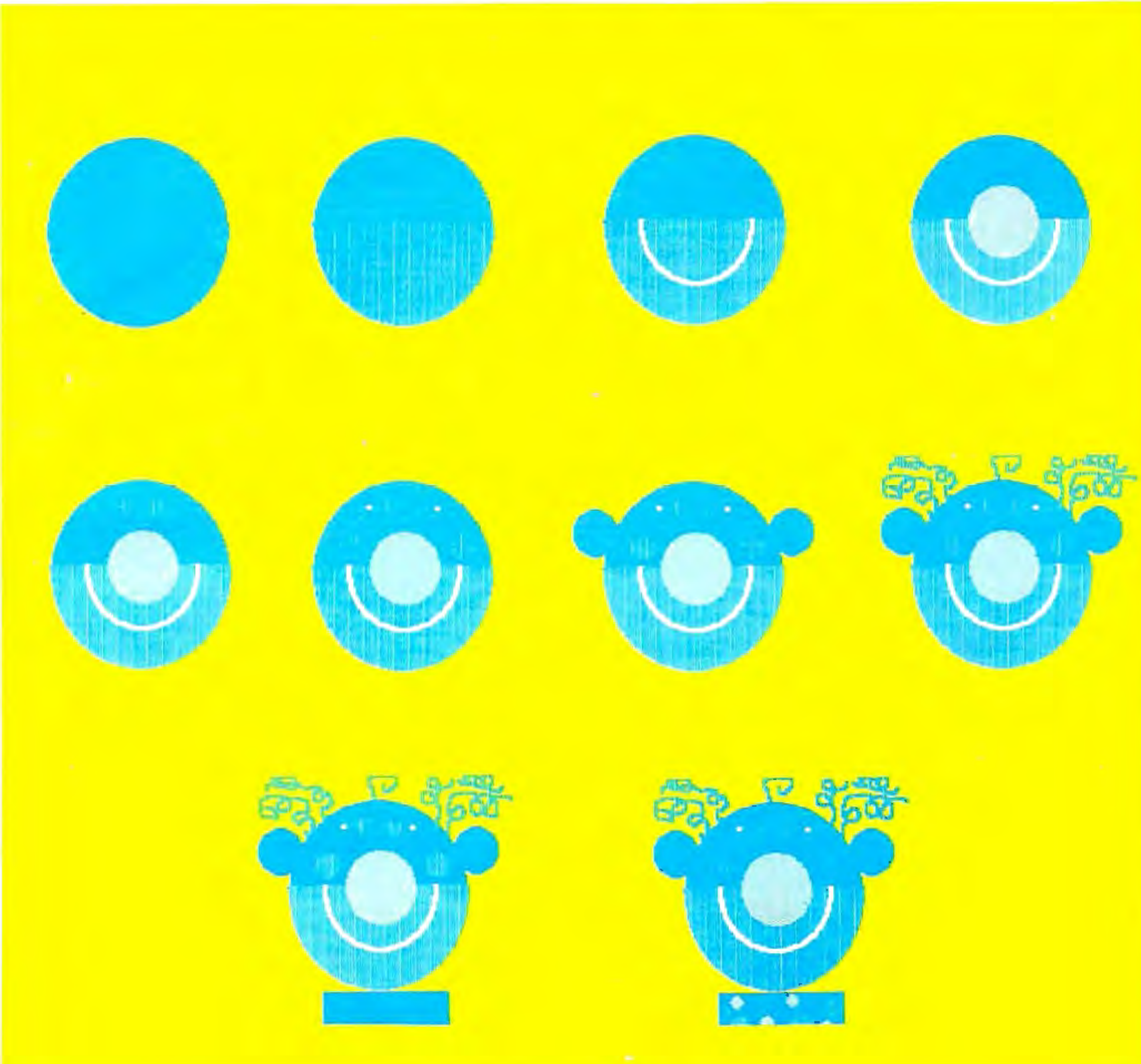
As an active proponent of turtle graphics and procedural problem-solving, I was delighted to find Ed Emberley's independent discoveries along these lines.

Ed Emberley has written several books on illustration for children. His books of particular interest to readers of this column would include: *Ed Emberley's Drawing Book of Animals*, *Ed Emberley's Drawing Book*, *Make a World*, *Ed Emberley's Big Orange Drawing Book*, and *Ed Emberley's Big Purple Drawing Book* (all published by Little, Brown and Co.).

Mr. Emberley's illustration technique is built on the idea that, just as words are created from an alphabet of letters, pictures can be created from an alphabet of shapes. He shows how to create myriad figures using circles, rectangles, arcs, lines, triangles, and other simple pieces. By building the figure piece by piece, the young artist is never overwhelmed by trying to deal with the whole figure at once. The following series of illustrations (courtesy of Mr. Emberley) shows how one can create a clown's head almost entirely from circles and circle parts.

If you were to create this figure using turtle graphics procedures, you would need only procedures for a circle, an arc, a rectangle, and the squiggles for the hair.

Ed Emberley does not normally use a computer to create his illustrations. The clown figures shown on the next page are a happy exception to that, as he created them on an Apple computer using the KoalaPad touch tablet with the Micro



Illustrator software. I am encouraging him to use Logo also to see how he likes it.

Just as Mr. Emberley's books can be a source of inspiration to those of us who build pictures using turtle graphics, they can also be wonderful tools for teaching procedural problem-solving – for teaching people how to solve larger problems by breaking them into bite-sized chunks. For this reason I encourage the use of his drawing books by teachers of computer programming. Not only are the children learning to solve problems with procedures, but they are also learning how to create charming illustrations at the same time.

I created the next figure myself to show that almost *anyone* can learn to make pictures in this manner.

For those of us who have been in the field a long time, the discovery of Mr. Emberley's excellent contributions is refreshing. Clearly, he is a Friend of the Turtle! ©

J.B. Shelton and Glenn M. Kleiman

Computers And Teaching Children To Read

Both authors of this report have long been interested in teaching children to read. Kleiman, previously a researcher at the National Center for the Study of Reading, attended the 1983 International Reading Association Convention. Shelton, a former reading tutor, has visited several schools where the IBM-sponsored program discussed in this column is being tested.

International Reading Association Convention

The International Reading Association (IRA) is the world's largest association of reading teachers and researchers. Its 1983 convention, held May 2-6 in Anaheim, California, reflected the size of the organization. The program booklet required 58 pages to list all the workshops, symposia, institutes, research report presentations, special interest group meetings, and other events. Thousands of teachers and researchers attended, and almost 250 companies exhibited their products.

Three years ago, at the 1980 convention, there was very little about computers. Only a handful of presentations focused on computers, and just a few companies exhibited computer-based products. Things have changed.

Three of the preconvention institutes focused upon computers, as did many conference presentations and symposia. A special interest group has been formed by people interested in using microcomputers to teach reading. The exhibit area included booths from Apple, Atari, Commodore, IBM, and Radio Shack. Perhaps most significant is that computer software was included in many of the exhibit booths, even from some of the major textbook publishing companies. Scott Foresman, Random House, Ginn, Houghton Mifflin, Scholastic, Milliken, Borg-Warner Educational Systems, Developmental Learning Materials, Walt Disney Educational Media, Educational Activities, Hartley, Follett Library Book Company, American Educational Software, Computer Curriculum Corporation, and other companies had computer materials on display.

In a display of IRA publications, there was a new book, *Computer Applications in Reading*, by George Mason, Jay Blanchard, and Danny Daniel. This book is a valuable resource for anyone interested in computers and reading. It describes college and university centers for computer-based reading programs, school applications in reading instruction, computer assessment of readability, sources of computer services and software, research on computers in reading, and background information about computers. Much of the book is taken up by annotated bibliographies, so it is a good starting point for finding out about computers and reading instruction. It is available from IRA, 800 Barksdale Road, Box 8139, Newark, DE 19714.

A new journal, *Computers, Reading and Language Arts*, was also being promoted. It contains articles, book reviews, software reviews, and news, all focusing on "the day-to-day use of computers in teaching basic skills in subjects like reading, writing, and spelling." For more information about this journal, contact Modern Learning Publishers, Inc., 6517 Liggert, Oakland, CA 94611.

At the IRA convention, there was clearly tremendous interest in using computers to teach reading. There were discussions of the potential of computers to help motivate children, to provide drill and practice in phonics and word recognition, to administer and score tests, and to improve comprehension skills. Many teachers, administrators, and researchers expressed optimism about the possible uses of computers.

However, there was far more said about

plans, potentials, and desires than about how computers are already being used. The presentations, for the most part, focused on the need for teacher training and the process of implementing computers in schools, and selecting and evaluating software – the beginning steps of using computers. Reading educators are just getting started with computers. It will be a few years before presentations at IRA can discuss what actually happens when computers are used in teaching children to read.

"Our kindergartners, four months into the school term, are writing sentences. They have to be enjoying, as well as learning from, the computers to show the dedication they do."

Writing To Read

One reading program already being used was shown at the IBM exhibit booth and at a separate display. The program is part of a comprehensive reading instruction package for kindergarten and first-grade children. Developed by Dr. John Henry Martin and called "Writing to Read," this approach to teaching reading is being tested by schools in eight states and in Washington, D.C., with 10,000 children participating. Wake County, North Carolina, with 2,900 kindergarten children in 34 schools, is the largest single participant in the national test. The program runs on IBM Personal Computers, and testing is funded by IBM. Dr. Martin's company, JHM Corporation, is supervising the program and, at the end of a two-year test period, Educational Testing Service will conduct an evaluation.

The introduction of Writing to Read into the schools was threefold: teachers and principals were given a two-day training seminar and a system management manual; parents viewed an orientation film and experienced a hands-on session; and the children were prepared in the classroom for the computer lab procedures.

The Program In Action

At Briarcliff Elementary School in Cary, North Carolina, the reading lab contains four IBM Personal Computers, one printer, eight electric typewriters, and ten tape recorders. Each child spends one hour per day in the reading lab. The computer segment lasts 15 minutes. The remaining time is spent in work center activities – children review their words, use typewriters to write, and listen to, and read, stories.

At the computer stations, the children work in pairs, using IBM Personal Computers with synthesized voice output and color graphics. Wearing headphones, the children listen and respond to the synthesized voice and graphics display.

The computerized lessons are designed to teach children letter-sound correspondences. The lessons use a "phonemic alphabet" which represents each of the 42 sounds of English by one symbol. That is, this approach uses a special teaching alphabet that makes the letter-sound correspondences consistent. For example, "cat" would be written as "Kat," and "through" as "throo." The children are introduced to all 42 phonemes by working with 30 words. The selected words include all the sounds and are represented by pictures – dog, cat, bed, rabbit, fish, and so on. The children learn the 30 words in ten cycles, each with three words.

The symbols for the sounds are displayed around the perimeter of the computer screen (these symbols are letters, letter pairs such as "th," and letters with markings, such as for long and short vowels). A color picture appears on the computer screen, with the word spelled both phonetically (rabit) and in standard English (rabbit). The computer (using a digitized female voice) intones: "Say rabbit." There is a pause for the children to say the word aloud. They are then instructed in the phonemic spelling, sound by sound. The voice requests an "r" and waits for the correct letter on the keyboard to be pressed. Incorrect key presses are simply ignored. When "r" is pressed, the phoneme "r" moves from its place at the perimeter of the screen to midscreen. The voice repeats, "Say rabbit," and the children again recite the word. This sequence continues until the whole word is spelled out. The procedure is very repetitive. As each new word is selected, the computer says the word, asks the children to repeat it, spells the word aloud, and asks the children to take turns spelling it several times. Since phonemic spellings are emphasized, the children must spell "rabbit" with only one "b" – the computer ignores any pressing of "b" after the first one.

After the learning phase, the computer work goes on to a "mastery test." Students are asked to spell the words they have just learned. If they make two errors, the computer takes them back to the learning phase. Following success on the mastery test, there is a "make words" phase, in which all the phonemes the children have learned so far are reviewed and combined into new words. There are also some game activities, such as one where the children must repeatedly type "mouse" quickly enough to keep a mouse pictured on the screen moving away from a cat.

The work-center activities complement and supplement the computer lessons. At the work-journal station, the children use workbooks which provide practice with the three basic words just learned, present additional words that have the same phonemes, and leave room for children to write new words. At the make-words center, the children write and illustrate words. At the classics listening center, they listen to stories on tape and read along with the actual books. This familiarizes them with standard spelling. At the typewriter station, children use IBM Selectric typewriters to write the words they have learned, combine them into sentences and, when they are ready, begin to write stories.

Program Effectiveness

Writing to Read has many similarities to other approaches to teaching reading. For example, it shares the use of a phonetic alphabet with approaches using the International Teaching Alphabet. The emphasis on letter-sound correspondences and on having children say the words and sounds repeatedly reminds us of such structured phonics approaches as the DISTAR method. However, the overall blend of computer and work-station activities is Dr. Martin's own synthesis, reflecting his experience as a teacher and school administrator and his philosophy of education.

Writing to Read has been enthusiastically received by the teachers and children at Briarcliff School. The principal, Mary Jane McReynolds, noted: "Our kindergartners, four months into the school term, are writing sentences. They have to be enjoying, as well as learning from, the computers to show the dedication they do."

Several minor problems, such as the mastery tests being too rapid and the headphones not working well, have been remedied. Kathleen Burt, head of Briarcliff School's Kindergarten program, reports: "We're more than pleased; we have high hopes. You can't separate reading and writing, and this is a logical approach toward teaching them together. It combines many of the best teaching methods we use in the classroom with the technological advantages of the computer."

The Writing to Read approach is an eclectic one, combining phonics, writing, and the reading of classical children's stories. Some educators would object to certain aspects, such as using nonstandard phonetic spellings and having children begin learning letter-sound correspondences before they learn "whole-word" or "sight" recognition of common words. These are classic issues in the teaching of reading, issues that have been debated for many years. We will not enter into this debate here, but we do want to point out that the computer does not settle any of these issues. Computers can be used in any approach to teaching reading. But computers do not tell us how or what to teach, and they do not automatically solve children's reading problems.

Dr. Martin and IBM are to be commended for their efforts to integrate computers into a complete approach to teaching reading, and for subjecting their approach to large-scale testing and evaluation. However, they are using the computer for only one purpose – to teach letter-sound correspondences to beginning readers. We hope to see other methods of teaching reading begin to incorporate computers, and to see computers used to help children develop more advanced reading skills. ©

It's still around if you know where to look.

We are Kangaroo™ Inc., dedicated to just plain fun for kids 3 to 8 years old.

Jeepers Creatures™ - 30 basic animals with interchangeable heads, torsos, and legs or tails. Create an owligator or an octocatfish or one of over 26,000 funny colorful combinations in this goofy collection of mixed up animals.

My House-My Home™ - Let's move into the house on Somewhere Street. It's ready for furnishing and family. Oops, you put the kitchen sink in the bedroom and Aunt Nellie's bed in the kitchen. That's silly.

Jeepers Creatures and My House-My Home have no winners or losers, no right or wrong answers, no high scores to beat, just hundreds of hours of creative play.

Give your child's imagination a boost with Kangaroo games designed for Apple and Atari computers.

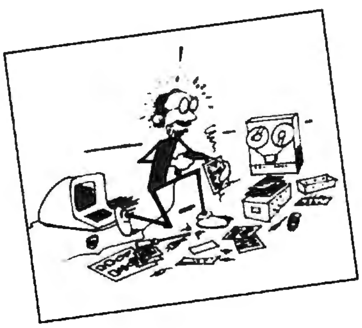
Jeepers Creatures and My House-My Home are each available for $34.95 at your software dealer or from Kangaroo, Inc., 332 South Michigan Ave. Suite 700, Chicago, IL 60604, (312) 987-9050. Visa, MasterCard and personal checks accepted.

KANGAROO™ *For a pocket full of fun.*

What ever happened to just plain fun?

Islands Of Learning

Fred D'Ignazio, Associate Editor



I am afraid that homes and schools are swiftly becoming islands of computer learning. I am afraid that unless something is done to coordinate this learning, the tremendous educational potential of personal computers might never be realized.

I've been speaking on this subject at several educational conferences, including the National Educational Computing Conference in Baltimore, the Florida Instructional Computing Conference in Tampa, and the Hollins College conference on "Computers in the Schools."

In Tampa, my speech was entitled "Linking Computers in the Classroom and the Home." After the speech, one of the people in the audience came up and introduced himself as Kenneth Komoski, Educational Director of the Educational Products Information Exchange (EPIE).

Ken and I sat down and chatted. We discovered that each of us independently had decided that the critical area in educational computing was neither the classroom nor the home. Instead it was the *connection* or the *gap* between the two.

To show how similar our thinking was, here's a quote from one of Ken's recent articles:

> It's important for parents to work cooperatively with educators and community leaders to make decisions about computers and educational software.... At stake is not only the quality of computer instruction in the schools, but also the full educational potential of microcomputers for your child, your family and your community. If parents and schools do not work together, much of this potential may be lost.
>
> Kenneth Komoski, *Network* (3/83).

Ken told me about the work he had been doing. He mentioned that his organization, EPIE, had recently joined forces with Consumers Union to evaluate educational computing products. Evaluations are published in a monthly *MICROgram*. (Subscriptions to *MICROgram* are available from EPIE-Consumers Union, P.O. Box 839, Watermill, NY 11976.)

Millions Of Programs

Hundreds of new educational software packages are appearing each month. Families and schools are buying these packages by the *millions*. In 1982, 1.4 million educational software packages were sold. Experts predict that 4 million packages will be sold this year.

Families are buying the software and schools are buying the software, but there is little or no coordination between the two. This is the source of Ken's and my concern.

The Miracle Machine

This lack of coordination creates a real problem. The biggest impact of computers will probably be on kids using computers at home. Yet families have the least experience in setting up a formal curriculum of instruction.

Many parents buying computers are hoping that the educational programs that their kids use will teach the kids without any parental guidance. Or that the programs at least won't do any harm.

But are these parents placing too much faith in the computer?

Ever since they were first invented, computers have been seen as a miracle solution to all sorts of problems. But people have learned the hard way that computers do nothing on their own. If computers are given the right instructions, they can help. But people still do most of the work.

The same is true for computer learning. The computer cannot teach a child on its own. Parents

and teachers still need to do most of the work. They need to create an environment and a curriculum fitted to the needs and age level of each child. Then the computer can be a valuable assistant.

The Impact Of Computer Learning

We have another problem. Computer learning at home is likely to be unguided and uncontrolled. Yet unless computer learning at school and at home is better coordinated, the impact of computer learning will be much greater at home.

Look at the ratio of kids to computers at home and in the schools. At home the ratio is great; at school it is miserable. This is not a nice thing to admit, but it's true. One reader recently wrote me that at her school the ratio was 400 kids to only one computer. "But," she admitted, "I suppose that's better than no computers at all."

At home the ratio is much smaller, of course. Either it's nonexistent, because the family doesn't want or can't afford a computer, or it's something like two or three kids per computer.

And when you think about how much exposure each child will get to educational software at home, the situation looks even worse.

Software prices are beginning to drop rapidly. Quality educational software will soon be available at affordable prices (from $5 to $30 a package). Yet at school the software must run on the same computer that is in demand by (perhaps) hundreds of schoolchildren. This makes it unlikely that a child using a computer at school will be able to sample and fully experience the full range of software that his or her school can afford.

At home, on the other hand, a child's family might buy two or three dozen packages, and the child will have the opportunity to fully experience *all of them*.

The lesson here is that the *impact* of computers will be much greater on kids learning at home.

A Riot Of Computer Learning

In recent columns, I have discussed educational computing in the classroom and in the home. In one column, I wrote about the "riot" of learning that will soon be taking place in homes that have computers. I called it a *riot* because computer learning at home will be spontaneous, uncontrolled, and, perhaps, destructive.

Parents have always depended on teachers to provide formal instruction for their children, especially in basics such as arithmetic, writing, and reading. Now there are dozens of math drill programs, typing tutors, spelling instructors, and reading "games."

Parents are buying these programs and turning the computer into a vehicle to teach these skills at home. But in most cases the computer "tutor" will not have a parent looking over its shoulder. The tutor will be teaching the child, but its lessons won't be integrated into a balanced curriculum especially suited for the child.

And this is just the beginning. The computer's effect on home learning will soon increase dramatically. At the end of the next twelve months there will be software on the market to teach every conceivable subject or skill to kids of almost every age. There will be geography programs, programs to teach chemistry, astronomy, art, dancing, songwriting, juggling, current events, and world religions.

Parents will buy this software, but kids will be expected to use it on their own.

The Cereal-Box Strategy

Schools and families will buy millions of educational programs. Schools will buy programs only after they have evaluated the programs' pedagogical soundness and relevance to teachers' existing curricula.

How will parents choose among the hundreds of educational programs on the market? They will buy programs the way they buy everything else. They will choose a particular package because:

- Their child responds to an ad on the back of a box of cereal by crying, "I want that program, Mommy!" And how can the mommy refuse? After all, the program is educational. And it's a lot better than getting the kid the laser blaster that was advertised on the box of cereal the family finished last week.
- Software will soon be on sale at special software stores and at almost every sales outlet imaginable, including local convenience marts, department stores, bookstores, drugstores, beauty salons, hardware stores, and supermarkets. It will be sold alongside how-to manuals, mass-market books, and other inexpensive consumer items. It will become as cheap and as convenient to buy as a detective novel, a romance novel, or a carton of milk.
- Software will soon be packaged as glossily as cosmetics and rock records. Ads on TV, on the radio, and in stores will make buying software as appealing as buying a new kind of lipstick, deodorant, sports car, or diet soda. It will look like an offer that can't be refused.
- Buying software will help people keep up with their neighbors who are also buying software for their kids. After all, a responsible parent can't sit still and watch her kids slip behind. And even the President of the United States admits they're not getting the education they need at school. The only way to get the kids educated is to give them an early start on learning at home. And the only way that can be managed is with a computer – and lots and lots of software.

YOUR PROBLEM IS SOLVED!

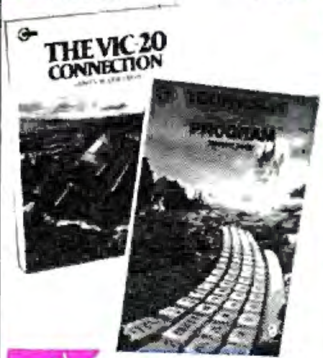
Now you can rely on *PACE* for ONE STOP shopping for all your Micro Computer needs. We have picked out the BEST 2000 Programs, Books and Accessories—covering all the major brands and put them into one friendly store. And, this is backed up by THOUSANDS of additional items we stock in our central warehouse, ready for overnight shipping to our stores. Magazines? You bet! We carry almost 60 different Micro Magazines on our racks! Plan to visit us soon. Can't visit? Then you can order from our gigantic product list—just write for your *FREE*, personal copy *TODAY!*

pace

For: VIC-20

NEW **THE VIC-20™ CONNECTION** By James W. Coffron. Learn the simple techniques for putting your VIC to work controlling external devices: lights, alarms, etc. Softcover. 7" X 9" format. 260 Pages.
4795-000128 $ 7.95

NEW **YOUR FIRST VIC-20™ PROGRAM** By Rodney Zaks. Write your first BASIC program in one hour! First time user guide, ages 8 to 88! Softcover. 7" X 11". 150 Pages.
4795-000129 $ 9.95



NEW **COMPUTERS FOR KIDS: VIC-20™ Edition** By Sally Greenwood Larsen. Tailored for even the youngest child, it has full instructions on writing programs for graphics, calculations and more! Special section of tips for teachers/parents. Softcover. 11" X 8½" format. 70-plus pages long.
4198-000042 $ 5.95

NEW **HOW TO USE THE COMMODORE® VIC-20™** By Jerry Willis and Deborrah Willis. The beginner's text. Includes a special Problem-Solving section for any difficulties you might encounter. Softcover. 5½" X 8½". 124 Pages.
4250-000134 $ 3.95

VIC-20™ PROGRAMMER'S REFERENCE GUIDE Commodore. Complete BASIC vocabulary guide, machine language guide, tips, tricks, short-cuts and general operations. Belongs beside your VIC! Softcover. Spiralbound. 290 Pages.
4760-021948 $16.95

COMPUTE!'s FIRST BOOK OF VIC™ A compilation of articles from Compute! Magazine.
4105-000007 Soft, 212 Pgs. $12.95

For: Atari

NEW **THE EASY GUIDE TO YOUR ATARI® 400/800™** By Joseph Kascmer. Jargon-free operating manual. Shows the easy way to use your Atari. . .word processing, math, budgeting, filing and more! The easiest way to start using your computer! Softcover. 5½" X 8¼" format. 160 Pages.
4795-000125 $ 7.95

NEW **BASIC EXERCISES FOR THE ATARI®** By J. P. Lamoitier. Learn true style and subtleties of Atari BASIC through actual practice. Graduated exercises in math, business, operations research, games and statistics. Each exercise contains a statement and analysis of the problem, a solution with flowchart, and a program with detailed explanation. Softcover, 7" X 11", 251 Pages.
4795-000101 $12.95

NEW **YOUR FIRST ATARI® PROGRAM** By Rodney Zaks. Write your first BASIC program in 1-hour! For first time users, ages 8 to 88. See how simple learning to write a program can be! Softcover. 7" X 11". 150 Pgs.
4795-000130 $ 9.95

For: Timex/Sinclair

NEW **CONTROL THINGS WITH YOUR TIMEX®/SINCLAIR®** By Robert L. Swarts. Looking for low cost ways to expand capabilities? With a few inexpensive parts your TS1000 can be taught to control lights, alarms, appliances and more! Softcover. 5½" X 8½". 80 Pages. Average project costs under $10!
4250-000127 $ 6.95

PRICES & AVAILABILITY SUBJECT TO CHANGE

For: Commodore 64

NEW **HOW TO USE THE COMMODORE® 64™** By Jerry Willis & Deborrah Willis. All you'll need to know to successfully operate your C-64. Good beginning text and includes a Problem-Solving section for any difficulties you might encounter. Softcover. 5½" X 8½" format. 124 Pages.
4250-000133 $ 3.95

NEW **THE COMMODORE® 64™ BASIC HANDBOOK** By Douglas Hergert. Need to check or clarify a BASIC keyword or function? Do it quickly with this dictionary-style handbook. Clear descriptions and examples of the 64's vocabulary. Fully explained in easy to understand English. Softcover. 5½" X 8¼" format. 170 Pages.
4795-000116 $ 9.95

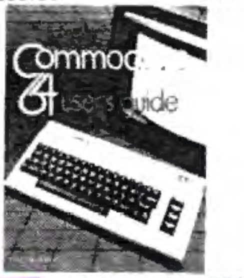
NEW **THE EASY GUIDE TO YOUR COMMODORE® 64™** By Joseph Kascmer. Jargon-free companion tells you how-to-do-it! Word processing, math calculations, budgeting, filing and more! Softcover. 5½" X 8¼". 160 Pgs.
4795-000126 $ 7.95

NEW **32 BASIC PROGRAMS FOR THE COMMODORE 64™** By Tom Rugg, Phil Feldman & Gene Moore. Written by experienced programmers, each of the 32 chapters is fully documented and bug-free. Can also be easily adapted using the authors' suggestions. And, if you've a working knowledge of BASIC, you'll be better able to devise and implement your own changes. Among some of the programs: Biorhythm, Checkbook, Kaleido, Sparkle, Math, Vocab, Curve, Decide, Birthday and more! Softcover. 5½" X 8½" format. 350 Pages.
4250-000112 $19.95
4250-000180 Book & Disk $29.95

NEW **USING THE COMMODORE® 64™ IN THE HOME** By William Behrendt & hank Liubrach. Beginner-expert, kids-adults, can now have 20 programs that include home management and games that would cost hundreds of dollars if bought separately. Helps teach basic programming techniques. Sample runs included. Softcover.
4690-940072 $10.95

NEW **COMMODORE® 64™ COMPUTING** By Ian Sinclair. Both an intro how-to and a reference manual. Discusses set up, operations and covers special features such as Sprite graphics, color commands and programming for sound. Illustrated with BASIC examples. Softcover.
4690-152306 $12.95

NEW **SPRITE GRAPHICS FOR THE COMMODORE® 64™** By Sally Larsen. For beginners, an intro to Sprite graphics on the Commodore 63—the documentation that is lacking when they buy their 64. Covers how to use Sprites for the practical tools they are and includes 25 ready-to-run programs. Softcover.
4690-838136 $15.95

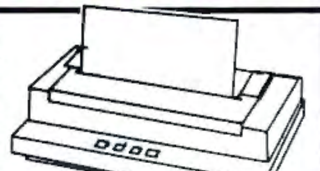


NEW **COMMODORE® 64™ USER'S GUIDE** By Tim Onosko. Brand new effort that is the only source covering all 3 versions of the C-64, with info on both classroom and portable models. Softcover.
4690-000380 $14.95

THE ELEMENTARY 64™ Datamost. Probably THE BEST book available to date on this superb computer. Easy-to-understand guide to operations, programming and more! Softcover. 224 Pages.
4560-000034 $14.95

PACE MICRO SOFTWARE CENTER
345 East Irving Park Road, Wood Dale, Il 60191
PHONE: (312) 595-3860

OLIVETTI® JET INK PRINTER



The Olivetti PR2300 Jet Ink Printer uses a single jet printing head to direct carbon particles onto regular paper in a 7 X 7 dot matrix. The desk top unit prints a full 96 ASCII characters with 7 other foreign language and symbol sets or software that are selectable. It prints at up to 50 bi-directional lines per minute for a full 80 columns. Unit offers normal, compressed, double height, double width and bold printing with either single or double-line underlining. Standard Centronics or RS232C Interface required/available. Call or write us for information.

5655-002300	Parallel Printer (Reg. 560.00) *SALE*	$529.99
5655-012230	Replacement Cartridge	$10.95

SPECIAL DISK SALE!

5320-001001	Elephant Single-Sided, Single Density, 5¼", Box/10	$17.99
5320-001002	Elephant Single-Sided, Double Density, 5¼", Box/10	$21.99
5320-001003	Elephant Double-Sided, Double Density, 5¼", Box/10	$25.99
4540-461910	3-M Single-Sided, Double Density, 5¼", Box/10	$22.99
4540-462210	3-M Double-Sided, Double Density, 5¼", Box/10	$31.99

SAVE UP TO 20% ON THESE POPULAR GAMES

GAMES/VIC-20

RECREATION 6 PAK Commodore. Casino-blackjack; Car Chase; Blue Meanies from Outer Space; Slither-Super Slither; Biorhythms and Space Math. 6-Cassettes.
4100-000106 (Reg. 59.95) $47.98

ASTROBLITZ Creative Software. Colorful and fast arcade "Defender"-style game. Cart.
4200-000129 (Reg. 39.95) $31.95

SIDEWINDER Tronix. Unlimited firepower is your only advantage as you protect the Earth's last radar tower! Joystick/Datasette.
5842-102004 Cass (Reg 29.95) $23.95

FAST EDDIE Sirius. New, fast action version of "Donkey Kong™" for the VIC™. Cartridge.
4770-022003 Cart (Reg 39.95) $31.95

TURMOIL FSirius. Fast-paced, arcade-style game for the VIC™. Joystick optional. Cart.
4770-022005 Cart (Reg 39.95) $31.95

DEADLY DUCK Sirius. Dodge the bricks as you try to shoot down the flying crabs! Watch out for the fireflies! Joystick optional. Cart.
4770-022004 Cart (Reg 34.95) $27.99

SNAKE BITE Sirius. New challenge for maze lovers—slither your way along, chomping apples—but watch out, cause you grow bigger and you can't touch the walls! Cart.
4770-022031 Cart (Reg 29.95) $23.98

RIVER RESCUE Thorne-EMI. Fast action, right to left scrolling game. Joystick-Cart.
4325-022001 (Reg. 39.95) $31.95

CHOPLIFTER Creative Software. Quick reaction helicopter rescue game. Cartridge.
4200-000119 (Reg. 39.95) $31.95

GAMES/COMMODORE 64

DEADLINE Commodore/Infocom. Keyboard/Text Murder Mystery Adventure! Disk.
4100-064628 Disk (Reg 49.95) $39.99

FROGGER Sierra-On-Line. Help Frogger survive the perils of highway and the rushing river! 64 version of the popular arcade game. Disk.
5763-254251 Disk (Reg 34.95) $27.99

REPTON Sirius. High-speed, graphically realistic space battle! Disk.
4770-030003 Disk (Reg 39.95) $31.95

FROGMAN O.E.M. Maze-type game for 1 or 2 players. Gather pearls before the sharks attack! Kill sharks and win more! Cartridge.
5648-000101 Cart. (Reg. 32.95) $26.39

SPACE RESCUE HESWARE. Benji helps exercise your knowledge of astro physics and the visible solar system. Educational fun! Disk. 64K required.
4428-000110 Disk (Reg 44.95) $35.95

CHOPLIFTER Broderbund. Quick, rescue the 64 captured citizens. Fight off enemy tanks, planes and satellites as you do!
5098-000231 Disk (Reg 44.95) $35.95

GAMES/ATARI 400/800

CASTLE WOLFENSTEIN Muse. You must escape from the Nazis in this popular adventure. Disk. Requires 32K.
5574-392422 (Reg. 29.95) $23.99

ZAXXON™ Datasoft. Atari version of the fast paced arcade action game. 16K Req.
5206-323330 Cass. (Reg 39.95) $31.99
5206-323030 Disk (Reg 39.95) $31.99

By the time that you read this ad, our first **PACE** store in the western suburbs of Chicago, at **345 East Irving Park Road, Wood Dale**, will be open, soon to be followed by many others. Until a **PACE** store opens in your city, you may order direct from the **PACE** central warehouse with confidence, where we currently stock over 6,000 microcomputer products.

COUPON Please Send Me:

QTY	NUMBER	DESCRIPTION	COST	TOTAL
			SHIPPING	$ 2.50
			TOTAL	

Illinois Residents Please Add 6% Sales Tax.
Foreign Orders, (All outside Continental US), Add 10% Shipping (Minimum $4.00)
Catalogs Shipped Postage Paid

PAYMENT ENCLOSED: ☐CASH ☐CHECK ☐MONEY ORDER
PLEASE CHARGE TO MY: ☐MASTERCARD ☐VISA (Min. Chg. $25)
CARD NUMBER ____________
EXPIRES ____________ INTRBNK# ____________
SHIP TO ____________
STREET ADDRESS ____________ APT ____
CITY ____________ STATE ________ ZIP ________

pace! micro software centers

Department: C-P
Lock Box 328
Bensenville, IL 60106

Help From Software Publishers

Can't parents depend on educational publishers and computer companies to advise them in making their purchases?

Not likely. Educational publishers and computer companies will likely have little or no interest in coordinating the software that children use at home and the software they use in school. Nor will they be concerned with how their software fits into a child's development and into his or her learning experiences.

That's why the average home's computer learning environment will be riotous. Most parents won't have the training or the time to properly select educational software. Nor will they have the skill to build that software into a comprehensive "Home Learning Program" for their children.

Schools have this training and skill, but they are not currently part of the home software selection and evaluation process.

Islands Of Learning

I see two islands of learning that will soon be created by computers. One island will be the school, the other the home.

In the school, computer learning will be structured, well organized, well thought out, properly sequenced, comprehensive, and pedagogically sound. Unfortunately, it will also be minimal, since each kid will have an absurdly small amount of time using any particular package. The effect of computer learning at school will be diluted by the huge number of kids trying to share a few computers.

In the home, computer learning will have a great impact, because kids will have the opportunity to spend a great amount of time on the computer and richly experience each program. Also, there is great potential for home learning to be exceptionally creative, open-ended, informal, self-motivated, and self-guided.

Unfortunately, this potential probably won't be realized. Children's home computer-learning will be blunted by a lack of any plan; by lack of coordination with a child's physical, emotional, social, and intellectual development; by a lack of non-computer learning activities that support the computer instruction; by improper sequencing of materials and concepts; and by the gaps in learning caused by a family's arbitrary purchase of software packages.

Looking Ahead

If present trends continue, and schools and homes become islands of computer learning, what will be the result? What will be the effect of computer learning on our children? What kind of kids will we be turning loose in the public schools? How frustrated will computer-literate kids become when they don't find the same computer resources at school that are available at home? Will non-mainstream kids be left behind? And, if so, how will they ever catch up? (They won't catch up after they leave school. In the future, the average workplace – factory or office – will be even more computerized than the average home.)

If present trends continue, there is the possibility that the effect of computer learning at school will be minimal. It is likely that the effect of computer learning at home will be profound, yet it may also be profoundly destructive. It is also possible that the enormous potential of computer learning at home and at school might never be realized.

Can we permit this? Are there any alternatives?

Building Bridges

There are alternatives. We can link computing at home and computing at school. We can build bridges between these islands of learning. The bridges will permit a two-way flow of expertise and resources. The sharing of computers, software, and learning strategies can make it possible to realize the computer's great potential as a learning tool for children and their entire family.

Ken Komoski's plan for building these bridges is similar to my own. Ken (in his *MICROgram* and elsewhere) stresses "concrete programs and policies." I emphasize communication. Both elements are important. What we need, at minimum, is:

1. *Community-wide Training*. Teachers, parents, and children need to attend common training sessions to learn how to use computers and how to evaluate, select, and get the most out of educational software.

2. *Community-wide Access*. Parent-teacher organizations should set up *computer cooperatives* to evaluate computer software and hardware sold by local vendors. Discounts (Ken's idea) should be offered to parents who buy the computers and software recommended by the cooperative. Low-income families in the community should get special deals: computers and software should be available for families to lease, lease-purchase, buy at a discount, or check out from a co-op library.

3. *Communication*. The co-op should hold regular public meetings and publish a monthly newsletter to make all the parents in the community aware of the diverse aspects of educational computing, including:

- Past, present, and planned applications of computers in the school (by teacher, subject, and grade).
- Educational applications of computers in local homes (by manufacturer, name of software, subject, age of children).
- Recommendation of the best new educa-

tional software and hardware; results of evaluations and tests performed at local schools and by local parents.

• Opportunities for Action: Community Computer Faires, Computer Flea Markets and Yard Sales, Parents' Brag Nights, involvement in software and hardware evaluation committees, donation of used hardware and software to the co-op library, etc.

4. *Sharing*. The co-op needs to set up libraries of hardware and software.

Families will be buying much more hardware and software than schools. *And they will be going through it much faster*. The co-op should get families to donate, loan, or sell (at a discount) the old software and hardware that their kids have outgrown. This strategy will create an enormous reservoir of materials that can be used in the classroom and by families who otherwise would have no access to them.

In addition, this strategy will create a *Supply Depot* of computer parts, including disk drives, cables, monitors, computer circuit cards, wires, plugs, and connectors. Since computer models change so rapidly, this Depot could become an invaluable resource for the schools and the community at large. Having backup computer supplies will make it possible to keep computers up and running for longer periods of time. It will make it possible to put more students on the computers and allow the students to use a wider variety of software packages.

What Do You Think?

I am convinced that the most important area in educational computing will be what is done to link educational computing in the classroom and in the home.

What do *you* think?

I would like to hear your ideas about this subject. Is it important? Will it become more important in the future? What are some of the dangers if nothing is done? What are some of the strategies we can adopt to link computing in the classroom and the home? What issues haven't I covered?

Write to me directly:

Fred D'Ignazio
2117 Carter Road, SW
Roanoke, VA 24015

Merging BASIC Programs From Commodore Disk

Jim Butterfield, Associate Editor

It's often very useful to be able to merge two programs. Here, Jim Butterfield steps through a disk merger program that is helpful also in understanding how programs are formatted and stored on disk. For all Commodore machines.

Programs can be merged using a curious technique with cassette tape – see "BASIC Program Merges: PET And VIC," COMPUTE!, June 1982, page 158. We can also do a disk merge in a much more straightforward manner.

Disk Advantages

With disk, we can have several files going at the same time. Thus, we can read two separate programs and write out the new combination program as a single activity.

With a disk unit, we can read in programs as if they were sequential files. This means that we can manipulate a program as if it were data; for that matter, we can write a data file which may be later used as a program. This opens the door to sophisticated activities, such as programs that analyze other programs, or programs that write programs.

The Merge Program

To show how it's done, and to provide a useful capability, we'll walk through some simple programming which will merge two programs. This MERGER program will work on all Commodore disk-based systems: VIC-20, Commodore 64, and PET/CBM. It's written in BASIC to enable you to see how everything works.

As we walk through the program lines, we'll point out special considerations that we need to take into account when reading and writing program files (as opposed to data files).

First, we identify the program:

```
100 PRINT "PROGRAM MERGER"
```

Each of the two input files will have individual working values. For example, C$ is the line of code we are working on; N is the line number. Let's make room:

```
110 DIM A$(2),B$(2),C$(2),N(2)
```

Let's open the error channel so we can spot problems:

```
120 OPEN 15,8,15
```

Now we'll ask for the name of the first program to be merged. We'll OPEN the file as a program (note the ,P for Program), then we'll check for problems and quit if we see trouble:

```
130 INPUT "PROGRAM 1";X$
140 OPEN 1,8,2,X$+",P,R"
150 INPUT#15,E,E$,E1,E2
160 IF E THEN PRINT E$: CLOSE 15: END
```

One extra thing to do here. The first two bytes of a program file contain the program's start address. We will assume that we won't need this information, since VIC and 64 programs will relocate automatically, and PET/CBM programs must start at address 1025. Finally, we'll grab the first line of the program by calling subroutine 420 (more on this when we reach it).

```
170 GET#1,A$,A$: J=1: GOSUB 420
```

We're ready to repeat the sequence for program two. In this case, we'll allow the user to reply "N" in order to specify that no program two exists. In such a case, we'll merge program one with nothing, and just copy program one. But there's a bonus which we'll discuss later.

```
180 INPUT "PROGRAM 2 (OR 'N')";X$
190 IF X$="N" THEN N(2)=1E9: GOTO240
```

```
200 OPEN 2,8,3,X$+",P,R"
210 INPUT#15,E,E$,E1,E2
220 IF E THEN PRINT E$: CLOSE 15: END
230 GET#2,A$,A$: J=2: GOSUB 420
```

Now we'll ask for the name of the new, merged file that we are about to create. We OPEN with "...,P,W"; in other words, program file in write mode.

```
240 INPUT "NEW FILE";X$
250 OPEN 8,8,8,"0:"+X$+",P,W"
260 INPUT#15,E,E$,E1,E2
270 IF E THEN PRINT E$: CLOSE 15: END
```

You may recall that we threw away the start addresses from our program input files. Before we start to write our output file, we must insert a start address so that the format is correct. VIC and 64 won't care, but PET/CBM needs an address of 1025, so that's what we'll supply in two bytes:

```
280 PRINT#8,CHR$(1);CHR$(4);
290 REM COMPARE LINES HERE
```

The above REMarks line says it all. We have a line from each program. The line numbers are held in N(1) and N(2). We wish to take the lowest line number; if they are the same, we'll take from file two:

```
300 X=2
310 IF N(1)<N(2) THEN X=1
```

When we reach the end of a file, we make the line number impossibly large. If both line numbers are way up there, we are finished and can wrap up our output file:

```
320 N=N(X):IF N>1E8 GOTO 380
```

Now we're ready to print the BASIC line from file X (X is 1 or 2). We'll need to know the format in more detail. The first two bytes of each line are called a "forward link." We don't need to work them out precisely: provided they are not zero bytes, they will be rebuilt when the program is loaded into the computer. So we can print a couple of CHR$(1)'s.

The next two bytes are the line number, in binary. We happen to have this information in bytes A$(X) and B$(X) – we obtained this in the subroutine at 420 – and we can just print them.

The line of BASIC follows. That's in C$(X); and the line must be followed by a binary zero, or CHR$(0), to terminate it properly. So the whole line goes:

```
330 PRINT#8,CHR$(1);CHR$(1);A$(X);B$(X);C
    $(X);CHR$(0);
```

Now we want to replace the line we've just used. We could read from file X with a call to 420 – but wait a moment. If both line numbers are the same, we want to replace them both. That's what a merge is about: one line overwriting another when the numbers match.

```
340 FOR J=1 TO 2
350 IF N(J)=N THEN GOSUB 420
360 NEXT J
```

We've handled the line in question. Let's go back and do some more.

```
370 GOTO 300
```

If we've reached the end – no more input lines – we terminate the output file with two zero bytes (a "null" forward link) and close down.

```
380 PRINT#8,CHR$(0);CHR$(0);
390 CLOSE 1: CLOSE 2: CLOSE 8: CLOSE 15
400 END
410 REM READ LINE OF BASIC
```

Here comes our subroutine to read from file number J (J may be 1 or 2). First, we grab the first two bytes (the "forward link"). We don't need these, except for one thing: if they are both zero bytes, we are at the end of the BASIC program.

Zero bytes arrive in an odd way. You'd think that the GET statement would receive the equivalent of CHR$(0), a single character containing zero bits. Nope. Due to an eccentricity of BASIC, they arrive as a "null string"; no bytes at all. So that's what we must test for:

```
420 GET#J,A$,B$
430 IF A$="" AND B$="" GOTO 540
```

Now we go for the line number, which we read into A$(J) and B$(J). If they happen to be null strings, we correct them to CHR$(0). Once we've done that, we can calculate the value of the line number and put it into N(J):

```
440 GET#J,A$(J),B$(J)
450 IF A$(J)="" THEN A$(J)=CHR$(0)
460 IF B$(J)="" THEN B$(J)=CHR$(0)
470 N(J)= ASC(A$(J))+ ASC(B$(J))*256
```

We've arrived at the BASIC line text itself. Let's gather it into a string called C$(J). As we collect the bytes, we must watch for the zero byte (or null string, to us) that flags end-of-line:

```
480 C$(J)=""
490 GET#J,A$
500 IF A$="" GOTO 550
510 C$(J)=C$(J)+A$
520 IF ST=0 AND LEN(C$(J))<254 GOTO 490
```

The above line checks for anomalies. If ST is not zero, we've reached the end of file, or we're having trouble with the disk interface. Either way, we want to warn the user or quit. And if C$(J) is getting too long, we must be into something that isn't really a BASIC program. In either case, we'll drop into a warning statement:

```
530 PRINT "PROBLEM FILE ";J
```

If we see the "forward link" of two zeros that flags end-of-BASIC, we set the line number to a ridiculously high value:

```
540 N(J)=1E9
```

And in any case, we return to the calling point:

```
550 RETURN
```

A Caution

When you type in this program, be sure that lines 330 and 380 end with a semicolon. If you miss this, you won't get a program; you'll get a mess.

Beginning programmers may not have seen statements such as GET#J,..., where a variable selects which file will be used. A little thought will reveal how it works and will possibly open up new trains of thought on the effective use of BASIC.

Free Bonus

Programs produced by MERGER will load into any Commodore machine. As mentioned before, VIC-20 and Commodore 64 (and the new B series) will automatically relocate programs to the proper address. For the PET and CBM, we have supplied the start address needed by these somewhat less flexible loaders.

This means that a simple run of MERGER with one input program file (replying "N" to the second file name) will convert a program into PET-loadable form.

This is not a wholesale conversion program, of course. A program may be loadable to another machine, but still won't run because the POKEs and PEEKs are incompatible.

Using MERGER

You might like to keep your favorite subroutines as small programs on disk, and merge them into other programs as needed. Be sure to keep your line numbers within compatible ranges so that the new program lines don't overwrite needed parts of a program.

Many calculation programs run using DATA statements. If these statements are saved as a program file, they can be merged into the calculation programs as needed, saving a great deal of retyping.

Perhaps most important of all: a careful reading of this program will reveal a good deal about how programs are formatted and stored on disk.

 ©

Jumpman

James Trunzo

Creating a good first impression isn't the easiest thing in the world, especially if the audience is made up of experts. An even more difficult task is to sustain or improve upon that good impression over a period of time. *Jumpman* succeeds in both cases.

In the face of cynical comments like "It's probably just like the rest of the climbing-motif games," *Jumpman* easily conquers that skepticism and establishes itself as a software classic. No true arcade-game fan who takes joystick in hand and begins to manipulate *Jumpman* around the screen will be able to walk away without adding this product to his Atari, Commodore 64, or Apple collection.

It's tempting to compare *Jumpman* to the much more widely ballyhooed *Miner 2049er*. The comparison would be unfair. *Jumpman* is much, much more. The basic premise is simple: you are the Jumpman, a superhero whose great leaping ability is needed to thwart the dastardly efforts of the Alienators. The Alienators have infiltrated the 30 levels of Jupiter headquarters and booby-trapped every floor with bombs. Utilizing your jumping abilities, you must defuse all the bombs on each level and save command headquarters. It all sounds rather simple and clear-cut. Not so.

30 Levels, 5 Variations

As hinted, *Jumpman* comes with 30 game levels, each one unique. The 30 levels are divided into three difficulty levels with five game variations in all. The three difficulty levels consist of a beginner's level, made up of eight "easy" floors; an intermediate level, consisting of ten moderately difficult floors; and the advanced level, containing twelve very complex floors. The other two game variations are Grand Loop, which lets you play all 30 levels consecutively, and Randomizer, which allows you to play a random selection of all levels.

It is this variation that makes *Jumpman* the excellent game that it is. Each level presents a different type of problem. Some are out-and-out hand-eye coordination tests, requiring excellent reflexes and quick thinking. Others, however, demand that the player call upon cleverness and imagination in addition to joystick gamemanship. For example, Hot Foot, one of the ten levels in the intermediate game, is impossible to complete unless you discover the correct sequence to defuse the bombs.

In the Atari version, reviewed here, the game begins with a full-blown title page, complete with delightful animation, excellent music, a game demo and, if you have played before, a review of previous high scores. Pressing SELECT starts your game.

You are first asked to select a game variation and difficulty level by using the OPTION key. Next, after pressing START, you indicate how many players, from one to four. Once the preliminaries are out of the way, the contest begins.

The first level scrolls down from the top, accompanied by the title of that particular screen. The title is then replaced by a variety of information: the number of the current player, the difficulty level, the number of remaining Jumpmen (shown by small renditions of Jumpman lined up eager to get into action), the player's current score, and the player's remaining bonus points.

Once the screen has settled into place and the necessary information is displayed, one of the seven Jumpmen allotted at the beginning of each game appears somewhere on the screen. What else appears depends on the level being played. It always is an arrangement of girders, but it might be accompanied by up-ropes, ladders (stationary or moving), and/or floating elevators. The number of combinations is amazing, and the configuration of the girders is always unique.

Off And Jumping

During play, Jumpman scurries about the structures on the screen with astonishing animation as he attempts to overcome obvious obstacles and copes with other hazards such as speeding bullets which come out of nowhere, floating blocks that home in and fire at him like a cannonshot, and swooping bats.

Jumpman is moved by pushing the joystick in the direction you wish him to travel. Make him jump by pressing the fire button and pushing the joystick

in the direction you want him to jump. Be forewarned: keep the joystick pushed in the given direction until the jump is completed. Otherwise, you may very well see Jumpman clinging desperately to the lip of a girder, legs kicking and arms flailing, only to fall to his demise.

Scores are achieved in a number of ways: by defusing bombs, killing creatures, discovering and defusing hidden bombs in the puzzle levels, and by completing levels. Also, points are awarded for any Jumpmen remaining at the end of the game. Also, bonus scoring can really add to your score. Each screen begins with a bonus score total of 1500. Every few seconds, 100 points are deducted. Any bonus points remaining at the completion of a screen are added to the total points score. Obviously, the quicker Jumpman defuses all the bombs on a given level, the more bonus points will be added to the grand total.

Jumpman comes with the several additional options. First, you have the option to change Jumpman's speed. Simply press the appropriate number on the keyboard, from 1 to 8, anytime during play. A medium speed is selected by default if you make no selection. Changing the speed of Jumpman changes your strategy. Jumpman is a little more difficult to control at high speeds.

High Scores Recorded

A vanity board allows players to record high scores. A High Score screen appears at the end of any game in which a player qualifies. He or she may enter via joystick up to three initials next to the score achieved. All information is then automatically saved to disk. In addition, a letter will appear to the right of the score indicating at what level the score was achieved. For example, a letter B will appear by the score if it was accomplished while playing at the beginner's level. High scores can also be cleared, assuming no write-protect tab was placed on the disk, by simply pressing the CLEAR key while the game is initially loading into memory.

All in all, *Jumpman* is a fine game. There are excellent hi-res graphics, intricate animation (wait until you see Jumpman scurry up or down a rope), and enjoyable but unobtrusive circus-like music that plays at the end of each level. These and other nice touches reveal the degree of attention the writers paid to small details and put *Jumpman* head and shoulders above most other games, climbing games in particular.

Epyx, better known for its adventure games (especially the Dungeonquest series), has created what should become an arcade classic.

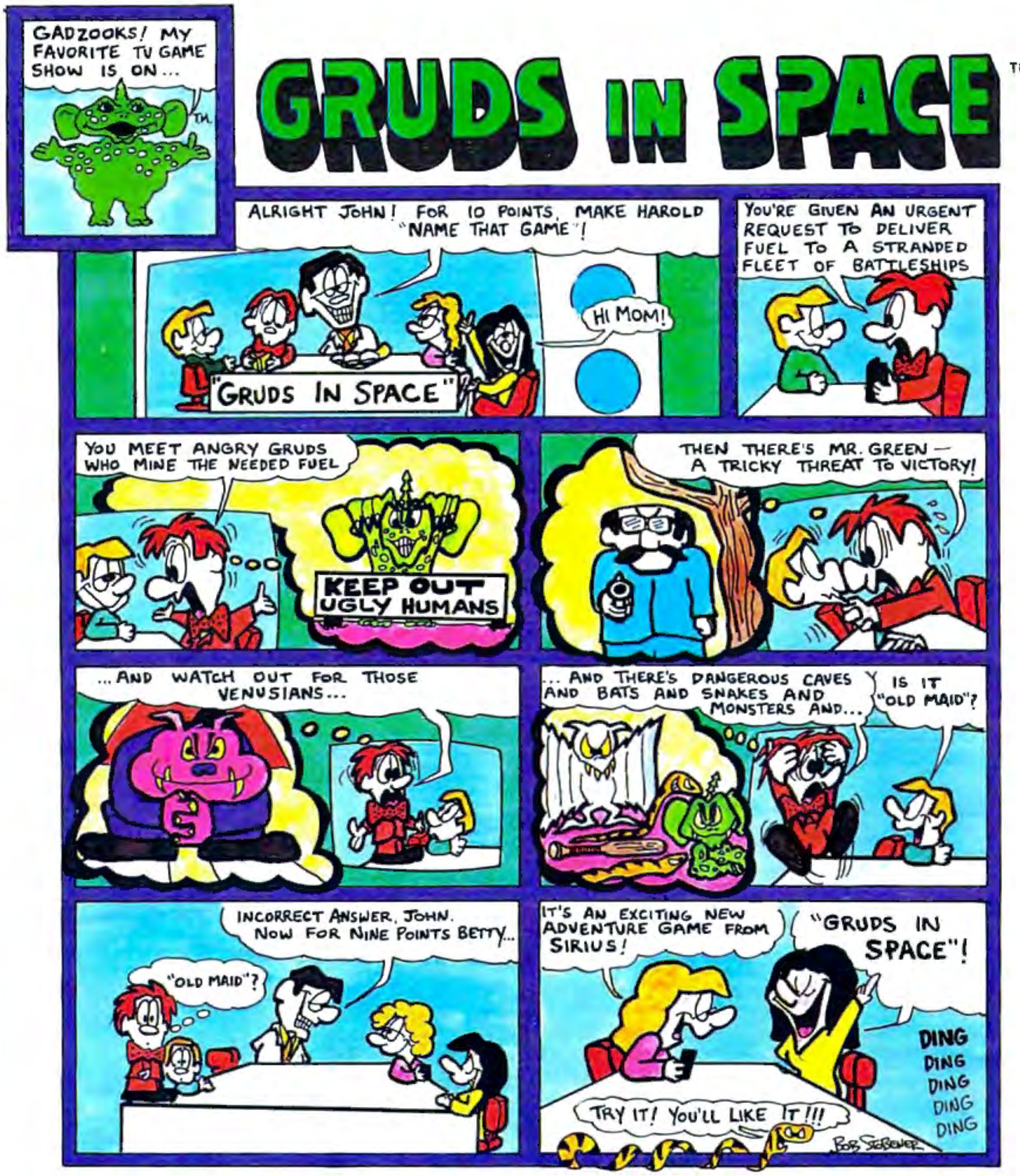
Jumpman
Epyx
Automatic Simulations, Inc.
Sunnyvale, CA 94086
Retail price: $39.95

Synthy 64

Richard Mansfield, Senior Editor

Music can be a complicated affair – you've got pitch, rhythm, tone, and duration/loudness factors to cope with for each musical moment. When you're *digitalizing* music (playing it from a computer), you've added to the complexity because you've got to program lots of numbers. The 64 has many built-in facilities for programming expressive, subtle music, but there's a price: the programmer must specify a good deal about each note.

Synthy 64 is an excellent solution to this dilemma. It allows you full access to the 64's extraordinary sound capabilities, but it greatly simplifies the programming for you. You can avoid



Sirius™

For more information contact your local Sirius dealer or contact Sirius directly at 10364 Rockingham Drive, Sacramento, CA 95827 (916) 366-1195.

Atari 800 & 1200 Disk
IBM-PC Disk
Apple II, II+ & IIe
Commodore 64 Disk

massive amounts of mysterious POKEs or DATA statements. Just as BASIC makes programming easier because you work with English words, *Synthy 64* makes music programming much more understandable. You work with the notes as they would apply to a piano, rather than with frequency numbers. You indicate a C in the third octave as C3, not as 1097.

This example, however, only touches on the efficiency with which you can enter complex music via *Synthy 64*. Because it is a *language*, like BASIC, you can make *Synthy 64* perform complicated tasks with relative ease. In fact, you program it very much the way you program BASIC: line numbers, LIST, RUN, SAVE, INPUT, error messages, and other BASIC-like commands all work the way you're familiar with. And all the music-specific commands are straightforward and easy to use. Voices, portamento, filtering, tempo, and envelope are simple to program, and even conveniences such as REPEAT a phrase and synchronizing are provided for in the *Synthy 64* language.

To give you an idea, here's one of the early examples from the software's documentation:

```
1  RUN
10 T120
20 C5/4 D E F G A6 B C
30 END
```

This will play a scale from the fifth octave C (C5) using quarter notes (/4). The octaves start with A, so you must put the /6 in, but only when changing to a new octave. Line 10 tells the computer how many quarter notes to play per minute. If you don't set a tempo, the default is 100.

All of *Synthy 64* is similarly well thought out and easy to learn and use. Even esoterica such as ring modulation and special bandwidth filtering become both comprehensible and usable. A 41-page booklet and three sample compositions round out the package. If you've been frustrated because you know your 64 has a lot of music inside it, but you've been unable to get at it – this might well be the tool you've been waiting for.

Synthy 64
Abacus Software
P.O. Box 7211
Grand Rapids, MI 49510
$35 (tape), $38 (disk)

VICFORTH From Human Engineered Software

Peter Busby

VICFORTH is an excellent cassette-based implementation of the Forth language for the Commodore VIC-20. Included in the sturdy 8K cartridge are such features as standard 16-line, 64-character-wide editing screens, many new Forth words (commands) that take advantage of the VIC-20 capabilities, ingenious automatic compensation for memory expansion and, best of all, the power to redefine fundamental Forth words.

A minor objection to VICFORTH – it cannot directly access the 3K memory expansion module. (Neither can BASIC when simultaneously expanded upward.) But without resorting to machine language, Forth can easily be persuaded to use the lower 3K RAM module for tables or even, with care, for dictionary space. This language is so flexible that it often presents several solutions to a problem.

Special Commands

On power-up or reset, VIC-

FORTH signs on with a cyan display and green border, indicating the normal operating environment in which programs are run, words executed and perhaps defined, and peripherals interfaced. Five special words control the tape cassette: WRITE/WRITES saves screens to tape, READ/READS loads screens from tape and LOADS loads and compiles screen 1 from consecutive blocks on tape, maximizing RAM usage with the cassette as virtual memory.

Entering 1 EDIT shifts the format to the EDITOR vocabulary, displaying screen 1 on the top 16 lines with a six-line workspace below, all with a white background. The editor includes many commands for finding, moving and eliminating blocks of material on the screens. Seven commands are programmed to the function keys, and the cursor controls are revectored to the 64-character-wide screen which scrolls horizontally.

Pressing INSERT changes the border to yellow and places the editor in direct screen function, in which the material is normally written to the screen. RETURN then leaves the INSERT mode, and STOP/RESTORE returns to the reset environment. This raises a second objection: the first key entered after reset is usually flagged an error. The habit quickly forms of pressing RETURN after reset, which causes the VICFORTH prompt to be displayed, "OK".

Limitations And Enhancements

Error codes conform to the Forth Interest Group's specifications. The 12 errors are indicated by an arrow and a number, cross-referenced to the instruction manual. Full error messages could be displayed at the cost of some memory, as shown in the manual, since MESSAGE is one of 13 *vectored* words provided. Vectoring allows the user to easily change VICFORTH's I/O port interfacing to recognize, for instance, a new printer configuration at the user port.

Other standard fig-Forth definitions missing here are 20 disk-handling words and a few redundant or virtually defunct words. An experienced programmer, with the fig-Forth installation manual in hand, could revector VICFORTH to handle disks. Also, the monitor is missing, but by dropping the memory limit variable EM appropriately and calling the VIC-20 Kernal machine language LOAD, a monitor such as Micromon may be installed. A new word DUMP displays the contents of a range of memory in four-byte lines.

Indeed, any machine language program, the Kernal (Commodore's set of subroutines called from a table), and even the BASIC subroutines in ROM are easily accessed with the word SYS, which has an enormous advantage over BASIC's SYS: the parameters for the accumulator, X and Y registers, and the carry flag can be passed back and forth (so to speak) – invaluable, as Forth often becomes very primitive in its operations.

About 50 words are added to fig-Forth, and many others are defined in the manual to access the VIC-20's features, including words for color and sound control, several predefined Kernal routines, and printer and user-port commands.

A third objection to Tom Zimmer's VICFORTH is the loss of Commodore's screen editor from the operating environment. With the VIC's BASIC editor, normally when RETURN is pressed the interpreter accepts what is on the display – what you see is what you get. In VICFORTH the sequence of keys pressed is acted upon regardless of the display: it is not possible to cursor into a previous line, correct or add to it and press RETURN to reexecute that line. This is partially compensated for by using the editing screens for defining and running procedures, but it seems a shame to lose that powerful interactive editor.

What about that favorite VIC-20 feature, programmable characters? As it happens, the editing screens almost coincide with the internal RAM required for redefining display characters. With straightforward manipulation, for instance, by moving the dictionary pointer the requisite amount (e.g., $1800 DP !) and, after compilation from the screens is complete, using the Kernal to load directly into that space, Greek, Katakana, or whatever characters desired may be made to appear.

Documentation

Finally, the manual: HES provides exceptional documentation. The 80-page instruction book – though not a beginner's handbook – has enough examples and detail in it to clarify much of VICFORTH. An example of the breadth of this concept is the eight pages devoted to adjustments to the recommended *Starting Forth*, from the Forth Interest Group, upgrading that tutorial manual to the VIC-FORTH version. A slight irritation is the use of £ for the symbol # throughout; otherwise the manual does seem to be error-free and complete.

Forth is a fast – very fast – compact, interactive, flexible language, though more arcane than BASIC, perhaps, and including less simple string manipulation. When for the sake of speed or complexity it is necessary to program closer to the level of machine language, Forth's power becomes decisive. This cartridge from Human Engineered Software is by far the most exciting "peripheral" for my computer. VICFORTH could become the center of programming action on your VIC-20.

VICFORTH
by Tom Zimmer
Human Engineered Software
71 Park Lane
Brisbane, CA 94005
$59.95

©

Flower Power Math Fun

Tony Roberts, Assistant Managing Editor

Flower Power Math Fun is an attractive, challenging math drill game that adapts itself to the abilities of the player.

The program includes exercises with whole numbers, fractions, and decimals, and it allows the player to choose addition, subtraction, multiplication, or division problems. A decimal-fraction conversion exercise is also included. The range of the game makes it effective for a child just learning math concepts, and challenging to an adult wishing to refine his ability to solve problems in his head.

The program automatically keeps track of each user's progress, with room for about 100 files on the disk, giving it added value in a classroom situation.

Flower Power Math Fun.

Planting The Garden

Each time a student runs the program, he is asked for his name. Let's say James is playing the game. After he types his name, the program searches the disk to see if any information

about James is available. If so, the screen will ask, "Are you the same James who played before?" If the answer is yes, the computer recalls the information about James's past experiences with *Flower Power Math Fun*. If this particular James hadn't played before, the computer would ask him to select a unique name for purposes of the program. Perhaps Jim, or James D.

Then it's time to choose the type of problems to solve, and the game begins. The object is to fill the garden at the bottom of the screen with flowers by answering math problems correctly. A flower sprouts if the answer is correct. If the answer is wrong, a weed shoots up, and the student is given two more chances to answer correctly. If the correct answer is supplied, the weed is replaced by a flower and the game continues. If each of the three tries is incorrect, the weed remains in the garden.

There is room in the garden for ten plants – flowers or weeds. Once the garden is full, the exercise is over, and the student is given the option of playing again.

Keeping Track Of Skill Levels

The first time each of the 13 games is played, the computer sets the skill level at 1.0. If problems are solved quickly and correctly, the skill level is increased and the problems become more difficult. When a student answers incorrectly, the skill level decreases. Through this method, the program keeps a student of any level working at the edge of his capabilities.

The program also awards points for correct answers, based on the time it takes to provide the right answer. Nothing is subtracted for a wrong answer, and nothing is added for an answer that is too long in coming.

At the end of each session, the student's skill level and score are saved on disk. The next time he plays, he'll be able to pick up where he left off.

The program requires precise answers. For example, if a fraction addition exercise had produced the problem $3/8 + 3/8$, the computer would not accept an answer of $6/8$. The program would prompt the player: "Reduce your fraction."

Unemotional Graphics

Flower Power Math Fun is not an arcade game. Aside from the garden, with its colorful flowers and green weeds, a sun, and a couple of birds in the sky, nothing flashy has been designed into the program. Officials at Softwave Productions use the word "unemotional" to describe *Flower Power Math Fun*'s graphics. "It's enough to keep them interested, but not enough to distract them."

There is no music, and there are no explosions to disturb others in a classroom. The only sound produced by the program is a series of notes which slide up on a right answer, down on a wrong answer.

A session with the program could last as long as time permits. If several students hope to have their chance to play, a student could complete one exercise in about five minutes.

The instructions that come with the program are sparse, but little guidance is needed. A report program included with the game allows a teacher to review the progress of each student either on screen or via a printout. The information available to the teacher includes the student's name, skill level, and high score for each of the 13 exercises. In addition, the report program can be used to erase one or all records from the disk.

A Challenge For All

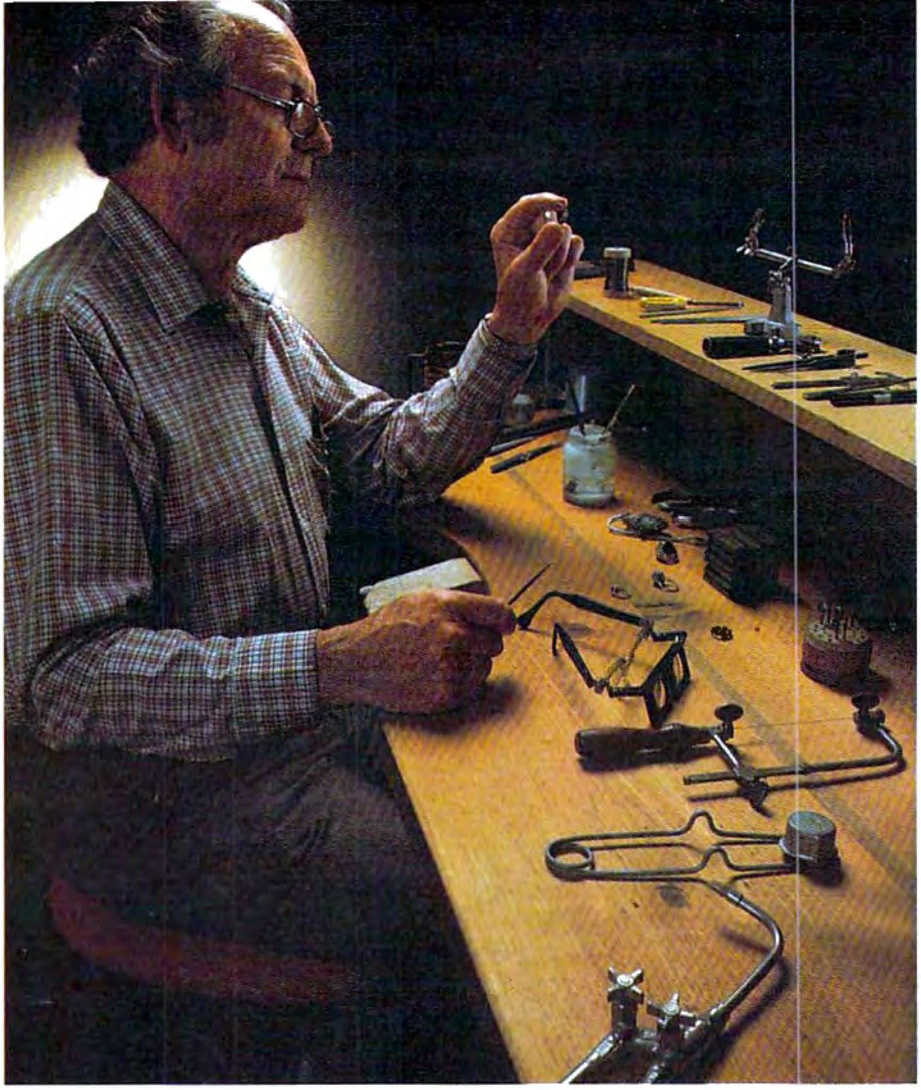
The package describes the game as fun for ages 5-15. A five-year-old would be able to handle many of the problems at lower skill levels, and the higher levels can present a challenge to older players, even those far past the age of 15. The problems at the higher levels of the whole number addition exercise, for example, have you adding 8- and 9-digit numbers without using pencil and paper.

Using *Flower Power Math Fun* regularly, for even just a short time, would sharpen almost anyone's ability to solve math problems mentally. This is not the kind of game that will keep kids riveted to the keyboard during all their after-school hours, but it can satisfy and challenge computer-loving kids in the more sedate setting of a schoolroom.

Flower Power Math Fun is available for the Commodore 64 and Apple II computers.

Flower Power Math Fun
Softwave Productions
156 Drakes Lane
Summertown, TN 38483
(615)964-3573
$39.95

©



Craftsmen Need Precision Tools . . .
Programmers! Demand Precision Software!

BASIC XL has twice the speed and twice the power of Atari® BASIC. And yet, as befits a fine craftsman's tool, BASIC XL is even easier to use and more dependable, while including such outstanding major additions as structured programming, string arrays, programming aids, enhanced graphics, and business capabilities.

Atari BASIC is a good starting point. We should know. We wrote it in 1978. Buy BASIC XL. Take advantage of five more years of experience!

So, prepare yourself for some exploration into imaginative programming with BASIC XL! Cartridge, excellent tutorial, reference manual . . . $99.

Functional, honest, and beautiful describe the simple lines of a craftsman's tools. For the jeweler these tools are an extension of the human hand to better execute complex designs. For you, the programmer, Precision Software tools keep complications out of your programming while allowing you to produce intricate programs.

See the complete collection of OSS Precision Software™ tools!

MAC/65: The fastest 6502 macro assembler/editor package on cartridge . . . $99.

BUG/65: A powerful debugger. On disk, with OS/A+ . . . $35.

C/65: The first native mode "small c" compiler for Atari and Apple® computers. On disk . . . $80.

ACTION!: The fastest, small computer language ever. A feature-packed cartridge at only . . . $99.

OSS™

Precision Software Tools

1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 · (408) 446-3099

ATARI and APPLE II, are trademarks of Atari, Inc. and Apple Computer, Inc., respectively. MAC/65, BUG/65, C/65, BASIC XL, OS/A+ and OSS PRECISION SOFTWARE™ are trademarks of O.S.S. INC. ACTION! is a trademark of Action Computer Products.

Super Software in a SuperCartridge!

Expand usable memory by as much as 50% ONLY with the OSS SuperCartridge™.

Pack up to 24,000 bytes of code into only 8K of your valuable memory—Thanks to the proprietary memory bank system of the OSS SuperCartridge™.

Only OSS can offer you these Precision Software Tools in the most advanced cartridge available.

BASIC XL

BUY THE BEST and only complete compatible enhancement of Atari BASIC! **BASIC XL** makes programming easier for both the beginner and experienced programmers by adding dozens of powerful features.

LEARN TO PROGRAM in only 30 days with our **FREE** tutorial—the best yet for ALL Atari® Home Computers.

BE MORE PRODUCTIVE by using automatic line numbering and renumbering, automatic string allocation, intelligent file name recognition, and more.

RUN PROGRAMS FASTER with **BASIC XL**'s exclusive **FAST** mode. Two to four times—or even more—faster than either of Atari's BASICs.

WRITE BETTER PROGRAMS with string arrays, structured programming controls, an advanced PRINT USING, extensive Player/Missile Graphics support, and much, much more.

FIND OUT MORE than we can possibly put in this ad by calling or writing for a complete catalog.

SUPER BONUS

Use **DOS XL** with a SuperCartridge and save an extra 5K of valuable RAM. Retail value $30. Only $10 with purchase of any two OSS SuperCartridges.

ACTION!

PROGRAM WITH ACTION!—the newest and fastest Precision Software Tool from OSS. **ACTION!** is an amazingly complete, consistent, and properly structured language which combines features from C, Pascal, Ada, and even BASIC.

WRITE THE FASTEST GAMES ever written in a high-level language for 6502-based computers. Speeds 100 to 200 times faster than BASIC are standard in **ACTION!** programs.

EDIT WITH EASE thanks to **ACTION!**'s built-in screen editor which compares favorably to even the best word processors.

PRODUCE MORE CODE faster and easier using **ACTION!**'s program monitor and built-in library of support routines.

GET A FOUR-IN-ONE DEAL unlike anything ever offered. Editor, compiler, monitor, and library all in an OSS SuperCartridge.

All OSS SuperCartridges are $99 each and will work in any Atari computer with at least 16K RAM. Disk or cassette highly recommended.

MAC/65

EXPERIENCE THE FASTEST 6502 macro assembler ever produced. **MAC/65** gives you more than speed. It makes assembly language programming easier and more productive than you thought possible.

USE SOPHISTICATED MACROS to take the drudge work out of assembly language. Build macros libraries to make easy and readable code.

EDIT, ASSEMBLE, AND DEBUG QUICKLY because the program line editor, macro assembler, and a very effective debugger are always just a command away in this OSS SuperCartridge.

COMPARE FEATURES. No other complete 6502 assembly language package offers all of **MAC/65**'s advantages. It even includes support for the 65C02 CMOS microprocessor at no extra charge.

ADDED BONUS: Buy **MAC/65** and get a 65C02 microporcessor for less than $20. Ask for details.

OSS™
Precision Software Tools
1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 • (408) 446-3099

Mothership For Timex/Sinclair

Derek Stubbs

Mothership is a fast and engaging space game for the Timex/Sinclair 1000 with 16K RAM. It has several features which make it unique.

You are the pilot of a Starlight fighter flying in a long canyon (called a Zarway), reminiscent of the last battle scene in *Star Wars*. The fast-moving graphics help to create this illusion.

Out in space, at the end of the canyon, is the giant mothership which launches waves of drone fighters along the Zarway toward you. You can evade them or blast them with your laser, but you must be careful not to crash into the sidewalls of the Zarway.

The object of the game is to destroy the Mothership, which can evade your lasers and will fire back if you get too close. The game has three levels – and can be played by one or two players.

Though *Mothership* is very difficult, the controls make it easier to play. You use groups of keys rather than single keys to move your fighter or fire your laser. This makes the game seem like an arcade game where you concentrate on the screen and just "feel" the controls. For example, the bottom line of keys is the fire button. You can use your thumb to tap this line at the right time while your other fingers guide your fighter.

This package, from Softsync, is presented well. The fast and realistic graphics and the easy controls make it one of the best Timex/Sinclair 1000 spacewar games I have seen. It is comparable to the excellent game *Winged Avenger* from Savage Software.

Depending on the difficulty level chosen, anyone over the age of five can spend many hours playing and enjoying *Mothership*.

Mothership
Softsync
P.O. Box 480
Murray Hill Station
New York, NY 10156
$16.95 plus $1.50 shipping and handling

©

Turmoil For VIC, 64, And Atari

Tony Roberts, Assistant Managing Editor

For quick-draw joystick artists and connoisseurs of shoot-and-run videogames, *Turmoil* (Sirius Software) may be the fulfillment of your dreams.

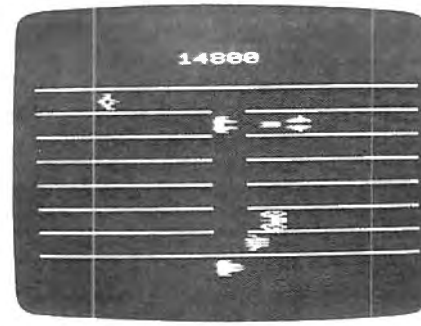
The game, which is available on disk for the Commodore 64 and on cartridge for the VIC-20 and Atari computers, is guaranteed to leave you with cramps in your joystick hand, and may temporarily short-circuit your ability to concentrate under pressure.

Turmoil was designed by Mark Turmell, who is responsible for several other successful Sirius Software games, including *Fast Eddie*, *Beer Run*, and *Free Fall*.

The object of the game is familiar: destroy the aliens before they destroy you. The complications include five alien ships, ghost ships, arrows, and prizes that turn into supersonic cannonballs if you fail to collect them.

Attacking The Aliens

Imagine your screen divided into seven horizontal traffic lanes which the aliens use to traverse your monitor. Down the center is a vertical lane, in which your ship flies.



Guide your ship up and down the center alley and fend off the aliens in Turmoil.

Fly your ship up and down the center alley and blast those aliens. They'll be on you quickly, so keep your thumb on the fire button. Each alien ship moves back and forth at a different pace, and the faster the ship moves, the more points it is worth.

In the first of the game's nine levels, things will be relatively simple. Zip up and shoot left, zip down and shoot right. Repeat. Repeat. Repeat. Watch the score soar.

Toss In A Few Curves

Before you have a chance to get bored with your success at shooting everything that moves, you'll be overrun with problems and wondering how you can save your skin.

Mixed in with the alien vessels will be a few arrows. If you fail to wipe out the arrows on their first pass across the screen, a metamorphosis occurs, and you have to cope with a tank – and a heavily armed one at that. Shooting a tank head-on won't destroy it, but it does knock it back a bit. To defeat a tank, let it pass and blast it from behind.

The Prize Eye

Occasionally you'll see something that looks like a flashing eye at the end of one of the aliens' traffic lanes. This is a prize, and it's worth your while to pick it up quickly.

Under normal circumstances, you'll remain in the center alley, flying up and down. However, when a prize is visible, you'll be allowed to fly down the traffic lane to pick it up.

If you fail to claim the prize, it mutates into a supersonic cannonball and behaves much like a pong-game ball that's gone haywire. It bounces back and forth so fast that getting a clean shot at it is nearly impossible. With a supersonic cannonball on the loose, it's usually just a matter of seconds before your ship has been hit and you're calling in the reserves.

If you can claim your prize, return quickly to the center alley, or you'll be smashed by an indestructible ghost ship.

Aliens, More And Faster

You begin the game with five ships, one in play and four in reserve. Wipe out all the aliens before they wipe you out, and you'll automatically move to the next level and receive a bonus ship to add to your reserves, up to a maximum of six.

As you advance to higher levels, the play becomes faster, and the aliens become more numerous. As an added treat, after you reach level four, the alien traffic lanes occasionally become invisible.

If the aliens get the best of you, as they are bound to, press the fire button to start a new game beginning at the level on which you started. It is possible to start at a higher level, or to change levels during the game, by pressing the f5 key on Commodore machines or the SELECT key on the Atari.

Turmoil is an appropriate name for this game. There are no patterns to memorize or complicated strategies to develop. Building a hefty point total depends solely on your ability to survive amid chaos.

Turmoil
Sirius Software, Inc.
10364 Rockingham Drive
Sacramento, CA 95827
(916)366-1195
Atari cartridge, Commodore 64 disk, $34.95
VIC-20 cartridge, $34.95 ©

The Commander – A Command Extension Facility

Jim Butterfield, Associate Editor

The Commander is a 4K ROM for PET/CBM computers with 4.0 BASIC, which extends the capabilities of BASIC programs. It's a little different from editing "enhancers" and is worth a few extra comments.

There are essentially two types of BASIC-helper programs: editing and runtime.

Editing Aids

Editing packages help you prepare a program. They may give you ease of program entry, renumbering, search and change capability, and other features to help you build the program and spot errors.

Once the program is complete, the editing package has done its job. The completed program will now run by itself. If you have prepared a program using BASIC-Aid, POWER, Sysres, or a similar package, the final program can run on any PET/CBM – even ones without the editing facility.

In other words, if you give a friend a copy of a program that you've written using one of the super-editors, he or she will be able to load it and run it. There's more: since the final program runs without using the editor, it will run at full BASIC speed.

Runtime Aids

Runtime packages help the program run. This means that everyone who wishes to run a program containing a runtime package feature must also have the package. A software house which created products using the Commander, for example, would need to specify that all purchasers of these packages must also have the appropriate runtime package.

A potential difficulty of runtime packages can be that programs using them may slow down, due to the extra overhead. Many packages use a "wedge" technique that slows everything down – sometimes a little, sometimes a lot. The Commander doesn't have this problem: its functions are invoked only when the program calls them in with a suitable SYS command.

The Objectives

An editing package tries to solve the question "What's missing from the machine's regular editing features?" A runtime package addresses the question "What's missing from BASIC?"

People may have different preferences and thoughts about what's missing, based on their experience with other BASICs and other languages. The Commander has chosen these:

PRINT USING: Formats a floating point variable into a string with a specific length and format;

WINDOW: Clears a specific area of the screen, on either 40- or 80-column machines;

FRAME: Prints a screen message, enclosed in a frame, for the user's benefit;

INSERT, APPEND, DELETE: Allows a program to be changed as it runs;

COM LITERALS: Repacks strings to help the above change features;

OVERLAY: Allows a program to call another program to continue the run;

RETURN CLEAR: Cancels all GOSUB and FOR/NEXT loop status;

COMPUTED GOTO: Allows a GOTO address to be calculated;

MAT PRINT#, MAT INPUT#, MAT INIT, MAT ZER: Allows arrays to be saved, recalled, and cleared quickly;

ENHANCED GET, STRING: Allows more powerful INPUT from keyboard or file.

Invoking The Commands

Using the Commander, you would invoke these functions with a SYS command. First, you'd be expected to set up variables which set the range of activities needed. For example, to delete lines 300 to 400 in a running program, you'd type in these lines:

```
360 BL =300       (Begin Line 300)
370 EL =400       (End before Line 400)
380 GT =400       (When complete,
                  GOTO 400)
390 SYS 40006     (Delete)
```

The SYS may be different, depending on the particular ROM chip fitted. When the program reaches line 390, it will delete all lines from 300 to just below 400 – including the above lines – and will continue running at line 400.

This is astonishing. We've made a significant change in a running program and yet the program can continue running, complete with live variables. We can't do that even if we stopped the program and made a change or deletion: our variables would disappear.

The Commander allows a program to modify itself as it runs, without loss of variables. That's amazing. It's also dangerous, and some computer scientists might argue that it's bad practice. The argument runs like this: BASIC is intended to be a "fixed framework" program, and all BASIC programs should be written that way. Self-modifying programs are foreign to BASIC: but with the Commander, you can have them.

Chaining And Such

Users may see this feature as one of the great conveniences of the Commander. Chaining and

Driving Commodore is getting easier

Steer your way to greater efficiency and meet all your administrative needs of the future with the SUPER range of quality software.

Whether you want a fast and flexible database, a comprehensive and easy to use word processor, or a completely integrated office 'manager', Precision Software has the answer. With software that harnesses the full power of the new Commodore "B" Business Computer (known as the 700 in Europe).

Superbase Database manager for all business environments where accurate and up-to-date information is the key to success.

SUPERBASE sets new standards in flexibility and ease of use, with large record sizes, fully re-definable multi-screen record formats, spreadsheet-like calculation facilities, fast 'B+ tree' keyaccess with selective retrieval, transaction linking, sorting and fully definable report formats. SUPERBASE can also be linked to SUPERSCRIPT II.

Superscript II Enhanced version of the popular full feature word processor. Unrivalled table handling with 240 column wide screen, scrolling in all directions, arithmetic, memory calculator, column manipulation and mail merge with record selection. Comprehensive editing and finish quality output formatting. Handles up to 2,100 lines of 80 column text. And it spells!

Versatile, fast, simple to learn and use, SUPERSCRIPT II processes letters, quotations, reports, mailshots and standard forms with professional ease, enhancing presentation and ensuring perfect copies every time.

Superspell Fast, effective spelling checker ensures 100% typographic accuracy. An integral part of SUPERSCRIPT II, SUPERSPELL checks documents against a standard 30,000+ word dictionary and displays a list of every unrecognised word. SUPERSPELL handles both English and American spelling variants. You can accept, ignore, correct or add new words, building up your own dictionary extension. SUPERSPELL also includes a word look- up facility.

Superoffice The ultimate integrated office administration system, with full records management, calculation, word processing and spell checking.

SUPEROFFICE combines the database handling of SUPERBASE with the document editing and formatting power of SUPERSCRIPT II.

The availability of SUPERSCRIPT II as an integral word processor enables lists selected from data files to be used for mailshots, tables, standard forms and labels. SUPEROFFICE includes a powerful programming capability, enabling you to build up a library of your own programs. System-wide help screens are supported by clear comprehensive manuals and tutorials.

Choose an off-the-shelf application to suit your business from a steadily expanding library.

Find out more about the Superseries range of Software. Contact your local Commodore Dealer or Precision Software at our U.K. office.

OFFICIALLY APPROVED BY COMMODORE

Precision Software

Precision Software Limited
Park House, 4 Park Terrace,
Worcester Park, Surrey, KT4 7JZ.
Telephone: 01-330 7166
Telex: 8955021. PRECIS G

overlaying are rather complex activities on a PET/CBM. The Commander gives you shortcuts which make the job easier. The OVERLAY command, for example, says "throw away the old program and variables and start the new one I have named." INSERT, APPEND, and DELETE commands, on the other hand, allow the user to modify or replace the running program with new programming and keep going, complete with variables in place.

The system takes care of most activity needed to preserve variables (using what the manual calls a "Common" facility), but one kind of variable needs a little extra work to preserve: "literal" strings. This is the type of string defined by a statement such as X$="PASS 1". The string is used where it lies in the program line; deleting the program line will make the string disappear. A special Common command may be used to move this kind of string into a safe place.

PRINT USING

The PRINT USING facility doesn't print; it just formats numbers into a string, from which you can print them. It has a reasonable complement of formatting facilities – punctuation, commas, etc. – but doesn't include advanced features such as floating dollar signs, asterisk fill, or special negative symbols such as "CR" or parentheses.

A small deficiency of the PRINT USING feature on the package I tested was that numbers are not rounded; they are truncated. This could cause problems if the user doesn't watch for it, since a financial number such as 6.35 might be held within the computer as 6.34999999; this would truncate to 6.34, which is wrong. The user can cover this possibility by adding .005 to the value before calling PRINT USING: but it would be better if the Commander covered this case.

Odds And Ends

The FRAME and WINDOW commands allow nice screen displays to be generated easily. CONVERT to ASCII can be useful to interface to standard ASCII (non-Commodore) printers, if the interface hardware doesn't do it already for you. RETURN CLEAR allows you to exit from subroutines and loops in an unorthodox manner, and computed GOTO works conveniently and well. The programmer should use these features with caution, since they could lead to "cute" programming that may cause debugging problems.

There are two special I/O commands. STRING INPUT allows information to be input from disk, including punctuation such as commas or colons; it's an improvement over BASIC, which trips up on these. There are MAT PRINT# and MAT INPUT# commands to save and recall arrays; these are written and read in a highly compacted form to save time and disk storage space. Files so written are not strongly compatible with other programs; you would have considerable difficulty getting them back with a conventional set of INPUT commands, so you'll want to use MAT INPUT# to do the corresponding input job.

The other MAT commands are a simple set of array commands which allow you to clear an array quickly (set to zero or null string). There's a gimmick which might be useful: you don't have to clear the whole array, since you can start partway through and clear from that point on.

Packaging

The Commander comes in two versions. Version 11-4 goes into socket UD11 (or UD4 on older machines) and has SYS addresses in the range 40000 to 40110; version 12-3 is for socket UD12 (or UD3 on older machines) and has SYS addresses in the range 44000 to 44110. If your computer already has one ROM chip in place, you can order the version which fits your blank socket; otherwise, it doesn't matter.

Documentation is complete, but a little fuzzy. The manual will seem cryptic to the beginner – for example, the Common facility is described in detail even though in most cases it's automatically invoked when needed. However, the Commander comes with a demonstration diskette, which serves as wonderful documentation. The user is invited to stop the demonstration programs and invoke various features. By using this diskette, the user will enjoy a well-paced training experience.

Programs written using the Commander will always need the Commander. You must be prepared to live with this; but once you do, you'll find a host of dynamic features at your fingertips.

Many of the features are useful; and the capability of having a program modify itself while running is exceptional. Using the Commander, there are new possibilities for your system.

The Commander
$70, including manual and demo diskette
Specify ROM expansion socket, and 4040 or 8050 demo diskette
Commander Systems Inc.
4505 Jackson Street
Hollywood, FL 33021

©

Picnic Paranoia For Atari

James V. Trunzo

Ants and more ants march relentlessly upon their objective. Spiders bar pathways while spinning silvery webs, knowing that the threat of their bite will make wary any who wish them harm. Wasps hover and dart about, their seemingly erratic flight

patterns only a guise that will better enable them to deliver a painful and harmful sting.

Is this a sci-fi movie about insects gone berserk, on TV at four o'clock in the morning? No, it's *Picnic Paranoia*, written by Russ Segal and produced by Synapse Software.

Picnic Paranoia is a refreshingly different hi-res arcade game that is much more than just another simple variation on a theme. The only similarity between *Picnic Paranoia* and, say, *Centipedes* or *Millipede* is that insects are the antagonists. But unlike those games and many others, the "villains" in this game aren't out to attack you directly. They are after something even better – your food.

As the title suggests, *Picnic Paranoia* deals with that universal summer pleasure, the picnic. The game puts you in the role of George, the beleaguered picnicker, and puts you right in the middle of four rectangular, food-covered picnic tables.

The tranquility of the moment is shattered, however, by the appearance of the first of many black ants intent upon pushing the food off the table, across the lawn, and off the screen. They are aided in their endeavors by spiders (a real nuisance) and wasps (downright vicious) – though neither spiders nor wasps have any desire for your food.

Armed with a flyswatter, you must dispatch the insects as quickly as possible and thus prevent them from carrying off your feast. It all sounds rather simple, doesn't it? Well, it isn't. Without quick reflexes, intent concentration, and some strategy, you are soon overwhelmed by the sheer number of insects scurrying all over the screen.

How George Swats

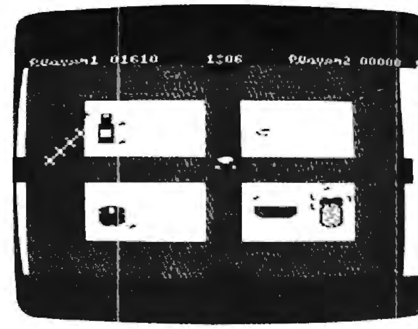
The mechanics of *Picnic Paranoia* are, for the most part, clean, and they function without a hitch. George, who is moved around the screen by a joystick, swats the assorted nasties in one of two ways. By depressing and holding the button on the joystick and moving the stick slightly in the direction George is facing, he achieves one slap with the flyswatter. By continuing to press the joystick in a given direction, he walks and swats at the same time. George walks faster when he isn't swatting – something to keep in mind if no insects are in his path.

The only aspect that takes some getting used to is the technique needed to return food to the table after it has been pushed onto the lawn by the ants. This is important because food on the table is worth more points than food on the ground, and at least one food item must be on a table at the end of each round or the game comes to an end.

To return food to the table, George must swat the food, drag or push it to a table, and release it by swatting it again. The fire button must be released before you attempt to drag or push the food.

Picnic Paranoia plays through a series of 90-second rounds. During this time, George can score points in two ways: (1) by killing insects, and (2) by preventing his food from being pushed off the screen. The ants, spiders, and wasps are worth 10, 20, and 40 points, respectively, multiplied by the round just completed. An ant killed on round five, for example, would be worth 50 points. The value of any food remaining at the end of each round can range anywhere from 30 to 100 points times the given round, depending upon the food's final position. Food still on the table is worth twice as much as food left on the ground. After the first 5000 points have been scored, and for every 10,000 points after that, you earn a bonus.

However, unlike most games, *Picnic Paranoia* does not reward you with an extra player



Swatting at hordes of marauding ants while dodging a bee in Picnic Paranoia.

(you are given three Georges at the beginning of the game). Instead, you are given a can of bug spray, which can be brought into play by hitting the space bar on the computer. The can of spray then traverses the screen three times, killing all insects on the screen at the time. Powerful stuff, but no points are awarded for insects sent to "bug heaven" by the spray. Special concepts like the bug spray will make owners of *Picnic Paranoia* enjoy the contest all the more.

Strategic Moves

Strategy enters the game in several ways. A red "door" is located in the middle of each of the four sides of the playing field. By entering a door, George is able to pop out on the other side of the screen. Making good use of the doors allows you to quickly maneuver him around the picnic grounds, arrive at trouble areas sooner, and often save food from being pushed off the screen.

A second strategy is so obvious that it might be overlooked. Each insect is unique and creates different problems. By recognizing each insect's characteristics, you can determine your best attack at any particular time during a round. The ants can't hurt George, but they are the only insect that can carry off the food. This must always be foremost in your mind.

Spiders can bite George, sending him to the middle of the screen in a state of temporary paralysis. However, the spiders are most annoying for the webs they spin, blocking pathways and thus making movement impossible until the webs are swatted and destroyed. Since speed is an integral part of the game, the webs can exert a significant influence on the eventual outcome of the contest.

Finally, the wasp's sting is more than twice as potent as the spider's bite. It, too, sends George to the middle of the screen, but leaves him immobile much longer. The wasp is difficult to hit, and as soon as one is eliminated, another appears.

A third bit of strategy involves the use of the bonus can of bug spray. For maximum benefit, the timing must be right. Use it too early, and you will still have the greater part of the round left to play. Use it too late, and you may have wasted an opportunity to save it for a more crucial moment in a following round.

The last strategy will be familiar to those who have played such games as *Missile Command*. In the later rounds, after you have lost several food items, as you undoubtedly will, do not waste your effort defending empty tables. Concentrate on attacking the insects surrounding tables that still have food on them. Just as in *Missile Command*, where it is of maximum importance to keep at least one city safe (and ignore empty earth), in *Picnic Paranoia* you must keep at least one piece of food on a table. Therefore, minimize the area you must cover by, for example, zeroing in on the two tables with the most food on them.

Picnic Paranoia offers several options for the gamer. In what has become standard practice with most video and computer games, *Picnic Paranoia* offers five levels of difficulty. Selecting a higher difficulty level simply starts the game off at an advanced round. Because each 90-second round features more and more insects, by beginning at a higher round, you face a greater challenge right from the start.

The game also offers a day or night option. By selecting the night option, you change the green lawn playing screen to a solid black. This is not to suggest, however, that the night option is simply cosmetic. When you are playing at night, you cannot see the picnic tables, and movement is difficult. It is easy to run George into a corner of one of the invisible tables. Playing at night increases the difficulty of *Picnic Paranoia* much more than just selecting a higher difficulty level.

Picnic Paranoia also offers some of the finest graphics and sound available today. From the opening song and graphics of the title page – a delightful animation accompanied by a rousing rendition of the "Flight of the Bumblebee" – to the game-ending animation where literally scores of ants push George off the screen and spell out (with their bodies) "THE END," the hi-res graphics are excellent.

Great attention has been paid to detail in all areas: the foods on the table look absolutely edible and are identifiable, not just shapes of familiar objects. George himself is a sight to behold, complete with a tiny brush mustache and hat. Even the insects are finely done, the wasp drawn with constantly fluttering wings, stripes, and stinger. George's face turns red with embarrassment when stung. A cacophony of sounds matches the graphics to round out this fine effort, and, as you would expect, all animation is flicker-free.

Picnic Paranoia is available from Synapse Software in either cassette or disk format. It runs in 16K.

Picnic Paranoia
Synapse Software
5327 Jacuzzi St. Suite 1
Richmond, CA 94804
$34.95

©

INSIGHT: Atari

Bill Wilkinson

Last month, I said that this month's column would include the final part of the series on writing self-relocatable code. Unfortunately, that project has turned out to be bigger than I thought it would be, so I am going to put it off a month and devote an entire column to it. However, as compensation, I will finally discuss the "new" Atari graphics modes I hinted at a couple of months back. Before I get to the juicy stuff, though, I'd like to continue a little of the ranting and raving that I started last month.

How To Shed 50 Kilobytes Without Even Trying

I heard (from two different sources) the official Atari "line" regarding the new 1050 disk drives. It seems that Atari chose to utilize only 128 bytes per sector and only 127K bytes of file space per drive in order to achieve "increased reliability." Honest. Do you believe it?

Actually, that's pure computer puckey (to paraphrase Colonel Sherman Potter). And it's ridiculous for several reasons.

First off, Atari is implying that double-density drives are unreliable. If that's true, then IBM, Radio Shack, Commodore, and a lot of other computer companies are in real trouble. Actually, Atari and Apple are the ony major computer companies still relying on single-density technology as their primary modus operandi. And, despite Atari's claims, even Atari's 1050 is actually using true double density.

It turns out – based on what we have gleaned from the specs of DOS III at this time – that Atari formats the 1050 drive with 40 tracks of 32 sectors each, with 128 bytes per sector. That's a total of 160K bytes. Most double-density manufacturers achieve either identically the same total (40 tracks times 16 sectors times 256 bytes) or slightly more (40 tracks times 18 sectors times 256 bytes equals 180K bytes – the format used by most Atari-compatible drives such as Percom, Astral, Micro-Mainframe, etc.). So why does Atari claim only 127K bytes?

Real simple: DOS III only supports 127K bytes. Shall I say that I don't know why Atari chose this limitation? With a relatively minor modification, and by using only another 64 bytes of memory per drive, DOS III could have supported a full 180K drive.

Now, as it turns out, I do happen to know the real reason Atari chose 128 bytes per sector. And I know this from the most reliable of sources, one of the DOS III's designers.

It seems that so many of Atari's own products violate Atari's own "rules" (especially those about respecting the LOMEM pointers), and so many other products also reach outside DOS to do direct sector disk I/O that Atari's planners were fearful of the impact of changing either LOMEM or the sector size. Hence the scheme of DOS III.

A secondary impact of the LOMEM problem was that it caused more and more of DOS III to be moved to the diskette from memory, to be called in as overlays when the user requested a function not in memory. Even the keyboard menu processor eventually got moved to disk. The result of all this? While DOS III may be the easiest-to-use DOS yet, it still suffers from the time-consuming swaps to a MEM.SAV file when you want to achieve something as simple as getting a disk directory.

(Of course, there is a very, very elegant way to completely avoid the LOMEM problem on the new Atari XL computers. Why not move the DOS into the as-yet-unused extra memory? Why waste 14K bytes of RAM? I probably shouldn't drop this idea in Atari's laps [I should sell it to them], but it will take them at least six months to even discuss it, so I figure it's OK.)

As I said last month, DOS III contains a nice little file manager. It's a crying shame that it wasn't released three or four years ago, since it seems ideally suited to an 810 size drive. But it doesn't look to me like a system for the long haul, when larger and larger drives become available for the new Atari computers.

And lest too many of my critics cry "foul" for my promoting OSS's version 4 DOS (which will allow up to 32 megabytes per disk drive), let me hasten to say that I am *not* suggesting that version 4 and the 1050 are necessarily the answer. What I *am* saying is that Atari could have achieved virtually the same results by sticking with DOS 2.0 and extending it to handle up to 120K bytes of file space (with 128 byte sectors – it will handle 240K bytes with 256 byte sectors).

Well, enough. I promise no more on this subject until I give you the patches to DOS 2.0s to give you 120K bytes on a 1050. In the meantime, ask yourself this question: if DOS III is limited to 127K bytes of file space, how will Atari handle the double-sided, double-density drive in the 1450XLD, which will have a capacity of at least 320K bytes? Atari, will you answer?

One more comment. I just want to say that, aside from the 1050, I am impressed with all of Atari's new hardware products. And I even like some of their new software. I think Atari is back on its feet and running hard.

Four Equals Seven

Many of the games currently on the Atari market use custom-designed character sets for background displays. The classic example of this is, of course, *Eastern Front* by Chris Crawford. That beautiful scrolling map he displays is actually composed of "characters." This works because a couple of the ANTIC graphic modes allow the programmer to treat each pair of bits within a character cell as one of four colors.

In fact, by controlling the high order bit of the character to be displayed, the programmer may choose two different sets of four colors. Which would be really nice except for the fact that only one of the colors can change between the two sets, thus there is a total of five displayable colors.

If you don't remember and/or understand all that, don't worry. There's a better way. A way which will get you seven colors! The method only works on machines with a GTIA installed, but I hope that all COMPUTE! readers have added a GTIA by now. (If you have purchased a machine in the last year and a half or so, you got a GTIA with your machine. If you have an old machine with a CTIA, the upgrade cost is nominal.)

The credit for finding and documenting this until now hidden feature of the Atari must go to Steve Lawrow, the author of our MAC/65 assembler. He did a nice job of investigating all the ramifications and provided me with the table which I've reproduced here. Before I go into the details of the table, though, let me briefly describe how one accesses two new Atari Graphics modes.

Getting At The New Modes

First, the new modes are variations on BASIC GRAPHICS 1 and GRAPHICS 2 (and, by extension, GR. 17 and GR. 18). And the method of producing the variations is so simple that it's almost funny that no one stumbled on it before. Simply turn on the GTIA's special color mapping mode. And what, pray tell, is that? In this case, it is the upper bit of GPRIOR, the priority select register.

GPRIOR is a hardware register that has its OS shadow location at $026F (decimal address 623). That means (for those of you not familiar with shadow locations) that by changing the RAM location $026F you cause the OS to change the appropriate hardware register for you. (And see COMPUTE!'s book *Mapping the Atari* if you need to know more.)

Briefly, then, you need simply to turn on the upper bit of GPRIOR in order to activate these new modes. There are, however, some caveats to be observed. Perhaps the easiest way to observe the toughest potential problem is to turn on your Atari, go into BASIC, and do a POKE 623,128.

What do you see? Garbage on the screen, if you have a GTIA. Unfortunately, activating the GTIA destroys the normal character display mode(s). More on this later.

Now, on to the table. When you tell BASIC to PRINT #6 in Graphics modes 1 and 2, it prints larger than normal characters to the upper portion of the screen. In particular, though, the characters are available in several different colors. Try this little program to see what I mean: GRAPHICS 2 : PRINT #6; "AaAa" (where the underlined characters are typed in inverse video).

And why do you get four different colors? Because the upper three bits of each of the characters are different. In particular, the upper three bits for the four characters shown are 010, 011, 110, and 111, respectively. Because you are in Graphics mode 2, all four characters came out as uppercase letters.

Now, the bytes which are put in screen memory are actually translations of the bytes which you PRINTed. In particular, when the bytes shown are translated to screen codes, they end up with upper bits of 00, 01, 10, and 11, respectively. The upper two bits of the bytes placed in screen memory determine the color to display; the bits in the character set determine which bits will be "turned on" on the screen.

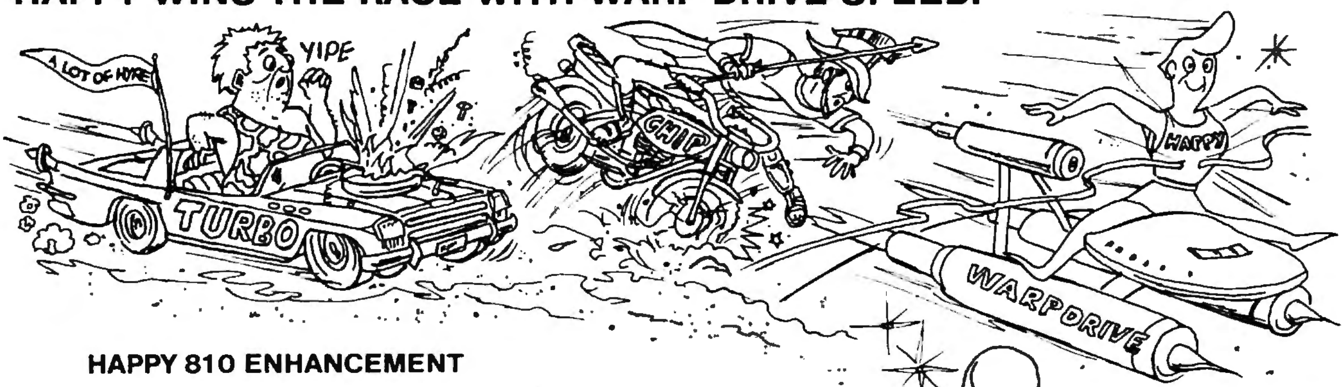
The concept used in our "new" graphics mode is similar. In particular, the upper two bits of the bytes placed in screen memory determine the color MAP to use. The actual bits in the character set determine which color will be selected from the appropriate map. In other words, we have added yet another level of color indirection to the Atari!

In GRAPHICS 10, memory is organized in groups of four bits. The value of the four-bit nybbles determines which color register is displayed. Thus, since there are nine color registers (five for the primary graphics and four for player/missile graphics), there are a maximum of nine independently displayable colors. (Yes, I know that you can get 16 colors in GRAPHICS 9 and 11; but in those cases the colors are not truly independent.)

In GRAPHICS 1+ and 2+ (well, I had to call them something, didn't I?), pairs of bits (instead of four-bit nybbles) determine the color register to use. Remember, though, that the pair of bits can only select a color from the particular MAP which has been selected by the two upper bits of the character on the screen.

And, finally, this implies that the other six

HAPPY WINS THE RACE WITH WARP DRIVE SPEED!



HAPPY 810 ENHANCEMENT

- The only change needed to run all WARP DRIVE software
- Plug in P.C. board requires no permanent modifications
- Proven reliable in thousands of installations, reduces disk drive wear
- Comes completely assembled and tested, just plug in and use
- Full one year parts and labor guarantee, compatible with existing software
- High quality printed circuit board with gold connectors

NO ONE ELSE HAS THIS PERFORMANCE
Unenhanced whole disk (ATARI rev B format) read time: 112 seconds
Unenhanced whole disk (ATARI rev C fast format) read time: 89 seconds
ENHANCED 810 whole disk (any format) read time with standard software: 68 seconds
ENHANCED 810 whole disk (any format) read time with WARP DRIVE software: 43 seconds
Standard software whole disk write and verify time: 238 seconds
WARP DRIVE software whole disk write and verify time: 62 seconds

NEW HAPPY WARP DRIVE SOFTWARE

WARP SPEED HAPPY BACKUP PROGRAM

- Completely automatic: nothing to figure out, insert disks and press return
- Only program on the market guaranteed to backup any disk
- Can write to a blank disk: format write and verify in one operation
- Automatic program tracing: copies only the tracks that are used
- Efficient memory utilization: reduces the number of disk insertions
- Requires only one ENHANCED disk drive, backups will work on a standard drive

WARP SPEED MULTI DRIVE HAPPY BACKUP PROGRAM

- Same features as above plus support of multiple ENHANCED drives
- Can be used with up to 4 ENHANCED drives
- Source and all destination drives read and write in parallel
- Format write and verify 3 complete disks in less than 3 minutes

WARP SPEED HAPPY COMPACTOR PROGRAM

- Reduces the number of disks required to backup your library
- Combines up to 8 self booting disks into 1 disk with a menu
- Compacted disks run only on an ENHANCED drive
- Pays for itself by saving on disks
- Single or dual ENHANCED drive operation

HAPPY WARP DRIVE DOS

- Improves ATARI DOS 2.0S to use warp speed reading and write with verify
- Use all features of BASIC, PILOT, FMS, and DUP at top warp speed
- Warp speed I/O software module available separate from DOS

HAPPY WARP DRIVE SECTOR COPY PROGRAM

- Standard format whole disk read, write and verify in 105 seconds
- Use with sngle or dual drives, mix ENHANCED and NON-ENHANCED drives

HAPPY CUSTOMIZER PROGRAM (sold separately $99.95)

- Creates custom format disks of any specification
- Any type bad sector, duplicate sector numbers, or interleave
- Easy to use but requires an advanced level user to interpret the results

REVIEWED IN POPULAR MAGAZINES

A.N.A.L.O.G. COMPUTING—July/August 1983 *"...The installation instructions for the Happy 810 Enhancement are among the best I have ever seen. ...The Happy 810 Enhancement is one of the most powerful hardware modifications available to ATARI computer owners."*

ANTIC—July 1983 *"The difference between a normal ATARI 810 disk drive and one equipped with Happy is like the contrast between mass transit and the automobile. A car costs you more initially, but improves the quality of your life. Similarly, if you use your disk drive a lot, installing Happy will markedly enhance your programming life."*

SPECIAL SUGGESTED RETAIL PRICE BEFORE DECEMBER 31, 1983: *Get the HAPPY 810 ENHANCEMENT with the single and multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE speed, including our diagnostic for $249.95. Existing registered ENHANCEMENT owners may upgrade to WARP DRIVE speed for $15.00 with no hardware changes!*

Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Please specify -H model for all drives purchased new after February 1982, call for help in ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

ATARI 810 is a registered trademark of Atari, Inc.

HAPPY COMPUTING • P.O. Box 32331 • San Jose, California 95152 • (408) 251-6603 bits of the character on the screen select the particular character from the character set memory, just as normal GRAPHICS 1 and 2 do.

Does that sound complicated? It should, because it is. Anyway, now is the time to look at the table. It shows the MAPs that are available.

Color Selection Under GRAPHICS 1+ And 2+

Bit Pairs Of Color Selectors In Character Memory

Upper Bits of Character on Screen (Map Selector)	00	01	10	11
00	704	704	704	704
01	704	705	712	709
10	704	706	712	712
11	704	707	712	711

And, you presumably ask, what are the numbers shown in the table? Simply the location of the color register which will be displayed when you use the given bits within the given map. For example, 704 is PMCOL0 (player color 0) and 712 is PFCOL5 (playfield color 5). However, the easiest way to change the color registers, in this instance, might be to refer to them via the locations shown in the table.

So, writing POKE 704,0 will make the background color black. Writing POKE 712,152 will give you a nice blue for bit pattern 10 in MAPs 01, 10, and 11. A little observation of the table will show you that MAP 00 is essentially useless: it always gives you the background color, regardless of the bit patterns in the character memory.

On the other hand, bit pattern 00 always gives you background color, regardless of the MAP used, so it may prove useful in many circumstances. For the rest, note that MAP 10 gives you only three colors, but it is the only MAP which gives color 2 (706). Sigh. The system is not totally flexible, but it is handy.

First thing next month we'll put this all together with a little BASIC program that demonstrates the capabilities of the new modes. ©

COMPUTE!'s
The Atari BASIC Sourcebook

Authors: Bill Wilkinson, Kathleen O'Brien, and Paul Laughton
Price: $12.95
On Sale: Now

If you program in BASIC, you know about commands like PRINT, GOSUB, IF-THEN, and others.

But did you know that each of these commands is actually a mini-program in itself? Atari BASIC is a collection of machine language routines that tell the computer what to do, how to do it, and what to do next.

Atari BASIC is a powerful and versatile language. Now available from **COMPUTE! Books**, *The Atari BASIC Sourcebook* offers Atari programmers a chance to look inside the language – directly to the source code that *is* Atari BASIC.

Authors Bill Wilkinson, Kathleen O'Brien, and Paul Laughton, the people who wrote Atari BASIC, take you on a tour through the language. They explain how it works and how you can make it work for you.

The Atari BASIC Sourcebook answers these questions (and more):

- When you RUN a BASIC program, what is really going on inside the computer?
- How does the computer know how to handle a FOR-NEXT loop? How does it RETURN from a subroutine?
- Where do ERROR messages come from? How does the computer know what's wrong?
- How does your Atari decide which mathematical operation to perform first?
- Why do some tasks take so long, while others happen almost instantly?
- Why does the computer sometimes lock up when you delete lines from a program?
- How does the computer interpret words and symbols like GOTO, INT, CHR$, *, and =?
- How can a machine language programmer take advantage of the sophisticated routines in Atari BASIC?

Intermediate to advanced Atari programmers will find a wealth of useful and interesting information in *The Atari BASIC Sourcebook*.

Much more than a simple source code listing, this book explains how BASIC works and why. All major routines are examined and explored. The authors go into detail about the internal design, the stack, input/output statements, and much more. When you finish reading this book, you will have an in-depth understanding of how to put Atari BASIC to work for you in ways you never thought possible.

The Atari BASIC Sourcebook is available at many computer stores and bookstores, and can also be ordered directly from **COMPUTE! Books**.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.
Add $2 shipping and handling. Outside the U.S. add $5 for air mail, $2 for surface mail. All orders prepaid, U.S. funds only.

64 EXPLORER

Larry Isaacs

We are pleased to welcome Larry Isaacs and his new column, "64 Explorer," to the pages of COMPUTE!. *Larry has extensive experience in programming with 6502-based machines.*

To get things started, here's the first of a two-part article on a little understood but important feature of BASIC, the STATUS variable.

This is the first of a two-part series dealing with a feature of BASIC which is not too well documented, the STATUS variable. It is used to detect the success or failure of input/output operations. I came to this realization while writing a disk copy program. At one point I thought I had the program fully operational. I was surprised to discover that the program was copying all but the last byte in the file, causing the copy to be one byte shorter than the original.

Fortunately I was able to quickly determine that the problem was not in my programming. The program worked exactly as I intended it to. Instead, my error was that I made an assumption concerning the STATUS variable which turned out to be incorrect. An error arising from an erroneous assumption can be a very tough one to find. Everything looks right, and doesn't work.

Tracking Down A Mistake

After discovering my error, I reread the documentation to find where I might have overlooked something relating to my mistake. (By documentation, I mean the *Commodore 64 Programmer's Reference Guide* and the *1541 Floppy Disk User's Manual*.) After a thorough reading, I wasn't able to find anything saying that my assumption wasn't valid. I even found an example program in the *1541 User's Guide* which makes the same mistake I did. Fortunately, I had some previous experience which led me to quickly suspect and correct my assumption. It can be quite frustrating if you don't have that experience, and have to acquire it the hard way. Since the STATUS variable is not documented very thoroughly, and is crucial to proper I/O (input/output) communications, perhaps we should try to discover the necessary information ourselves. The all-important question we need to answer is: "What does the STATUS variable really tell us?"

Before starting our investigation, a little introduction to the STATUS variable may prove helpful. The purpose of the STATUS variable is to provide an indication of the completion status of the last input/output operation. It is supposed to indicate, among other things, when an error occurs in the I/O operation, and when the end of the data is reached while reading a file. The end of the data in a file is more commonly called end-of-file, abbreviated *EOF*. My invalid assumption involved this EOF indication.

The STATUS variable acts like a normal variable in most respects. It may be used in an expression just like any other variable, and may be abbreviated ST. However, it differs from other variables in that *you* can't give it a value. It only returns a value. Here are a few examples showing how the ST variable might be correctly used:

```
1050 SS=ST:REM SAVE THE STATUS IN VARIABL
     E SS
2000 IF ST<>0 GOTO 9999
3110 EOI=ST AND 64:REM GET EOI STATUS
```

An example of an incorrect use of the ST variable would be:

```
100 ST=ST-64
```

where the statement attempts to assign a value to ST. This results in a SYNTAX ERROR.

STATUS Can Catch Errors

The value returned by the STATUS variable is used to detect if anything unusual happened during the last I/O operation. The unusual things that may occur will vary depending on which device is involved. The *Reference Guide* describes the meaning of the STATUS variable only for the cassette and serial bus devices, such as the 1541 disk drive. Therefore, we will concentrate on these. In either case, the value returned by the ST variable will be a *signed byte*. To keep things simple, think of this as a byte of memory holding a number which can range from -128 to +127. This differs from an *unsigned byte* which can hold values from 0 to 255.

Instead of getting into a discussion of bits and binary numbers, we will interpret the unsigned byte as being the sum of a unique combination of numbers from the group: 1, 2, 4, 8, 16, 32, 64, -128. An important restriction is that these


READ THE LABEL
THE FILING ASSISTANT
DATA BASE MANAGEMENT for the COMMODORE 64
This data management package allows you to create record formats, sort information, and produce reports. It includes the following features:
Design records up to 256 characters, up to 25 fields, (up to 35 characters per field)
Designate up to 10 two-level sorts
"Step through" records while displaying selected fields to screen
Add, copy, change, or delete records
"Find" a record by key field, or "locate" by record number
Set function keys to insert a frequently used word, phrase, zip code, etc.
No programming or code necessary. On-screen prompts lead to full operation in one to two hours.
THE FILING ASSISTANT contains a built-in mail label program that allows you to print a 1 to 6 line label. You can include more than one field per line and print up to four labels across the page. Labels to be printed can be selected on one or more fields.
The program also allows you to design your own report formats. Your reports may include:
Automatic record counting
Computation between fields
Totals and/or sub-totals per column
Record selection by one or more fields
TECHNICAL NOTE
This program supports the serial or the user port —you can configure just about any printer. It can be used with more than one 1541 disk drive or with the "C-64 Link-IEEE Interface" which allows you to work with the Commodore 2031, 4040, and 8050 disk drives.
THE FILING ASSISTANT is part of a series. It will integrate with THE WRITER'S ASSISTANT (word processing) and THE SPREADSHEET ASSISTANT (forecasting and financial analysis).
Even if you are a beginner THE FILING ASSISTANT will allow you to manage personal and business records. The ASSISTANT SERIES provides you with the most complete and practical information management package available for the Commodore 64.
RAINBOW Computer Corporation
490 Lancaster Pike
Frazer, Pennsylvania 19355

numbers may appear in the sum only once. The presence of one of these numbers in the sum equivalent to the value of ST will indicate the presence of a particular condition. Let's explain this a little further.

You can find which numbers make up the sum by repeating the following steps, using the ST value as the initial remainder: (1) Subtract the next number in the group from the current remainder, starting with -128 and proceeding toward 1. Subtracting -128 is the same as adding +128. (2) If the result is positive and less than 128, include the subtracted number in the sum and use the result as a new remainder. (3) If the result is negative or greater than or equal to 128, don't include the subtracted number in the sum, and then keep the old remainder. When the remainder becomes zero, you will have the numbers which make up the sum. Here are a couple of examples to show how it's done.

```
ST = 115
   - 64   (-128 IGNORED, 115 + 128>128)
     51   (64 INCLUDED)
   - 32
     19   (32 INCLUDED)
   - 16
      3   (16 INCLUDED)
   -  2   (8 AND 4 IGNORED)
      1   (2 INCLUDED)
   -  1
      0   (1 INCLUDED)
115 = 64 + 32 + 16 + 2 + 1

ST = -120
   +128
      8   (128 INCLUDED)
   -  8   (64,32,16 IGNORED)
      0   (8 INCLUDED, 4,2,1 IGNORED)
-120 = -128 + 8
```

Using STATUS With Tape

Since more readers are likely to have the cassette unit instead of a disk drive, we will first investigate the STATUS variable as it relates to cassette. To begin, let's take a look at what the STATUS variable indicates when accessing the cassette.

VALUE	MEANING
1	not used
2	not used
4	SHORT BLOCK
8	LONG BLOCK
16	UNRECOVERABLE READ ERROR
32	CHECKSUM ERROR
64	END OF FILE
-128	END OF TAPE

(Note: This information can be found in the *Commodore 64 Programmer's Reference Guide* on page 85.)

Since the END OF FILE indication was the one I had trouble with, let's begin there. The primary question we want to answer is "When will the STATUS variable indicate we are at EOF?" Will EOF be indicated as the last byte of the file is read, or will EOF be indicated when you try to read one byte beyond the last byte?

To answer this, all we need is a simple little test program. This test program should first write a cassette file containing a few bytes. Then it should instruct us to rewind the cassette. Finally, it should read the cassette file, displaying each byte and the ST variable as each byte is read. Here is a test program to do this, which will be called "TEST 1":

```
100 OPEN 1,1,2,"TEST"
110 PRINT#1,"ABC";:CLOSE 1
200 PRINT "REWIND THE CASSETTE."
210 PRINT "PRESS RETURN WHEN READY."
220 INPUT Z$
300 OPEN 1,1,0,"TEST"
310 FOR I=1 TO 5
320 GET#1,Z$:PRINT I,Z$,ASC(Z$+CHR$(0)),S
    T
330 NEXT I:CLOSE 1
```

As you can see, lines 100-110 write the file, 200-220 ask you to rewind the cassette, and 300-330 read the file. The 2 in the OPEN command in line 100 specifies that the file is being opened for writing with an END OF TAPE marker to be written when the file is closed. You might note that the program reads five bytes from the file, though only three bytes are written. This is done so we might answer another question, namely, "What happens if you read past the end of the file?" Executing TEST 1 will cause the following to appear on the display screen:

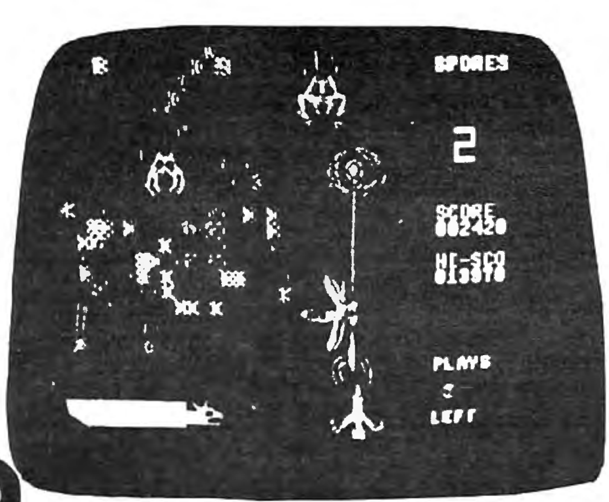
```
PRESS RECORD & PLAY ON TAPE
OK
REWIND THE CASSETTE.
PRESS RETURN WHEN READY.
?
PRESS PLAY ON TAPE
OK
 1     A     65     0
 2     B     66     0
 3     C     67     64
 4           0      0
 5     T     84     0
```

This includes all of the prompts which occur during program execution. Since it is the displayed data which will answer our questions, we'll limit ourselves to that in the tests which follow.

The first column of the displayed data shows a count of the bytes in the file. The second column displays the character, with the third column giving the numeric value (called the *ASCII value*) of the character. The last column shows the value of the ST variable after the GET command which fetched the character.

From this data, we are now prepared to answer the first question. The EOF indication appears with the reading of the last byte of the

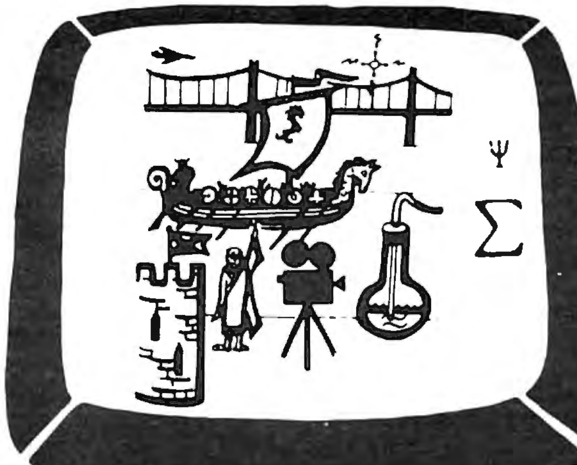
COMMODORE 64



SPORE WAR®

disk $19.95
cassette $14.95

DIGITAL ARTS Corp.



TRIVIA!®

GENERAL TRIVIA + MOVIE TRIVIA
OR
SCIENCE TRIVIA + HISTORY TRIVIA

disk $19.95
cassette $14.95

DIGITAL ARTS Corp.

COMMODORE 64

AIRBASE CONTROLLER®

a simulation of military air control training

disk $19.95
cassette $14.95

DATA/SCAN®

for
mailing lists,
inventories,
phone numbers, etc.

DATA/SCAN

Stores, scans and prints one or three line records and scrolls them forward or backward quickly for you to read or edit Perfect for mailing lists, incoming data, phone numbers or temporary records Stop action lets you see each line, automatically sorts and alphabetizes

disk $29.95
cassette $29.95

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

WE SHIP C.O.D. HONOR VISA AND MASTER CHARGE
ADD $3.00 SHIPPING FOR C.O.D. ADD $2.00 MORE
SPECIAL SERVICES:
One Day — Express Mail add $10.00

cassette file.

Concerning the second question, we now have some test results to examine. First, note that the test program read the two additional bytes with no apparent ill effects – no errors occurred, etc. Second, we can see the EOF indication went off once the next byte was read. And third, notice the first byte following the last one we wrote (i.e., the C) is suspiciously a zero. Since a zero byte doesn't correspond to a displayable character, there is a blank space in the second column where a character would have been. Zero bytes are often used when there is need of a byte which marks the end of something. Thus, it is not too surprising to find one here. But if a zero byte is being used to mark the end of a file, what is going to happen if a zero byte is written as part of the data in the file? To answer this question, let's modify the TEST 1 program to make another test program. Make the following changes to TEST 1 to make "TEST 2":

```
100 PRINT#1,"A";CHR$(0);"C";:CLOSE 1
```

As you can see, TEST 2 will write a zero byte in place of the "B" written by TEST 1. Executing the TEST 2 program displays the following results:

1	A	65	64
2		0	0
3	C	67	64
4		0	0
5	T	84	0

Ah ha! This time we got two EOFs. Since it's not possible for a file to have two ends, one must conclude that what is given as an EOF indication, strictly speaking, doesn't indicate EOF. Instead it is a "next byte is a zero" indication. Provided you do not write any zeros as part of your data, then the zero byte at the end (which is added automatically by the 64) might properly provide the EOF indication. However, if you should accidentally read past the end of your data, you could be in trouble. It doesn't appear that the GET# command can tell if you've passed the end of the file.

The Cassette Buffer

At this point, you might be wondering where the data is coming from once you read past the end of the file. The answer to this is fairly simple. Whenever data is read or written to a cassette file, it will be done in groups of 191 bytes, called blocks. There is an area of memory reserved to hold this block, called the *cassette* tape I/O *buffer*. In the case of writing a cassette file, the data is stored in the cassette buffer until the buffer becomes full (that is, contains 191 bytes). At this point the BASIC program will temporarily stop executing while the cassette motor is turned on and the buffer contents written to the cassette.

Now the buffer is considered empty, and execution of the BASIC program resumes. When the buffer becomes full again, another block is written. When the program closes the cassette file, a zero byte is placed following the most recent byte stored in the buffer and the entire buffer written as the last block in the file. This means there may be bytes following the zero byte which are left over from the previous buffer full.

A similar process occurs for reading a cassette file. Blocks are read from the cassette into the cassette buffer, and then read from the buffer until it becomes empty. As you might have guessed by now, the bytes we are reading after the end of the file are the leftover bytes in the buffer.

So far, our two test programs wrote only three bytes to the file. This implies that our cassette file contains only one block of data. Since our test programs wrote an END OF TAPE marker following the file, it might be interesting to find out what will happen if we read past the one block. To find out, make the following change to the TEST 2 program to make "TEST 3":

```
310 FOR I=1 TO 196
```

This reads five bytes past the one block of data in the cassette file. Executing the TEST 3 program displays the following as the last five lines.

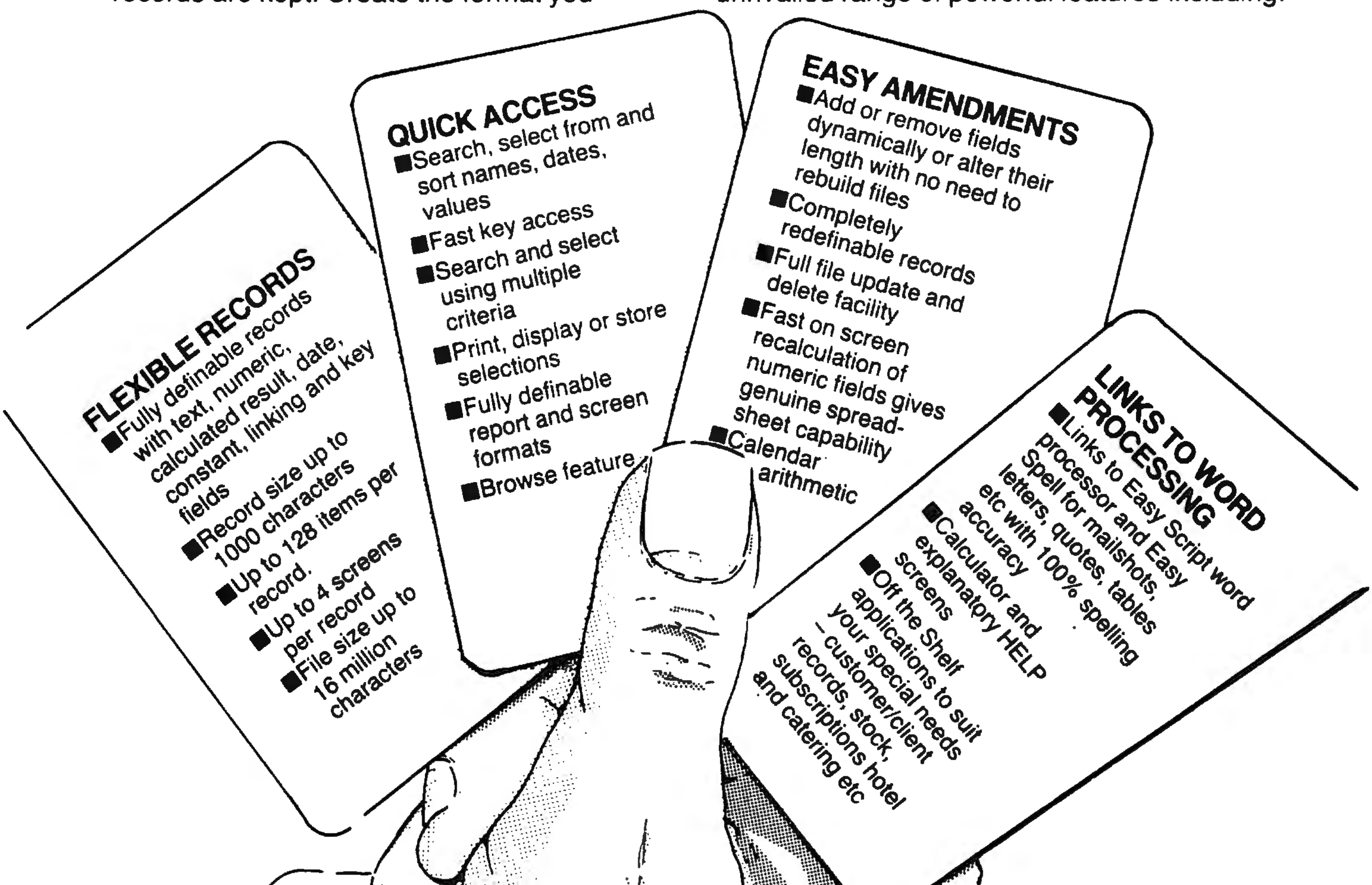
192	<	60	0
193		3	0
194	{C}	252	0
195		3	0
196		32	0

The {C} is the character which would be entered by pressing the COMMODORE key plus C. The first four numbers turn out to be the beginning and ending addresses of the cassette I/O buffer. From this I would assume that the END OF TAPE marker is simply an additional 191-byte block written after the file. It seems to have no terminating effect while data is being read. I suspect that the END OF TAPE marker has an effect only if encountered while the 64 is searching for a file with a specific name.

Well, that's about all the experimenting we have space for in this column. In next month's column we will continue with a few more experiments on the cassette, then look into the disk drive. ©

Strengthen your hand with Superbase 64

The complete information control system for the Commodore 64. Ideal for any home, business or professional environment where records are kept. Create the format you need and enter your records. If the layout or data field sizes are not quite right, correct them and carry on. Superbase gives you an unrivalled range of powerful features including:



Find out more about Superbase 64. Contact your local Commodore Dealer or Precision Software at our U.K. office.

Precision Software Limited
Park House 4 Park Terrace
Worcester Park
Surrey KT4 7JZ England
Telephone: 01-330 7166
Telex: 8955021 PRECIS G

Precision Software

OFFICIALLY APPROVED BY COMMODORE

SPIRALIZER

Chayim Avinor

Based on geometrical principles, this program uses high-resolution graphics to let you create some dazzling patterns. A screen menu makes it very easy to use. Written for the Apple II, versions are also included for the VIC, 64, Atari, and Color Computer.

"Spiralizer" is a program for making patterns on the high-resolution screen of the Apple II. It makes patterns very much like those made on the non-computer game called Spirograph, which has tooth wheels of different sizes. However, the spiralizer can create a far greater number of combinations.

The patterns are actually made by two radii – one of them is turning around a stationary or linearly moving center (depending on your input), and the center of the other radius is the free edge of the first one.

You are given control of the relative speed and length of the radii and some additional handy features.

Running The Program

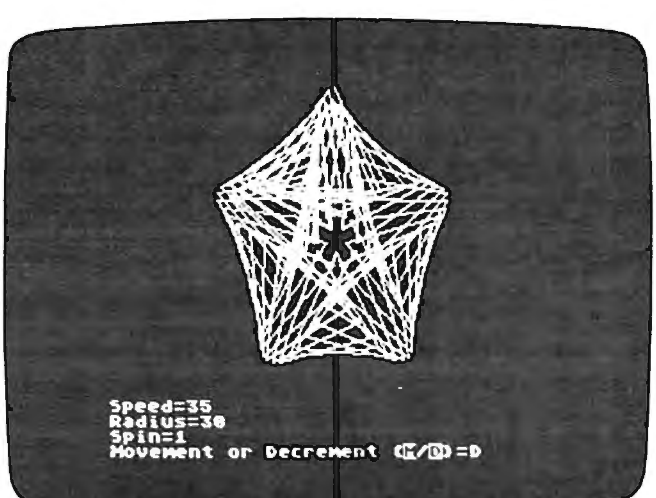
After typing RUN and RETURN, you are asked to type in the relative speed between the two radii. This is actually the number of loops the pattern is going to have. You can choose an answer between -50 and 50. If you type 4 and RETURN, your pattern will have four complete loops. If you type -6 there will be six loops, but they'll be on the inner side of the pattern.

Use the back arrow to delete a character. If you simply press RETURN without typing in a number, the program will default to a value of 5, and will print it on the menu. Large numbers cause the program to draw straight segments, because of the large steps. The patterns thus produced are quite nice, too.

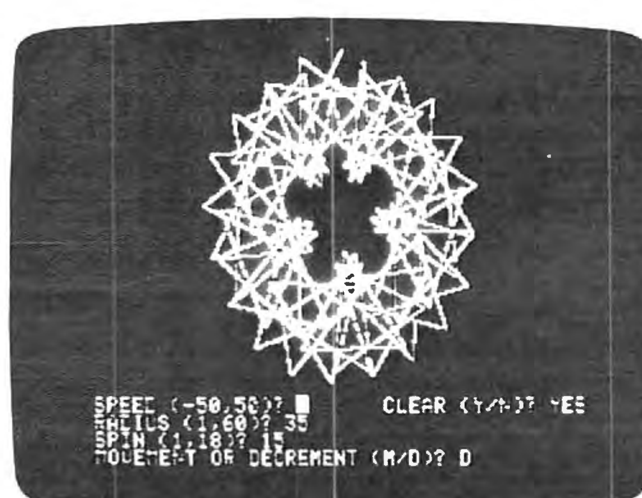
A pattern with three loops is easily understood, but what would a pattern with two loops look like? How about one loop? Could a pattern possibly have zero loops? Try them and see.

The second number you are asked to enter is the radius. This determines the ratio of the radii. You can choose any number between 1 and 60. A small number would make the inner radius small and the outer radius large, and vice versa. Like the first input, you can simply press RETURN and take the default value of 35.

Next you are asked for the spin. An answer larger than 1 will make the pattern rotate while it



"Spiralizer," Atari version.



Another variation available in "Spiralizer," Apple version.

is drawn, and, of course, the number of loops will change. You can choose between 1 (no spin) and 18. When spinning, the lines remain smooth and curvy, but it takes more time to draw the complete pattern. If you decide to quit while a pattern is being drawn, press any key and the program will return to the menu. To escape from the program, use RESET.

Added Features

Now things become more complicated. You are asked, "MOVEMENT OR DECREMENT (M/D)?" If you choose M, the whole pattern will move while it is being drawn. If the spin is 1, the pattern will be drawn five times while it moves. If the spin is greater than 1, the pattern will move until it finishes rotating. If the spin is greater than 1 but less than 9, you will not be asked for this input.

Pressing D will cause the pattern to decrease in size while being drawn. The rules here are the same as above. If you press RETURN, the default value is NONE, and none of the above actions will take place.

Last, you are asked if you want to clear the screen. If you decide not to, then the new pattern

VIC, Atari, And Color Computer Notes

Patrick Parrish, Editorial Programmer

To create different spiral patterns with "Spiralizer," you must first type in several variables (speed, radius, etc.). Try different values for these variables or simply press RETURN (ENTER on the Color Computer) to use default values.

For the VIC version of Spiralizer, you need the Super Expander cartridge. With this version, you can overlay as many as five spiral figures on the high-resolution graphics screen. The Atari and Color Computer versions allow you to overlay as many patterns as you wish.

As with the Apple version, the Atari and VIC versions let you return to the beginning of the program by pressing any key if you are not pleased with the patterns being drawn.

64 Notes

Gregg Peele, Editorial Programmer

Like the Apple, Atari, and Color Computer, the Commodore 64 is capable of high-resolution graphics. Unlike those computers, however, there are no specific commands in the 64's standard BASIC to create graphics on the high-resolution screen. At the time of this writing, the Video Support Package cartridge promised by Commodore is not yet available. We are left with a system of complex POKEs and formulas to accomplish simple hi-res graphics on the 64.

Program 5, the Commodore 64 version, consists of routines to plot points, draw lines, and clear the hi-res screen. All these routines are in machine language and are accessed with the SYS command. The hi-res screen in these programs is located at 8192 ($2000) and may be accessed from BASIC by two POKEs:

```
POKE 53272,PEEK(53272)OR 8
POKE 53265,PEEK(53265)OR 32
```

These POKEs put the screen in bitmap mode with a 320 x 200 high-resolution screen.

To access the plot routine, LOAD and RUN Program 5, and set up a BASIC program which begins with the two POKEs previously mentioned. Next, POKE the X coordinate into 53242. If the X coordinate is greater than 255, then POKE 53243 with 1 and POKE 53242 with X-256. If the X coordinate is less than 256, then POKE 53243 with 0. Now POKE the Y coordinate of the point into location 53240 and POKE a 0 into location 53241. Finally, SYS to location 49710 and your point will appear on the screen. To clear the screen, merely SYS 50039 and all points plotted will vanish. If you want to return to text screen, POKE 53272,21 and POKE 53265,27.

If you want to draw a line between two sets of coordinates, then POKE the first endpoint coordinate (X1) into 53220. As before, if the coordinate is greater than 255, then POKE 53221 with 1 and POKE 53220 with X1-256. If X1 is less than 256, then POKE 53220,X1 and POKE 53221,0. POKE the first Y coordinate into 53222 and POKE 53223 with 0. The coordinates of the other endpoint are now POKEd in starting with X2. If X2 is less than 256, then POKE 53224,X2 and POKE 53225,0. If X2 is greater than 255, then POKE 53224,X2-256 and POKE 53225,1. POKE Y2 into 53226 and POKE 53227 with 0. Once you have these coordinates in memory, use SYS49152 to plot your line on the high-resolution screen.

To use "Spiralizer" on the 64, run Program 5 and then Program 6. Program 6 does all these POKEs for you and will create endless variations of designs for you on the high-resolution screen.

will be drawn on the previous one. This feature allows you to make overlays of patterns.

For a nice sample, I suggest you try the following INPUTs: for speed, enter 7; for radius, 50; for spin, 18; then choose M for movement and clear the screen.

Experiment with different values, and you'll see some stunning designs.

Program 1: Spiralizer – Apple Version

```
10  ONERR  GOTO 90
60 R$ =  CHR$ (8): HCOLOR= 3:H$ = "
     " + R$ + R$ + R$ + R$
70  HOME : HGR
80  VTAB 9: HTAB 15: FLASH : PRINT "SPI
    RALIZER": NORMAL
90  POKE  - 16368,0:Z = 5: HTAB 1: VTAB
    21: PRINT "SPEED (-50,50)? ";: GOSUB
    380:K = Z
100  IF Z <  - 50 OR Z > 50 THEN 90
110 K = K - 1
120 Z = 35: PRINT "RADIUS (1,60)? ";:
    GOSUB 380:R = Z
130  IF Z < 1 OR Z > 60 THEN  VTAB 22:
    GOTO 120
140 R = R + 13:S = 1
150 Z = 1: PRINT "SPIN (1,18)? ";: GOSUB
    380
160  IF Z < 1 OR Z > 18 THEN  VTAB 23:
    GOTO 150
170 A = 1 / Z: IF Z > 1 AND Z < 9 THEN
    240
180 SM = 1:M = 2: PRINT "MOVEMENT OR DE
    CREMENT (M/D)? "H$;
190  GET X$: IF X$ =  CHR$ (13) THEN M =
    0:SM = 0: PRINT "NONE";: GOTO 240
200  IF X$ = "M" THEN SM = 0: GOTO 230
210  IF X$ = "D" THEN M = 0: GOTO 230
220  GOTO 190
230  PRINT X$;
240  VTAB 21: HTAB 24: PRINT "CLEAR (Y/
    N)? "H$;: GET T$: IF T$ <  > "N"
    THEN PRINT "YES": TEXT : CALL 62450:
    HGR
250  IF T$ = "N" THEN  PRINT "NO"
260 W = 1:Z = 139: IF M = 2 THEN Z = 80
    : IF A = 1 THEN W = 5:M = 1:Z = 12
    2
270  IF SM - A = 0 THEN W = 5
280  IF A < 1 THEN K = K + A
290 C = 0.001: IF A < 1 / 9 THEN M = M /
    2:C = C / 2
300 J = R:I = 79 - R
310  HPLOT Z,0
320  REM -MAIN LOOP-
330  FOR T = 0 TO 6.2831 / A * W STEP 0
    .06283:F =  PEEK ( - 16384): POKE
     - 16368,0: IF F > 127 THEN 90
340  IF SM THEN J = R * S:I = 79 * S -
    J:S = S - C
350  HPLOT  TO Z + T * M -  SIN (T) * J
     +  SIN (T * K) * I,79 -  COS (T) *
    J -  COS (T * K) * I: NEXT
360  GOTO 90
370  REM -INPUT SUBROUTINE-
380 L0 = 0:L1 = 1:B$ = ""
390  PRINT H$;: GET A$
400  IF A$ = "-" AND L0 = 0 THEN  PRINT
    A$;:B$ = A$:L0 = 1:L1 = 2: GOTO 39
    0
410  IF A$ =  CHR$ (13) AND L0 > 0 THEN
    Z =  VAL (B$): PRINT : RETURN
420  IF A$ =  CHR$ (13) THEN  PRINT Z: RETURN
430  IF A$ = R$ AND L0 > 1 THEN  PRINT
    A$;:B$ =  LEFT$ (B$, LEN (B$) - 1)
    :L0 = L0 - 1: GOTO 390
440  IF A$ = R$ AND L0 = 1 THEN  PRINT
    A$;:B$ = "":L0 = 0: GOTO 390
450  IF L0 > L1 THEN 390
460  IF A$ < "0" OR A$ > "9" THEN 390
470  PRINT A$;:B$ = B$ + A$:L0 = L0 + 1
    : GOTO 390
```

Program 2: Spiralizer – VIC Version

```
100 IF FL=1 THEN:GRAPHIC 4
110 FL=1:PRINT"{CLR}":POKE 36879,28:PRINT
    "{DOWN}{3 RIGHT}{YEL}U
    ****************I":PRINT"
    {3 RIGHT}-{14 SPACES}-"
120 PRINT"{3 RIGHT}-{PUR}{2 SPACES}SPIRAL
    IZER{2 SPACES}{YEL}-":PRINT"{3 RIGHT}
    -{14 SPACES}-"
130 PRINT"{3 RIGHT}J
    ****************K{BLU}"
140 POKE198,0:Z=1:PRINT"{2 DOWN}{RED}HOW
    {SPACE}MANY SPIRAL":PRINT"FIGURES (1-
    5) ?{4 SPACES}{3 LEFT}";:GOSUB550:F=Z
150 IF F<1 OR F>5 THEN PRINT"{4 UP}";:GOT
    O 140
160 FORP=1TOF:PRINT"{DOWN}{GRN}SPIRAL FIG
    URE #";P:GOSUB 690
170 FOR I=7TO0 STEP-1:FORJ=1TO50:NEXT J:P
    OKE 38680,I:NEXT I
180 Z=5:PRINT"{OFF}{BLU}{DOWN}SPEED (-50,
    50) ?{4 SPACES}{3 LEFT}";:GOSUB 550:K
    (P)=Z
190 IF Z<-50 OR Z>50 THEN PRINT"{2 UP}";:
    GOTO 180
200 K(P)=K(P)-1
210 Z=35:PRINT"{DOWN}RADIUS (1,60) ?
    {4 SPACES}{3 LEFT}";:GOSUB 550:R(P)=Z
    *7
220 IF Z<1 OR Z>60 THEN PRINT"{2 UP}";:GO
    TO 210
230 S=1
240 Z=1:PRINT"{DOWN}SPIN (1,18) ?
    {4 SPACES}{3 LEFT}";:GOSUB 550
250 IF Z<1 OR Z>18 THEN PRINT"{2 UP}";:GO
    TO 240
260 FOR I=8120 TO 8163:POKE I,32:NEXT I
270 A(P)=1/Z:IF Z>1 AND Z<9 THEN PRINT"
    {2 DOWN}":GOTO 350
280 SM(P)=1:M(P)=2:PRINT"{DOWN}MOVEMENT /
     DECREMENT{2 SPACES}(M/D) ?{5 SPACES}
    {4 LEFT}";
290 GET X$:IF X$="" THEN 290
300 IF X$=CHR$(13) THEN M(P)=0:SM(P)=0:PR
    INT"NONE":GOTO 350
310 IF X$="M" THEN SM(P)=0:GOTO 340
320 IF X$="D" THEN M(P)=0:GOTO 340
330 GOTO 290
340 PRINT X$
350 PRINT"{12 UP}":NEXT P:C5=INT(RND(0)*6
    )+2
360 FOR P=1 TO F
370 W=1:Z=400:IF M(P)=2 THEN Z=401:IF A(P
    )=1 THEN W=5:M(P)=1:Z=370
380 IF SM(P)-A(P)=0 THEN W=5
390 IF A(P)<1 THEN K(P)=K(P)+A(P)
```

Finally! An Affordable Full-Size, Full-Feature PRINTER

For your VIC-20®, C-64® ATARI®

SUG. LIST $299

$229.95!!

Centronics Parallel Types
And RS-232 Serial Types

BASIC PRINTER
(Requires one
Option Below)

FEATURES:

- Full graphics capability.
- In the graphic mode, a column of graphic data can be repeated as many times as you want with a single command.
- Double width character output under software control (5 char. per inch).
- Print position addressable by character or dot (positioning control).
- Graphic character and double width character modes can be intermixed on a single line.
- Automatic printing. When the text exceeds the maximum line length no data is lost due to overflow.
- Self-test printing mode.
- Paper width is adjustable up to 10 inches. Standard plain paper.
- 50 cps print speed.
- 80 characters per line.
- 5 × 7 dot matrix.
- Full 2 yr. Warranty.
- Foreign character sets For U.S., U.K., Sweden, and Germany.

Any of these **Options** allow you to connect and print - cables included.

APROPRINT-2064™ (pictured) . . . .Add: $35.95
For Commodore **VIC-20 & C-64** - Cable included.

APROPRINT-4080™ . . . . .Add: $45.95
For **all Atari Computers** - Cable included.

APROPRINT-1000™ . . . . .Add: $29.95
RS-232-Serial - Name your computer

APROPRINT-8000™ . . . . .Add: $29.95
Centronics type Parallel - Name your computer

ADD: $8.00 shipping (cont. USA), $25.00 (Canada, HI, AK)

(All other foreign orders Add $55.00 (shipped by Air)

The ONE VIC-20® Memory Expansion Board that DOES IT ALL!

Maximum Memory allows you to use more powerful programs for:
- EDUCATION • ENTERTAINMENT • MAIL LISTS
- BUSINESS APPLICATIONS • FINANCIAL RECORDS

A perfect investment to give your family and yourself more enjoyment and use from your home computer! The ease of operation, the neat appearance, and the real POWER it adds to your VIC at this low price makes it a MUST for every VIC home!

SPECIAL LOW PRICE!

Only $124.95

Price includes shipping and handling within Continental USA. Foreign orders please add $15.00. Calif. Residents add 6% sales tax.

10 DAY MONEY-BACK GUARANTEE
If not satisfied, simply return in original condition for your money back.

RAMAX Jr.™

Already own an 8k Expander? Get the NEW **RAMAX** Jr.™! Identical to the RAMAX™ except with 19k instead of 27k. Our instructions will show you how to use your 8k as BLK 3 with Jr. to get the full complement of memory!

Special Only $109.95
Shipping included

To equal the total memory of RAMAX™ you would have to buy a 16k Memory Expansion, PLUS an 8k Expansion, PLUS 3k Expansion. THEN you would need a "mother board". With RAMAX™ you buy just ONE piece ... at ABOUT HALF THE PRICE!

RAMAX™ Features and Specifications:

- Adds up to a full 27k bytes of additional RAM to the standard VIC-20's internal RAM of 5k.
- Built-in switch allows User selection of any combination of 5 areas or RAM memory*:
  BLK 1 (8k: Adr. 8192-16383
  BLK 2 (8k: Adr. 16384-24575)
  BLK 3 (8k. Adr. 24576-32767)
  BLK 5 (Adr. 40960-49151, allows/disallows 8k ROM games)
  RAM (3k: Adr. 1024-4095)
  RESET (Resets computer without power off/on)
- Built-in electrical Fuse to protect equipment
- Totally self-contained. No external power supply needed.
- Two (2) extension connectors allow ANY additional cartridges and/or devices designed for the VIC expansion port
- Very low power consumption (.175 amp usual)
- High reliability gold-plated connectors are designed for long life
- Complete Operating Manual
- 6 month parts and labor warranty to original purchaser
- Factory service

*Many VIC-20 cartridges and programs require certain configurations of the memory (i.e. certain games will only run on the unexpanded VIC while others require the upper portion of the expanded memory). With RAMAX™ you have switches that turn-on and turn-off portions of the memory to provide the right area of memory - all without plugging or unplugging. It's so easy!

TO ORDER:

Send Check or Money Order For the Total
Calif. residents add 6% tax.
Or Contact your Local Dealer
Phone orders Call **(805) 482-3604**

VISA All Prices U.S. Dollars MasterCard

DEALER INQUIRIES WELCOME

WE SERVICE WHAT WE SELL

VIC-20 & Commodore-64 are registered trademarks of Commodore International. Atari is a trademark of Atari Inc.

APROPOS TECHNOLOGY
1071-A Avenida Acaso
Camarillo, CA 93010

New Product!

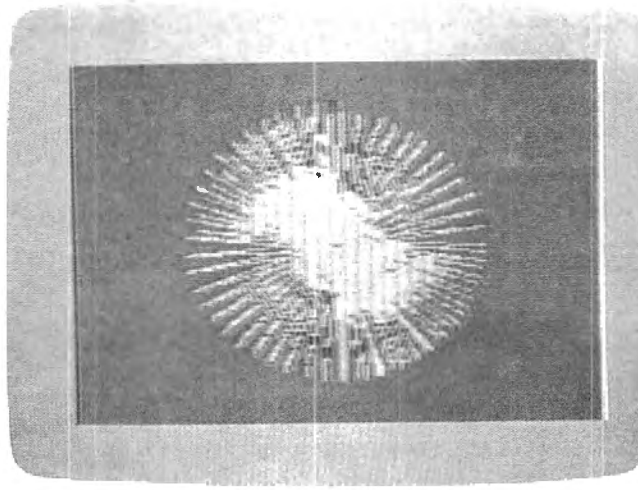
APROSPAND-64™ Gives your Commodore 64 full expandability. This superbly designed expansion module plugs into the 64 & gives you 4 switchable (singly or in any combination) expansion connectors - plus fuse protection - plus a reset button! only $69.95
Shipping Included

APROPOS TECHNOLOGY

```
400 C=.001:IF A(P)<1/9 THEN M(P)=M(P)/2:C
    =C/2
410 J=R(P):I=400-R(P)
420 GRAPHIC 2:COLOR 1,2,C5,C5
430 POINT C5,125+(Z+100)/1.3,130
440 FOR T=0TO 6.2831/A(P)*W STEP .06283
450 IF PEEK(198)<> 0 THEN 100
460 IF SM(P) THEN J=R(P)*S:I=400*S-J:S=S-
    C
470 X=125+(100+Z+T*M(P)-SIN(T)*J+SIN(T*K(
    P))*I)/1.3
480 Y=520-COS(T)*J-COS(T*K(P))*I
490 IF X<0 OR Y<0 THEN 650
500 DRAW 1 TO X,Y:NEXT
510 NEXT P:GOSUB 690
520 GET R$:IF R$="" THEN 520
530 GOTO 100
540 REM -INPUT ROUTINE-
550 L0=0:L1=1:B$=""
560 GET A$:IF A$="" THEN 560
570 IF A$="-" AND L0=0 THEN PRINT A$;:B$=
    A$:L0=1:L1=2:GOTO 560
580 IF A$=CHR$(13) AND L0>0 THEN Z=VAL(B$
    ):PRINT:RETURN
590 IF A$=CHR$(13) THEN PRINT Z:RETURN
600 IF A$=CHR$(20) AND L0>1 THEN PRINT A$
    ;:B$=LEFT$(B$,LEN(B$)-1):L0=L0-1:GOTO
     560
610 IF A$=CHR$(20) AND L0=1 THEN PRINT A$
    ;:B$="":L0=0:GOTO 560
620 IF L0>L1 THEN 560
630 IF A$<"0" OR A$>"9" THEN 560
640 PRINT A$;:B$=B$+A$:L0=L0+1:GOTO 560
650 REM ERROR TRAPPING ROUTINE
660 GRAPHIC 4:PRINT"{CLR}{2 DOWN}COORDINA
    TES ARE OUT{3 SPACES}OF RANGE"
670 PRINT"{2 DOWN}TRY ANOTHER FIGURE":PRI
    NT"{2 DOWN}RETURNING TO MAIN MENU"
680 FOR I=1 TO 4000:NEXT:GOTO 110
690 FORV=15 TO 0 STEP -.5:SOUND245,0,0,0,
    V:NEXT:RETURN
```

Program 3: Spiralizer – Atari Version

```
100 DIM A$(3),X$(4)
110 GRAPHICS 2:SETCOLOR 2,0,0:SETCOL
    OR 0,9,10:POSITION 0,5:PRINT #6;
    "S P I R A L I Z E R":FOR I=1 TO
     750:NEXT I
120 OPEN #1,4,0,"K:"
130 GRAPHICS 8:COLOR 1:SETCOLOR 2,0,
    0
140 ? CHR$(125);:G=4:Z=5:POKE 752,0:
    POKE 656,0:POKE 657,2:PRINT "Spe
    ed (-50 to 50)?{3 SPACES}{3 LEFT}
    ";:GOSUB 450:K=Z
150 IF Z<-50 OR Z>50 THEN PRINT "
    {UP}";:GOTO 140
160 K=K-1
170 G=3:Z=35:PRINT "Radius (1 to 60)
    ?{3 SPACES}{3 LEFT}";:GOSUB 450:
    R=Z
180 IF Z<1 OR Z>60 THEN PRINT "{2 UP}
    ":GOTO 170
190 R=R+13:S=1
200 Z=1:PRINT "Spin (1 to 18)?
    {3 SPACES}{3 LEFT}";:GOSUB 450
210 IF Z<1 OR Z>18 THEN PRINT "{2 UP}
    ":GOTO 200
220 A=1/Z:IF Z>1 AND Z<9 THEN 290
230 SM=1:M=2:PRINT "Movement or Decr
    ement (M/D)?{4 SPACES}{4 LEFT}";
```



A sunburst of color can be seen in the VIC version of "Spiralizer."

```
240 GET #1,X:X$=CHR$(X):IF X=155 THE
    N M=0:SM=0:X$="None":? X$;:GOTO
    290
250 IF X$="M" THEN SM=0:GOTO 280
260 IF X$="D" THEN M=0:GOTO 280
270 GOTO 240
280 PRINT X$;
290 POKE 656,1:POKE 657,24:POKE 752,
    1:PRINT "Clear (Y/N)?{3 SPACES}
    {3 LEFT}";
300 GET #1,V:A$=CHR$(V):IF A$<>"N" T
    HEN GOSUB 550:GOTO 320
310 PRINT "No";
320 W=1:Z=139:IF M=2 THEN Z=80:IF A=
    1 THEN W=5:M=1:Z=122
330 IF SM-A=0 THEN W=5
340 IF A<1 THEN K=K+A
350 C=1.0E-03:IF A<1/9 THEN M=M/2:C=
    C/2
360 J=R:I=79-R
370 PLOT Z+20,0
380 REM -MAIN LOOP-
390 FOR T=0 TO 6.2831/A*W STEP 0.062
    83:IF PEEK(764)<255 THEN POKE 76
    4,255:GOTO 140
400 IF SM THEN J=R*S:I=79*S-J:S=S-C
410 TRAP 420:DRAWTO (Z+T*M-SIN(T)*J+
    SIN(T*K)*I)+20,79-COS(T)*J-COS(T
    *K)*I
420 NEXT T
430 GOTO 140
440 REM -INPUT SUBROUTINE-
450 ZL=1
460 GET #1,V:IF V<>155 THEN 500
470 IF ZL=1 THEN ? Z:RETURN
480 SIGN=1:IF A$(1,1)="-" THEN A$=A$
    (2):SIGN=-1
490 Z=0:FOR I=1 TO LEN(A$):Z=Z*10+AS
    C(A$(I))-48:NEXT I:Z=SIGN*Z:? :R
    ETURN
500 IF V=126 THEN IF ZL>1 THEN ZL=ZL
    -1:PRINT CHR$(126);:GOTO 460
510 IF (V<48 OR V>57) AND V<>45 AND
    V<>46 THEN 460
520 IF ZL=G THEN 460
530 PRINT CHR$(V);:A$(ZL)=CHR$(V):ZL
    =ZL+1
540 GOTO 460
```

Byte Book Club™

FORMERLY COMPUTER PROFESSIONALS' BOOK CLUB

INTRODUCING THE UNIX SYSTEM. *By* H. McGilton & R. Morgan. *256 pp., 50 illus. soft.* This book helps you over the initial hurdles of learning this new and powerful computertool—even if you have never used the Unix system or penetrated any of its literature.
450/013 $18.95

OPERATING SYSTEMS *By* H. Lorin & H. M. Dietel
582354-3 $24.95

MICROCOMPUTER GRAPHICS AND PROGRAMMING TECHNIQUES. *By* H. Katzan, Jr. *240 pp., 100 illus. and tables.* Here's a stimulating introduction to computer graphics for small computers. It covers all the advances to date in color coding and computer graphics technology and—best of all—it's written for information professionals who can't draw! Includes actual graphics programs worth hundreds of dollars!
582576-7 $18.95

MINICOMPUTER AND MICROPROCESSOR INTERFACING. *By* J. C. Cluley. *266 pp., 73 illus. and tables.* Unless you are content to have your information processing system simply talk to itself, you need the intense coverage of interfacing provided so brilliantly by this compact volume. In addition to discussing the logical design of interfaces assembled from small-scale integrated circuits, the book gives you a lucid picture of the interface packages designed for microprocessor systems and the way in which they are used.
582585-6B $27.50
(Counts as 2 of your 3 books)

ASSEMBLERS, COMPILERS, AND PROGRAM TRANSLATION *By* P. Calingaert
582110-9 $24.95

INTERFACE PROJECTS FOR THE TRS-80 *By* R. C. Hallgren
582466-3 $18.95

PROGRAMMING WITH ADA: An Introduction By Means of Graduated Examples *By* P. Wegner
789/24X $19.95

AN INTRODUCTION TO VISICALC® MATRIXING FOR APPLE® AND IBM®. *By* H. Anbarlian. *252 pp., illus., softcover.* Enables you to use VisiCalc matrixes—also known as templates and models—to put your Apple or IBM personal computer to productive use almost immediately. It describes the actual process of developing matrixes for such applications as expense vouchers, price/earnings ratios, payrolls, stock portfolios, and more.
016/054 $22.95

TALKING COMPUTERS AND TELECOMMUNICATIONS. *By* J. A. Kuecken. *237 pp., 94 illus.* A look at a future that's already upon us—and a guide to help you put this exciting technology to work for you! Coverage ranges from the basic techniques of telephony to linear predictive coding, both the hardware and software techniques used in Moser encoding, and the problems of speech recognition.
582801-4B $26.50
(Counts as 2 of your 3 books)

APPLE PASCAL GAMES *By* D. Hergert *and* J. T. Kalash. *371 pp., illus., softcover.* Most fun anybody's had with an Apple since Eve! These 27 games range from the simple to more advanced games such as Numconvert, Wumpus, and Blackbox.
582521-X $19.95

POWERFUL TOOLS! POWERFUL SAVINGS!

Take any 3 books for only $1.00 each*

Values up to $79.90

LEARNING WITH LOGO *By* D. H. Watt. *230 pp., spiralbound, softcover.* Makes LOGO come alive! This introductory Byte Book teaches basic LOGO programming, offers numerous fascinating projects that reveal the powerful ideas underlying this dynamic language, and shows how to avoid common bugs and pitfalls.
685/703 $14.95

MICROPROCESSOR APPLICATIONS HANDBOOK. *Edited by* D. F. Stout. *472 pp., 284 illus.* This BIG book on SMALL chips will help you make your systems timely, versatile, and cost-effective. The 16 expert contributors provide in-depth treatments of both hardware and software so you can completely analyze, design, construct, and program.
617/988B $39.95
(Counts as 2 of your 3 books)

COMPUTER PERIPHERALS FOR MINICOMPUTERS, MICROPROCESSORS, AND PERSONAL COMPUTERS *By* C. L. Hohenstein
294/518 $23.95

THE SMALL COMPUTER CONNECTION: Networks for the Home and Office. *By* N. L. Shapiro. *256 pp.* Shows you how to use existing hardware and software to link your small computer to other computers—large and small—and to a vast universe of databases.
564/124 $15.95

ELECTRONICS ENGINEERS' HANDBOOK, 2/e *By* D. G. Fink & D. Christiansen
209/812A $79.90
(Counts as 3 of your 3 books)

STRUCTURED PROGRAMMING: Theory and Practice *By* R. C. Linger, H. D. Mills & B. I. Witt
788/537B $26.95
(Counts as 2 of your 3 books)

DATA COMMUNICATIONS FOR MICROCOMPUTERS: With Practical Applications and Experiments *By* E. A. Nichols, J. C. Nichols, *and* K.R. Musson. *352 pp., 103 illus.* If your micro lacks the "installation support" needed to establish data communications between equipment within a system as well as between separate systems, you'll welcome this information-packed guide! It covers everything from constructing a simple cable to writing I/O.
464/804 $16.95

Z80 USERS MANUAL. *By* J. Carr. *326 pp., with diagrams, charts, and tables.* Takes you through every opportunity the ZAP can offer! It covers Z80 pin definitions, CPU control signals, support chips, interfacing peripherals, and much more.
582336-5 $21.95

THE BUSINESS GUIDE TO SMALL COMPUTERS *By* L. Calmus. *230 pp., 22 illus.* Written by a long-experienced computer consultant, this is a treasury of information on both hardware and software—requirements, costs, maintenance, implementation, and systems design.
096/627 $19.95

MICROCOMPUTER INTERFACING *By* B. Artwick
789/436B $28.00
(Counts as 2 of your 3 books)

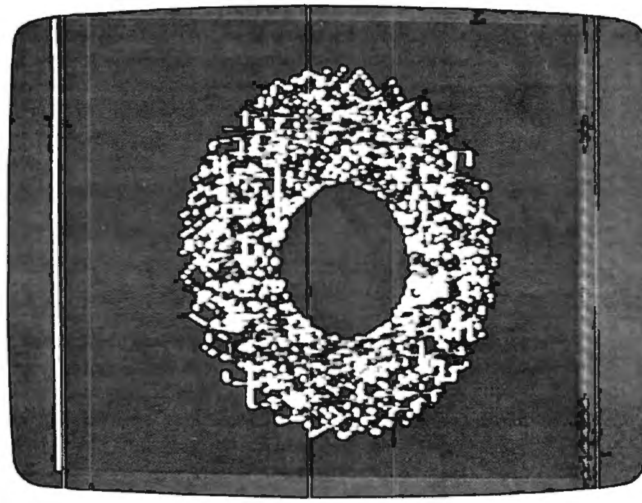
ELECTRONICS DICTIONARY, 4/e *By* J. Markus
404/313B $32.95
(Counts as 2 of your 3 books)

MICROPROCESSOR DATA BOOK. *By* S. A. Money. *350 pp., 220 illus.* A truly awesome collection of data about virtually every chip available today!
427/062B $42.50
(Counts as 2 of your 3 books)

```
550 REM -TEXT WINDOW WRITE-
560 GRAPHICS 8:SETCOLOR 2,0,0
570 PRINT "Speed=";K+1:? "Radius=";R
    -13:? "Spin=";Z:POKE 752,1:PRINT
     "Movement or Decrement (M/D)=";
    X$;
580 RETURN
```

Program 4:
Spiralizer – Color Computer Version

```
10 CLS 0
20 PRINT@200,"SPIRALIZER";
25 FORT=1TO1000:NEXT
30 CLS 3
35 K=5
40 INPUT "SPEED (-50 TO 50)";K$
41 K=VAL(K$)
42 IF K$=""THEN K=5
45 IFK<-50ORK>50THEN40
47 K=K-1
48 R=35
50 INPUT "RADIUS (1-60)";R$
51 R=VAL(R$)
52 IF R$=""THENR=35
55 IF R<1 OR R>60 THEN50
57 R=R+13:S=1
60 INPUT "SPIN 1-18";Z$
61 Z=VAL(Z$)
63 IF Z$=""THENZ=1
65 IF Z<1OR Z>18THEN 60
70 A=1/Z:IF Z>1AND Z<9THEN200
80 SM=1:M=2
85 INPUT "MOVEMENT OR DECREMENT";X$
90 IF LEFT$(X$,1)<>"M" ANDLEFT$(X$,
   1)<>"D"THENM=0:SM=0:PRINT:PRINT
   "NONE";:GOTO200
95 IF LEFT$(X$,1)="M"THENSM=0:GOTO1
   90
100 IF LEFT$(X$,1)="D"THENM=0:GOTO1
    90
190 PRINT X$
200 PRINT:INPUT"CLEAR (Y/N)";T$
205 IF LEFT$(T$,1)<>"N"THENPRINT"YE
    S";:E=0:REMPUT CLEAR OF HIRES
210 IF LEFT$(T$,1)="N"THENPRINT"NO"
    {26 SPACES}
220 W=1:Z=139:IF M=2 THEN Z=80:IF A
    =1 THENW=5:M=1:Z=122
230 IF SM-A=0THENW=5
235 IF A<1THENK=K+A
238 C=0.001:IFA<1/9THENM=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z:Y0=20{10 SPACES}
260 REM MAIN LOOP
270 FOR T=0 TO 6.2831/A*W STEP 0.06
    283
280 IF SM THENJ=R*S  :I=79*S-J:S=S-
    C
290 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I
    {9 SPACES}
300 Y1=79-COS(T)*J-COS(T*K)*I:Y1=Y1
    +20
310 GOSUB 1000:X0=X1:Y0=Y1
320 NEXT
330 A$=INKEY$:IF A$=""THEN330
340 GOTO30
1000 SCREEN 1,1
1010 IF LEFT$(T$,1)="Y"ANDE=0THENPC
     LS:E=1
1020 LINE (X0,Y0)-(X1,Y1),PSET
1030 RETURN
```



"Spiralizer," Color Computer version.

Program 5:
Plot And Drawto Routine (64 Version)

by Gregg Peele, Editorial Programmer

```
10 I=49152:CK=0
20 READA:CK=CK+A:PRINT"{CLR}";CK:IF A=256
   THEN40
30 POKE I,A:I=I+1:GOTO 20
40 IFCK<>167356 THENPRINT"ERROR IN DATA S
   TATEMENTS"
49152 DATA 173,234,207,141,236,207,173
49160 DATA 232,207,141,238,207,173,235
49168 DATA 207,141,237,207,173,233,207
49176 DATA 141,239,207,173,236,207,56
49184 DATA 237,230,207,141,240,207,173
49192 DATA 237,207,237,231,207,141,241
49200 DATA 207,176,24,173,240,207,73
49208 DATA 255,141,240,207,173,241,207
49216 DATA 73,255,141,241,207,238,240
49224 DATA 207,208,3,238,241,207,173
49232 DATA 238,207,56,237,228,207,141
49240 DATA 244,207,173,239,207,237,229
49248 DATA 207,141,245,207,176,24,173
49256 DATA 244,207,73,255,141,244,207
49264 DATA 173,245,207,73,255,141,245
49272 DATA 207,238,244,207,208,3,238
49280 DATA 245,207,56,173,236,207,237
49288 DATA 230,207,141,192,207,173,237
49296 DATA 207,237,231,207,13,192,207
49304 DATA 240,15,144,24,169,1,141
49312 DATA 222,207,169,0,141,223,207
49320 DATA 76,169,192,169,0,141,222
49328 DATA 207,141,223,207,76,169,192
49336 DATA 169,255,141,222,207,141,223
49344 DATA 207,56,173,238,207,237,228
49352 DATA 207,141,193,207,173,239,207
49360 DATA 237,229,207,13,193,207,240
49368 DATA 15,144,24,169,1,141,246
49376 DATA 207,169,0,141,247,207,76
49384 DATA 224,192,169,0,141,246,207
49392 DATA 141,247,207,76,224,192,169
49400 DATA 255,141,246,207,141,247,207
49408 DATA 173,230,207,141,248,207,173
49416 DATA 228,207,141,250,207,173,231
49424 DATA 207,141,249,207,173,229,207
49432 DATA 141,251,207,56,173,244,207
49440 DATA 237,240,207,141,211,207,173
49448 DATA 245,207,237,241,207,13,211
49456 DATA 207,176,15,173,240,207,141
49464 DATA 252,207,173,241,207,141,253
```

Educational News

Orange Plus Educational News located at 23801 Calabasas Road/Suite 2050/Calabasas, CA 91302/(213) 999-5210

Legal Apple[Type] Compatible

New Computer runs CP/M & Apple[Type] Software

CALABASAS — Now instructors can teach Apple-type programs such as LOGO in one class and professional-style word processing in another without any hardware changes. Dual microprocessors (Z80-A & 6502, two computers in one case), with the radical new Orange Plus development the "EuroROM", allows the machine to read/write/work with Apple-type software as well as CP/M programs and access either CPU via the keyboard.

The ***"ORANGE+TWO"***™ is a brand new direction in the evolution of the personal computer. ORANGEFORTH-83, a derivative of Fig-FORTH — a readily available public domain language, is resident in the ROM*. Also included is CP/M 3.0, Digital Research's latest CP/M version. There is also a built-in disk drive controller for two Apple-type drives, a cassette interface, joystick port, color graphics and ASCII keyboard with numeric keypad. For expert word processing, the keyboard features upper and lower case (lower case characters are true descenders) with auto repeat.

This new breed of computer is a breakthrough for educators and school systems throughout the world. The ability to run both Apple-type and CP/M software on the same machine relieves financially-pressed educators from expensive equipment burdens, allowing them to spend more money where it counts...**on the teachers.**

*Available on disk at extrta cost.

MORE FOR LESS

The ***"ORANGE+TWO"***™ 64K computer includes these standard features:

Built-in CP/M 2.2 compatibility	**no charge**
Digital Research's CP/M 3.0	**no charge**
Built-in disk drive controller for two Apple-type drives	**no charge**
Numeric keypad with separate return key	**no charge**
110/220 volt switch selectable power supply, 50-60 Hz	**no charge**
Full function ASCII keyboard with auto repeat	**no charge**
Cassette interface	**no charge**
Fully grounded metal base plate	**no charge**
Adjustable audio volume control	**no charge**
ORANGEFORTH-83, Z80-A FORTH language, resident in ROM	**no charge**
Z80-A CPU, a second computer	**no charge**
6 slot double sided logic board	**no charge**
Bank switchable RAM, fully socketed, expandable to 256K	**no charge**
Programmable 2764 EPROMs	**no charge**
TOTAL: ***"ORANGE+TWO"***™	**$1095.00**

*Compare these features included on the **"ORANGE+TWO"**™ with the competition's over $2000 for equivalent product.*

THE LEASING ALTERNATIVE

CALABASAS — Through select leasing companies, Orange Plus Computer Systems will offer the following lease programs to qualified corporations and educational institutions: An ***"ORANGE+TWO"***™ computer with a green monitor and one disk drive for only $59.55 per month...36 month closed end lease. Also included is a full maintenance program and all revisions and upgrades that may be available during the term of the lease. Subject to credit approval. Call for additional information.

DISTRIBUTORS: CALL (213) 999-5210

CALABASAS — Due to the tremendous amount of individual inquiries and dealer applications, qualified stocking distributors are needed. Select areas available!

MORE PRODUCTS FROM ORANGE PLUS COMPUTER SYSTEMS TO ENHANCE YOUR *"ORANGE+TWO"*™, APPLE II/II PLUS, OR FRANKLIN ACE 1000

Disk-based ORANGEFORTH-83 language with full documentation	**$99.95**
Digital Research's™ CBASIC with disk and documentation	**$49.95**
Orange Plus KoalaPad™ Touch Tablet w/Micro Illustrator™	**$124.95**
Orange Plus Joystick (works on "ORANGE+TWO", Apple II, II+, & //e)	**$29.95**
Orange Plus self-centering joystick	**$39.95**
10MB (Formatted) 5¼" half-height Winchester Hard Disk Drive	**$1495.00**
Controller & Interface for Hard Disk**	**$395.00**

***Includes everything necessary to be installed in **"Orange+Two"**, Apple or Franklin Computer*

The above are suggested retail prices. Prices may vary from state to state.

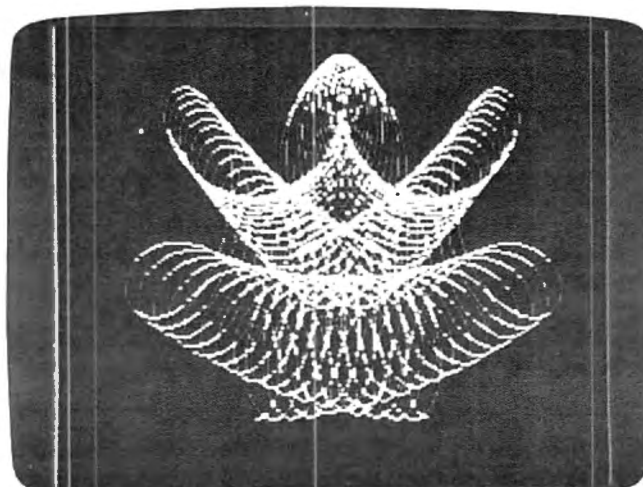
Collins International Trading Corporation,
23801 Calabasas Road, Suite 2050, Calabasas, CA. 91302
(213) 999-5210

```
49472 DATA 207,76,40,193,173,244,207
49480 DATA 141,252,207,173,245,207,141
49488 DATA 253,207,173,252,207,141,254
49496 DATA 207,173,253,207,141,255,207
49504 DATA 56,173,252,207,237,244,207
49512 DATA 141,211,207,173,253,207,237
49520 DATA 245,207,13,211,207,208,17
49528 DATA 173,255,207,74,141,161,207
49536 DATA 173,254,207,106,141,160,207
49544 DATA 76,104,193,173,255,207,74
49552 DATA 141,227,207,173,254,207,106
49560 DATA 141,226,207,173,252,207,13
49568 DATA 253,207,240,27,173,253,207
49576 DATA 48,22,24,173,160,207,109
49584 DATA 240,207,141,160,207,173,161
49592 DATA 207,109,241,207,141,161,207
49600 DATA 76,142,193,76,45,194,56
49608 DATA 173,160,207,237,254,207,141
49616 DATA 211,207,173,161,207,237,255
49624 DATA 207,13,211,207,144,38,56
49632 DATA 173,160,207,237,254,207,141
49640 DATA 160,207,173,161,207,237,255
49648 DATA 207,141,161,207,24,173,248
49656 DATA 207,109,222,207,141,248,207
49664 DATA 173,249,207,109,223,207,141
49672 DATA 249,207,24,173,226,207,109
49680 DATA 244,207,141,226,207,173,227
49688 DATA 207,109,245,207,141,227,207
49696 DATA 56,173,226,207,237,254,207
49704 DATA 141,193,207,173,227,207,237
49712 DATA 255,207,13,193,207,144,38
49720 DATA 56,173,226,207,237,254,207
49728 DATA 141,226,207,173,227,207,237
49736 DATA 255,207,141,227,207,24,173
49744 DATA 250,207,109,246,207,141,250
49752 DATA 207,173,251,207,109,247,207
49760 DATA 141,251,207,32,46,194,206
49768 DATA 252,207,173,252,207,201,255
49776 DATA 240,3,76,104,193,206,253
49784 DATA 207,76,104,193,96,173,251
49792 DATA 207,240,8,173,250,207,201
49800 DATA 65,144,1,96,173,248,207
49808 DATA 201,199,144,1,96,173,249
49816 DATA 207,240,1,96,173,250,207
49824 DATA 141,212,207,173,251,207,141
49832 DATA 213,207,173,248,207,141,214
49840 DATA 207,173,249,207,141,215,207
49848 DATA 173,215,207,74,141,217,207
49856 DATA 173,214,207,106,141,216,207
49864 DATA 173,217,207,74,141,217,207
49872 DATA 173,216,207,106,141,216,207
49880 DATA 173,217,207,74,141,217,207
49888 DATA 173,216,207,106,141,216,207
49896 DATA 173,213,207,74,141,219,207
49904 DATA 173,212,207,106,141,218,207
49912 DATA 173,219,207,74,141,219,207
49920 DATA 173,218,207,106,141,218,207
49928 DATA 173,219,207,74,141,219,207
49936 DATA 173,218,207,106,141,218,207
49944 DATA 173,214,207,41,7,141,220
49952 DATA 207,173,216,207,10,46,217
49960 DATA 207,10,46,217,207,10,141
49968 DATA 210,207,46,217,207,173,217
49976 DATA 207,141,211,207,173,210,207
49984 DATA 10,46,217,207,10,46,217
49992 DATA 207,109,210,207,141,216,207
50000 DATA 173,211,207,109,217,207,141
50008 DATA 217,207,173,216,207,10,46
50016 DATA 217,207,10,46,217,207,10
50024 DATA 46,217,207,141,216,207,173
50032 DATA 218,207,10,46,219,207,10
50040 DATA 46,219,207,10,46,219,207
50048 DATA 141,218,207,24,173,216,207
50056 DATA 109,218,207,141,208,207,173
50064 DATA 217,207,109,219,207,141,209
50072 DATA 207,24,173,220,207,109,208
50080 DATA 207,141,208,207,169,0,109
50088 DATA 209,207,141,209,207,24,169
50096 DATA 32,109,209,207,141,209,207
50104 DATA 173,208,207,133,251,173,209
50112 DATA 207,133,252,173,212,207,41
50120 DATA 7,141,225,207,56,169,7
50128 DATA 237,225,207,141,225,207,169
50136 DATA 0,141,206,207,56,173,225
50144 DATA 207,46,206,207,206,225,207
50152 DATA 16,245,160,0,177,251,13
50160 DATA 206,207,145,251,96,169,147
50168 DATA 32,210,255,169,0,162,0
50176 DATA 157,0,32,157,0,33,157
50184 DATA 0,34,157,0,35,157,0
50192 DATA 36,157,0,37,157,0,38
50200 DATA 157,0,39,157,0,40,157
50208 DATA 0,41,157,0,42,157,0
50216 DATA 43,157,0,44,157,0,45
50224 DATA 157,0,46,157,0,47,157
50232 DATA 0,48,157,0,49,157,0
50240 DATA 50,157,0,51,157,0,52
50248 DATA 157,0,53,157,0,54,157
50256 DATA 0,55,157,0,56,157,0
50264 DATA 57,157,0,58,157,0,59
50272 DATA 157,0,60,157,0,61,157
50280 DATA 0,62,157,0,63,232,208
50288 DATA 157,169,16,162,0,157,0
50296 DATA 4,157,0,5,157,0,6
50304 DATA 157,0,7,232,208,241,96,256
```

Program 6: Spiralizer – 64 Version

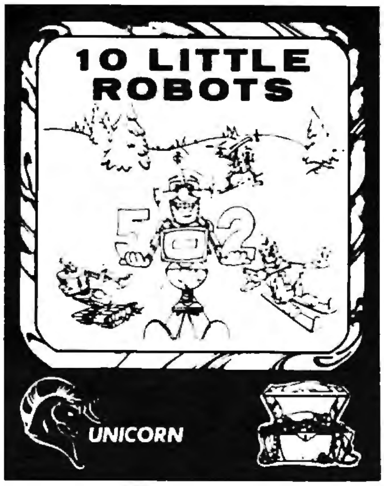
Translated by Gregg Peele, Editorial Programmer

```
10 POKE53281,1:PRINT"{CLR}{9 DOWN}
   {7 RIGHT}";
20 PRINT"{BLK}{2 SPACES}UCCCCCCCCCCCCCCCI"
30 PRINT"{9 SPACES}B{15 SPACES}B"
40 PRINT"{9 SPACES}B{2 SPACES}SPIRALIZER
   {3 SPACES}B"
50 PRINT"{9 SPACES}B{15 SPACES}B"
60 PRINT"{9 SPACES}B{15 SPACES}B"
70 PRINT"{9 SPACES}JCCCCCCCCCCCCCCCK"
80 PRINT"{7 DOWN}{12 RIGHT}PRESS SPACE";:
   POKE198,0:FORT=1TO300:NEXT
100 WAIT197,64,64:POKE53272,21:POKE53265,
    27:PRINT"{CLR}";
```



Complex designs are possible with "Spiralizer," 64 version.

UNICORN TREASURES MAKE LEARNING A PLEASURE

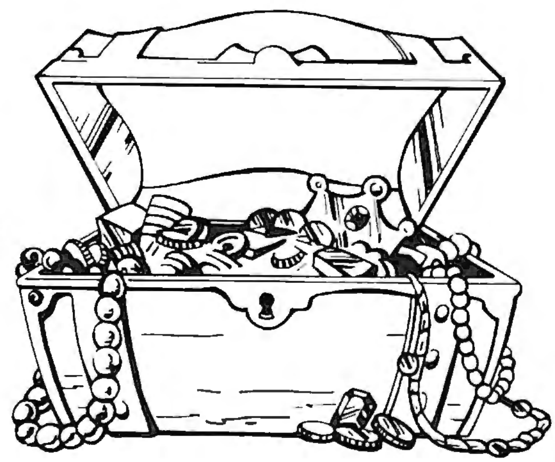


Each Unicorn educational game teaches as it entertains. All our treasures have been developed and tested at The Computer Learning Center for Children. Written by experts who make them educational and fun, our games feature colorful, high-resolution graphics, multiple difficulty levels, beautiful music, and are completely user-friendly with simple on-screen instructions.

Unicorn's educational games are unique in their flexibility. Parents will be delighted to be able to use them year after year as their child's educational needs change. No need to spend a lot of money on software that children will master in a short time and not use again. Our superior packaging will protect our treasures for years to come.

10 LITTLE ROBOTS™— Ages 2-7. The most delightful way to introduce your young learner to the computer. 10 Little Robots has four different games to keep your child's avid attention. There is upper and lower case letter recognition, counting the robots, robot addition, and an interactive storybook tale. The storybook tale introduces the concept of subtraction and serves as a motivational tool for the beginning reader.

Available for Atari 400/800/1200, 48K RAM and one disk drive. Soon to be available for Commodore 64 and IBM PC.



FUNBUNCH™— The most flexible language arts program on the market today. Available on three levels, elementary (grades 1-6), intermediate (junior high school), and college board preparatory (high school). Each level includes over 2000 words and phrases. Within the elementary level the word list can be accessed by grade. The program also allows you to enter your own words and phrases. You can adjust the length of time the words and phrases are displayed on the screen, making Funbunch an excellent tool for *speed reading* as well as *remediation*. There is a built-in printer option which allows you to list the vocabulary for further review. Funbunch also contains a computer doodle drawing game for creative fun. Please specify Funbunch (elementary), Funbunch (intermediate) or Funbunch (college board preparatory) when ordering.

SHIPS AHOY™— Ages 5-13. Outstanding graphics and sound makes this program an entertaining way for children to practice their basic math facts. The object of the game is to sail your ship across the ocean avoiding the treacherous mine hidden beneath the sea. Ships Ahoy allows you to select beginner, intermediate or advanced levels within the four basic math functions. The *flexibility* of this program lies in the option of choosing to be timed or not enabling the academically talented student to practice *speed math*.

RACE CAR 'RITHMETIC™— Ages 5-Adult. A math game that the entire family can play together. Each person can compete on their own level in a race to the finish line. Children and adults alike will have hour upon hour of educational fun together and can practice the four basic math functions, each choosing their own operation, difficulty level and optional time factor. Race Car 'Rithmetic fully utilizes your Atari computer's superb graphics and sound capabilities. From 1 to 4 players.

Available at your local computer store. If you can't find them there, you can order directly from Unicorn Software. All programs $39.95. Please enclose $2.00 for shipping and handling. Visa and Mastercard welcomed.

Atari, Commodore 64, and IBM PC are trademarks of Atari Inc., Commodore Electronics LTD and International Business Machines Corp., respectively.

UNICORN™ SOFTWARE

DEALER INQUIRIES INVITED

UNICORN SOFTWARE™
1775 East Tropicana Avenue
Las Vegas, Nevada 89109
(702) 798-2727 or (702) 798-3085

```
110 Z=5 :PRINT"SPEED(-50 TO 50)? ";"
    {3 SPACES}{4 LEFT}";:GOSUB500:K=Z
115 IFZ<-50OR Z>50 THEN110
125 K=K-1
127 Z=35:PRINT"RADIUS (1 TO 60)? ";"
    {3 SPACES}{4 LEFT}";:GOSUB500:R=Z
130 IFZ<1ORZ>60THEN127
133 R=R+13:S=1
135 Z=1:PRINT"SPIN(1 TO 18)? ";"
    {3 SPACES}{4 LEFT}";:GOSUB500
137 IFZ<1ORZ>18 THEN135
140 A=1/Z:IFZ>1ANDZ<9THEN200
143 SM=1:M=2:PRINT"MOVEMENT OR DECREMENT
    {SPACE}(M/D)? ";"{3 SPACES}{4 LEFT}";
145 GETX$:IFX$=""THEN145
148 IF X$=CHR$(13)THEN M=0:SM=0:PRINT:PRI
    NT"{DOWN}NONE";:GOTO200
150 IFX$="M"THENSM=0:GOTO190
160 IF X$="D"THEN M=0:GOTO 190
170 GOTO145
190 PRINTX$
200 PRINT:PRINT"CLEAR (Y/N)?
201 GETT$:IFT$= ""THEN 201
203 IFT$<>"N"THENPRINT"YES":PRINT"{CLR}";
    :SYS[redacted]50039
205 IF T$="N"THENPRINT"NO" :SYS[redacted]50147
210 W=1:Z=139:IF M=2 THEN Z=80:IF A=1 THE
    N W=5:M=1:Z=122
215 IF SM-A=0THENW=5
220 IF A<1THEN K=K+A
230 C=0.001:IF A<1/9THENM=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z+20 :Y0=30:REM:SYS49152
260 REM MAIN LOOP
270 FORT=0TO6.2831/A*WSTEP0.06283
280 IF SMTHENJ=R*S:I=79*S-J:S=S-C
290 X1=Z+T*M-SIN(T)*J+SIN(T*K)*I :X1=X1+2
    0
300 Y1=79-COS(T)*J-COS(T*K)*I:Y1=Y1+30
308 GOSUB1000:X0=X1:Y0=Y1
310 NEXT :GOTO100
500 L0=0:L1=1:B$=""
510 PRINTH$;
520 GETA$:IF A$=""THEN520
530 IF A$="-"AND L0=0THENPRINTA$;:B$=A$:L
    0=1:L1=2:GOTO510
540 IFA$= CHR$(13)AND L0>0THENZ=VAL(B$):P
    RINT:RETURN
550 IF A$=CHR$(13)THENPRINTZ:RETURN
560 IF A$=R$ANDL0>1THENPRINTA$;:B$=LEFT$(
    B$,LEN(B$)-1):L0=L0-1:GOTO510
570 IF A$=R$AND L0=1THENPRINTA$;:B$="":L0
    =0:GOTO510
580 IFL0>L1THEN510
590 IFA$<"0"ORA$>"9"THEN510
600 PRINTA$;:B$=B$+A$:L0=L0+1:GOTO510
900 POKE53272,21:POKE53265,27:PRINT"{CLR}
    ";:END
1000 POKE53272,PEEK(53272)OR8:POKE53265,P
     EEK(53265)OR32
1005 X0=INT(X0):IFX0<=255THEN POKE53220,X
     0:POKE53221,0
1010 IF X0>255THENPOKE53220,(X0-256):POKE
     53221,1
1030 POKE53222,Y0:POKE53223,0
1040 X1=INT(X1):IFX1<=255THEN POKE53224,X
     1:POKE53225,0
1050 IF X1>255THENPOKE53224,(X1-256):POKE
     53225,1
1060 POKE53226,Y1:POKE53227,0
1200 SYS49152 :RETURN
```

©

Commodore EXEC

Edwin King

There are times when you'd benefit from being able to access subroutines directly from a disk file. They're not in your program (taking up space), but they can be accessed from a main program, executed, and then the main program continues. Called the EXEC command on those few versions of BASIC which have it, this technique is worth adding to your programmer's bag of tricks. For all Commodore machines. We'll go through the process step by step so you can try the technique and watch it in operation.

Here's a way to store all of your favorite subroutines on disk and have programs call them when they're needed, without having to retype or append or use up memory space.

The Technique

The idea behind the EXEC command (as found in Applesoft BASIC; Commodore Microsoft has no such thing) is to execute a subroutine from disk as if it were typed directly into the computer. Just call a command from the disk, in the form of a character string, and start POKEing to the *dynamic keyboard*.

For those not familiar with the dynamic keyboard concept, let me review. Every time a key is pressed, the computer stores the ASCII code representation of it in a place called the "keyboard buffer." It keeps doing this until you press RETURN (which also goes into the buffer), then it goes back to evaluate and execute what you just typed in. Lest the computer forget some of the things you typed, it also keeps track of how many characters you typed before (and including) RETURN.

Now, if we are in immediate mode, we can make the computer think we typed something in by PRINTing it, then RETURNing over what was printed on screen. The dynamic keyboard routine involves PRINTing a command on the screen and then POKEing a few carriage returns (13) into the buffer to make the computer think *we* typed in the command and the carriage returns. This way we only need to POKE one carriage return for every line we want entered.

There are a few drawbacks to this system. First of all, it only works in immediate mode, not as an executing, RUNning program. So, we have to PRINT the command and PRINT a GOTO to get us back into the program. This requires the cursor to be very carefully positioned each time we execute a command – which means no PRINT statements can be anywhere in our EXEC file. Second, INPUT, INPUT#, GET, and GET# are illegal in immediate mode and therefore cannot be used in our EXEC file. And last, since typing in a line with a line number causes that line to be added to the program, our EXEC file will have no line numbers. This means that any use of GOTOs or GOSUBs will call lines in the *program*, not in the EXEC file. Be very careful if you use these commands.

The Program

"EXEC-file" was written on a VIC-20 and will also run, as is, on a Commodore 64 (you may want to change the "22" in line 85 to "40"). It can easily be modified to run on other Commodore machines (more on this later).

Lines	Function
40-60	get input and store file to disk
70-100	call and execute the file
1000-1002	check for disk error

When creating an EXEC file, be sure to type in the EXEC file commands *without* line numbers. Numbers will almost guarantee a crash when you later EXEC the file. The file-call routine (lines 70-100) can easily be lifted and relocated to be used in another program.

Modifications

Users of other Commodore machines should find this program very easy to modify for their system. There are only two changes to be made.

First, change the exit code (the key you press to stop creating the file and get on to other things). The exit code is in quotes on line 55; change the prompt in line 40 accordingly.

Next, the keyboard buffer and the "how-many" (number of characters currently contained in keyboard buffer) location are in different places on different machines. The chart below should assist you in changing this (in line 90).

	VIC/64	Original ROM PETs	BASIC 4.0 PETs
buffer	631-640	527-536	623-632
how-many	198	525	158

Thus, on a PET 4032, line 90 would become:

```
90 POKE 623,13:POKE624,13:POKE625,13:POKE
   626,13:POKE158,4:STOP
```

VIC and 64 owners may also wish to make the print color the same as the screen color before calling this routine, so the user is unaware of the EXEC taking place on screen.

Testing The EXEC

The program here contains both the filemaking and EXEC routines. To test the EXEC function, you must first answer YES when asked if you want to "Create A File?" and then type something like A=51:B=17:F$="Mirabelle" or whatever you want to pass to the program from the disk. It could be a POKE to change screen color or to change the character set, anything you like.

Then, the special file will be on your disk under whatever name you gave it during the filemaking phase. To try out the file, RUN the program again, but answer NO when asked if you want to create a file. This time, the program will move down to line 70 and EXEC the file. When you use the EXEC function in a program, you'll probably want to replace F$ in line 70 with the actual name of the file you want to EXEC: OPEN2, 8,2,"NAME,U,R". The technique of *adding* a string to a quoted name (F$+",U,R") is the way to specify variable file names, but in a real program you'll know in advance the file name that you intend to EXEC.

Pay special attention to the key in quotes in line 55. If you are using a PET/CBM, for example, you'll want to change this to the back-arrow key or something. PET/CBM has no function keys so you could never signal the end of your INPUT when creating an EXEC file.

EXEC-file

```
10 REM *COMMODORE*
15 REM *EXEC-FILE*
20 K$="":A$="":F$=""
25 INPUT"{CLR}{2 DOWN}CREATE A FILE";A$
30 INPUT"{2 DOWN}FILE NAME";F$
35 IFLEFT$(A$,1)<>"Y"THEN70
38 REM**CREATE EXEC FILE **
40 PRINT"{CLR}{3 DOWN}{RVS}{6 SPACES}F1
   TO END{7 SPACES}"
45 OPEN2,8,2,"@0:"+F$+",U,W":OPEN15,8,15
   :GOSUB1000
50 GETA$:IFA$<>""THENPRINTA$;:PRINT#2,A$
   ;
55 IFA$<>"{F1}"THEN50
60 CLOSE2:CLOSE15
66 REM**{3 SPACES}EXECUTE FILE{2 SPACES}
   **
70 OPEN2,8,2,F$+",U,R":OPEN15,8,15:GOSUB
   1000
75 PRINT"{CLR}{4 DOWN}":GET#2,A$:IFA$<>C
   HR$(13)AND(ST<>64)THENK$=K$+A$:GOTO75
80 IF (ST)AND64 THEN 100
85 PRINTK$"{3 DOWN}":PRINT"GOTO75":PRINT
   "{10 UP}"+LEFT$("{7 UP}",INT(LEN(K$)/
   22)):K$=""
90 POKE631,13:POKE632,13:POKE633,13:POKE
   634,13:POKE198,4:STOP
100 CLOSE2:CLOSE15:END
911 REM..{3 SPACES}CHECK DISK ERROR
    {2 SPACES}...
1000 INPUT#15,EN,EM$,ET,ES
1001 IFDS>20THENPRINTEN,EM$:STOP
1002 RETURN
63003 A=PEEK(B)+256*PEEK(B+1):IFA=0THENC
      LR:END
```

Atari Master Disk Directory

Joseph M Apice

With this program you create a single disk "library" incorporating the contents of all your directories. The menu gives you six options – the program is truly multipurpose.

Master directories are an essential part of any computer system. We often take them for granted in the larger minicomputers simply because they exist. These multi-user systems utilize some kind of central library containing a list of all the user directories and their files.

In our smaller home computers, we do not have this luxury. And after working on a mini all day, I find it difficult to do without a master directory so I decided to incorporate some of the nicer features of the larger system into my personal computer. Though it is impossible to exactly duplicate the features, I found I could make a reasonable addition.

I had read several articles dealing with various types of master directory programs. All of them were good, but many required the constant swapping of disks. I needed something that could quickly display the contents of any directory in my library as well as locate any file that I wanted to use without searching through my entire library.

With this in mind, I used the Atari forced read mode to load the contents of every directory in my library as a series of DATA statements in the "Master Disk Directory" program. I could then use the program to examine the contents of any disk, search for any file, and even print labels for my disks without loading any other disk.

The program is menu driven and structured so that each menu function is a subroutine. This allows the user to follow what is being done and to make any desired changes.

Running The Program

After you load the program and type RUN, a menu will display the six options available. Enter the number preceding your selected option and press RETURN.

1. *Directory Update.* This first option is selected each time you enter a new disk or update the listing of a previous disk into the master directory. At the prompt, simply enter the disk name or label and press RETURN. Any additional files which may have existed in the previous disk are automatically deleted when the most recent copy is installed.

2. *Disk Search.* Use this option to review the contents of any disk directory previously installed. Enter the name of the disk you wish to view, and the contents of that disk directory will be displayed to the screen.

3. *File Search.* One interesting feature of the program is that it can quickly locate any named file and its resident disk. The wild card feature is always active if the full name is not specified. Multiple listings of any file will be displayed along with their disk locations. The message NO MATCH FOUND will be displayed if the named file does not reside on any disk.

4. *Print Labels.* Those of you who own a Gemini 10 or Epson MX-80 compatible printer can use this option to print directory labels. The program will allow up to 24 files and one header on any standard (4 x 1-7/16 inch) label. Additional files are printed on the next label.

5. *Install Update.* When you have completed the transfer of all the directories, use this option to install the most recent update into the Master Disk Directory program. The SAVE feature is automatic; when it is completed, the program will return you to the main menu.

6. *Exit*. This option allows you to exit the program and return to BASIC. A word of caution here: This option should be used after option 5 if any updates are being made as it will erase the Master Disk Directory program from memory when it is selected.

DATA Locations

Each directory group of DATA statements is allowed a maximum of 64 lines. This corresponds to the maximum number of data files allowed by DOS on any one disk.

Line 2000 will be the first DATA line. Do not renumber the program without making the necessary changes to the variables LINE, LINCNT, and FIRST.

Master Disk Directory

```
120 DIM A$(20),F$(18),R$(1),SP$(2),L
    B$(20),TAB$(8)
130 STP=65:FIRST=2000:TAB$="
    {6 SPACES}"
150 GRAPHICS 0:POKE 710,146:POKE 712
    ,146:POKE 752,1
160 POSITION 9,4:? "MASTER DIRECTORY
     FILE"
190 FOR PAUSE=1 TO 500:NEXT PAUSE
200 ? CHR$(125):POSITION 13,2:? "MEN
    U SELECTION"
210 POSITION 13,5:? "1..DIRECTORY UP
    DATE"
220 POSITION 13,7:? "2..DISK SEARCH"
230 POSITION 13,9:? "3..FILE SEARCH"
240 POSITION 13,11:? "4..PRINT LABEL
    S"
250 POSITION 13,13:? "5..INSTALL UPD
    ATE":?
260 POSITION 13,15:? "6..EXIT"
280 ? :? :? "ENTER CHOICE---->";:INP
    UT CHOICE
290 ON CHOICE GOSUB 310,670,980,1180
    ,1540,1620
300 GOTO 200
310 REM ** DIR. UPDATE ROUTINE **
320 ? CHR$(125):POSITION 2,6:? "INSE
    RT DISK TO CATALOG IN DRIVE 1"
330 POSITION 2,10:? "DISK LABEL--->
    ";:INPUT LB$
340 TRAP 390:LINCNT=FIRST
360 RESTORE LINCNT:READ A$,N
370 IF A$=LB$ THEN 390
380 LINCNT=LINCNT+STP:GOTO 360
390 LINE=LINCNT:TRAP 520
410 FILCNT=1:OPEN #1,6,0,"D:*.*"
420 INPUT #1,A$
430 ? CHR$(125)
440 ? :? LINCNT+FILCNT;" DATA ";A$:G
    OSUB 470
450 FILCNT=FILCNT+1:GOTO 420
470 ? :? :? "CONT"
480 POSITION 0,0
490 POKE 842,13:STOP
500 POKE 842,12
510 RETURN
520 TRAP 40000:CLOSE #1
530 ? CHR$(125):? :? LINE;" DATA ";L
    B$;",";FILCNT-1
540 GOSUB 470
550 IF FILCNT>=N THEN 630
560 DLINE=LINCNT+FILCNT
570 FOR I=FILCNT-1 TO N
580 ? CHR$(125)
590 ? :? DLINE
600 GOSUB 470
610 DLINE=DLINE+1
620 NEXT I
630 ? CHR$(125):POSITION 8,6:? "ANY
    MORE DISKS";:INPUT A$
640 IF A$="Y" OR A$="YES" THEN 320
650 POSITION 6,12:? "REMOVE DISK PRE
    SS--->RETURN";:INPUT A$
660 LINCNT=FIRST:RETURN
670 REM ** DISK SEARCH ROUTINE **
680 ? CHR$(125):POSITION 2,2:? "SEAR
    CH WHICH DISK-->";:INPUT LB$
690 ? CHR$(125)
700 SP$="  ":LINCNT=FIRST:TRAP 950
730 RESTORE LINCNT:READ A$,N
740 IF A$=LB$ THEN 760
750 LINCNT=LINCNT+STP:GOTO 730
760 L=LEN(A$):CENT=20-INT(L/2)
770 POSITION CENT,0:? A$:POSITION CE
    NT,1
790 FOR I=1 TO L:? "-";:NEXT I
800 POSITION 13,4:? "DISK DIRECTORY"
    :?
810 FOR I=1 TO N
820 IF I>=10 THEN SP$=" "
830 IF I=17 OR I=34 OR I=51 THEN 930
840 READ A$
850 IF A$(4,5)=" F" THEN 910
860 IF A$(1,2)<>"* " THEN 890
870 PRINT TAB$;SP$;I;" ";A$
880 GOTO 900
890 PRINT TAB$;SP$;I;"{3 SPACES}";A$
900 NEXT I
910 ? :? TAB$;"{3 SPACES}";A$:GOTO 9
    40
930 ? :? "PRESS---> RETURN TO CONTIN
    UE";:INPUT R$:? "{CLEAR}":GOTO 8
    40
940 ? :? :? "PRESS---> RETURN TO CON
    TINUE";:INPUT R$:RETURN
950 TRAP 40000
960 ? CHR$(125):? :? :? "DISK ---> "
    ;LB$;"{5 SPACES}NOT FOUND"
970 ? :? "PRESS RETURN TO CONTINUE--
    >";:INPUT A$:RETURN
980 REM ** FILE SEARCH ROUTINE **
990 ? CHR$(125):POSITION 2,2:? "SEAR
    CH WHICH FILE-->";:INPUT F$
1000 LINCNT=2000:? CHR$(125)
1010 RESTORE LINCNT:TRAP 1150
1020 READ A$,N
1030 PRINT "SEARCHING DISK---> ";A$:
     ?
1040 FOR I=1 TO N
1050 READ A$
1060 IF A$(1,2)<>"* " THEN 1090
1070 IF A$(3,LEN(F$)+2)=F$ THEN PRIN
     T "FILE LOCATED---> ";A$:? :FLA
     G=1
1080 GOTO 1100
1090 IF A$(1,LEN(F$))=F$ THEN PRINT
     "FILE LOCATED---> ";A$:? :FLAG=
     1
1100 NEXT I
1120 LINCNT=LINCNT+STP:GOTO 1010
1140 GOTO 1020
1150 IF FLAG THEN 1170
1160 ? CHR$(125):POSITION 8,16:? "--
     NO MATCH FOUND--"
```

```
1170 FLAG=0:? :? "LIST EXHAUSTED ---
     ->RETURN";:INPUT R$:RETURN
1180 REM ** DISK LABEL ROUTINE **
1190 ? CHR$(125):POSITION 8,4:? "LOA
     D PRINTER WITH LABELS"
1200 POSITION 7,6:? "PUT 850 INTERFA
     CE-ON LINE-"
1210 POSITION 10,7:? "PUT PRINTER-ON
      LINE-"
1220 POSITION 8,10:? "PRESS RETURN W
     HEN READY";:INPUT A$
1230 TRAP 1520:ROW=0:COL=0:INC=12
1240 CLOSE #2:OPEN #2,8,0,"P:"
1250 LPRINT CHR$(27);"@":GOTO 1280
1260 PUT #2,27:PUT #2,51:PUT #2,18
1270 ? #2;CHR$(15);:? #2;CHR$(27);CH
     R$(83);:RETURN
1280 ? CHR$(125):POSITION 2,4:? "PRI
     NT LABELS FOR WHICH DISK ":? :?
      "---> ";:INPUT LB$
1300 LINE=FIRST:TRAP 1530
1310 RESTORE LINE:READ A$,N
1320 IF A$=LB$ THEN 1340
1330 LINE=LINE+STP:GOTO 1310
1340 PUT #2,27:PUT #2,71
1350 ? #2;" DISK= ";A$
1360 PUT #2,27:PUT #2,64
1370 GOSUB 1260:PRINT #2
1380 ROW=ROW+3
1390 FOR I=1 TO N
1400 READ A$
1410 IF A$(4,5)=" F" THEN 1480
1420 IF A$(1,2)<>"* " THEN 1450
1430 ? #2;A$;"   ";:COL=COL+1:ROW=ROW
     +1:IF COL=3 THEN PRINT #2:COL=0
1440 GOTO 1460
1450 ? #2;"   ";A$;"   ";:COL=COL+1:RO
     W=ROW+1:IF COL=3 THEN PRINT #2:
     COL=0
1460 IF ROW=24 THEN ROW=0:PRINT #2:P
     RINT #2
1470 NEXT I
1480 PRINT #2;"   ";A$
1490 SKIP=INT(ROW/3)
1500 FOR I=1 TO INC-SKIP:PRINT #2:NE
     XT I
1510 RETURN
1520 TRAP 40000:GOTO 1190
1530 TRAP 40000:GOTO 1280
1540 REM ** EXIT TO BASIC & SAVE UPD
     ATED PROGRAM **
1550 ? CHR$(125):POSITION 3,8:? "INS
     ERT DISK CONTAINING MASTER.DIR"
     :POSITION 14,10:? "IN DRIVE #1"
     :POSITION 3,13
1560 ? "PRESS ---> RETURN{5 SPACES}W
     HEN READY";:INPUT A$
1570 ? CHR$(125):? :? "SAVE ";CHR$(3
     4);"D:MASTER.DIR"
1580 GOSUB 470
1590 ? "{CLEAR}":POSITION 12,4:? "UP
     DATE INSTALLED"
1600 FOR PAUSE=1 TO 500:NEXT PAUSE
1610 RETURN
1620 GRAPHICS 0:NEW
```

©

Runway 180

Using Sprites In TI Extended BASIC

James Dunn

The efficient, remarkable sprite-handling ability of TI Extended BASIC is clearly evident in this game. The author discusses creating sprites and explores sprite manipulation. There are several valuable pointers here for those interested in graphics, animation, or game programming on the TI.

Your plane is on final approach. "Runway 180," TI version.

Using Sprites In TI Extended BASIC

One of the biggest problems in designing an arcade-type game in BASIC is that BASIC can move only one character at a time, usually slowly and usually not very smoothly. Ideally, we need the ability to move an object independently of the operation of the main program. Once set in motion, the object would continue in motion until acted upon by a new command from the main program. Sprites accomplish this.

Although a sprite is a type of subprogram that runs concurrently with a main program, the main program first must create the sprite, define its shape, and set it in motion. A sprite then continues its motion without requiring continuous control from the main program, except that the main program may at any time test the sprite for position, change the color or pattern, delete, or change its motion.

Included in TI-99/4A Extended BASIC are 11 commands to control sprites: CALL COLOR, CALL CHAR, CALL SPRITE, CALL PATTERN, CALL MAGNIFY, CALL MOTION, CALL POSITION, CALL LOCATE, CALL DISTANCE, CALL COINC, and CALL DELSPRITE. To illustrate the use of these commands, we'll look at an airplane landing game, "Runway 180." Try some examples for yourself to get a feel for sprite programming.

Creating Sprites

Certain considerations must be taken into account before sprites are created. If a special graphics character is to be used for the sprite, the character must be created by use of CALL CHAR. For example, in the game there are three special characters defined for the aircraft. One is with the wheels up (lines 430)-460), one is with the wheels down (lines 510-540), and one is debris after a crash (lines 550-580).

To create a special character, it is necessary to redefine an existing standard character. The standard characters correspond to the numbers 30 through 143 (part of what's called the ASCII number code). The new pattern is created by using CALL CHAR and is referenced by its ASCII number.

Before we choose which ASCII number to use, we must examine some other factors. CALL MAGNIFY can enlarge a sprite to one of four magnification factors. Factor four is used in the game (line 630). This enlarges the sprites to double-size pixels and uses a block of four sequential characters. The ASCII number used to define the sprite must be evenly divisible by four and represents the upper-left character in the block of four. The next three ASCII numbers represent the lower-left, upper-right, and lower-right characters respectively in the block of four.

The sprite may be colored independently of the other characters in the same character set. In addition, the sprite with the lower sprite number (this is a different number than the ASCII number) will pass in front of (that is, *over*) the higher numbered sprite. Since the aircraft should pass in front of the tower, it should have a lower sprite number for each of its three configurations (line 610).

To set up a list of sprites, first number the lines on a sheet of paper from 30 to 143. Then, beside each number, write what set it belongs to (set 0 to 14). Since you may want to use letters or numbers in a screen display at the same time, mark out ASCII numbers 48 through 57 and 65 through 90. The remaining ASCII numbers can be used to define special characters for graphics and sprites.

For sprites, using CALL MAGNIFY (4), select four sequential numbers starting at one of the numbers evenly divided by four. Now you are ready to use CALL SPRITE.

CALL CLEAR will not remove a sprite from the screen. To completely clear the screen, you must also use CALL DELSPRITE (line 1350).

Sprites In Motion

Now that the sprite has been created, there are two ways of moving it around the screen. Let's call these two methods *absolute* and *relative*. The absolute method uses exact row and column positions via the CALL LOCATE command. The relative method uses row and column motion values via the CALL MOTION command.

The absolute method uses a loop with CALL JOYST to increment row and column variables, and then a CALL LOCATE to move the sprite one step each time the loop is executed. This is analogous to nonsprite methods of animation. The drawback in using this method is that the sprite does not move independently; the main program causes the move. A modified form of this method is used for the stall subroutine (line 1470) and the new approach routine (line 1380).

The relative method is similar, using a loop with CALL JOYST to increment row and column *motion* variables which are used in a CALL MOTION command. This allows the sprites to continue moving independently of the main program. By this method, the runway stripe is moved horizontally only (line 680) and the aircraft vertically only (also line 680).

The sprite's shape may be changed anytime during the program by using CALL PATTERN to substitute a different ASCII character number and therefore a different pattern. When the fire button is depressed (line 1130), the aircraft landing gear comes down (line 1190). The pattern is changed again if the aircraft crashes (line 1720).

Testing For Game Conditions

During the operation of the program, it may become necessary to test for certain conditions. For example, we see if the aircraft has touched down on the runway (line 690), if the tower has reached the left side of the screen (line 700), or if the aircraft is going off the top of the screen (line 710). CALL COINC is used to test for these conditions.

However, there is a problem with this method. Since the main program tests for coincidence only when CALL COINC is executed and since the sprite moves independently of the main program, it is quite possible to miss an exact coincidence when it occurs. For this reason a tolerance factor is included in CALL COINC. So the test is really for a range of + or - tolerance. If the tolerance is too large, coincidence can be returned too early. If the tolerance is too small, coincidence can be missed altogether. How large the tolerance should be depends upon two things: the speed of the sprite and the speed of the loop which is testing for coincidence.

The test for the tower reaching the left side of the screen is in both the main loop (line 700) and the stall loop (line 1480). The tolerance in the stall loop is much smaller because the execution speed is so fast and the sprite moves so slowly that coincidence is actually read twice before the sprite leaves the tolerance range. Trial and error is the only way to find out how large the tolerance should be.

However, after programming this game, it is obvious that very fast-moving sprites will require tolerance ranges that will make arcade-style, fast-action games nearly impossible in Extended BASIC. The problem is that the coincidence test is executed from the main program. If it were part of the sprite subprogram instead, it would be possible to keep the tolerance very small.

CALL POSITION and CALL DISTANCE both suffer from the same problem as CALL COINC. By the time a position or distance can be computed and returned to the main program, the sprite has moved elsewhere. But it is possible to stop the sprite by using a CALL MOTION before using CALL POSITION or CALL DISTANCE (line 1330), then to restart whatever motion is required.

Despite a few shortcomings, the sprite capabilities in Extended BASIC are remarkable. For true arcade-type play, machine language is still necessary, but Extended BASIC sprites will carry the programmer a lot closer to this goal.

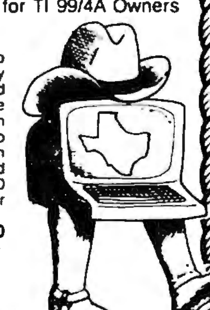
Runway 180

```
130 CALL CLEAR :: CALL SCREEN(5):: CALL COLOR(1,16,1,2,16,1,3,16,1,4,16,1,5,16,1,6,16,1,7,16,1,8,16,1)
140 DISPLAY AT(10,9):USING "RUNWAY 180"
150 FOR B=0 TO 30 STEP 2 :: CALL SOUND(-10,110,30,110,30,2500,30,-8,B):: CALL SOUND(-10,110,30,110,30,4000,30,-8,B):: NEXT B
160 CALL CLEAR :: DISPLAY AT(10,9):USING "PRESS" :: DISPLAY AT(12,9):USING "I-FOR INSTRUCTIONS"
170 DISPLAY AT(14,14):USING "OR" :: DISPLAY AT(16,9):USING "G-FOR GAME"
180 CALL KEY(0,K,S):: IF S<>1 THEN 180
190 IF K=71 THEN 330
200 IF K=73 THEN 220
210 PRINT "ALPHA LOCK MUST BE ON" :: PRINT :: PRINT "TRY AGAIN" :: FOR DELAY=1 TO 200 :: NEXT DELAY :: GOTO 160
220 CALL CLEAR :: PRINT "YOU ARE PILOTING A JET" :: PRINT :: PRINT "AIRCRAFT WHICH HAS BEEN " :: PRINT :: PRINT "CLEARED TO LAND ON": :
230 PRINT "RUNWAY 180." :: PRINT :: PRINT :: GOSUB 310
240 CALL CLEAR :: PRINT "USE YOUR JOYSTICK TO CONTROL" :: PRINT :: PRINT "SINK RATE AND AIRSPEED.": :
243 PRINT "JOYSTICK CONTROL-" :: PRINT
245 PRINT "LEFT: ACCELERATE" :: PRINT "RIGHT: BRAKE" :: PRINT "UP: DECREASE SINK RATE"
247 PRINT "DOWN: INCREASE SINK RATE" :: PRINT
250 PRINT "FIREBUTTON CONTROLS LANDING" :: PRINT :: PRINT "GEAR." :: PRINT :: PRINT :: GOSUB 310 :: CALL CLEAR
260 PRINT "TO RECOVER FROM A STALL" :: PRINT :: PRINT "INCREASE AIRSPEED ABOVE 60." :: PRINT :: PRINT "IF YOU CANNOT STOP BEFORE": :
270 PRINT "TOWER REACHES LEFT SIDE OF" :: PRINT :: PRINT "SCREEN, INCREASE AIRSPEED" :: PRINT
280 PRINT "TO 60 AND LIFT OFF FOR " :: PRINT :: PRINT "ANOTHER PASS." :: PRINT :: PRINT :: GOSUB 310 :: CALL CLEAR
290 PRINT "YOU MAY HAVE FOUR PASSES" :: PRINT :: PRINT "AT THE RUNWAY....." :: PRINT :: PRINT "BEWARE OF THE WIND SHIFTS!" :: PRINT :: PRINT
300 PRINT "GOOD LUCK!!!!" :: PRINT :: PRINT :: PRINT :: PRINT :: GOSUB 310 :: GO TO 330
310 PRINT :: DISPLAY AT(24,1):USING "HIT ANY KEY TO CONTINUE"
320 CALL KEY(0,R8,S8):: IF S8<>1 THEN 320 ELSE RETURN
330 A1=1
340 REM INITIALIZE
350 A=0 :: B=-75 :: LG=0 :: CALL SCREEN(2)
360 CALL CLEAR :: CALL CHAR(33,"FFFFFFFFFFFFFFFF"):: CALL COLOR(1,8,1)
370 LC=0 :: FOR Z=1 TO 16 :: CALL HCHAR(Z,1,33,32):: NEXT Z
380 CALL CHAR(42,"FFFFFFFFFFFFFFFF"):: CALL COLOR(2,13,1)
390 FOR Z=17 TO 20 :: CALL HCHAR(Z,1,42,32):: NEXT Z
400 RANDOMIZE
410 REM DEF CHAR
420 CALL CHAR(96,"00000000FFFFFFFFFFFFFFF0000000000000000FFFFFFFFFFFFFFFF")
430 CALL CHAR(120,"0030181C3F1F0700")
440 CALL CHAR(121,"000000")
450 CALL CHAR(122,"00000000FCFF8000")
460 CALL CHAR(123,"00000000")
470 CALL CHAR(104,"00000000071F151F")
480 CALL CHAR(105,"0203030203030203")
490 CALL CHAR(106,"00008080E0F8A8F8")
500 CALL CHAR(107,"C040C0C040C0C0C0")
510 CALL CHAR(124,"0030181C3F1F07050000")
520 CALL CHAR(126,"00000000FCFF88840000")
530 CALL CHAR(125,"00000000")
540 CALL CHAR(127,"00000000")
550 CALL CHAR(128,"0000000021F3B00")
560 CALL CHAR(129,"00000000E56E300")
570 CALL CHAR(130,"00000000")
580 CALL CHAR(131,"00000000")
590 REM DRAW DISPLAY
600 CALL SPRITE(#1,96,2,180,1,0,B):: CALL COLOR(#1,16)
610 CALL SPRITE(#2,120,2,10,245,A,0):: CALL COLOR(#2,7)
620 CALL SPRITE(#3,104,2,110,250,0,-2)
630 CALL MAGNIFY(4)
640 FOR C5=1 TO 40 :: CALL LOCATE(#2,10,C5):: NEXT C5 :: GOSUB 870
650 REM MAIN LOOP
660 GOSUB 1120 :: GOSUB 890
670 IF J=0 THEN 690
680 CALL MOTION(#1,0,B,#2,A,0)
690 CALL COINC(#2,170,40,9,T)
700 CALL COINC(#3,110,1,4,DA)
710 CALL COINC(#2,240,40,9,E):: IF E=-1 THEN A=1 :: GOSUB 890 :: GOTO 680
720 IF DA=-1 THEN 1320
730 IF T<>-1 THEN 660
740 CALL MOTION(#2,0,0)
750 IF A>1 THEN GOSUB 920 :: GOSUB
```

```
   960 :: GOTO 1660
760 IF LG=0 THEN 1660
770 GOTO 1760
780 REM UPDATE DISPLAY
790 IMAGE SINK RATE: ###
800 IMAGE RUNWAY ENDS ### YDS
810 IMAGE AIRSPEED: ###
820 IMAGE TOUCH DOWN
830 IMAGE SINK RATE TOO HIGH
840 IMAGE AIRSPEED TOO HIGH
850 IMAGE CRASH LANDING
860 IMAGE STALL WARNING!
870 DISPLAY AT(1,10)SIZE(20):USING
    "ATTEMPT NO. #":A1
880 RETURN
890 DISPLAY AT(3,10)SIZE(20):USING
    790:A
900 DISPLAY AT(5,10)SIZE(20):USING
    810:-B
910 RETURN
920 DISPLAY AT(7,5)SIZE(20):USING 8
    30
930 RETURN
940 DISPLAY AT(7,5)SIZE(20)BEEP:USI
    NG 840
950 DISPLAY AT(9,5)SIZE(20):USING "
    BOUNCE" :: RETURN
960 DISPLAY AT(9,5)SIZE(20):USING 8
    50
970 RETURN
980 CALL HCHAR(7,5,33,27):: DISPLAY
     AT(9,5)SIZE(20):USING 820
990 RETURN
1000 DISPLAY AT(9,5)SIZE(20):USING
     "WARNING "
1010 DISPLAY AT(11,5)SIZE(20):USING
      800:RE
1020 RETURN
1030 CALL HCHAR(7,5,33,27):: RETURN
1040 CALL HCHAR(9,5,33,27):: RETURN
1050 CALL HCHAR(11,5,33,27):: RETUR
     N
1060 DISPLAY AT(9,5)SIZE(20):USING
     "LIFT OFF" :: CALL HCHAR(11,5,
     33,27):: RETURN
1070 DISPLAY AT(3,10):USING "END OF
      RUNWAY " :: DISPLAY AT(5,10):
     USING "NEW APPROACH" :: DISPLA
     Y AT(7,10):USING "NECESSARY"
1080 RETURN
1090 PRINT "THAT'S 5 PASSES AT THE"
      :: PRINT :: PRINT "RUNWAY. TU
     RN IN YOUR" :: PRINT :: PRINT
     "PILOT LICENSE AND PUT": :
1100 PRINT "SOMEONE ELSE IN THE" ::
      PRINT :: PRINT "COCKPIT" :: P
     RINT :: RETURN
1110 DISPLAY AT(7,9)BEEP SIZE(20):U
     SING 860 :: RETURN
1120 REM JOYST/ LANDING GEAR
1130 CALL KEY(1,RV,ST):: IF RV=18 A
     ND LG=0 THEN 1190
1140 CALL JOYST(1,X,Y):: IF X=0 AND
      Y=0 THEN GOSUB 1210 :: RETURN
1150 A=A-Y/4 :: B=B+X/4
1160 IF ABS(A)>127 THEN A=127*SGN(A
     )
1170 IF B>-50 THEN 1430
1180 J=1 :: RETURN
1190 CALL PATTERN(#2,124)
1200 A=A+3 :: B=B+20 :: LG=1 :: GOT
     O 1160
1210 REM COMPLICATIONS
1220 CP=INT(RND*16)
1230 IF CP=1 THEN B=B-1 :: GOTO 130
     0
1240 IF CP=6 THEN B=B+1 :: GOTO 130
     0
1250 IF CP=10 THEN A=A-1 :: GOTO 12
     80
1260 IF CP=15 THEN A=A+1 :: GOTO 12
     80
1270 J=0 :: RETURN
1280 IF ABS(A)>127 THEN A=127*SGN(A
     )
1290 GOTO 1310
1300 IF B<-127 THEN B=-127
1310 J=1 :: RETURN
1320 REM NEW APPROACH
1330 CALL MOTION(#2,0,0):: CALL POS
     ITION(#2,R4,C4)
1340 IF A1>4 THEN 1400
1350 CALL DELSPRITE(#1,#3):: CALL C
     LEAR
1360 GOSUB 1070
1370 CALL PATTERN(#2,120)
1380 FOR X=C4 TO 255 :: CALL LOCATE
     (#2,INT(R4),X):: R4=R4-(R4/(25
     5-C4)):: NEXT X
1390 A1=A1+1 :: GOTO 340
1400 CALL DELSPRITE(ALL):: CALL CLE
     AR
1410 GOSUB 1090
1420 FOR DELAY=1 TO 900 :: NEXT DEL
     AY :: GOTO 1970
1430 REM STALL
1440 GOSUB 1110
1450 CALL MOTION(#2,0,0)
1460 CALL POSITION(#2,SR,SC)
1470 CALL LOCATE(#2,SR,SC)
1480 CALL COINC(#2,170,40,2,T)
1490 CALL COINC(#3,110,1,2,DE):: IF
      DE=-1 THEN A1=A1+1 :: GOSUB 8
     70 :: IF A1>4 THEN 1400
1500 IF T=-1 THEN 1660
1510 SR=SR+4
1520 CALL KEY(1,RV,ST)
1530 IF RV=18 AND LG=1 THEN 1610
1540 CALL JOYST(1,X,Y):: IF X=0 AND
      Y=0 THEN 1470
1550 B=B+X/4
1560 REM
1570 IF B<-60 THEN 1640
1580 CALL MOTION(#1,0,B)
1590 GOSUB 890
1600 GOTO 1470
1610 CALL PATTERN(#2,120)
1620 A=A-3 :: B=B-22 :: LG=0
1630 GOTO 1560
1640 GOSUB 1030
1650 RETURN
1660 REM CRASH
1670 CALL MOTION(#1,0,0,#2,0,0,#3,0
     ,0,#4,0,0)
1680 CALL SOUND(1000,-7,0)
1690 FOR P=1 TO 10
1700 CALL SCREEN(2)
```

```
1710 CALL SCREEN(16):: NEXT P :: CA
     LL SCREEN(2)
1720 CALL PATTERN(#2,128)
1730 FOR DELAY=1 TO 400 :: NEXT DEL
     AY
1740 CALL DELSPRITE(ALL)
1750 GOTO 1970
1760 REM TOUCHDOWN/BRAKE/T&G
1770 GOSUB 980 :: IF B<-53 THEN 194
     0
1780 CALL JOYST(1,X,Y):: B=B+X/2
1790 IF B>-1 THEN 1880
1800 CALL MOTION(#1,0,B)
1810 CALL COINC(#3,110,1,4,DA)
1820 IF DA=-1 THEN RE=0 :: GOSUB 10
     10 :: GOTO 1660
1830 CALL DISTANCE(#3,110,1,RQ)
1840 RE=INT(SQR(RQ)):: GOSUB 1000 :
     : GOSUB 900
1850 CALL KEY(1,RV,ST):: IF RV=18 A
     ND B<-60 THEN GOSUB 1060 :: A=
     A-2 :: GOTO 187
0
1860 GOTO 1780
1870 CALL MOTION(#2,A,0):: FOR DELA
     Y=1 TO 200 :: NEXT DELAY :: GO
     TO 650
1880 REM SCORING
1890 CALL MOTION(#1,0,0,#2,0,0,#3,0
     ,0,#4,0,0)
1900 CALL DELSPRITE(ALL):: CALL CLE
     AR
1910 PRINT "CONGRATULATIONS !": :
1920 PRINT "YOUR SCORE IS :";(RE/A1
     )*10: :
1930 GOTO 1990
1940 A=A-2 :: CALL MOTION(#2,A,0)::
      GOSUB 940
1950 FOR DELAY=1 TO 20 :: NEXT DELA
     Y
1960 A=A+2 :: GOSUB 1030 :: GOSUB 1
     040 :: GOTO 650
1970 REM PLAY AGAIN
1980 CALL CLEAR
1990 PRINT "PLAY AGAIN (Y/N)?"
2000 CALL KEY(2,RV,SV)
2010 IF SV=0 THEN 2000
2020 IF RV=15 THEN 2050
2030 IF RV=18 THEN 330
2040 GOTO 1990
2050 END
```

©



How To Create A Data Filing System

Part 4: The Main Program

Jim Fowler

In the final installment of this series, the author looks at ways to approach the overall logic of a final system. Safeguards and auxiliary programs are also discussed.

Now you have most of the detail work on your data file system finished. You know what kind of files you want and how they are formatted. The nature of the output functions (and searches) has determined the coding and index files needed, and this has pretty much dictated the input part of the program. Now we're ready to put it all together.

Make It Modular

You probably already know the advantages of writing programs with lots of subroutines, each doing a single task. In data filing systems this advantage is particularly obvious. A subroutine to input a string from a particular device is much more useful than one to input a string from the keyboard and another to read a string from the disk. That subroutine can be used in other subroutines, to input an author's name, and again to input the title of a work, the date of publication, and so on. The subroutine to input an author's name and encode a part for the index file can also be used to input a key used to search for a particular author. This can go on and on. Whenever possible, make subroutines so all-purpose that they can be called throughout the program.

In the accompanying flowchart, I have illustrated the design for my author-subject file of books and articles. How you want your data displayed, what you want printed, and what you want on the screen will depend on your individual situation. Some people want a printout of their input as well. It is also easier for some people to proofread text on paper than on a screen, so customize it for your needs. Your requirements will differ from mine, so your flowchart will be different, too. However, you probably should use subroutines in a modular fashion.

Preventing Disaster

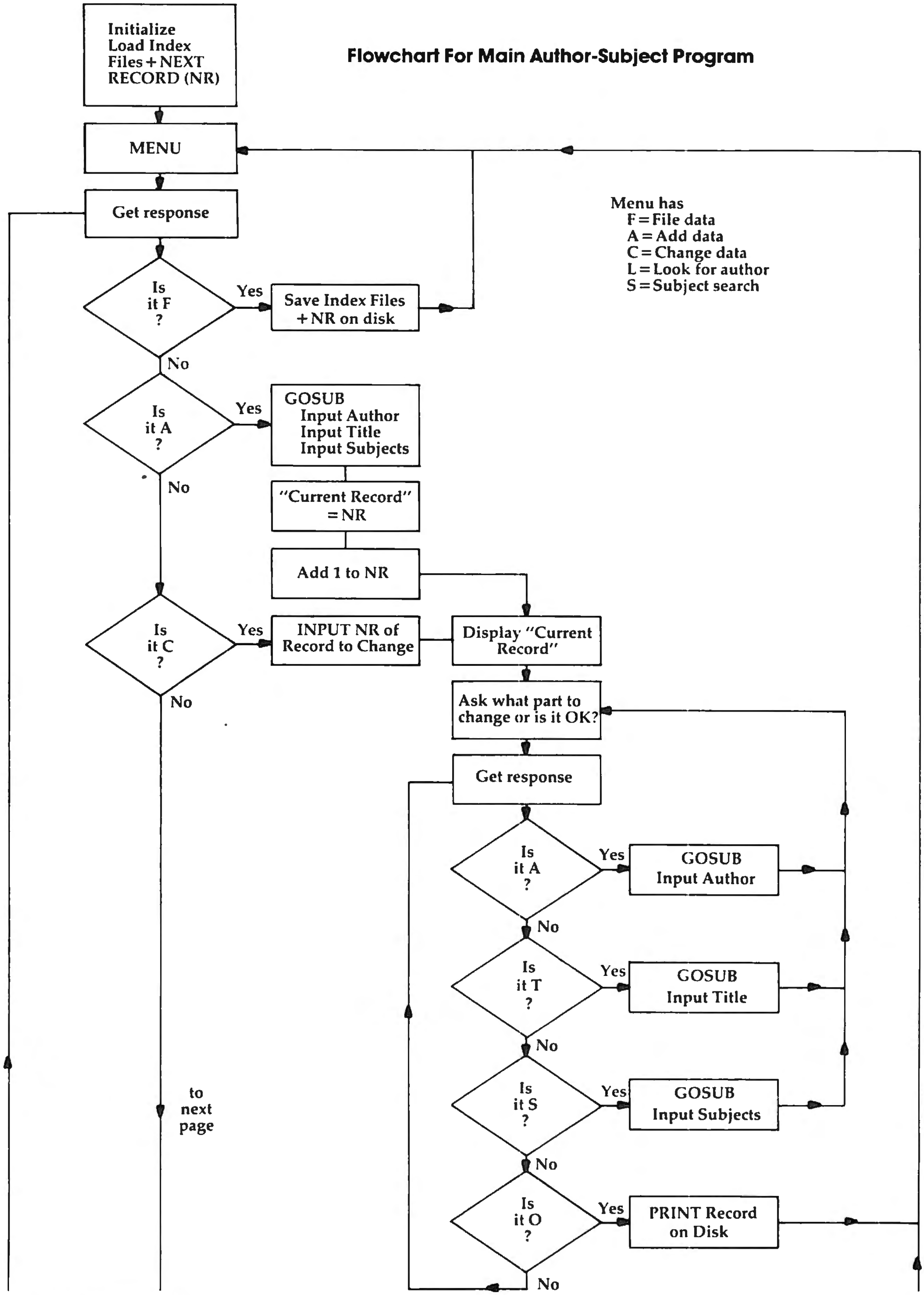
You should include fail-safe methods to prevent disastrous errors. For instance, suppose you have just finished entering a hundred records and you turn off the system without saving the index files. This disaster breaks down into two problems: reminding the user to save the file before quitting, and reconstructing the lost files from the data on disk in the main records. Both are easy to solve, but you must solve them – preferably in advance. Even if your method of reconstructing files is crude or your warning to the user lacks elegance, the important thing is to have these provisions in the program.

You cannot prevent certain disasters, although you can reduce the seriousness of the damage. These include a power interruption, hardware failure, or a bad spot on a disk. To minimize the damage from these troubles, you need good operating procedure. For data files, this means making backup copies frequently. You could, for example, make backup copies after every twentieth entry into the file, then put a counter in the program. When it "goes off," have the program tell you to insert a disk into drive X and "press return" – and there is your backup.

Satellite Programs

You may find, as I did, that you will need one or more other programs to augment your data file system. For example, you probably will have to write a program to prepare the disk for the records to be written. It should allocate (and fill with nulls) perhaps one thousand relative records. These nulls (zeros) are then replaced as real data is written into the system. A program that does this is a *satellite program.* It is not part of the main system

Flowchart For Main Author-Subject Program



COMMODORE 64 \$189.95*

COMMODORE 64 Computer—only \$189.95*
*when purchased with any of these three packages

COMMODORE 64			\$189.95
	with the purchase of		
1	1541	Disk Drive	\$299.95
1	1525E	Printer	\$289.95
		ALL FOR	\$779.85

COMMODORE 64			\$189.95
	with the purchase of		
1	1541	Disk Drive	\$299.95
1	1701	14" color monitor	\$299.95
		ALL FOR	\$789.85

COMMODORE 64			\$189.95
	with the purchase of		
1	1541	Disk Drive	\$299.95
1	1526	Fast printer-includes IFC/cable direct connect to 64	\$349.95
		ALL FOR	\$839.85

COMMODORE VIC-20 \$74.95*

1	*with the purchase of	
1	Datasette program recorder	\$69.95
	Gortek educational software	\$24.95
	All for	\$169.85

	COMMODORE 64 Computer	\$239.95
1541	Disk drive	279.95
1525E	Printer/direct connect to 64	289.95
1626	Printer/direct connect to 64	339.95
1701	14" color monitor	289.95
1530	Datasette program recorder	79.95
1600	Telephone modem for VIC-20/Com. 64	69.95
1650	Modem, new-auto. dial/answer	109.95

Software for Commodore 64 & VIC-20 at discount! Call us!

DATA 20 Accessories

• Z-80 card	For Commodore 64	\$269.95
• 80 column card	For Commodore 64	169.95
• 40 column card	For VIC-20	99.95
• 40 column card	w/8K mem built-in for VIC-20	139.95
• Video Pack 64K	For VIC-20	329.95

*FREE with the purchase of each of the above (*) products—Word Master/word processing software, Mailing List software & Telecommunications software.

NEW ROYAL 64K ALPHATRONIC Computer

CP/M, color graphics, 3 video outputs—RGB for hi res 80 col color displays composite video:use with any regular monitor; modulated RF for use with your TV, Centronic printer port, 40/80 col display, KC std audio cass. port RE:\$695.00 Your cost: \$489.95

Optional 320 5¼" disk drive, \$379.95

TI COMPACT COMPUTER CC-40 only \$199.95

Most advanced portable, fits in your briefcase!
Will perform as well as the \$1000 unit!
One set of A batteries = 200 hours of operation (appx)

HX-1000 Printer/plotter	\$179.95
HX-2000 Wafertape (digital tape drive)	\$124.95
RS-232 Interface	89.95
HX-3000 RS-232 int. w/parallel IFC	109.95

Solid state software/wafertape software-call us!

Olympic Sales Company

SERVING YOU SINCE 1947

Telex 67 34 77 Toll-Free Phone Orders:
Toll-free *(in CA)* 800-252-2153 800-421-8045 (out of CA)
Order Desks open 6 days a week! 7:00 AM to 6:00 PM Mon-Sat
P.O. Box 74545 216 So. Oxford Ave. Los Angeles, CA 90004
Phone: (213) 739-1130 Cable "OLYRAV" LSA

DISKETTES	In packs of 10	
Memorex 5¼"	SS/DD	\$24.85
Verbatim	SS/DD	29.45
Maxell	SS/DD	33.95

VISA

We carry close to \$5,000,000 inventory at all times. Corp. accts. invited.

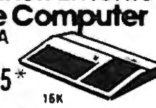
Good subject to availability, this ad supercedes all previous ads, fob our warehouse, prices subject to change without notice, not responsible for typographical errors, all orders subject to verification, minimum shpg & hndlg \$5.95. Send \$2 (for \$5 foreign) for our famous catalog

COMPUTER PRINTERS & MONITORS

Diablo 620	Letter quality, 25 cps	\$ 999.95
Diablo 620	Letter quality, 40 cps	1799.95
NEC 8023A	100 cps with tractor	499.95
Transtar	Color printer, 30 shades, 50 cps	499.95
Okidata 82A	Graphics 120 cps bi-directional	424.95
Okidata 83A	Par/Ser., 120 cps, friction/tractor	664.95
Okidata 92P	160 cps, graphics, bi-directional	539.95
Okidata 93P	160 cps, graphics, frict/tract, letter quality	889.95
Star Gemini 10	100 cps, 2 3K buffer, Epson comp.	369.95
Star Gemini 15	100 cps, 15", 136 col, 2.3K buffer	489.95
Zenith	12" green monitors, good resolution	99.95
Sanyo	15" b/w monitor, high res-below cost!	169 95
IDS	Model 480	529.95
IDS	80 column color printer/all options	1595.00
IDS	132 columns color/all options	1695.00

Texas Instruments Home Computer

TI-99/4A



\$99.95* 16K

NET after \$50 rebate from TI
You pay us \$149.95!

Peripheral expansion box	\$199.95
RS-232 card	139.95
Disk controller card	199.95
Expansion Sys. disk drive	319.95
Memory expan. card (32K)	229.95
P Code card (req. exp. card)	199.95
Telephone modem	159.95
Color monitor	339.95
Extended Basic	79.95
LOGO	89.95

HEWLETT PACKARD

HP-75C	Handheld comp.	\$719.95
HP-10C	Scientific calc	59.95
HP-11C	Scientific calc	76.90
HP-12C	Financial calc	99.95
HP-15C	Scientific calc	99.95
HP-16C	Prog. sce scien. calc	99.95
HP-97	Prog. scien. w/print	595.00
HP-41C	Prog. calculator	169.95
HP-41CV	Prog. calculator	219.95

DON'T BUY ANY PORTABLE COMPUTER UNTIL YOU SEE THE NEW COMMODORE!

* * * COMMODORE'S SX-100 PORTABLE! * * *

COMPLETELY PORTABLE!

- Full 64K
- Color Monitor built-in
- Disk drive built-in
- FREE software package
- Completely compatible with the Commodore 64
- LOTS OF SOFTWARE AVAILABLE!

\$995.00

SANYO

NEW FABULOUS SANYO COMPUTER
MBC-555 IBM PC Compatible, uses IBM software, 8088 CPU, 128K memory, 160K disk storage, color graphics, centronic port, 80 column, MS/DOS Basic, diagnostics, utilities, speaker, joystick port, word processing & spread sheet software included.
Retail: \$995.00 call for best price! Optional:
128K RAM expansion; 320K to 640K drives,8087 number cruncher, RS-232 hard disk available & more!

SOFTWARE for ALL MICROCOMPUTERS

3000 titles available at great prices

1-2-3 Lotus for IBM	\$397.00
PFS:File	114.00
PFS Report	99.95
Multiplan	219.00
D Base II	529.00
Wordstar and more	397.00
Recreation— and more	
Zaxxon	37.95
Miner: 2049er	36.95
Frogger	34.95
Deadline	46.95
Zork II	39.95
Zork III	39.95

EPSON

MX-80FT \$449.95

FX-80	160 cps	call for price
RX-80	100 cps	call for price
MX-100		call for price

TIMEX

TS-1000 \$44.95

16K RAM	45.95
Printer by Timex	89.95

Call us for Timex software—buy 4 software/get 1 software free!

brother

Personal Electronic Typewriter
EP-20 Incl. batt
FITS IN A BRIEFCASE - AC/DC
\$159.95

COMPUTER CASSETTES

100% Error-Free • Fully Guaranteed

LENGTH	12 PACK	24 PACK
C-05	79¢	69¢
C-10	89¢	79¢
C-20	99¢	89¢
Boxes	26¢	21¢

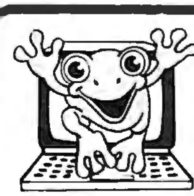
UPS \$3.00 Pkg. \$18.00 Case

C-10's 39¢
(Min. 500 Case Lot)
w/labels ADD 4¢,
w/boxes ADD 13¢

FOR ORDERS ONLY
1-800-528-6050
Extension 3005

VISA

MICRO-80™ INC.
2665-C Busby Road
Oak Harbor, WA 98277



FREE

Educational Software Catalog for Your Family

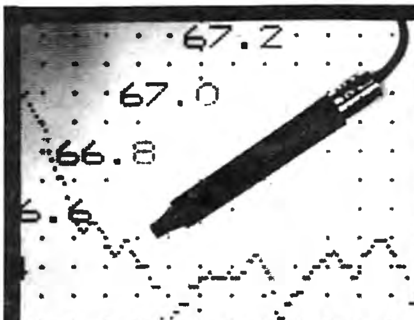
Just Released! Sunburst's new catalog, featuring the best home educational software . . . for all ages . . . selected from the nation's top publishers. Reading, math, programming, problem solving and more . . . to motivate your children and make learning come alive.

Call Toll-Free 800-431-1934

☐ RUSH MY FREE CATALOG

Name ______________________
Address ______________________
City ______________________
State ____________ Zip ____________
Type of computer ______________________

SUNBURST
Dept YK 7
39 Washington Ave.
Pleasantville, NY 10570



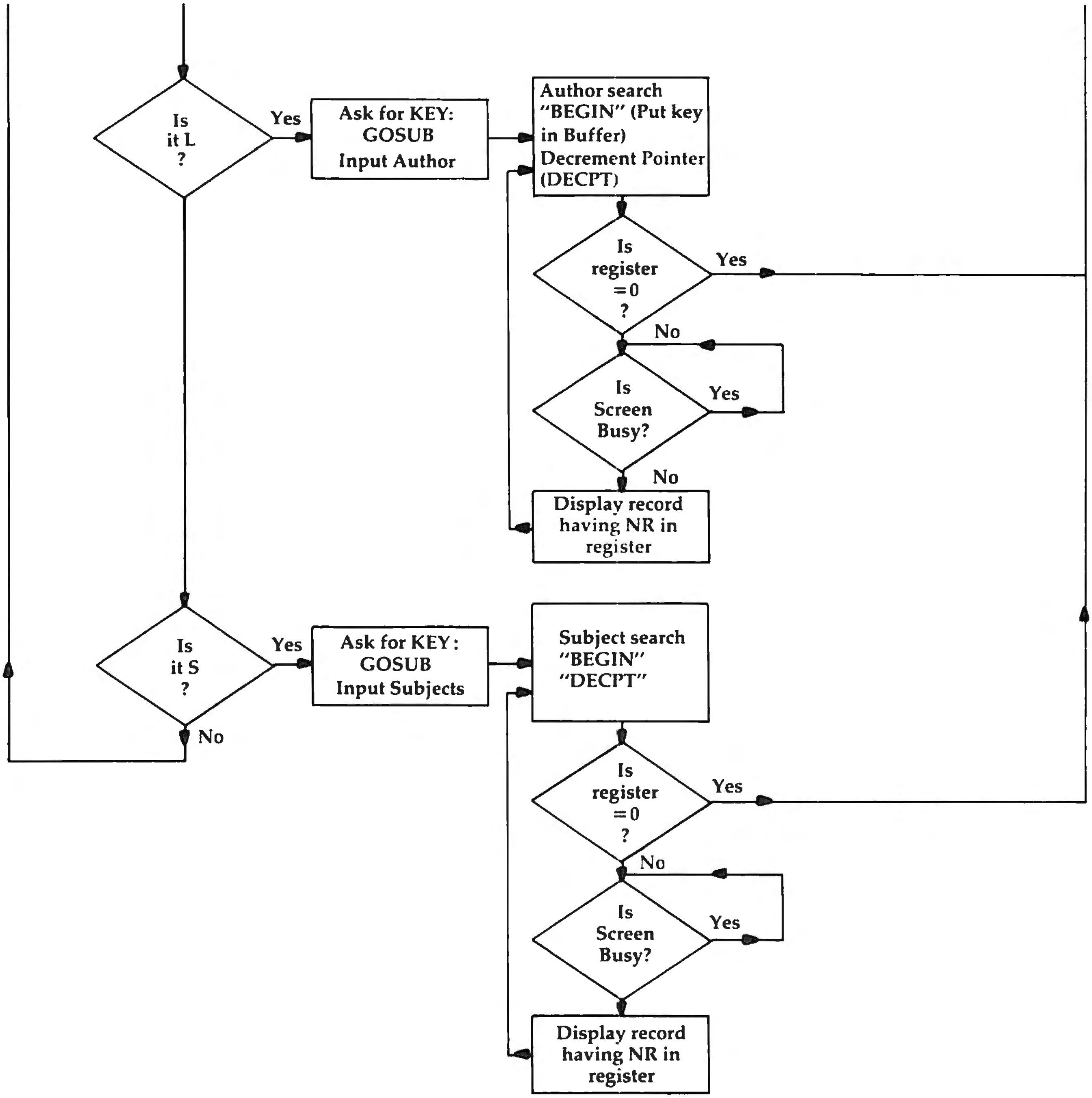
APPLE / ATARI / COMMODORE

MEASURE & CONTROL TEMPERATURE

DISPLAY GRAPHICS
HARD COPY OUTPUT
ALARMS & SETPOINTS
DISK FILE DATA STORAGE

1–256 Sensors
Precise to 1/100 Degree
Complete Software
\$129.00 Package

American Data Cable, Inc.
2864 Ray Lawyer Drive, #205
P.O. Box 2212 • Placerville, CA 95667
(916) 622-3465



program, but has no function except in that system. You can simply include this program on the same disk as the main program where it is handy.

I had to create a satellite program when my main data file forced me to make a second data system. Many of the records I entered were titles of magazine or journal articles. There is no point in spelling out *The Journal of Embryology and Experimental Morphology* when everybody in the business knows it as "JEEM." Every periodical has an official abbreviation, but how to remember them all? I had to make a dictionary of journal names and their abbreviations. Of course, that meant another data file. If I had had the foresight, I could have incorporated the dictionary into the main system. Fortunately, my half-megabyte disks have lots of room, but I really do not need a second system with its files and program when it could be ancillary to the main one. Maybe you will think far enough ahead and avoid the rather clumsy solution I had to adopt.

As we've stressed throughout this series, ingenuity, careful planning, and foresight are the key ingredients to a good system. Although it may be frustrating at times, writing the system is almost as worthwhile as using it. ©

HOFACKER

Books + Software for ATARI VIC-20 OSI SINCLAIR TIMEX

PAYMENT: check, money order, VISA, MASTER CARD, Eurocheck, ACCESS, Interbank
Prepaid orders add $3.50 for shipping (USA)
$5.00 handling for C.O.D.
All orders outside USA: add 15 % shipping, California residents add 6.5 % sales tax.

ATARI is a registered trademark of ATARI Inc.
VIC-20, CBM are registered trademarks of Commodore
APPLE is a registered trademark of APPLE Computer, Inc.

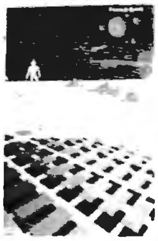
ELCOMP PUBLISHING, INC
53 Redrock Lane
Pomona, CA 91766
Phone: (714) 623 8314

PRODUCTS FOR ATARI* 400/800 FROM ELCOMP

BOOKS for ATARI Computers

ATARI BASIC – Learning by Using
An excellent book for the beginner. Many short programs and learning exercises. All important features of the ATARI computers are described (screen drawings, special sounds, keys paddles, joysticks, specialized screen routines, graphics, sound applications, peeks, pokes, and special stuff). Also suggestions are made that challenge you to change and write program routines.
Order #164 $7.95

Games for the ATARI Computer
This book describes advanced programming techniques like player-missile-graphics and use of the hardware-registers. Contains many ready to run programs in BASIC and one called GUNFIGHT in machine language.
Order #162 $7.95



How to program your ATARI in 6502 Mach.Lang.
Introduction to machine language for the BASIC programmer
Order #169 $9.95

FORTH on the ATARI – Learning by Using
Order #170 $7.95
Order No. 7319 $22.00 only!

A Look into the Future – ASTROLOGY
Order #171 Incl. listing of the program $9.95

Our catalog is free with every order. Send $1.00 and SASE for catalog only.

SUPERMAIL
500 addr. on 1 disk! Completely written in FORTH. Comes ... disk ... cartridge DOS ... FORTH language required!
Order# 7312 $49.00

SUPERINVENTORY
... items per disk. Completely written in FORTH. Same as ... (Disk only)
Order# 7320 $49.00

BUSIPACK-1
(written in FORTH) Complete order entry, inventory mailing and invoicing. (Disk only)
Order #7313 $98.00

ATAMEMO
Datablock to keep track of your appointments (D+C)
Order #7310 $29.95

ATCASH
Convert your ATARI ... into a powerful cash register. (Disk only)
Order #7307 $49.95

Invoicing progr. i. BASIC
Order #7201 (C) $29.95
Order #7200 (D) $39.95

Mailing List in BASIC
Order #7212 (C) $19.95
Order #7213 (D) $24.95

Inventory control in BASIC
Order #7214 (C) $19.95
Order #7215 (D) $24.95

Microcomputer Hardware Handbook (845 pages)
Descriptions, pinouts and specifications of the most popular microprocessors and support chips.
A MUST for the hardware buff.
Order-No. 29 $14.95

SOFTWARE IN MACHINE LANGUAGE for ATARI

ATMONA-1
This is a machine language monitor that provides you with the most important commands for programming in machine-language. Disassemble, dump (hex and ASCII), change memory location, block transfer, fill memory block, save and load machine-language programs, start programs. Printer option via three different interfaces.

Order #7022	cassette version	$19.95
Order #7023	disk version	$24.95
Order #7024	cartridge version	$59.00

ATMONA-2
This is a tracer (debugger) that lets you explore the ATARI RAM/ROM area. You can stop at previously selected address, opcode, or operand. Also very valuable in understanding the microprocessor. At each stop, all registers of the CPU may be changed. Includes ATMONA-1.

Order #7049	cassette version	$49.95
Order #7050	disk version	$54.00

ATMAS
Macro-Assembler for ATARI-800/48k. One of the most powerful editor assemblers on the market. Versatile editor with scrolling. Up to 17k of source-Code. Very fast, translates 5k source-code in about 5 seconds. Source code can be saved on disk or cassette. (Includes ATMONA-1)

Order #7099	disk version	$89.00
Order #7999	cartridge version	$129.00

ATAS
Same as ATMAS but without macro-capability. Cassette-based

Order #7098	32k RAM	$49.95
Order #7998	48k RAM	$49.95

ATEXT-1
This wordprocessor is an excellent buy for your money. It features screen oriented editing, scrolling, string search (even nested), left and right margin justification. Over 30 commands. Text can be saved on disk or cassette.

Order #7210	cassette version	$29.95
Order #7216	disk version	$34.95
Order #7217	cartridge version	$69.00

GUNFIGHT
This game (8k machine-language) needs two joysticks. Animation and sound. Two cowboys fight against each other. Comes on a bootable cassette.
Order #7207 $19.95

FORTH for the ATARI

FORTH from Elcomp Publishing, Inc. is an extended Fig-Forth-version, Editor and I/O package included. Utility package includes decompiler, sector copy, Hexdump (ASCII), ATARI Filehandling, total graphic and sound, joystick program and player missile. Extremely powerful!
Order #7055 disk $39.95

Floating point package with trigonometric functions (0 – 90°)
Order #7230 disk $29.95

Learn-FORTH from Elcomp Publishing, Inc.
A subset of Fig-Forth for the beginner. On disk (32k RAM) or on cassette (16k RAM).
Order #7053 $19.95

Expansion boards for the APPLE II

The Custom Apple + Other Mysteries
A complete guide to customizing the Apple Software and Hardware
Order-No. 680 $24.95

We also stock the boards which are used in the book The Custom Apple (bareboards)

6522 I/O Board	No. 605	$39.00
EPROM Burner	No. 607	$49.00
8K EPROM/RAM Board	No. 609	$29.00
Prototyping board for the Apple II	No. 604	$29.00
Slot repeater board for the Apple II	No. 606	$49.00

Order two boards and get the book free!

Care and Feeding of the Commodore PET
Eight chapters exploring PET hardware. Includes repair and interfacing information. Programming tricks and schematics.
Order #150 $9.95

Hardware – ADD-ONS for ATARI

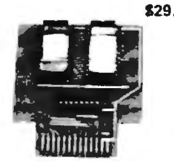
PRINTER INTERFACE
This construction article comes with printed circuit board and software. You can use the EPSON printer without the ATARI printer interface. (Works with gameports 3 and 4).
Order #7211 $19.95

RS-232 Interface for your ATARI 400/800
Software with connector and construction article.
Order #7291 (5V TTL-Level) $19.95

EPROM BURNER for ATARI 400/800
Works with gameports. No additional power supply needed. Comes compl. assembled with software (2716, 2732, 2532).
Order #7042 $179.00

EPROM BURNER for ATARI 400/800 KIT
Printed circuit board incl. Software and extensive construction article.
Order #7292 $49.00

EPROM BOARD (CARTRIDGE)
Holds two 4k EPROMs (2532). EPROMs not included.
Order #7043 $29.95



EPROM BOARD KIT
Same as above but bare board only with description.
Order #7224 $14.95

ATARI, VIC-20, Sinclair, Timex and OSI

NEW – for your ATARI 400/800
Astrology and Biorhythm for ATARI (Disk only!)
Order-No. 7223 $29.95
Birth control with the ATARI (Knaus ...)
Order-No. 7222 Disk only! $29.95

Books + Software for VIC-20 (requires 3KRAM Exp.)
No. 4870 Wordpr... VIC 20, 8KRAM $19.95
No. 4883 Ma... List ... VIC 20, 8K RAM $14.95
No. 176 Tricks for VICs (book, 115 pages) $ 9.95

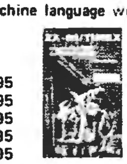
Miniassembler for VIC-20 No. 4896 $19.95
Runfill for VIC, No. 4894 $9.95
TIC TAC VIC, No. 4880 $9.95
GAMEPACK I (3 Games) No. 4881 $14.95

Progr. in 6502 Machine Language on your PET+CBM
2 complete Editor/Assemblers (Source code 3 hex dump + description plus a powerful machine language monitor (Hexdump). Order-No. 166 $19.95

Universal Experimenter Board for the VIC-20
(Save money with this great board). This board plugs right into the expansion slot of the VIC-20.
Order #4844 $18.95

Software for SINCLAIR ZX-81 and TIMEX 1000
#2399 Machine Language Monitor $9.95
#2398 Mailing List $19.95

Programming in BASIC and machine language with the ZX-81 (82) or TIMEX 1000.
Order-No. 174 (book) $ 9.95



BOOKS FOR OSI
No. 157 1. Book of Ohio $7.95
No. 158 2. Book of Ohio $7.95
No. 159 3. Book of Ohio $7.95
No. 160 4. Book of Ohio $7.95
No. 161 5. Book of Ohio $7.95

#151 8K Microsoft BASIC Ref. Man. $9.95
#152 Expansion Handbook for 6502 and 6802 $9.95
#153 Microcomputer Appl. Notes $9.95

Complex Sound Generation
New revised applications manual for the Texas Instruments SN 76477 Complex Sound Generator.
Order #154 $6.95

Small Business Programs Order #156
Complete listings for the business user. Inventory, Invoice Writing, Mailing List and much more. Introduction to Business Applications. $14.90

Jim Butterfield, Associate Editor

Bagel Break, Part 3

We've looked at some of the planning that goes into organizing a machine language game of "Bagels." Let's put the final touches together, and discuss some less obvious aspects of the way the program fits with BASIC.

We chose to start the machine language at $033C, decimal 828. The main reason for this was to make it universal – the same space is available in PET, CBM, VIC and Commodore 64 computers. It is the memory address of the cassette tape buffer (on PET/CBM, the buffer for cassette 2).

But that space is not always free and clear. If we wished to save the program to cassette tape, we might need this buffer space. The SAVE command would begin by staging the program "header" block in this area; the program would be destroyed before it was written. If we should try any BASIC 4.0 disk commands, this area would also be invaded; a simple CATALOG command would wreck our program.

For safety's sake, we should pop our machine language program into place just before we use it. What better way than to build the program as a series of BASIC DATA statements, and POKE it into its working area?

That's exactly what we do in the program here. If we examine the numbers in the DATA statements, we'll be able to spot our original program. The first two numbers, for example, are 169 and 0. These decimal numbers would translate to $A900, and that's our first instruction, LDA #$00, or, "Load the A register with the actual value of hex 00." We could trace through all of the instructions of the original program in this fashion.

It Is BASIC

How did we get the DATA statement values in lines 100-180? We could do it by painstaking hand translation, but there are easier ways. After all, we have a computer to do the routine calculations for us. One way would be to put the hex program in place, and then write a loop using PEEK to print out the decimal values. For example,

```
FOR J = 828 TO 848: PRINT PEEK(J);:NEXT J
```

would yield a series of decimal values. Using screen editing, we could insert the commas and prefix the values with a line number and the word DATA.

Thus, we have a program that's totally BASIC. When it runs, we manufacture a machine language program and then call it. But the program handles like BASIC, lists like BASIC, and may be loaded and saved like BASIC – because it is BASIC.

A few comments on the BASIC program itself. Line 290 causes the random number generator to be scrambled, or "randomized." When we use the value zero as an argument, i.e., RND(0), the random number seed is scrambled against the clock time so that all following numbers, called with RND(1), will be unpredictable.

Lines 300 to 320 generate four random numbers, each from 65 (the ASCII letter A) to 70 (letter F) inclusive. These values are POKEd into memory for the machine language program to use.

After the call to machine language, PEEK(577) will tell us whether or not the player got the solution. Location 577 ($0241), tells us about the "exact matches": four is a correct solution, of course. If the count is less than four, we must tell the player what the solution was by PEEKing the characters back out from addresses 580 to 583 – that's where we put them.

We have looked at a simple game which uses BASIC and machine language working together. The emphasis this time was on working the problem through and commenting on the various tools that a programmer might bring to the task.

The program could well have been written entirely in BASIC. After all, Bagels doesn't need super-speed to run. But you may notice that for this sort of job, machine language brings a clean elegance to the program. The programmer often feels that machine language gives a more total control over the programming.

Alspa Computer, Inc.

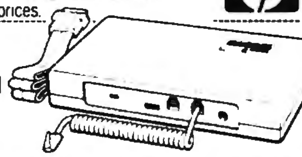
Price-performance leader. Includes Z80A, 8" ds/dd drives, 3 serial + 1 parallel port, winchester port, networking. Prices start below $1500. DEALER / OEM inquiries invited.

SPECIALS on INTREGATED CIRCUITS

6502	7.45	10/ 6.95	50/ 6.55	100/ 6.15
6502A/6512A	8.40	10/ 7.95	50/ 7.35	100/ 6.90
6520 PIA	5.15	10/ 4.90	50/ 4.45	100/ 4.15
6522 VIA	6.45	10/ 6.10	50/ 5.75	100/ 5.45
6532	7.90	10/ 7.40	50/ 7.00	100/ 6.60
2114-L200		2.45	25/ 2.30	100/ 2.15
2716 EPROM		4.90	5/ 4.50	10/ 4.00
2532 EPROM		6.90	5/ 6.75	10/ 6.45
6116 2KX8 CMOS RAM		6.90	5/ 6.75	10/ 6.45
4116 RAM		8 for 14		
Zero Insertion Force 24 pin Socket (Scanbe)				2.00

Hewlett Packard

Write or call for prices.



Anchor Automation Signalman Modems

FREE SOURCE MEMBERSHIP WITH SIGNALMAN

All Signalman Modems are Direct Connect, and include cables to connect to your computer and to the telephone. Signalman Modems provide the best price-performance values, and start at less than $100. **Dealer and OEM inquiries invited.**

Mark I RS232	(99)	59
Mark II for Atari 850	(99)	79
Mark IV for CBM/PET with software	(169)	89
Mark V for Osborne (software available)	(129)	79
Mark VI for IBM Personal Computer	(279)	175
Mark VII Auto Dial/Auto Answer	(159)	89
Mark VIII Bell 212 Auto Dial/Answer	(399)	299

DC HAYES Smartmodem	**219**
DC Hayes Smartmodem 1200	545

PROM QUEEN for VIC	**170**
Apple Emulator for Commodore 64	**89**
Screenmaker 80 COLUMN CARD for C64	**145**
Solid Oak 2 Level Stand for C64 or VIC	**29**
C64/VIC Switch (networking)	125
BACKUP V1.0 tape copier for C64 or VIC	**20**
CARDBOARD/6 Motherboard - VIC	64
CARDAPTER/1 Atari VCS Adapter - VIC	69
CARDPRINT Printer Interface - C64/VIC	64
CARDBOARD/3s Motherboard - VIC	32
CARDCO C64/VIC Calculator Keypad	32
CARDRAM/16 RAM Expansion - VIC	64
Complete CARDCO Line in stock	
CIE and VIE IEEE Interfaces in stock	
BASM Compiler/Assembler for C64	89

APPLE—FRANKLIN ITEMS

KRAFT Apple Joystick	**40**
16K RAM Card for Apple	**59**
Solid Oak 2 Level Stand for Apple	**29**
Serial Card for Apple	99
MPC RAM/80 column card for IIE (AP/TXT)	**139**
Z80 Softcard and CP/M (Microsoft)	235
RANA Elite I with Controller	389
Parallel Printer Interface/Cable	79
Apple Dumpling (Microtek) Printer Interface	**115**
Apple Dumpling with 16K Buffer	**160**
Grappler + Interface	140
Kraft Products for Apple in stock	
DC Hayes Micromodem II	299
PFS: File	**100**
PFS: Report	**100**
Videx 80 Column Card	209
Hayden Software for Apple 20% OFF	
PIE Writer Word Processor	**120**

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN I $245 base price

Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

COMPACK/STCP $115

Intelligent Terminal Package for PET, CBM, C64
Includes ACIA Hardware / STCP Software

VE-2 IEEE to Parallel Interface 110

Includes case, power supply, full 8-bit transmission, and switch selectable character conversion to ASCII.

VIDEO ENHANCER for Commodore 64 89
Realize video quality equal or better than composite monitor using standard color TV.

SCREENMAKER 80 Column Adapter for C64 145
Provides big screen capability for business applications.

VIC 20 Products		VIC Sargon II Chess	32
BACKUP V1.0	**20**	**VIC GORF**	**32**
VIC RAM Cards in stock		Meteor Run (UMI)	39
VIC SuperExpander	52	VIC Radar Ratrace	24
VIC 16K RAM	95	Amok (UMI)	20
Thorn EMI Software		Snakman	15
HES Software		Rubik's Cube	13
VIC Omega Race	32	Programmers Reference	15
Spiders of Mars (UMI)	39	**FROGGER**	**25**
Programmers Aid	45	VIC Adventure Series	

VICTORY Software for VIC and C64

Street Sweepers (VIC)	12	Kongo Kong (VIC)	16
Night Rider (VIC)	11	Cosmic Debris (VIC)	12
Annihilator	16	Adventure Pack I	16
Adventure Pack II	16	Metamorphosis	11
Educational Pack I	11	Trek	12
Strategy Pack I	16	Grave Robbers	12

PAPER CLIP Word Processor	**109**
ORACLE Data Base Batteries Included	**125**
Commodore 64 Programmers Reference Guide	16
MicroChess for C64—8 levels of play	**19**
Compute!'s First Book of PET/CBM	11
C64 or VIC SWITCH	**125**
POWER ROM Utilities for PET/CBM	78
WordPro 3+/64	**69**
WordPro 4+ - 8032, disk, printer	295
SPELLMASTER spelling checker for WordPro	
VISICALC for PET, ATARI, or Apple	189
PET-TRAX PET to Epson Graphics Software	**40**
SM-KIT enhanced PET/CBM ROM Utilities	**40**
Programmers Toolkit - PET ROM Utilities	35
EASY CALC for C64	**65**
PET Spacemaker II ROM Switch	36
COPYWRITER Word Processor for C64	**69**
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, ADA1450, SADI in stock)	
Programming the PET/CBM (Compute!) — R. West	**20**
Compute! First Book of VIC	11
HES MODEM with Software	**65**
HES Software and Hardware in stock	
UMI products in stock	

GRIDRUNNER (HES) VIC or C64			**29**
COCO (HES) Tutorial for C64			39
Gridrunner (HES) C64 or VIC			29
Agressor (HES)			29
HES Turtle Graphics	VIC 29		C64 49
HES Writer	VIC 30		C64 34
Grand Master Chess (UMI) C64 or VIC			**27**
Renaissance (UMI)	VIC 29		C64 24
Synthesound (HES)			44
HES MON	VIC or C64		29
6502 Professional Developement System (HES)			24
Robot Panic 29	Pirate's Peril 29		Retro Ball 29

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.20	50/ 2.00	100/ 1.95
Scotch (3M) 5" ds/dd	10/ 3.05	50/ 2.80	100/ 2.75
Scotch (3M) 8" ss/sd	10/ 2.30	50/ 2.10	100/ 2.06
Scotch (3M) 8" ss/dd	10/ 2.85	50/ 2.70	100/ 2.65

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

Sentinal 5" ss/dd	10/ 1.90	50/ 1.85	100/ 1.80
Sentinal 5" ds/dd	10/ 2.55	50/ 2.50	100/ 2.45

We stock Dysan disks

Wabash 5" ss/sd	10/ 1.65	50/ 1.60	100/ 1.55
Wabash 5" ss/dd	10/ 1.95	50/ 1.90	100/ 1.85
Wabash 8" ss/sd	10/ 2.00	50/ 1.95	100/ 1.85

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages	10 for $5	Hub Rings 50 for $6
Disk Library Cases	8"—3.00	5"—2.25
Head Cleaning Kits	11	

CASSETTE TAPES—AGFA PE-611 PREMIUM

C-10	10/ .61	50/ .58	100/ .50
C-30	10/ .85	50/ .82	100/ .70

DATASHIELD BACKUP POWER SOURCE 265
Battery back up Uninterruptible Power Supply with surge and noise filtering. The answer to your power problems.

Zenith ZVM-121 Green Phosphor Monitor	98
Zenith new color and monochrome monitors in stock	
Peachtext 5000 Software Package	**219**
PFS Software for IBM and Apple in stock	
VOTRAX Personal Speech System	280
VOTRAX Type-N-Talk	160
VOICE BOX Speech Synthesizer (Apple or Atari)	
CompuServe Subscription (5 hours free)	32
Dynax (Brother) DX-15 Daisy Wheel Printer	**469**
Prowriter Parallel Printer	379
Panasonic 1090 Printer with Correspondence Mode	**319**
USI CompuMOD 4 R F Modulator	39
Daisywriter 2000 with 48K buffer + cable	1150
Many printers available (Gemini-Star, Brother, OKI, etc.)	
We Stock AMDEK Monitors	
Amdek DXY-100 Plotter	590
A P Products	15% OFF
Watanabe Intelligent Plotter 990	6-pen 1290
BROOKS 6 Outlet Surge Suppressor/Noise Filter	**54**
We stock Electrohome Monitors	
Synertek SYM-1 Microcomputer	**189**

ALL BOOK and SOFTWARE PRICES DISCOUNTED

Panasonic 12" Monitor (20 MHz) with audio	135
Panasonic CT-160 Dual Mode Color Monitor	259

USI Video Monitors—Green or AMBER 20 MHz hi-res. Dealer and OEM inquiries invited

ZENITH data systems

HERO 1 Robot (factory assembled)	2145
Z29 Terminal (DEC and ADM compatible)	**680**
ZT-1 Intelligent Communications Terminal	**369**
ZT-10 Intel. Terminal with Serial Port	**340**
Z100 16-bit/8-bit Systems in stock	**CALL**
We stock entire Zenith line.	

ATARI SPECIALS

WE STOCK ENTIRE LINE—write for prices.
Software for Atari in stock.

215-822-7727
252 Bethlehem Pike
Colmar, PA 18915

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

OAK STAND-C64, VIC, Apple, Atari 29

Beautiful natural solid oak two-level stand. Rests on table above computer. Holds disk drives/cassette deck, as well as your monitor/TV.

KMMM Pascal for PET/CBM/C64 $79

A subset of standard Pascal with extensions. Includes Machine Language Pascal Source Editor, Machine Language P-Code Compiler, P-Code to machine language translator for optimized object code, Run-time package, Floating Point capability, User Manual, and sample programs.

Requires 32K — Please specify configuration.

EARL for PET (disk file based) $65

Editor, Assembler, Relocater, Linker

Generates relocatable object code using MOS Technology mnemonics. Disk file input (can edit files larger than memory).

Paper Clip (Batteries Included) $109

Extremely comprehensive word processor for Commodore and Commodore 64 computers. Has features of WordPro, plus advanced functions like horizontal scroll, column move, column arithmetic, column sort, and comprehensive printer support.

Delphi's ORACLE (Batteries Included) $125

Comprehensive Data Base, Report Writer, Mail Label system allowing large record size (over 8000 characters) with the number of records in a file limited only by disk capacity (7.5 MB on 9090 drive). Fast machine language routines, including full multilevel sorts.

RAM/ROM for PET/CBM

4K or 8K bytes of soft ROM optional battery backup.

Use RAM/ROM as a software development tool to store data or machine code beyond the normal BASIC range, or to load a ROM image to avoid ROM socket conflicts. Possible applications include machine language sort (such as SUPER-SORT), universal wedge, Extramon, etc.

RAM/ROM — 4K	$75
RAM/ROM — 8K	90
Battery Backup Option	20

SUBSORT for PET/CBM $35

Excellent general purpose machine language sort routine.

PROGRAM YOUR OWN EPROMS $75

Branding Iron EPROM Programmer for PET/CBM software for all ROM versions. Includes all hardware and software to program or copy 2716 and 2532 EPROMs.

PORTMAKER DUAL RS232 SERIAL PORT $63

Two ports with full bipolar RS232 buffering. Baud rates from 300 to 4800. For PET/CBM, AIM, SYM.

COMAL Package for CBM $25

Includes software on disk, and Comal Handbook

SuperGraphics 2.0

NEW Version with TURTLE GRAPHICS

SuperGraphics, by John Fluharty, provides a 4K machine language extension which adds 35 full featured commands to Commodore BASIC to allow fast and easy plotting and manipulation of graphics and shapes on the PET/CBM video display.

SOUND commands allow you to initiate notes or songs from BASIC, and then play them in the background mode without interfering with your BASIC program.

Additionally, seven new TURTLE commands open up a whole new dimension in graphics.

Specify machine model (and size), ROM type (BASIC 3 or 4)

SuperGraphics in ROM ($A000 or $9000) $45

Volume discounts available for schools.

FLEX-FILE for PET/CBM Computers

NEW VERSION 2 now for C64

FLEX-FILE is a set of flexible, friendly programs to allow you to set up and maintain a data base. Includes versatile Report Writer and Mail Label routines, and documentation for programmers to use Data Base routines as part of other programs.

RANDOM ACCESS DATA BASE

Record size limit is 256 characters. The number of records per disk is limited only by record size and free space on the disk. File maintenance lets you step forward or backward through a file, add, delete, or change a record, go to a numbered record, or find a record by specified field (or partial field). Field lengths may vary to allow maximum information packing. Both subtotals and sorting may be nested up to 5 fields deep. Any field may be specified as a key. Sequential file input and output, as well as file output in WordPro and PaperMate format is supported. Record size, fields per record, and order of fields may be changed easily.

MAILING LABELS

Typical mail records may be packed 3000 per disk on 8050 (1400 in 4040). Labels may be printed any number wide, and may begin in any column position. There is no limit on the number or order of fields on a label, and complete record selection via type code or field condition is supported.

REPORT WRITER

Flexible printing format, including field placement, decimal justification and rounding. Define any column as a series of math or trig functions performed on other columns, and pass results such as running total from row to row. Totals, nested subtotals, and averages supported. Complete record selection, including field within range, pattern match, and logical functions can be specified.

FLEX-FILE 2 by Michael Riley $110

CBM64, PET/CBM/VIC—32K, Disk. Specify configuration.

SCREEN MAKER (cgrs microtech) $149

80 Column Adapter for Commodore 64

Expand your computer for business applications. Provides 80 column × 24 line display in a 2K video RAM. Linking software provided.

Copy-Writer Word Processor $69

Full-featured professional word processor with over 800 lines of text per memory load on C64. Has features not available in many word processors such as double column printing, built in graphic capability, shorthand notations, and ability to support all printer codes.

SPECIAL COMBINATION PACKAGE $200

Includes SCREEN MAKER AND Copy Writer for C64

DISK I.C.U. $40

Intensive Care Unit by L.C. Cargile

COMPLETE DISK RECOVERY SYSTEM FOR CBM DRIVES

- edit disk blocks with ease
- duplicate disks, skipping over bad blocks
- complete diagnostic facilities
- unscratch scratched files
- check and correct scrambled files
- recover improperly closed files
- extensive treatment of relative files
- optional output to IEEE488 printer
- comprehensive user manual (an excellent tutorial on disk operation and theory).

Furnished on copy-protected disk with manual.
Backup disk available, $10 additional.

CBM Software

PAPER CLIP Word Processor	110
BASIC INTERPRETER for CBM 8096	$95
PEDISK II Systems from cgrs Microtech available.	
FILEX IBM 3741/2 Data Exchange Software available.	
JINSAM Data Base Management System for CBM.	
COPY-WRITER Word Processor for PET/CBM	$159
CASH MANAGEMENT SYSTEM	$45
Petspeed BASIC Compiler	120
Integer BASIC Compiler	120
CMAR Record Handler	110
UCSD Pascal (without board)	135
Wordcraft 80 or 8096	265

FORTH for PET now for C64

BY L.C. Cargile and Michael Riley **$50**

Features include:

- full FIG FORTH model.
- all FORTH 79 STANDARD extensions.
- structured 6502 Assembler with nested decision making macros.
- full screen editing (same as when programming in BASIC).
- auto repeat key.
- sample programs.
- standard size screens (16 lines by 64 characters).
- 150 screens per diskette on 4040, 480 screens on 8050.
- ability to read and write BASIC sequential files.
- introductory manual.
- reference manual.

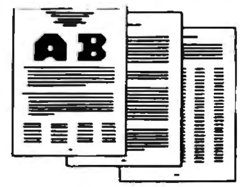
For Commodore 64, or any 16K/32K PET/CBM with ROM 3 or 4, and CBM disk drive. Please specify configuration when ordering.

Metacompiler for FORTH **$30**

Simple metacompiler for creating compacted object code which can be run independently (without FORTH system).

PageMate 60 COMMAND WORD PROCESSOR

by Michael Riley



Paper-Mate is a full-featured word processor for Commodore computers. Page-Mate incorporates 60 commands to give you full screen editing with graphics for all 16K or 32K machines (including 8032), all printers, and disk or tape drives. Many additional features are available (including most capabilities of WordPro 3).

Page-Mate functions with all Commodore machines with at least 16K, with any printer, and either cassette or disk.

To order Page-Mate, please specify machine and ROM type.

Page-Mate (disk or tape) for PET, CBM, VIC, C64 **$40**

SM-KIT for PET/CBM **$40**

Enhanced ROM based utilities for BASIC 4. Includes both programming aids and disk handling commands.

Commodore 64

Hunter-Killer - Commodore 64	15
- authentic naval warfare game (complete with sonar)	
Submarine Warfare (Clockwork Computers)	29
PAPER CLIP Word Processor	109
BASM Compiler and Assembler	85
WordPro 3+/64	75
Vanilla PILOT with Turtle Graphics	27
- also includes sound, Toolkit, joystick support	
Commodore 64 Programmer Reference Guide	16
CCI Submarine Warfare	24
Laser Command	15
EARLY GAMES for Young Children	25
PETSPEED Compiler C64	120
CALC RESULT Spread Sheet Package	135
1000 Miles (Mille Bornes Game	9
MicroChess	19
Adventure (disk)	9
PILOT 64	45
LOGO 64	45
Easy Calc 64	65
MAE Assembler - C64	85
Assembly Language Tutorial - C64/VIC	27
Abacus Software in stock	
Synthy-64 music and sound synthesizer	26
Tiny BASIC Compiler	17
ScreenGraphics-64 adds BASIC Graphics	22
Draw Poker	5

VICTORY SOFTWARE®

All Victory Software in stock.

215-822-7727
252 Bethlehem Pike
Colmar, PA 18915

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

Bagels

```
90 REM BAGELS ML
100 DATA 169,0,141,64,2,238,64,2,173,64,2
    ,201,10,240,5,32,81,3,208,241,96
110 DATA 9,48,32,210,255,169,32,32,210,25
    5,162,0,142,65,2,142,66,2,142,67,2
120 DATA 32,228,255,201,65,144,249,201,71
    ,176,245,32,210,255,174,67,2
130 DATA 238,67,2,157,76,2,189,68,2,157,7
    2,2,224,3,208,223,189,72,2
140 DATA 221,76,2,208,11,238,65,2,169,0,1
    57,72,2,157,76,2,202,16,234,160,0
150 DATA 162,0,185,72,2,240,16,221,76,2,2
    08,11,238,66,2,169,0,153,72,2
160 DATA 157,76,2,232,224,4,144,230,200,1
    92,4,144,223,162,0,169,32
170 DATA 32,210,255,189,65,2,9,48,32,210,
    255,232,224,2,144,238
180 DATA 169,13,32,210,255,173,65,2,201,4
    ,96
200 FOR J=828 TO 990
210 READ X: T=T+X
220 POKE J,X
230 NEXT J: IF T<>18169 THEN STOP
240 PRINT "BAGELS"
250 PRINT ".. GUESS MY SECRET CODE"
260 PRINT ".. I'LL TELL YOU HOW MANY"
270 PRINT ".. EXACT MATCHES AND OTHER"
280 PRINT ".. MATCHES YOU GET..."
290 X=RND(0)
300 FOR J=0 TO 3
310 X=INT(RND(1)*6)+65
320 POKE 580+J,X
330 NEXT J
340 SYS 828
350 IF PEEK(577)=4 THEN PRINT "GOT IT!":G
    OTO 400
360 PRINT "THE CODE WAS: ";
370 FOR J=0 TO 3
380 PRINT CHR$(PEEK(580+J));
390 NEXT J: PRINT
400 INPUT "WANT ANOTHER GAME";X$
410 IF X$="Y" OR X$="YES" GOTO 300
```

©

C Regena

Playing Music On The TI

Musicians, rejoice! Here is a computer that plays music. You can play a single tone to tune an instrument or get a pitch for a song, or you can play complex rhythms with three-part harmony. Compose to your heart's content and let the computer perform your masterpiece.

Nonprofessional musicians – if you'd like to learn more about music, your computer can be a great teacher. Learn the notes on a keyboard, learn to read music, learn definitions, or learn to distinguish chords. The computer with music capabilities makes learning fun.

For those of you who prefer not to write your own programs, the TI *Music Maker* command module is available. Here's a quick review. You may compose music by choosing various types of notes or rests (quarter, eighth, half, etc.) and placing them on the staff. Choose notes for accompaniment if you wish. Build a song a measure at a time. The computer makes sure the timing works out correctly. Oh yes, you can choose your key signature, time signature, and tempo. At any time you can play or edit your composition, then save it on cassette or disk if you like.

Another section of the module is made especially for nonmusicians. You may draw lines up and down the screen at different levels for a "sound graph," then hear the computer play relational tones. Add second and third voices if you wish. This command module is really quite versatile with many options and can help you learn about music.

CALL SOUND

To program your own music on the TI, use the CALL SOUND statement. The basic form is

CALL SOUND(***duration, frequency, volume***)

The *duration* is a numeric expression (number, variable, or algebraic expression which will evaluate to a number) which is the number of milliseconds you wish to play the tone. For example, 1000 would be one second. The number may be from 1 to 4250 or from -4250 to -1.

The *frequency* is a numeric expression that indicates what tone to play. The frequency is the cycles per second and may be from 110 to 44733, which is from low A on the bass staff to out-of-human-hearing range. The "Musical Tone Frequencies" table in the Appendix of the *User's Reference Guide* lists the musical notes with the corresponding frequencies. Note that you can specify numbers that are between the normal musical tones.

The *volume* is a numeric expression that indicates loudness. The volume may vary from 0 to 30, where 0 is the loudest. The volume also depends on the audio setting of your monitor or television, but you can control relative volumes of the tones with this parameter.

Try this command:

```
CALL SOUND(500,440,2)
```

The computer plays the tone of A (440) for 500 milliseconds (half a second) at a volume level of 2.

Now, if you want to tune your band instrument, just run this program.

```
100 CALL SOUND(4250,440,0)
110 GOTO 100
```

You may specify one, two, or three notes to be played in one CALL SOUND statement. Each statement has one duration, then a frequency with a volume for each note desired. Here is an example of the three notes in the C major chord:

```
CALL SOUND(1000,262,6,330,4,392,2)
```

The chord will play for 1000 milliseconds. The notes played are C at a volume 6, E at a volume 4, and G at a volume 2. Try a few chords with different frequency and volume numbers.

If you play a solo instrument, you might enjoy programming the computer to play the accompaniment chords. Tune your instrument with the computer, then you can play with the computer as your accompanist.

Using Sheet Music

If you use three tones in the CALL SOUND statement, they may be in any order. I like to use the first frequency and volume as the melody tone, then the second and third frequencies and volumes as the accompaniment tones. This way I can keep track of which number is the melody. Also, if I start to run out of memory in a piece, I can go back to the CALL SOUND statements and delete accompaniment tones by keeping only the first frequency and volume in each statement.

You may work from a copy of written music to try out the musical capabilities of the TI. The top note is usually the melody. You may choose any two notes written directly under the melody note for the accompaniment or the other two notes in your CALL SOUND statement. To emphasize the melody, use a louder volume for the melody note and softer volumes for the accompaniment notes. For example:

```
CALL SOUND(400,262,1,196,6,159,8)
```

If you have two CALL SOUND statements together which specify the same frequencies and volumes, the notes may sound like one long note rather than two separate notes. To make the notes sound distinct, just change the volume number for one of the notes:

```
300 CALL SOUND(200,262,2,196,6,165,8)
310 CALL SOUND(200,262,3,196,6,165,8)
```

To make a bass note sound tied or held while two different melody notes are played, keep the frequency and the volume numbers the same in both statements:

```
500 CALL SOUND(300,262,2,165,8)
510 CALL SOUND(300,330,2,165,8)
```

Other statements may be executed while a note is being played. You may define graphics, draw graphics, or make calculations between CALL SOUND statements. This feature allows you to have fun choreographing pictures with music to present a musical dramatization. You do need to experiment so you don't get too many statements between the music statements or there will be gaps in the music.

A note will keep playing for its specified duration, and the computer will execute statements until either the duration runs out or another CALL SOUND statement is encountered. If another CALL SOUND statement needs to be executed, the computer waits until the first duration is finished before starting the next sound. If you prefer to have the computer go ahead with the next sound statement, use a negative number for the second statement's duration. Here is an example.

```
100 CALL SOUND(200,392,2)
110 CALL SOUND(200,330,2)
120 CALL SOUND(200,262,2)
130 CALL SOUND(200,330,2)
140 CALL SOUND(400,392,2)
150 END
```

The computer starts with the tone of G and plays for 200 milliseconds. Next the tone of E plays for 200 milliseconds, then C for 200 milliseconds, then E for 200 milliseconds, then G for 400 milliseconds. During the last note the program will end, but the note will keep playing for the 400 milliseconds.

Now change to negative durations in lines 110-140:

```
100 CALL SOUND(200,392,2)
110 CALL SOUND(-200,330,2)
120 CALL SOUND(-200,262,2)
130 CALL SOUND(-200,330,2)
140 CALL SOUND(-400,392,2)
150 END
```

This time, the computer starts by playing G. As soon as the computer comes to line 110, a CALL SOUND statement with a negative duration, the computer immediately starts the new sound – no matter what the previous duration was. Line 140 starts the sound of G as soon as the computer comes to that statement, then continues the sound for 400 milliseconds since there is not a following sound statement with a negative duration. Try running these two programs to hear the difference.

A technique I like to use in programming music is to use a variable name for the duration, and specify the numeric value of that duration variable near the beginning of the program. For example, I often use T for "tempo" or "time" or M for "metronome marking" or N for "note." If I use T to represent the duration for a quarter note, then T/2 would be an eighth note and 4*T would be a whole note. You can get exact timing in your music and let the computer calculate the durations.

Note: Avoiding using Q for "quarter note," especially on the TI-99/4, because the key combination of SHIFT Q is "quit." This is comparable to the FCTN (quitting on the TI-99/4A). An accidental SHIFT Q will wipe out your program and return to the title screen. With a shifted parenthesis before the variable and a shifted comma after the variable, it's too easy to get an accidental SHIFT Q.

Variable Durations

Another advantage to using a variable duration is that you can write your song in terms of the variable, then change the tempo of the song by changing only one line (the line defining the duration) rather than each CALL SOUND statement. Here is a short example.

```
100 T=400
110 CALL SOUND(T,262,2)
120 CALL SOUND(T,294,2)
```

```
130 CALL SOUND(2*T,330,2)
140 CALL SOUND(3*T/4,349,2)
150 CALL SOUND(T/4,392,2)
160 CALL SOUND(T/2,440,2)
170 CALL SOUND(T/2,494,2)
180 CALL SOUND(T*4,523,1)
190 END
```

100 Duration of quarter note=400
110 Quarter note
120 Quarter note
130 Half note
140 Dotted eighth note
150 Sixteenth note
160 Eighth note
170 Eighth note
180 Whole note

RUN the program. Now change line 100 to T=800. The song is twice as long, but each note stays in the exact proportion. Change line 100 to T=200. The song is faster, but still in proportion.

If you need to learn a song with a difficult rhythm, program the computer to play the song. Use a variable such as T for the duration. You can set the duration to a slower note, then as you learn the song you can speed it up by changing just the one line.

You may prefer to use variables for the different kinds of notes in this manner:

```
100 T=400
110 E=T/2
120 H=T*2
130 CALL SOUND(H,523,2)
140 CALL SOUND(E,494,3)
150 CALL SOUND(E,440,3)
160 CALL SOUND(T,392,2)
```

100 Quarter note duration
110 Eighth note
120 Half note

You may also want to set up a list of variables for the note names before you use them in CALL SOUND statements:

```
100 T=400
110 C=262
120 D=294
130 E=330
140 CALL SOUND(T,E,2)
150 CALL SOUND(T,D,2)
160 CALL SOUND(T,C,2)
```

You may also use a variable for the volume, such as CALL SOUND(T,D,V).

Just as in other programming, you can use FOR-NEXT loops and GOSUB and GOTO statements to help write your music. For example, if you have a musical phrase between repeat bars, you can use a FOR-NEXT loop to play it twice. If you have a common phrase used several times within a song, use a GOSUB procedure.

Beethoven Medley

The following program, "Ludwig," illustrates the use of CALL SOUND statements to create a medley of familiar Beethoven pieces. Line 120 sets the duration of a quarter note to 400 milliseconds for the first tune, an excerpt from "Ode to Joy" of the Ninth Symphony. Lines 170-660 play this melody. In between the CALL SOUND statements are graphics statements. Lines 180-340 define graphics characters and colors, then later CALL HCHAR and CALL VCHAR statements draw a picture. The CALL SOUND statements in lines 170-400 illustrate the "tied" bass note, or a bass note held while two melody notes are played. Most of the notes are quarter notes, but line 610 has a dotted quarter note, line 650 has an eighth note, and line 660 has a half note.

Line 860 resets the duration variable T to 200 milliseconds. This time T represents an eighth note for phrases from "Ecossaises." The excerpt here is taken from music that is within repeat bars but has a first ending and a second ending. The common part of the repeat is in the subroutine at lines 1860-2230. Line 890 GOSUB 1860 plays the common phrase, then lines 920-980 play the first ending. Line 1010 repeats the common phrase with GOSUB 1860, then lines 1040-1100 contain the second ending.

Lines 1260-1420 play the third melody, "Für Elise." This example shows GOSUB commands within a FOR-NEXT loop. The subroutine for the common notes is contained in lines 2240-2420.

The final melody (lines 1430-1840) is an excerpt from the second movement of Beethoven's Fifth Symphony. Line 1430 defines the new duration T to be 800 milliseconds for an eighth note at an andante tempo. U is defined as three-fourths of an eighth note, or a dotted sixteenth note. T/4 is used for a thirty-second note. Character 128 is defined as a graphic musical note, and the embedded CALL HCHAR statements among the CALL SOUND statements place the notes on the screen.

Line 1850 (GOTO 1850) holds the picture on the screen. Press CLEAR (FCTN 4 on the TI-99/4A or SHIFT C on the TI-99/4) to stop the program.

If you prefer to save the typing time, you can obtain a copy of this program by sending $3, a stamped, self-addressed mailer, and a blank tape or disk to: REGENA, P.O. Box 1502, Cedar City, UT 84720. Please specify the name of the program.

Ludwig

```
100 REM  BEETHOVEN MEDLEY
110 REM
120 T=400
130 CALL CLEAR
140 CALL SCREEN(3)
150 PRINT TAB(6);"BEETHOVEN MEDLEY"
    ;
160 CALL COLOR(1,2,8)
170 CALL SOUND(T,330,2,131,6)
180 CALL CHAR(96,"FFFFFFFFFFFFFFFF"
    )
```

```
190 CALL CHAR(97,"FF7F3F1F0F070301")
200 CALL SOUND(T,330,3,131,6)
210 CALL CHAR(98,"FFFEFCF8F0E0C08")
220 CALL CHAR(104,"FFFFFFFFFFFFFFFF")
230 CALL SOUND(T,349,3,131,6)
240 CALL CHAR(105,"0103070F1F3F7FFF")
250 CALL CHAR(106,"80C0E0F0F8FCFEFF")
260 CALL SOUND(T,392,2,131,6)
270 CALL CHAR(120,"00003C3E3E1E0F03")
280 CALL CHAR(121,"003878F8F8F0C0BC")
290 CALL SOUND(T,392,3,147,6)
300 CALL CHAR(113,"FF7F3F1F0F070301")
310 CALL CHAR(114,"FFFEFCF8F0E0C08")
320 CALL SOUND(T,349,3,147,6)
330 CALL COLOR(10,8,8)
340 CALL COLOR(11,3,11)
350 CALL SOUND(T,330,3,147,6)
360 CALL VCHAR(13,15,104,7)
370 CALL VCHAR(13,16,104,7)
380 CALL VCHAR(13,17,104,7)
390 CALL VCHAR(13,18,104,7)
400 CALL SOUND(T,294,3,147,6)
410 CALL HCHAR(19,14,105)
420 CALL HCHAR(19,19,106)
430 CALL HCHAR(20,13,105)
440 CALL HCHAR(20,14,104,7)
450 CALL HCHAR(20,21,106)
460 CALL SOUND(T,262,2,165,6)
470 CALL HCHAR(21,11,105)
480 CALL HCHAR(21,12,104,10)
490 CALL HCHAR(21,22,106)
500 CALL SOUND(T,262,3,165,7)
510 CALL HCHAR(22,9,105)
520 CALL HCHAR(22,10,104,14)
530 CALL HCHAR(22,24,106)
540 CALL SOUND(T,294,2,175,6)
550 CALL HCHAR(23,7,105)
560 CALL HCHAR(23,8,104,18)
570 CALL HCHAR(23,26,106)
580 CALL SOUND(T,330,2,176,5)
590 CALL CHAR(115,"80A2A2AAEEEFFFFF")
600 CALL CHAR(99,"80A2A2AAEEEFFFFF")
610 CALL SOUND(T*1.5,294,2,196,7)
620 CALL HCHAR(24,1,99,32)
630 CALL CHAR(122,"0F3F7F7D790101")
640 CALL CHAR(123,"FEFE9EC0E0E0E0E")
650 CALL SOUND(T/2,262,3,196,7)
660 CALL SOUND(2*T,262,4,165,7,131,8)
670 CALL COLOR(9,3,8)
680 CALL COLOR(10,11,8)
690 CALL HCHAR(24,6,115,22)
700 CALL HCHAR(12,15,113,2)
710 CALL HCHAR(12,17,114,2)
720 CALL HCHAR(11,13,97)
730 CALL HCHAR(11,14,96,6)
740 CALL HCHAR(11,20,98)
750 FOR I=10 TO 4 STEP -1
760 READ A,B
770 CALL HCHAR(I,A,97)
780 CALL HCHAR(I,A+1,96,B)
790 CALL HCHAR(I,A+B+1,98)
800 NEXT I
810 DATA 12,9,9,14,7,18,6,20,4,24,3,26,1,28
820 CALL HCHAR(1,1,96,94)
830 CALL HCHAR(3,31,98)
840 CALL HCHAR(2,32,32)
850 CALL HCHAR(1,32,98)
860 T=200
870 CALL COLOR(2,16,3)
880 CALL COLOR(12,3,3)
890 GOSUB 1860
900 CALL HCHAR(3,6,120)
910 CALL HCHAR(3,7,121)
920 CALL SOUND(T,466,3,117,8)
930 CALL HCHAR(4,6,122)
940 CALL HCHAR(4,7,123)
950 CALL SOUND(T,831,4,698,8)
960 CALL HCHAR(5,10,120)
970 CALL HCHAR(5,11,121)
980 CALL SOUND(T*2,831,3,698,7,233,9)
990 CALL HCHAR(6,10,122)
1000 CALL HCHAR(6,11,123)
1010 GOSUB 1860
1020 CALL HCHAR(2,15,120)
1030 CALL HCHAR(2,16,121)
1040 CALL SOUND(T,349,3,294,7,117,9)
1050 CALL HCHAR(3,15,122)
1060 CALL HCHAR(3,16,123)
1070 CALL SOUND(T,466,3)
1080 CALL HCHAR(4,26,120)
1090 CALL HCHAR(4,27,121)
1100 CALL SOUND(T*2,466,2,294,6,233,8)
1110 CALL HCHAR(5,26,122)
1120 CALL HCHAR(5,27,123)
1130 CALL COLOR(12,16,3)
1140 CALL HCHAR(6,20,120)
1150 CALL HCHAR(6,21,121)
1160 CALL HCHAR(7,20,122)
1170 CALL HCHAR(7,21,123)
1180 CALL HCHAR(8,16,120)
1190 CALL HCHAR(8,17,121)
1200 CALL HCHAR(9,16,122)
1210 CALL HCHAR(9,17,123)
1220 CALL HCHAR(2,22,120)
1230 CALL HCHAR(2,23,121)
1240 CALL HCHAR(3,22,122)
1250 CALL HCHAR(3,23,123)
1260 FOR I=1 TO 2
1270 GOSUB 2240
1280 CALL SOUND(T,415,3)
1290 CALL SOUND(T,494,3)
1300 CALL SOUND(T,523,2,110,15)
1310 CALL SOUND(T,165,4)
1320 CALL SOUND(T,220,4)
1330 CALL SOUND(T,330,3)
1340 GOSUB 2240
1350 CALL SOUND(T,523,3)
1360 CALL SOUND(T,494,4)
1370 CALL SOUND(T,440,4,110,14)
1380 CALL SOUND(T,165,10)
1390 CALL SOUND(T,220,7)
1400 NEXT I
1410 CALL SOUND(T,330,6)
1420 CALL SOUND(T*3,440,6)
```

```
1430 T=800
1440 U=T*3/4
1450 CALL CHAR(128,"080C0A0A0878F87
     ")
1460 CALL COLOR(13,2,6)
1470 CALL SOUND(1,9999,30)
1480 CALL SOUND(U,156,6)
1490 CALL COLOR(1,2,6)
1500 CALL COLOR(9,3,6)
1510 CALL COLOR(10,11,6)
1520 CALL SOUND(T/4,208,5)
1530 CALL SOUND(T,262,3)
1540 CALL HCHAR(17,4,128)
1550 CALL SOUND(U,262,4)
1560 CALL HCHAR(15,8,128)
1570 CALL SOUND(T/4,233,4)
1580 CALL SOUND(U,208,3)
1590 CALL HCHAR(13,12,128)
1600 CALL SOUND(T/4,262,4)
1610 CALL SOUND(T+U,175,3,139,10)
1620 CALL HCHAR(13,21,128)
1630 CALL SOUND(T/4,220,3)
1640 CALL SOUND(U,233,3)
1650 CALL HCHAR(15,25,128)
1660 CALL SOUND(T/4,262,2)
1670 CALL SOUND(U,277,2,233,8)
1680 CALL HCHAR(17,29,128)
1690 CALL SOUND(T/4,262,3)
1700 CALL SOUND(U,233,2,196,8)
1710 CALL SOUND(T/4,277,2)
1720 CALL SOUND(U,196,2,156,8)
1730 CALL SOUND(T/4,233,2)
1740 CALL SOUND(U,165,3,131,8)
1750 CALL SOUND(T/4,196,3)
1760 CALL SOUND(T+U,262,2)
1770 CALL SOUND(T/4,233,4)
1780 CALL SOUND(U,220,4,175,10)
1790 CALL SOUND(T/4,175,4)
1800 CALL SOUND(U,233,2,117,10)
1810 CALL SOUND(T/4,277,3)
1820 CALL SOUND(U,196,4,156,10)
1830 CALL SOUND(T/4,156,4)
1840 CALL SOUND(2*T,208,2)
1850 GOTO 1850
1860 CALL SOUND(T,392,3,156,8)
1870 CALL HCHAR(2,2,42)
1880 CALL SOUND(T,466,3)
1890 CALL HCHAR(4,29,42)
1900 CALL SOUND(2*T,466,2,233,6,196
     ,8)
1910 CALL HCHAR(6,14,42)
1920 CALL SOUND(T,523,3,392,6,156,8
     )
1930 CALL HCHAR(8,11,42)
1940 CALL SOUND(T,466,3)
1950 CALL HCHAR(2,26,42)
1960 CALL SOUND(T*2,466,2,392,6,196
     ,8)
1970 CALL HCHAR(3,4,42)
1980 CALL SOUND(T,622,1,392,6,156,8
     )
1990 CALL HCHAR(2,19,42)
2000 CALL SOUND(T,466,2)
2010 CALL HCHAR(7,23,42)
2020 CALL SOUND(T*2,466,1,392,5,196
     ,8)
2030 CALL HCHAR(3,12,42)
2040 CALL SOUND(T,523,1,392,5,156,8
     )
2050 CALL HCHAR(9,19,42)
2060 CALL SOUND(T,466,3)
2070 CALL HCHAR(6,7,42)
2080 CALL SOUND(T*2,466,2,392,5,196
     ,8)
2090 CALL HCHAR(5,24,42)
2100 CALL SOUND(T,349,1,294,5,117,8
     )
2110 CALL HCHAR(5,17,42)
2120 CALL SOUND(T,466,3)
2130 CALL HCHAR(2,9,42)
2140 CALL SOUND(T*2,466,2,294,6,175
     ,8)
2150 CALL HCHAR(4,20,42)
2160 CALL SOUND(T,392,2,311,5,117,8
     )
2170 CALL HCHAR(2,30,42)
2180 CALL SOUND(T,466,3)
2190 CALL SOUND(T*2,466,2,311,7,196
     ,8)
2200 CALL SOUND(T,415,3,349,6,117,8
     )
2210 CALL SOUND(T,466,4)
2220 CALL SOUND(T*2,466,3,349,6,208
     ,8)
2230 RETURN
2240 CALL SOUND(T,659,6)
2250 CALL SOUND(T,622,6)
2260 CALL SOUND(T,659,6)
2270 CALL SOUND(T,622,5)
2280 CALL SOUND(T,659,4)
2290 CALL SOUND(T,494,3)
2300 CALL SOUND(T,587,4)
2310 CALL SOUND(T,523,5)
2320 CALL SOUND(T,440,6,110,15)
2330 CALL SOUND(T,165,8)
2340 CALL SOUND(T,220,6)
2350 CALL SOUND(T,262,4)
2360 CALL SOUND(T,330,4)
2370 CALL SOUND(T,440,4)
2380 CALL SOUND(T,494,4,131,15)
2390 CALL SOUND(T,165,4)
2400 CALL SOUND(T,208,4)
2410 CALL SOUND(T,330,4)
2420 RETURN
2430 END
```

©

Invisible Disk Directory For VIC And 64

Kevin E Gough

If you have a VIC or 64 and a 1540 or 1541 disk drive, this utility program can be very helpful. Once loaded, a simple SYS 828 will let you display your disk directory yet retain a program in memory.

The "Invisible Disk Directory Loader" is not really invisible. It only seems to be. A BASIC program POKEs the loader into the cassette buffer as machine language. Beginning at 828 ($033C) and ending at 971 ($03CB), the loader uses 144 bytes. A knowledge of BASIC is all you need to enter and use this program.

Using The "Invisible" Loader

If it were not for Jim Butterfield's article, "The Confusing Catalog" (COMPUTE!, March 1983), I probably would not have written the loader. I saw how easily he could load the disk directory from a program, as a file. Just OPEN 1,8,0,"$0" , input the bytes, do some manipulation, and there you have it. With the Invisible Directory Loader, you can display the directory and have any program in memory at the same time. You will no longer have to LOAD"$",8 as a program. Just type SYS 828 and press RETURN. The directory of your disk scrolls onto the screen. Use the CTRL key to slow the scroll when listing programs, or press the space bar to stop the listing.

This program will also give you the number of blocks each file uses and the number of blocks free on your disk. The directory cannot be listed on your printer.

You can also load and save cassette programs and not destroy the loader in the cassette buffer. This is because the loader also changes the start of the cassette buffer pointer at 178 ($B2) for you. Where it used to be 60 ($3C), it is now 204 ($CC), thus the buffer now starts at 972 ($03CC).

Loading Hints

After loading a program from cassette, you will get a load error. This can be remedied with POKE 45,PEEK(174): POKE 46,PEEK(175). This indicates to the VIC or 64 the end of your program or the start of variables. Data files will not load properly with the Invisible Loader in place.

If you SYS 828 and your disk drive is not on, then nothing will happen. Turn the drive on and an error message appears on the screen. You must restore the VIC or 64 by pressing the STOP and RESTORE keys. Insert a disk, type SYS 828, RETURN, and there it is. If you do not have a disk in the drive, then the red light will flash. The screen will also scroll up with nothing on it. STOP/ RESTORE, insert a disk, and type SYS 828. Before running it, be sure to save a copy to your disk or cassette. Call it "DIR".

I use this disk utility more than any other. Rarely do I LOAD "$",8. I just load "DIR",8 and run it and forget about it. It really seems invisible.

If you would rather not type in the program, I have the VIC version available. Send a blank cassette, an SASE mailer, and $3 to:

Kevin Gough
24 Daisy Lane
Wappingers Falls, NY 12590

Invisible Disk Directory

```
10 I=828
20 READ A:IF A=256 THEN 40
30 POKE I,A:I=I+1:GOTO 20
40 IF PEEK(65440)=135 THEN POKE 924,189:REM
   924 HOLDS 221 ON VIC, 189 ON 64
828 DATA 169,1,32,195,255,169,36
835 DATA 141,240,3,169,48,141,241
842 DATA 3,169,1,162,8,160,0
849 DATA 32,186,255,169,2,162,240
856 DATA 160,3,32,189,255,32,192
863 DATA 255,169,64,32,144,255,162
870 DATA 1,32,198,255,32,144,255
877 DATA 32,207,255,32,207,255,32
884 DATA 207,255,32,207,255,201,0
891 DATA 240,58,32,204,255,32,228
898 DATA 255,201,32,208,3,32,196
905 DATA 3,162,1,32,198,255,32
912 DATA 207,255,168,32,207,255,72
919 DATA 152,170,104,32,205,221,169
926 DATA 32,32,210,255,32,207,255
933 DATA 201,0,208,8,169,13,32
940 DATA 210,255,76,115,3,32,210
947 DATA 255,76,162,3,169,1,32
954 DATA 195,255,32,204,255,169,204
961 DATA 133,178,96,32,228,255,201
968 DATA 32,208,249,96,256
```

©

A Multicolor Atari Character Editor

Charles Brannon, Program Editor

This program makes colorful animation easy and fun. You'll find "ANTIC Aerobics" to be an invaluable tool for working with four-color characters. There's also a submarine chase game to illustrate these techniques, an entertaining game in its own right.

Remember the last time you played an arcade game? You probably controlled a realistic-looking ship, plane, race car, or even a Q*bert. If you tried to program such a game and ended up discouraged, here's the answer. Using one of the Atari's least used (and possibly most interesting) graphics modes, you can animate multicolor objects with simple PRINT statements.

If you've been programming for a while, you know about most of the Atari's 14 graphics modes (17 if you count GTIA modes). For example, GRAPHICS 8 is the high-resolution screen with the smallest controllable "dots." GRAPHICS 3 uses the least memory and gives you four-color graphics in a 40x24 format (each "dot" is as large as the text cursor). And there are the text modes, such as GRAPHICS 1 (double-wide), GRAPHICS 2 (double-wide and twice as high as normal text), and of course, GRAPHICS 0, the normal white-on-blue text screen.

Silicon Symbiosis

All these graphics modes are supported by the ANTIC chip, which has been called a video microprocessor. ANTIC's job is to tell the GTIA, an essentially "dumb" chip, how to display a TV screen. Your job is to tell the ANTIC how to format a screen. Fortunately, the Atari's operating system already knows how to set up graphics screens for the ANTIC.

But this doesn't mean that you can't "do it yourself." In fact, it's rather easy to create your own custom screens with all kinds of graphics modes mixed together. Although we won't go into detail here, you can refer to Craig Chamberlain's "How to Design Custom Graphics Modes" in *COMPUTE!'s First Book of Atari Graphics* if you'd like more information.

Hidden Modes

ANTIC can generate more graphics modes than most people think. For example, there is a special variation on GRAPHICS 0 that lets you design characters within a 9x8 matrix for true descenders (the "tail" on a g,j,p,q, or y). There's a special graphics mode "between" GRAPHICS 7 and GRAPHICS 8 that is a four-color mode with a resolution of 160x192 (some call it GRAPHICS 7½).

ANTIC 4 And 5

However, let's limit ourselves here to the five-color character modes. In GRAPHICS 1 and 2, you get four colors of text (for example, A, a, inverse A, and inverse a). Each character can have a different color, but you can have only 64 characters, and you are limited to one color per character. But two special ANTIC modes, ANTIC 4 and ANTIC 5 (or IRG 4 and 5 according to the hardware manual), allow four colors per character.

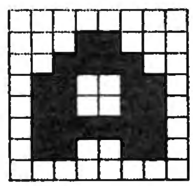
Unfortunately, the use of these modes is not intuitively obvious. It helps if you know binary (base two arithmetic). You don't really have to understand how to program characters in these modes to write games with them, as long as you have a utility to do it for you ("ANTIC Aerobics," found at the end of this article). But for those with an inclination to understand the details, the following discussion should be illuminating. Otherwise, you can skip ahead to "Using The Program."

Assumptions

Let's start by making a few assumptions for the sake of brevity: that you understand binary numbers, know how to create custom character sets, understand the relationship between COLOR and SETCOLOR, and have a good working knowledge of BASIC.

You know that when defining a normal Atari character you get eight bits or pixels horizontally and eight bytes vertically. The letter A would be defined in binary as (we'll use open boxes for zeros, and solid squares for ones):

```
........
...oo...
..oooo..
.oo..oo.
.oo..oo.
.oooooo.
.oo..oo.
........
```



Every bit represents one pixel (picture element, or "dot"). In the multicolor modes, it takes two bits to represent four colors (00,01,10,11), so the bits are "paired up." You still use only one byte per line, so you get only four pixels horizontally, although you still get eight lines vertically. Since the size of the character is the same as a GRAPHICS 0 character (in ANTIC 4), this implies that each pixel is twice as wide as a single-color pixel.

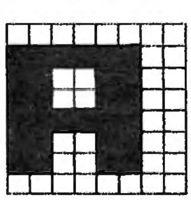
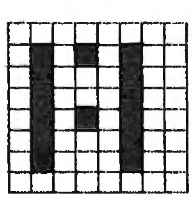
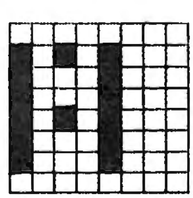
If you're using a standard character editor such as *SuperFont* or *Instedit*, you must remind yourself that you must reserve two bits per pixel.

When designing a four-color character, use the following combinations:

(00) = background color (COLOR 0, SETCOLOR 4)
(01) = COLOR 1 (SETCOLOR 0)
(10) = COLOR 2 (SETCOLOR 1)
(11) = COLOR 3 (SETCOLOR 2)

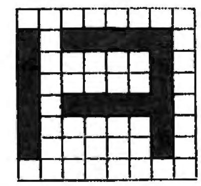
In addition, if you print the character in inverse video, the COLOR 3 bit pattern (binary 11) will be displayed with the color in the fourth color register (SETCOLOR 3). Here's what three different colored A's would look like:

```
........    ........    ........
oooooo..    .o.o.o..    o.o.o...
oo..oo..    .o...o..    o...o...
oo..oo..    .o...o..    o...o...
oooooo..    .o.o.o..    o.o.o...
oo..oo..    .o...o..    o...o...
oo..oo..    .o...o..    o...o...
........    ........    ........
COLOR 3     COLOR 2     COLOR 1
```

Now there's nothing to keep you from combining all the colors in a single character. For example, here's an A with its left side in COLOR 1, the right side in COLOR 2, and the top and middle segment in COLOR 3. To the right is the same character with bit pattern 00 shown as a period, bit pattern 01 as an *, bit pattern 10 as an @, and bit pattern 11 as a #:

```
........   ----
o.ooooo.   *##@
o.....o.   *--@
o.....o.   *--@
o.ooooo.   *##@
o.....o.   *--@
o.....o.   *--@
........   ----
```



If you try to program other shapes, however, such as an alien invader or a race car, you will find that you need more than one character per shape. Four pixels don't give you much to work with. But if you put two characters side by side, you're back in eight-bit business. As long as you're doing that, you can create matrices of two-by-two characters, or any size you like. You can create "building block" characters, "primary" shapes that you use to build larger objects. But if you try to make larger, more complex "pictures," you'll probably discover that the task of designing each character and piecing the characters together can be rather maddening. That's where ANTIC Aerobics comes in.

Using The Program

ANTIC Aerobics lets you draw a free-hand shape or picture that is 32 pixels wide and 16 pixels high. You can then "compile" the shape into a set of 16 characters. You display the shape as two rows of eight characters. If you put the shape into the character set on top of the alphabet, you could show it on the screen with a statement like:

```
200 PRINT "ABCDEFGH":PRINT "IJKLMNOP"
```

You can also place each shape into a string. When you PRINT the string, the shape appears. The string is made of eight characters, a cursor down, and eight cursor-lefts to back up the cursor under the first eight, then eight more characters. Following the COMPUTE! listing conventions, it would look like:

```
C$="ABCDEFGH{DOWN}{8 LEFT}GHIJKLMNOP"
```

If you also had a string filled with blanks (eight spaces, cursor down, eight cursor-lefts, and eight spaces), you could PRINT the blank string on top of the shape to blank it out. Animation made simple! Program 2 is a submarine game using shapes developed with ANTIC Aerobics. Take a look at the line-by-line explanation for more ideas on animation.

Display List Dickering

To go into the special ANTIC modes 4 or 5, you have to change the display list. Fortunately, this is fairly simple with a mode 4 display; you just change all the 2's in the display list to 4's. You can also POKE in 5's for the double-height ANTIC 5 mode. These two lines will do either one:

ANTIC 4:

```
GRAPHICS Ø:DL=PEEK(56Ø)+256*PEEK(561)+4
FOR I=2 TO 24:POKE DL+I,4:NEXT I:POKE DL-
  1,68
```

ANTIC 5:

```
GRAPHICS Ø:DL=PEEK(56Ø)+256*PEEK(561)+4
FOR I=2 TO 12:POKE DL+I,5:NEXT I:POKE DL-
  1,69
```

You might also want to disable the cursor with POKE 752,1.

How To Use ANTIC Aerobics

Use a joystick to draw. Press the trigger to set a point in the current color. To change colors, press either 0,1,2, or 3 (0 is used to erase). If you want to change a color, hold down SHIFT and type the number key. You will see a cursor above a 16-color bar (GTIA only; you'll see 16 densities of vertical lines if you have a CTIA, but you can still use the program). Move the joystick left or right to the color you want. Then push up or down to change the luminance (brightness). When you're through, press the trigger.

There are several other commands to make drawing easier and more fun. For example, to draw a line between two points, press P to set the first point (think "Plot"), then move the cursor to the second spot and press D ("Drawto"). If you move the cursor again and press D, another line will emanate from the original center point. If you want to draw from one line to another without having to reset the starting point, use CTRL-D. Each time you press CTRL-D, a line will be drawn from the last line. This makes it easy to draw lines at odd angles.

What if the cursor is too fast for you? The M command will give you a cursor speed from 0 (fast) to 9 (slow). Just press M and then the appropriate number key.

You can also use the insert line and delete line key (SHIFT-INSERT, SHIFT-DELETE) to insert or delete lines.

Use the S and L keys to either SAVE a shape or LOAD one previously saved. Enter the complete filename, i.e., C: for tape, or D: name for disk. If you see an error message, like "ERROR 162 ON SAVE", press a key to try again. The SAVE command will not work until you "compile" the shape with the C key (see below), since it SAVEs the character bytes, which aren't defined until you compile the shape. You can use the I (Index) command to view the disk directory. Press a key after each displayed name to view the next.

When you press C, the computer will scan the picture you've drawn and convert it into 16 characters. It will then show you what the picture would look like in ANTIC modes 4 (top) and 5 (bottom). To the right is the shape drawn in inverse video, so all bit-pair three's will be in another color (usually pink). Notice that ANTIC mode 5 has vertical pixels twice as high as ANTIC 5 – in fact, each pixel is the same size as a GRAPHICS 7 pixel. What we've got is GRAPHICS 7 resolution (or better) without the exorbitant memory consumption.

DATA Creator

The last option lets you create DATA statements from the characters you've defined. As with the SAVE command, you can write data only after you've compiled the shape. You will be asked to choose at which line number you want to start the DATA statements and the filename for the program you want to create.

The Atari will then write a series of line numbers and DATA statements to tape or disk. To merge these lines later with your own programs, use the ENTER command (ENTER "D:name" or ENTER"C:"). You have to write the lines that READ the data and POKE it into your character set wherever you want it to go (also see Program 2). Since each shape requires 16 characters, you can fit eight shapes into one character set.

If you ever get stuck, the program has a built-in Help function that gives a quick reference list of the commands. Press H, "?", or the Help key on the 1200XL. The commands will be given one at a time at the bottom of the screen. Press a key to advance each command. When you're ready to exit the program, press CTRL-Q.

Sub Attack Program Analysis

Here's a line-by-line explanation of "Sub Attack" (Program 2). We'll look at its structure in some detail as well as explore some programming tips and tricks.

Line 130: Lines 580-890 are the initialization routine. SUBS keeps track of how many "lives" you have. Line 575 prints from one to three miniature subs at the bottom of the screen. Each minisub is formed from custom characters.

Line 140: This line clears out applicable variables at the start of each game.

Line 150: This is part of the main loop. The hardware random number generator is used (53770) to decide on a 50/50 chance whether or not to put a ship on the screen. Ships are always spaced at least ten characters apart. Since each ship is eight characters long (including spaces), the closest two ships would be is two spaces.

Line 160: Here, one of the ships is picked. The characters for the ships are stored in a string. The statement is equivalent to R = INT(4*RND(1))* 2.

Line 170: POKE 766,1 disables cursor controls, since some of the ships contain control characters which PRINT would execute instead of display.

Line 180: This is the first line in the main loop. It continually checks to see if the high score has been topped. HSCR is initialized to 500 in line 590. If the high score is beat, a special subroutine is called, but (due to BEAT) only once per game.

Line 190: This checks for a change in the score. Many different routines add to or subtract from the score, so this one statement is responsible for noting a change from the previous score (OPTS, for "Old Points") and updating the score line. The POSITION statement centers the score.

Line 200: If there is no mine falling, and if the number in the hardware random number generator is greater than 200 (a 55/255 chance), then a mine position is chosen. If MS = 0, no mine will fall. Otherwise, MS holds the offset from the upper-left corner of screen memory.

Line 210: First, we reset 766, so we can execute control characters. This one line is the core of a tricky animation technique. Instead of moving each ship by drawing and erasing, each ship is placed at the right of the screen. The CHR$(254)'s are CTRL-DELETEs, which pull the line to the left. By PRINTing two of these, we can "scroll" the line to the left. It's possible to use INSERTs to push a line to the right.

Line 220: It's preferable to use short variables like JS (joystick) and FB (fire button) than the longer statements over and over again.

Line 230: FB = 0 if the button is pressed. If no torpedo is in "flight," then we set one up. If TORP is zero, no torpedo will be displayed or updated. Otherwise, TORP holds the actual screen memory location of the torpedo. FIRSTMOVE is set when the player makes any move, such as moving the sub or firing a torpedo. FIRSTMOVE is used to disable the mines falling until the player has begun to move. POKE 77,0 kills attract mode.

Line 240: JS is used as an index into an array containing -1's, 0's, and +1's. The appropriate offset (+1 for 7, right, -1 for 11, left) is added to the X (horizontal) position of the submarine. Only nonzero offsets are accepted, so the sub is always moving.

Line 250: Similar to 240, except for the vertical position of the sub. We also have a check for the START button here in case the player wants to restart the game in progress.

Line 260: If the submarine moves up or down, or changes direction, we erase the submarine before the new one is PRINTed.

Line 270: The submarine horizontal variable, SUBX, is updated and checked for wraparound. A single phrase: SUBX = 32-ABS(SUBX) will reverse the illegal -1 and 32 to the legal 31 and 0.

Line 280: If the vertical variable is out of range, we just remove the offset.

Line 290: SUB$ contains the characters for both directions of the submarine. The characters include a leading space if the sub is moving right, and a trailing space if the sub is moving left. The leading or trailing space erases the previous character when the sub moves, without having to erase the whole submarine (which is somewhat "flickery").

Line 300: This is the routine for moving the falling mine. If no mine is selected, then it's skipped.

Line 310: First, we erase the previous mine (if any). The mine is two characters wide, so this makes things complicated. The next position of the mine is found by adding 40 (each screen line is 40 characters long) to the mine position. If the mine has hit the bottom of the screen, it is removed from execution (since it hasn't hit anything).

Line 320: Shorthand, again. Using LOC over and over again is shorter than using SCR + MS. SCR holds the starting address of screen memory. We "look ahead" before we POKE in the mine's characters to check for a "collision."

Line 330: If nothing is hit (PEEK returned a zero for SPACE), the mine is POKEd into its new position, and we go on to the next routine at 480 (which updates the torpedo).

Line 340: Sound effect and explosion time. We assume we've hit the submarine or a torpedo. Color register three is POKEd with random colors, causing anything on the screen drawn using this register to flash and glow. The sixtieth of a second timer is set to zero.

Line 350: A loop to wait a sixtieth of a second. Not really necessary, but we want to slow down this part because the game was compiled.

Line 360: EXL(0-3) contains the left side of three explosion scenes, and EXR(0-3) holds the right side. The sound effect is arbitrary, but the volume is stepped progressively down.

Line 370: The mine is now removed.

Line 380: If the mine hit a torpedo, then both the torpedo and the mine are removed, and the player gets 100 points.

Lines 390-410: Otherwise, the sub was hit, and we flip it back and forth to illustrate its demise.

Line 415: One less submarine, but was it the last?

Line 420: Not if this line is executed. The "dead" sub is erased, and some variables are reset. Line 575 updates the number of little submarine symbols displayed.

Line 430: Start of the "game over" section. Check for high score.

Line 440: Mode 2 without text window. Turn off display list interrupt (to be safe). Messages.

Line 450: Score line. Notice that all four colors are used, upper/lowercase, inverse and normal video.

Line 460: A loop to wait for either START or the fire button to be pressed.

Line 470: Kill attract mode again, restart the game.

Line 480: Check to see if we should move the torpedo.

Line 490: Erase the old torpedo, if indeed the torpedo was where it should be (sometimes the scroll routine will pull a ship into the space where the torpedo was).

Line 500: Move the torpedo up (minus 40 characters per line). Check to see if the torpedo has gone off the "top" of the ocean. If so, deduct ten points, but don't let the score fall below zero.

Line 510: There is no sane reason to use LOC in place of TORP, but I'd done it, and didn't want to change lines 510-550 when I realized the redundancy. Anyway, we check to see if the new position is occupied (meaning a ship). If not (=0), the torpedo is placed into the spot *if* the torpedo position is under the water still.

Line 520: The mine checks to see if it hit the torpedo, and here the torpedo checks to see if it hit either the left or the right side of the mine. If so, we just reuse part of the mine explosion routine.

Lines 530-570: A different, complex sound and explosion. The explosion moves left and right from the collision until it runs out of ship characters to blow up. It *is* complicated. The score depends on how high your ship is in the water.

Line 575: A simple FOR/NEXT loop to print from one to three "subettes" as symbols of how many lives you have left.

Line 580: The start of the really Atari-specific stuff, where characters are initialized, arrays are set up, machine language is read in, and the display list modified. Actually, line 580 is merely a useless REM statement.

Line 590: The game is not started over with RUN, since this would clear out the high score. Instead, we just make sure that we do our DIMensions only once, and then set a flag (DIMMED) to make sure it won't happen again. If you're a novice player, or a 6000-pointer, you can modify HSCR here as a goal to reach.

Line 600: We modify the display list of the 24-line GR.0 display to make it a 24-line mixed-mode display (convenient). The cursor is also turned off here.

Line 610: These SETCOLOR statements come from the ANTIC Aerobics program, which generates them with the WRITE DATA STATEMENTS option.

Line 620: The top line (DL-1) is ANTIC 5, multicolor and double-height. Lines 2 to 23 are ANTIC 4, and the last line is ANTIC 6 (a.k.a. GRAPHICS 1). Line 10 is flagged for ANTIC as where the display list interrupt will occur.

Line 630: Screen memory.

Line 640: The character set is placed eight pages (2K) behind the top of memory, about 1K beneath the screen display. The character set pointer now causes ANTIC to display *our* character set.

Line 650: This important line checks to see if the character set has already been POKEd in previously. If so, why bother to do it again?

Line 660: We put the 128-byte character set up on the screen as four rows of 32 characters so you can watch the characters as they're being redefined.

Line 670: The long list of character set data starts at 1040. Four 8x2 shapes are READ into, and thereby replace, the lowercase and graphics symbols (for this game, no big loss).

Line 680: But then we overlay most of the punctuation and math symbols with the submarine characters. We still have the alphabet, the numbers, and a few punctuation marks free.

Line 690: And here we POKE the alphabet and numbers into the character set from the default ROM set at $E000 (57344).

Lines 700-710: Here we define a couple of characters, including the small submarine used on the score line. You can use the same character set in different modes, although the multicolor characters look odd in GRAPHICS 1, and the text is hard to decipher in ANTIC 4.

Line 730: This machine language section is used for the purely cosmetic purpose of dividing the screen into two parts, sea and sky, and giving us four separate colors for each half. The display list interrupt is easy to understand. It is just a bunch of LDAs (like PEEK, Load Accumulator, a special 6502 "variable") with each color and STA (Store Accumulator into memory, like POKE) into the hardware color registers. The ANTIC chip lets us synchronize this color change with any screen line we choose, and we chose line 10 in line 620 (128 is added to the mode byte).

Line 740: We clear the screen (PUT#6,125) and draw the characters for the sun and clouds.

Line 750: We tell the operating system where our display list routine is, low byte 0, high byte 6=$0600, 1536, "page six." A single POKE to 54286 tells ANTIC to "start interrupting."

Lines 760-840: The characters for each ship, arranged as eight characters for the top half of the ship and eight for the bottom, are concatenated into a single string.

Line 850: The string holding the characters for the submarine is set up.

Lines 860-870: We READ in the +1,0, and -1 values for the joystick.

Line 880: The initial position of the submarine is set, and the explosion characters are read.

Line 890: That's it for initialization!

Line 900: DATA for the explosion characters.

Lines 910-1030: A special subroutine when you beat the high score.

Lines 1040-1846: Last, but certainly not least, the DATA statements for over 530 bytes of custom character data. This is where almost all your typing mistakes will be made.

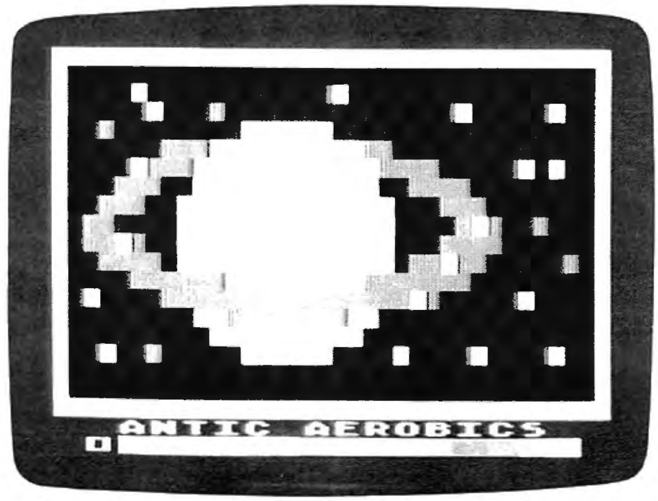
Line 1860: Here are the bytes for the small machine language display list interrupt routine. Initially, we do a store into \$D40A (any write to \$D40A). This makes ANTIC "hold down" the 6502's READY line, effectively freezing the microprocessor until the TV scanning beam hits the right edge of the screen. We don't want to change the colors in the middle of a line, or it would be quite jagged. This handy feature lets us wait until the beam is off the left side of the TV before we make the color change. See Program 3 for a disassembly of the display list interrupt routine (it looks long, but it's only 26 bytes).

Program 1: ANTIC Aerobics

```
100 REM ANTIC Aerobics
110 CHSET=(PEEK(106)-8)*256
120 GOSUB 300:GOSUB 430:SPEED=4
130 LOCATE X,Y,Z
140 COLOR 1+(Z=1)
150 PLOT X,Y
160 ST=STICK(0)
170 IF PEEK(20)<SPEED THEN 160
180 POKE 20,0
190 COLOR Z:PLOT X,Y
200 IF PEEK(732) THEN POKE 732,0:POK
    E 764,102
210 IF PEEK(764)<255 THEN GOSUB 510
220 IF STRIG(0)=0 THEN COLOR CURR:PL
    OT X,Y:LET COMPILED=0:IF ST=15 T
    HEN 130
230 IF ST=15 THEN 140
240 X=X+DX(ST):Y=Y+DY(ST)
250 IF X<XL THEN X=XH
260 IF X>XH THEN X=XL
270 IF Y<YL THEN Y=YH
280 IF Y>YH THEN Y=YL
290 GOTO 130
300 DIM DX(15),DY(15),ML$(20),COL(5)
    ,FN$(20)
310 CURR=1:XL=4:XH=35:YL=4:YH=19:X=X
    L:Y=YL:SX=XL:SY=YL:GOSUB 1320
320 RESTORE :FOR I=5 TO 15:READ A:DX
    (I)=A:NEXT I
330 FOR I=5 TO 15:READ A:DY(I)=A:NEX
    T I
340 DATA 1,1,1,0,-1,-1,-1,0,0,0,0
350 DATA 1,-1,0,0,1,-1,0,0,1,-1,0
360 OPEN #1,4,0,"K:"
370 FOR I=0 TO 15:READ A:POKE CHSET+
    768+I,A:NEXT I
380 DATA 0,255,240,240,240,240,255,0
    ,0,240,240,240,240,240,240,0
390 IF PEEK(CHSET+257)<>60 THEN FOR
    I=0 TO 511:POKE CHSET+I,PEEK(573
    44+I):NEXT I
400 FOR I=1 TO 6:POKE CHSET+504+I,25
    5:NEXT I:POKE CHSET+504,0:POKE C
    HSET+511,0
410 IF PEEK(CHSET+784)<>17 THEN FOR
    I=1 TO 15:FOR J=0 TO 7:POKE CHSE
    T+I*8+776+J,I+I*16:NEXT J:NEXT I
420 RETURN
430 PMBASE=(PEEK(106)-16)*256:GRID=1
440 POKE 54279,PMBASE/256
450 POKE 53277,3:POKE 559,62:POKE 62
    3,4
460 FOR I=0 TO 3:POKE 704+I,2:POKE 5
    3248+I,64+I*32:POKE 53256+I,3:NE
    XT I
470 P0=PMBASE+1024:BP=85:IF PEEK(P0+
    64)=BP THEN RETURN
480 FOR I=64 TO 190 STEP 8
490 FOR J=0 TO 7:POKE P0+I+J,BP:POKE
     P0+256+I+J,BP:POKE P0+512+I+J,B
    P:POKE P0+768+I+J,BP:NEXT J:BP=2
    55-BP
500 NEXT I:RETURN
510 GET #1,A:POKE 711,70
520 IF A=ASC("W") THEN IF COMPILED T
    HEN 1810
530 IF A=87 THEN A=83:GOTO 800
540 IF A=17 THEN GRAPHICS 0:POKE 532
    77,0:FOR I=0 TO 3:POKE 53248+I,0
    :NEXT I:END
550 IF A=ASC("G") THEN GRID=1-GRID:F
    OR I=0 TO 3:POKE 53248+I,(64+I*3
    2)*GRID:NEXT I:RETURN
560 IF A>47 AND A<52 THEN CURR=A-48:
    POKE 711,PEEK(707+CURR+5*(CURR=0
    )):RETURN
570 IF A=125 THEN GOSUB 1320:LET COM
    PILED=0:GOSUB 430:RETURN
580 IF A=ASC("C") THEN LET COMPILED=
    1:POP :GOTO 1020
590 IF A=7 THEN CREG=704:A=PEEK(CREG
    ):GOTO 620
600 IF A<33 OR A>35 AND A<>41 THEN 7
    40
610 CREG=708+A-33-4*(A=41):A=PEEK(CR
    EG)
620 C=INT(A/16):L=A-C*16:POKE 87,1:C
    OLOR 32:PLOT 0,11:DRAWTO 19,11
630 POSITION C+2,11:PUT #6,95
640 T=C+DX(STICK(0)):IF T<0 OR T>15
    THEN T=16-ABS(T)
650 L=L-2*DY(STICK(0)):IF L<0 OR L>1
    4 THEN L=16-ABS(L)
660 A=C*16+L:POKE CREG,A:IF CREG=704
     THEN POKE 705,A:POKE 706,A:POKE
     707,A
670 IF STICK(0)<15 THEN POSITION C+2
    ,11:? #6;" ";:C=T
680 IF STRIG(0)=0 THEN 710
690 IF PEEK(20)<SPEED THEN 680
700 POKE 20,0:GOTO 630
710 IF STRIG(0)=0 THEN 710
720 GOSUB 1550:POSITION 3,11:? #6;"E
    ntic aerobics":POKE 711,PEEK(707
    +CURR+5*(CURR=0))
730 POKE 87,3:RETURN
740 IF A<>ASC("P") THEN 760
750 SX=X:SY=Y:COLOR CURR:PLOT X,Y:FO
```

```
   R W=15 TO 0 STEP -1:SOUND 0,W,12
   ,W:NEXT W:POP :LET COMPILED=0:GO
   TO 130
760 IF A=ASC("D") THEN COLOR CURR:PL
   OT SX,SY:DRAWTO X,Y:LET COMPILED
   =0:POP :GOTO 130
770 IF A=4 THEN COLOR CURR:PLOT SX,S
   Y:DRAWTO X,Y:SX=X:SY=Y:POP :LET
   COMPILED=0:GOTO 130
780 IF A=ASC("L") THEN 1560
790 IF A=ASC("S") THEN IF COMPILED T
   HEN 1450
800 IF A=ASC("S") THEN POKE 87,1:GOS
   UB 1550:POSITION 4,11:? #6;"not
   compiled":GET #1,A:GOTO 1520
810 IF A<>ASC("M") THEN 840
820 GET #1,A:IF A<48 OR A>57 THEN RE
   TURN
830 SPEED=A-48:RETURN
840 IF A<>ASC("?") AND A<>ASC("H") T
   HEN 890
850 RESTORE 1980:POKE 87,1
860 READ FN$:IF FN$="END" THEN 880
870 GOSUB 1550:POSITION 10-LEN(FN$)/
   2,11:? #6;FN$:GET #1,A:GOTO 860
880 GOTO 1520
890 IF A<>156 THEN 930
900 FOR ROW=Y*10 TO 180 STEP 10:FOR
   COL=1 TO 8:POKE SCR+ROW+COL,PEEK
   (SCR+ROW+10+COL):NEXT COL:NEXT R
   OW
910 FOR COL=1 TO 8:POKE SCR+ROW+COL,
   0:NEXT COL:LOCATE X,Y,Z
920 RETURN
930 IF A<>157 THEN 970
940 FOR ROW=190 TO Y*10+10 STEP -10
950 FOR COL=1 TO 8:POKE SCR+ROW+COL,
   PEEK(SCR+ROW-10+COL):NEXT COL:NE
   XT ROW
960 Z=0:GOTO 910
970 IF A<>ASC("I") THEN 1010
980 TRAP 1000:OPEN #2,6,0,"D:*.*":PO
   KE 87,1
990 INPUT #2,FN$:GOSUB 1550:POSITION
    1,11:? #6;FN$;:GET #1,A:GOTO 99
   0
1000 CLOSE #2:GOTO 1520
1010 RETURN
1020 FOR I=0 TO 3:POKE 53248+I,0:NEX
    T I
1030 SCR=PEEK(88)+256*PEEK(89)
1040 FOR ROW=4 TO 19
1050 FOR COL=1 TO 8
1060 LOC=SCR+ROW*10+COL:A=PEEK(LOC)
1070 POKE LOC,255-A
1080 C=COL-1:R=ROW-4:IF R>7 THEN R=R
    +56
1090 POKE CHSET+512+C*8+R,A
1100 POKE LOC,A
1110 NEXT COL:NEXT ROW
1120 FOR I=0 TO 4:COL(I)=PEEK(708+I)
    :NEXT I
1130 GRAPHICS 0:SCR=PEEK(88)+256*PEE
    K(89):DL=PEEK(560)+256*PEEK(561
    )+4
1140 POKE 752,1:POKE 756,CHSET/256
1150 FOR I=0 TO 4:POKE 708+I,COL(I):
    NEXT I:POKE 711,70
1160 POKE DL-1,4+64:FOR I=2 TO 10:PO
    KE DL+I,4:NEXT I:POKE DL+11,5:P
    OKE DL+12,5:POKE DL+13,5:POKE D
    L+14,6
1170 POKE DL+15,65:POKE DL+16,PEEK(5
    60):POKE DL+17,PEEK(561)
1180 FOR I=0 TO 1:FOR J=1 TO 8:FOR K
    =0 TO 1:FOR L=0 TO 1
1190 POKE SCR+I*40+L*10+120*K+J+284,
    63+I*8+J+L*128:NEXT L:NEXT K:NE
    XT J:NEXT I
1200 POSITION 0,13:? "PRESS fire TO
    RETURN"
1210 IF STRIG(0) THEN 1210
1220 REM RESTORE
1230 GOSUB 1320:FOR I=0 TO 4:POKE 70
    8+I,COL(I):NEXT I
1240 SCR=PEEK(88)+256*PEEK(89)
1250 FOR ROW=4 TO 19
1260 FOR COL=1 TO 8
1270 C=COL-1:R=ROW-4:IF R>7 THEN R=R
    +56
1280 A=PEEK(CHSET+512+C*8+R)
1290 POKE SCR+ROW*10+COL,A
1300 NEXT COL:NEXT ROW
1310 GOTO 130
1320 REM SET UP GR.3+16 SCREEN
1330 RESTORE 1350:FOR I=1 TO 16:READ
     A:ML$(I)=CHR$(A):NEXT I
1340 POKE 513,INT(ADR(ML$)/256):POKE
     512,ADR(ML$)-256*PEEK(513)
1350 DATA 72,169,192,141,10,212,141,
    27,208,169,10,141,26,208,104,64
1360 GRAPHICS 3+16:POKE 559,0:SCR=PE
    EK(88)+256*PEEK(89)
1370 COLOR 1:PLOT XL-2,YL-2:DRAWTO X
    H+2,YL-2:DRAWTO XH+2,YH+2:DRAWT
    O XL-2,YH+2:DRAWTO XL-2,YL-2
1380 DL=PEEK(560)+256*PEEK(561)+4
1390 POKE DL+23,6+128:POKE DL+24,2:P
    OKE 54286,192
1400 POKE 87,1:POSITION 3,11:? #6;"E
    ntic aerobics":POKE 87,3
1410 FOR I=1 TO 15:POKE SCR+244+I*2,
    97+I:POKE SCR+245+I*2,97+I:NEXT
     I:POKE SCR+244,96:POKE SCR+245
    ,97
1420 POKE 756,CHSET/256:POKE 559,62
1430 FOR I=0 TO 3:POKE 53248+I,64+I*
    32:POKE 53256+I,3:NEXT I
1440 RETURN
1450 REM SAVE ROUTINE
1460 POKE 87,1:GOSUB 1550:POSITION 0
    ,11:? #6;"s":GOSUB 1650
1470 TRAP 1500:OPEN #2,8,0,FN$
1480 FOR I=0 TO 127:PUT #2,PEEK(CHSE
    T+512+I):NEXT I:FOR I=0 TO 4:PU
    T #2,PEEK(708+I):NEXT I
1490 PUT #2,PEEK(704):CLOSE #2:GOTO
    1520
1500 GOSUB 1550:POSITION 1,11:? #6;"
    ERROR ";PEEK(195);" ON SAVE":CL
    OSE #2
1510 GET #1,A
1520 GOSUB 1550
1530 POSITION 3,11:? #6;"antic aerob
    ics"
1540 POKE 54286,192:POKE 87,3:TRAP 3
    2767:RETURN
1550 COLOR 32:PLOT 0,11:DRAWTO 19,11
    :RETURN
1560 REM LOAD ROUTINE
1570 POKE 87,1:GOSUB 1550:POSITION 0
    ,11:? #6;"l":GOSUB 1650
1580 TRAP 1620:OPEN #2,4,0,FN$
1590 FOR I=0 TO 127:GET #2,A:POKE CH
```

```
      SET+512+I,A:NEXT I:FOR I=0 TO 4
      :GET #2,A:COL(I)=A:NEXT I
1600 GET #2,A:FOR I=0 TO 3:POKE 704+
      I,A:NEXT I
1610 CLOSE #2:POKE 54286,192:POP :TR
      AP 32767:LET COMPILED=1:GOTO 12
      20
1620 GOSUB 1550:POSITION 1,11:? #6;"
      ERROR ";PEEK(195);" ON LOAD":CL
      OSE #2
1630 GET #1,A
1640 GOTO 1520
1650 REM FILENAME INPUT
1660 POSITION 1,11:? #6;"fn";CHR$(15
      9);
1670 ZL=1
1680 POSITION 4+ZL,11:PUT #6,223
1690 GET #1,A
1700 IF A=155 THEN 1790
1710 IF A=126 THEN IF ZL>1 THEN ZL=Z
      L-1:COLOR 32:PLOT 5+ZL,11:GOTO
      1680
1720 IF NUM AND (A<48 OR A>57) THEN
      1690
1730 IF NUM=0 AND ZL=1 AND A<65 OR A
      >90 THEN 1690
1740 IF A=42 OR A=46 OR A=58 THEN 17
      60
1750 IF (A<48 OR A>57) AND (A<65 OR
      A>90) THEN 1690
1760 IF ZL=15 THEN 1690
1770 POSITION 4+ZL,11:PUT #6,A:FN$(Z
      L)=CHR$(A):ZL=ZL+1
1780 GOTO 1680
1790 NUM=0:IF ZL=1 THEN POP :GOTO 15
      20
1800 FN$=FN$(1,ZL-1):RETURN
1810 REM WRITE DATA
1820 POKE 87,1:GOSUB 1550:POSITION 1
      ,11:? #6;"ln";CHR$(159);:NUM=1:
      GOSUB 1670
1830 LN=0:FOR I=1 TO LEN(FN$):A=ASC(
      FN$(I))-48:IF A>=0 AND A<10 THE
      N LN=LN*10+A:NEXT I
1840 IF I<LEN(FN$) THEN POP
1850 GOSUB 1550:GOSUB 1650
1860 TRAP 1950:OPEN #2,8,0,FN$
1870 PRINT #2;LN;" ";:FOR I=0 TO 4:A
      =PEEK(708+I):C=INT(A/16):L=A-C*
      16
1880 PRINT #2;"SE.";I;",";C;",";L;:I
      F I<4 THEN PUT #2,58
1890 NEXT I:PRINT #2:LN=LN+10
1900 FOR I=0 TO 127 STEP 8
1910 PRINT #2;LN;" DATA ";
1920 FOR J=0 TO 7:PRINT #2;PEEK(CHSE
      T+512+I+J);:IF J<7 THEN PUT #2,
      44
1930 NEXT J:PRINT #2:LN=LN+10:NEXT I
1940 TRAP 32767:CLOSE #2:GOTO 1520
1950 GOSUB 1550:POSITION 1,11:? #6;"
      #";PEEK(195);" ON WRITE"
1960 GET #1,A
1970 CLOSE #2:GOTO 1520
1980 DATA C :COMPILE
1990 DATA 1-3 :COLOR
2000 DATA shift 1-3 :SETCOLOR
2010 DATA S :SAVE
2020 DATA L :LOAD
2030 DATA I :DISK INDEX
2040 DATA W :WRITE DATA STMTS
2050 DATA M :MOTION (0-9)
```



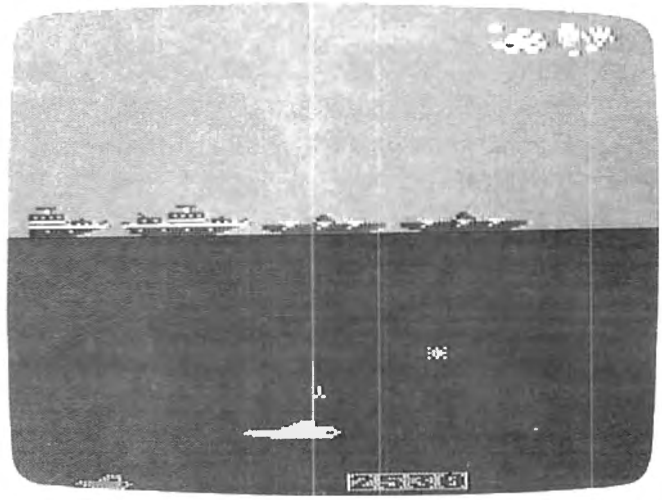
A shape resembling the planet Saturn being edited with the ANTIC Aerobics Editor.

```
2060 DATA P :PLOT
2070 DATA D :DRAWTO
2080 DATA ctrl D :DRAWTHRU
2090 DATA G : GRID ON/OFF
2100 DATA ctrl G:GRID COLOR
2110 DATA shift INSERT:LINE
2120 DATA shift DELETE:LINE
2130 DATA ctrl Q :QUIT
2140 DATA END
```

Program 2: Sub Attack – An Example Game

```
100 REM  SUB ATTACK
110 REM
120 REM
130 GOSUB 580:SUBS=3:GOSUB 575:REM N
     UMBER OF LIVES
140 PTS=0:BEAT=0:DX=0:OPTS=PTS:FIRST
     MOVE=0
150 X=1:IF PEEK(53770)<128 THEN 180
160 R=INT(4*PEEK(53770)/256)*2
170 POKE 766,1:POSITION 31,8:? SHIP$
     (R*8+1,R*8+8):POSITION 31,9:? SH
     IP$(R*8+9,R*8+16)
180 IF PTS>HSCR AND BEAT=0 THEN GOSU
     B 910
190 IF PTS<>OPTS THEN POSITION 10-LE
     N(STR$(SCORE))/2,23:? PTS;"  ";:
     OPTS=PTS
200 IF MS=0 AND FIRSTMOVE AND PEEK(5
     3770)>200 THEN MS=INT(30*PEEK(53
     770))/256+400
210 POKE 766,0:POSITION 0,9:? CHR$(2
     54);CHR$(28);CHR$(254);:X=X+1:IF
      X=10 THEN 150
220 JS=STICK(0):FB=STRIG(0)
230 IF FB=0 AND TORP=0 THEN TORP=SCR
     +SUBY*40-40+SUBX+7*(DX=1):FIRSTM
     OVE=TORP:POKE 77,0
240 IF DX(JS)<>0 THEN DX=DX(JS):POKE
      77,0
250 DY=DY(JS):IF PEEK(53279)=6 THEN
     100
260 IF DY OR DX<>ODX THEN POSITION S
     UBX,SUBY:? "{8 SPACES}";:ODX=DX:
     FIRSTMOVE=DX:POKE 77,0
270 SUBX=SUBX+DX:IF SUBX<0 OR SUBX>3
     1 THEN POSITION SUBX-DX,SUBY:? "
     {8 SPACES}";:SUBX=32-ABS(SUBX)
280 SUBY=SUBY+DY:IF SUBY<11 OR SUBY>
```

```
22 THEN SUBY=SUBY-DY
290 P=1+8*(DX<1):POSITION SUBX,SUBY:? SUB$(P,P+7);"{LEFT}";
300 IF MS=0 THEN 480
310 POKE SCR+MS,0:POKE SCR+MS+1,0:MS=MS+40:IF MS>919 THEN MS=0:GOTO 480
320 LOC=SCR+MS:P1=PEEK(LOC):P2=PEEK(LOC+1)
330 IF P1=0 AND P2=0 THEN POKE LOC,102:POKE LOC+1,103:GOTO 480
340 FOR V=12 TO 0 STEP -3:FOR I=0 TO 2:POKE 711,PEEK(53770):POKE 20,0
350 IF PEEK(20)<1 THEN 350
360 POKE LOC,EXL(I):POKE LOC+1,EXR(I):SOUND 0,70,8,V:SOUND 1,PEEK(53770),0,V:NEXT I:NEXT V
370 POKE LOC,0:POKE LOC+1,0
380 IF P1=112 OR P2=112 THEN MS=0:TORP=0:PTS=PTS+100:GOTO 180
390 FOR I=150 TO 0 STEP -5:POSITION SUBX,SUBY:? SUB$(FL*8+1,FL*8+8);:FL=1-FL
400 FOR J=I TO 1 STEP -20
410 SOUND 0,J/10,8,I/10:NEXT J:NEXT I
415 SUBS=SUBS-1:IF SUBS=0 THEN 430
420 POSITION SUBX,SUBY:? "{8 SPACES}";:SUBX=16:SUBY=22:DX=0:DY=0:MS=0:GOSUB 575:GOTO 180
430 IF PTS>HSCR THEN HSCR=PTS
440 GRAPHICS 18:POKE 54286,64:SETCOLOR 4,9,14:POSITION 5,0:? #6;"GAME OVER"
450 POSITION 5,5:? #6;"SCORE:";PTS:POSITION 5,11:? #6;"press start"
460 IF PEEK(53279)<>6 AND STRIG(0) THEN 460
470 POKE 77,0:GOTO 100
480 IF TORP=0 THEN 180
490 IF PEEK(TORP)=112 THEN POKE TORP,0
500 TORP=TORP-40:IF TORP<SCR+360 THEN TORP=0:PTS=(PTS-10)*(PTS>10):GOTO 180
510 LOC=TORP:IF PEEK(LOC)=0 THEN POKE LOC,112*(TORP>SCR+400):GOTO 180
520 IF PEEK(TORP)=102 OR PEEK(TORP)=103 THEN LOC=SCR+MS:P1=112:GOTO 340
530 FOR V=14 TO 0 STEP -2:FOR I=2 TO 3:L=0:R=0
540 FOR UP=0 TO 40 STEP 40:POKE 711,PEEK(53770)
550 POKE LOC-L-UP,EXL(I):POKE LOC+R-UP,EXR(I):A1=(PEEK(LOC-L-1)<>0):A2=(PEEK(LOC+R+1)<>0):NEXT UP:L=L+A1:R=R+A2
560 SOUND 0,L+R,0,V:IF A1 OR A2 THEN 540
570 NEXT I:NEXT V:TORP=0:PTS=PTS+40+(23-SUBY)*5:GOTO 180
575 POSITION 1,23:? "{7 SPACES}";:FOR I=1 TO SUBS:POSITION I*2-1,23:? "[\";:NEXT I:RETURN
580 REM Initialization
590 IF NOT DIMMED THEN DIM DX(15),DY(15),SHIP$(128),SUB$(16),EXL(3),EXR(3),MSG$(100):LET DIMMED=1:HSCR=500
```



An example of edited graphics in the Atari game "Sub Attack."

```
600 GRAPHICS 0:DL=PEEK(560)+256*PEEK(561)+4:POKE 752,1
610 SETCOLOR 0,11,4:SETCOLOR 1,0,12:SETCOLOR 2,1,10:SETCOLOR 3,4,6:SETCOLOR 4,10,8
620 POKE DL-1,69:FOR I=2 TO 23:POKE DL+I,4:NEXT I:POKE DL+10,128+4:POKE DL+24,6
630 SCR=PEEK(88)+256*PEEK(89)
640 CHSET=(PEEK(106)-8)*256:POKE 756,CHSET/256
650 IF PEEK(CHSET+20)=85 THEN 740
660 FOR I=0 TO 3:FOR J=0 TO 31:POKE SCR+I*40+80+2+J,I*32+J:NEXT J:NEXT I
670 RESTORE 1040:FOR I=512 TO 1023:READ A:POKE CHSET+I,A:NEXT I
680 FOR I=0 TO 127:READ A:POKE CHSET+I,A:NEXT I
690 FOR I=128 TO 511:POKE CHSET+I,255-PEEK(57344+I):NEXT I
700 FOR I=0 TO 7:POKE CHSET+208+I,255:NEXT I:POKE CHSET+214,239
710 FOR I=0 TO 15:READ A:POKE CHSET+472+I,A:NEXT I
730 FOR I=0 TO 25:READ A:POKE 1536+I,A:NEXT I
740 PUT #6,125:POSITION 30,0:FOR I=0 TO 7:PUT #6,I:NEXT I
750 POKE 512,0:POKE 513,6:POKE 54286,192
760 RESTORE 770:FOR I=1 TO 64:READ A:SHIP$(I)=CHR$(A):NEXT I
770 DATA 160,160,160,160,160,160,160,160
780 DATA 8,9,10,11,12,13,14,15
790 DATA 32,32,18,19,20,21,22,23
800 DATA 24,25,26,27,28,29,30,31
810 DATA 32,97,98,99,100,101,32,32
820 DATA 104,105,106,107,108,109,110,111
830 DATA 160,160,160,160,160,160,160,160
840 DATA 120,121,122,123,124,125,32,32
850 FOR I=1 TO 16:SUB$(I)=CHR$(I+31):NEXT I:SUB$(16)=CHR$(32)
860 FOR I=5 TO 15:READ A:DX(I)=A:READ A:DY(I)=A:NEXT I
```

```
870 DATA 1,1,1,-1,1,0,0,0,-1,1,-1,-1
    ,-1,0,0,0,0,1,0,-1,0,0
880 SUBX=16:SUBY=22:FOR I=0 TO 3:REA
    D A,B:EXL(I)=A:EXR(I)=B:NEXT I
890 RETURN
900 DATA 246,247,208,209,254,255,0,0
910 HSCR=PTS:MSG$="::::::::::::::::
    :::":POKE 711,90
920 MSG$(21)="CONGRATULATIONS::::new
    :::high:::score::::":MSG$(LEN(MS
    G$)+1)=STR$(PTS)
930 MSG$(LEN(MSG$)+1)=":::::::::::::
    ::::::::":CX=31
940 FOR I=1 TO LEN(MSG$)-20
950 SOUND 0,10,8,8
960 POSITION 1,23:? MSG$(I,I+17);:PO
    KE 20,0
970 IF PEEK(20)<2 THEN 970
980 SOUND 0,30,8,8
990 POSITION CX,2:? "qrstu ";:POKE 2
    0,0:CX=CX-1:IF CX=0 THEN POSITIO
    N CX,2:? "{7 SPACES}";:CX=31
1000 IF PEEK(20)<2 THEN 1000
1010 NEXT I:BEAT=1
1020 COLOR 32:PLOT 0,23:DRAWTO 18,23
     :PLOT 18,23:PLOT 0,2:DRAWTO 39,
     2:SOUND 0,0,0,0:GOTO 575
1030 RETURN
1040 REM  GAME CHARACTERS FOLLOW
1050 DATA 168,0,0,0,42,0,0,0
1060 DATA 0,2,42,170,10,168,0,0
1070 DATA 0,160,2,170,160,42,0,0
1080 DATA 0,0,128,0,160,0,0,0
1090 DATA 15,63,63,63,15,0,0,0
1100 DATA 192,240,242,240,192,0,160,0
1110 DATA 0,34,10,136,2,168,0,0
1120 DATA 0,32,160,136,0,0,0,0
1130 DATA 0,0,0,0,85,85,5,0
1140 DATA 0,0,192,63,95,117,85,21
1150 DATA 0,0,0,192,85,85,85,85
1160 DATA 1,5,53,31,87,93,85,85
1170 DATA 64,80,84,244,213,85,85,85
1180 DATA 0,0,48,15,87,93,85,85
1190 DATA 0,0,0,240,213,85,85,85
1200 DATA 0,0,0,0,85,85,80,0
1210 DATA 128,8,3,143,3,128,8,0
1220 DATA 2,48,240,252,224,200,0,128
1230 DATA 0,0,0,0,0,0,5,7
1240 DATA 0,0,8,85,42,42,85,119
1250 DATA 0,130,24,64,0,0,84,116
1260 DATA 0,2,33,0,0,0,0,0
1270 DATA 0,2,24,145,2,0,0,0
1280 DATA 0,4,0,32,80,0,0,0
1290 DATA 0,0,21,10,5,0,0,0
1300 DATA 0,0,85,170,117,85,21,5
1310 DATA 7,10,85,170,215,85,85,85
1320 DATA 119,170,85,170,93,85,85,85
1330 DATA 116,170,85,170,117,85,85,85
1340 DATA 0,0,85,170,215,85,85,85
1350 DATA 0,0,84,170,85,84,80,64
1360 DATA 0,0,0,128,0,0,0,0
1370 DATA 0,0,0,0,0,0,0,0
1380 DATA 0,0,0,0,0,0,64,16
1390 DATA 0,0,0,0,0,0,0,0
1400 DATA 0,0,0,5,0,0,85,119
1410 DATA 0,0,0,84,128,128,85,119
1420 DATA 0,0,0,0,0,0,64,64
1430 DATA 0,4,1,5,1,4,0,0
1440 DATA 0,68,80,84,80,68,0,0
1450 DATA 0,0,80,170,21,5,5,0
1460 DATA 85,117,85,170,85,213,85,21
1470 DATA 80,208,80,170,85,93,85,85
1480 DATA 85,170,85,170,85,85,85,85
1490 DATA 85,170,85,170,85,213,85,85
1500 DATA 64,129,65,170,85,93,85,85
1510 DATA 16,85,221,170,85,85,80,0
1520 DATA 0,1,4,170,80,0,0,0
1530 DATA 0,16,16,16,16,136,0,0
1540 DATA 1,0,0,5,31,31,5,0
1550 DATA 84,5,4,85,213,86,85,0
1560 DATA 0,80,0,80,85,89,84,0
1570 DATA 0,0,0,1,85,64,0,0
1580 DATA 4,17,4,84,64,0,0,0
1590 DATA 32,2,51,15,131,8,128,0
1600 DATA 34,0,200,240,194,48,0,8
1610 DATA 0,0,0,0,5,1,0,0
1620 DATA 0,0,0,2,85,93,85,5
1630 DATA 20,20,20,170,85,93,85,85
1640 DATA 20,20,20,170,85,93,85,85
1650 DATA 0,0,0,128,85,93,85,84
1660 DATA 0,0,0,0,84,64,0,0
1670 DATA 0,128,8,32,15,131,3,2
1680 DATA 0,0,194,192,242,252,192,50
1690 DATA 0,0,0,0,0,0,0,0
1700 DATA 0,0,0,0,5,42,1,0
1710 DATA 0,0,0,0,85,165,85,0
1720 DATA 0,0,0,3,85,85,85,85
1730 DATA 0,5,21,255,85,85,85,85
1740 DATA 85,85,105,255,85,85,85,85
1750 DATA 0,0,0,255,85,84,85,85
1760 DATA 0,0,0,192,80,20,80,64
1770 DATA 0,0,0,3,5,20,5,1
1780 DATA 1,1,1,255,85,21,85,85
1790 DATA 84,85,165,255,85,85,85,85
1800 DATA 0,64,80,255,85,85,85,85
1810 DATA 0,0,0,240,85,85,85,85
1820 DATA 0,0,0,0,85,90,85,64
1830 DATA 0,0,0,0,80,168,64,0
1840 DATA 0,0,0,0,0,0,0,0
1845 DATA 0,0,1,63,127,31,7,0
1846 DATA 120,248,200,254,251,254,25
     2,0
1850 REM Type the following carefull
     y--Machine Language!
1860 DATA 72,169,6,141,10,212,141,22
     ,208,169,40,141,23,208,169,10,1
     41,24,208,169,128,141,26,208,10
     4,64
1870 END
```

Program 3:
Disassembly Of The Display List Interrupt Routine

```
72           PHA          ;Since this is an interrupt, we want to
                           save any registers we use so that when
                           we return from the interrupt, the
                           original routine won't notice anything.
                           PHA means to "push" the accumulator
                           onto the stack. The stack will hold the
                           previous value in the accumulator
                           until we "pull" it off.
169 6        LDA #6       ;Grey (0*16+6)
141 10 212   STA $D40A    ;WSYNC (wait for synchronization)
141 22 208   STA $D018    ;Color register zero (hardware)
169 40       LDA #40      ;2*16+8, light orange
141 23 208   STA $D019    ;Color register one. Since ML is so fast,
                           the TV beam still hasn't reappeared.
                           We'll be able to make all our changes
                           without having to store to WSYNC
                           again.
169 10       LDA #10      ;Light white
141 24 208   STA $D01A    ;Color register two
169 128      LDA #128     ;8*16+0, dark blue, for the ocean
141 26 208   STA $D01C    ;Background color register
104          PLA          ;Restore accumulator
64           RTI          ;Return from Interrupt (like RETURN
                           from a GOSUB)
```

©

High Speed Mazer

Gary E Marsa

This update of previously published "Maze Generator" uses machine language to construct a random maze in less than two seconds – for PET, VIC, and 64. Also, there's "Munchmaze," a fast-action strategy game to show off the maze utility, with versions for the PET and 64.

If you tried Charles Bond's "Maze Generator" in the December 1981 COMPUTE!, you'll remember how fascinating it was to watch the maze being constructed on the screen right before your eyes. It's a clever program and lacks only one thing – speed. It takes my PET about 38 seconds to construct a full-screen maze. After watching it make several mazes, it occurred to me that a machine language version would be much faster.

The machine language maze generator was written on an Upgrade PET, and conversions for Original ROMs and 4.0 ROMs were incorporated into the loader program (Program 1). Also included are versions for the VIC-20 (Program 2) and the 64 (Program 3).

The PET version uses 176 bytes and will fit into one of the cassette buffers. It uses the second cassette buffer, but 4.0 BASIC users may prefer to use the first cassette buffer. If so, change the value of S in line 120 to 634. If you would like to use one or both buffers for utility programs, instructions for loading machine language into high RAM are given at the end of the loader (lines 450 on).

The VIC-20 version occupies 201 bytes and must be loaded into high memory because it's too large for the cassette buffer. The extra bytes in this version are needed to handle color. Screen and border are both white, and the maze color is chosen randomly. All colors except black and white are used.

While typing in the loader program, make special note of DATA items beginning with an asterisk (*) or a plus sign (+). Be sure to include these symbols. When you've finished typing, be *sure* to SAVE the program before RUNning it. When the program is RUN, it first POKEs the machine language into memory and then offers a demonstration. Mazes will be constructed on your screen as long as you keep pressing keys.

Speeding Up The Maze

Converting Charles Bond's algorithm from BASIC to machine language was accomplished by a nearly line-by-line translation of the original BASIC program. Although the machine language program executes far faster than the original BASIC program, the maze does not appear on the screen instantaneously. But the motion is so fast it's hard to follow with your eyes. I timed the PET maze construction at 1.65 seconds. The VIC and 64 versions take about half as much time.

The mazes are 39 columns by 23 rows on the PET and 64, and 21 columns by 21 rows on the VIC. These are maximum sizes. Changing the maze dimensions is possible, but not particularly easy, especially if you want to center the maze on the screen. You can try this:

PET & 64:	POKE S+37,C	(where C is >10 or <40)
	POKE S+53, R	(where R is >10 or <24)
VIC:	POKE S+62, C	(where C is >10 or <22)
	POKE S+78, R	(where R is >10 or <22)

C is the number of columns, R the number of rows, and S the SYS address *minus eight*. C and R *must be odd numbers*. Mazes smaller than the maximum size will not be centered, but will start in the upper-left-hand corner of the screen.

Munchmaze

Shortly after converting the maze generator to machine language, I wrote a machine language game called "Munchmaze," in which a character hurries through the maze dropping bread crumbs as it goes. You move your character around with the appropriate keys and try to munch as many of the bread crumbs as you can before the character catches you. The game ends when the two characters collide or when you accumulate 10,000 points.

There are three speed levels: slow, moderate, and fast. Both characters move at the same speed, but the computer character beats you on the corners. Also, you have to change directions manu-

ally; it doesn't. The computer moves its character according to the same "left-turn rule" used by the mouse in Charles Bond's original maze generator program. You must be aware of this in order to find temporary hiding places.

There's another tricky feature, too. Sometimes, when the two characters are moving from opposite directions toward each other, the computer character goes right on by and no collision occurs. Just breathe a sigh of relief and continue munching – you were lucky.

The maze in Munchmaze is not constructed on the screen, but in another area of RAM. It is then transferred to the screen, where the maze appears all at once; then there is a one-second delay before the action begins. If you break out of the program for any reason just type SYS 12311 to restart.

Programs 4 and 5 are versions of Munchmaze for 4.0 and Upgrade PETs, respectively. Program 6 is a 64 version of Munchmaze.

If you would rather not type these programs yourself, I'll make copies for you. Send a self-addressed, stamped mailer, a blank cassette, and $3 to:

Gary Marsa
320 Terrace, Apt. 2-S
Flushing, MI 48433

I have available the Maze Generator for all PETs and the VIC, and Munchmaze for Original, Upgrade or 4.0 PETs. Please tell me which version(s) you want.

Special Note To 8032 And Fat Forty Owners

Because of keyboard differences between "old style" 40-column PETs and "Fat Forties," Munchmaze will not work properly on Fat Forties, or 8032s. Your machine is a "Fat Forty" if a bell rings when you turn it on.

Munchmaze 4.0 will work properly on these computers if you type in these two lines instead of the ones that appear in the listings:

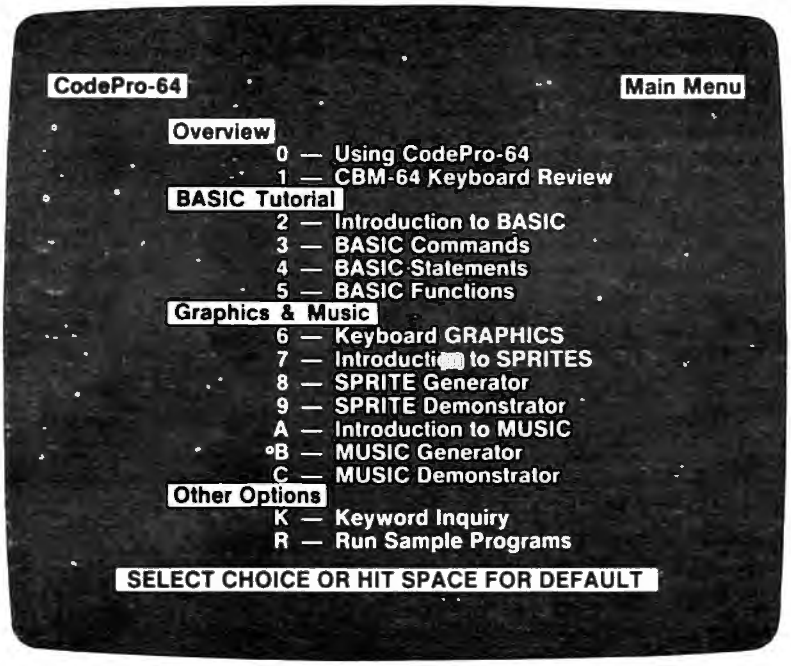
```
13314 DATA 255,255,255,40,0,182
13320 DATA 184,180,178,160,32,58
```

Program 1:
Maze Generator For 40-Column PET With Original, Upgrade, Or 4.0 ROMs

```
50 REM FOR 40-COLUMN PET/CBMS WITH ORIGI
   NAL, UPGRADE, OR 4.0 ROMS
100 CLR: POKE 59468,12: X=RND(-TI)
110 P=PEEK(50003): Z=84-82*(P=0)
120 S=826: A=S
130 PRINT "{CLR}{2 DOWN}LOADING...
    {2 DOWN}"
140 READ X$: IF X$="XXX" THEN 200
150 R=ASC(X$): Q=VAL(MID$(X$,1-(R<48)))
160 IF R=42 THEN X=Z+Q: GOTO 190
170 IF R<>43 THEN X=Q: GOTO 190
180 Y=S+Q: X=INT(Y/256): Y=Y-256*X: POKE
     A,Y: A=A+1
190 POKE A,X: A=A+1: GOTO 140
200 IF P=0 THEN POKE S+63,69: POKE S+66,
    222
210 IF P=160 THEN POKE S+63,41: POKE S+6
    4,210
220 PRINT "ACTIVATE WITH {RVS}SYS"; S+8
230 PRINT "{2 DOWN}PRESS ANY KEY FOR DEM
    ONSTRATION MAZES."
240 PRINT "{2 DOWN}PRESS 'Q?' WHEN YOU WA
    NT TO QUIT.{3 DOWN}": GOTO 260
250 SYS S+8: PRINT "{HOME}PRESS KEY..."
260 GET X$: IF X$="" THEN 260
270 IF X$<>"Q" THEN 250
280 DATA 1, 0, 216, 255, 255, 255, 40, 0
    , 169, 81
290 DATA 133, *0, 169, 40, 133, *2, 169,
     128, 133, *1
300 DATA 133, *3, 169, 147, 32, 210, 255
    , 162, 0, 160
310 DATA 0, 169, 160, 145, *2, 200, 192,
     39, 208, 249
320 DATA 24, 165, *2, 105, 40, 133, *2,
    144, 2, 230
330 DATA *3, 232, 224, 23, 208, 229, 160
    , 0, 169, 4
340 DATA 145, *0, 32, 127, 223, 165, 140
    , 41, 3, 133
350 DATA 1, 170, 10, 168, 24, 185, +0, 1
    01, *0, 133
360 DATA *4, 185, +1, 101, *1, 133, *5,
    24, 185, +0
370 DATA 101, *4, 133, *2, 185, +1, 101,
     *5, 133, *3
380 DATA 160, 0, 177, *2, 201, 160, 208,
     18, 138, 145
390 DATA *2, 169, 32, 145, *4, 165, *2,
    133, *0, 165
400 DATA *3, 133, *1, 76, +62, 232, 138,
     41, 3, 197
410 DATA 1, 208, 189, 177, *0, 170, 169,
     32, 145, *0
420 DATA 224, 4, 240, 26, 138, 10, 168,
    162, 2, 56
430 DATA 165, *0, 249, +0, 133, *0, 165,
     *1, 249, +1
440 DATA 133, *1, 202, 208, 238, 76, +62
    , 96, XXX
450 REM MAKE THESE ADDITIONS & CHANGES T
    O LOAD MACHINE CODE INTO HIGH RAM:
460 REM
470 REM{2 SPACES}70 P=PEEK(50003): M=52-
    82*(P=0)
480 REM{2 SPACES}80 Y=PEEK(M)+256*PEEK(M
    +1)-177: X=INT(Y/256): Y=Y-256*X
490 REM{2 SPACES}90 POKE M,Y: POKE M+1,X
    : POKE M-4,Y: POKE M-3,X
500 REM{2 SPACES}110 P=PEEK(50003): Z=84
    -82*(P=0): M=52-82*(P=0)
510 REM{2 SPACES}120 S=PEEK(M)+256*PEEK(
    M+1): A=S
```

Program 2: Maze Generator For VIC

```
10 REM MAZE (VIC)
20 REM MAZE GENERATOR IN MACHINE LANGUAG
   E
```



NEW! For the Commodore 64™ ANNOUNCING

CodePro-64™

A new concept in interactive visual learning . . .

Now you can learn to code in BASIC and develop advanced programming skills with graphics, sprites and music—**visually.** You learn by interacting with CodePro-64, a new concept in interactive visual learning.

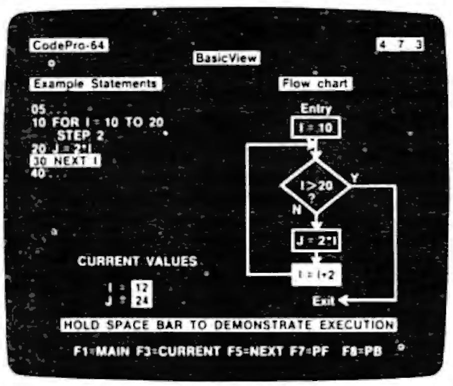
SEE PROGRAM EXECUTION

Imagine actually seeing BASIC statements execute CodePro-64 guides you through structured examples of BASIC program segments You enter the requested data or let CodePro-64 do the typing for you (It will not let you make a mistake.)

After entering an example you invoke our exclusive **BasicView™** which **shows you how the BASIC program example executes.**

You step through and actually see the execution of sample program statements by simply pressing the space bar CodePro-64 does the rest

You see statements with corresponding **flow chart graphics** and variable value displays You learn by visual examples



EXTENSIVE TUTORIAL

CodePro-64's extensive tutorial guides you through each BASIC command program statement, and function You get clear explanations Then you enter program statements as interactive examples Where appropriate you invoke BasicView to see examples execute and watch their flow charts and variables change

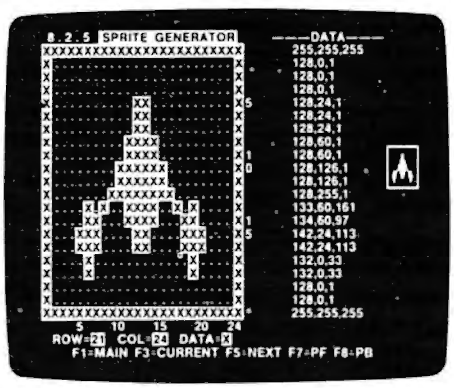
By seeing graphic displays of program segment execution you learn by visual example You **learn faster and grasp programming concepts easier** with CodePro-64 because you immediately see the results of your input

You control your learning You can go through the tutorial sequentially or return to the main menu and select different topics, or **use keywords** to select language elements to study You can page back and forth between screens within a topic at the touch of a function key

CodePro-64 **lets you follow your interests** and practice with interactive examples But you can never get "lost" F1 will always return you to the main menu Once you have practiced and mastered the BASIC language elements you move on to more advanced concepts You learn about sprite and music programming

SPRITE GENERATOR & DEMONSTRATOR

CodePro-64's sprite generator lets you **define your own sprites** on the screen You learn how to define sprites and what data values correspond to your sprite definitions. (You can then use these values to write your own programs) You can **easily experiment** with different definitions and make changes to immediately see the effects



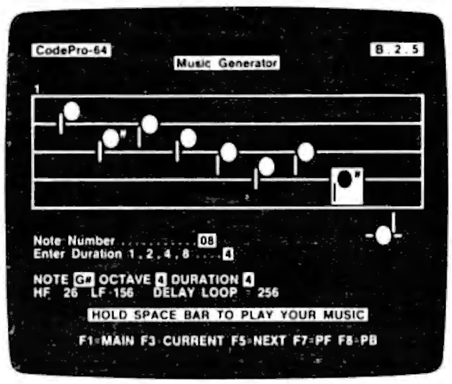
We also help you learn to program with sprites by giving you a **sprite demonstrator** so you can see the effect of changing register values You can experiment by moving your sprite around in a screen segment change its color or priority, and see the effects of your changes You learn by visual examples

MUSIC GENERATOR & DEMONSTRATOR

To teach you music programming CodePro-64 gives you an interactive music generator and demonstrator First we help you set all your SID parameters (attack/decay, sustain/release, waveform, etc) Then you enter notes to play and **we show your tune graphically as it plays,** note by note, on the scale You learn by seeing and hearing the results of your input

OUR GUARANTEE

We guarantee your satisfaction. You must be satisfied with CodePro-64 for the Commodore 64. Try it for 10 days and if for any reason you are not satisfied return it to us (undamaged) for a full refund. **No risk.**



Our music demonstrator **lets you experiment** with various combinations of music programming parameters and hear the results You can **quickly modify any of the SID register values** to hear the effects of the change For example, you could easily change waveform and attack/decay values while holding all other SID values constant By seeing your input and hearing the result you quickly learn how to create new musical sounds and special sound effects

AND MORE . . .

We don't have enough space to tell you everything CodePro-64 offers You need to see for yourself BASIC tutorials, graphics sprites music keyboard review sample programs—the main menu shown above gives you just a summary of the contents of this powerful educational product

Whether you're a beginning programmer or an experienced professional CodePro-64 will help you improve your Commodore 64 programming skills We're sure because CodePro-64 was developed by a team of two professionals with **over 25 years** of software development experience

CodePro-64 is a professional quality educational program for the serious student of personal computing And it's **fully guaranteed.** Order yours today

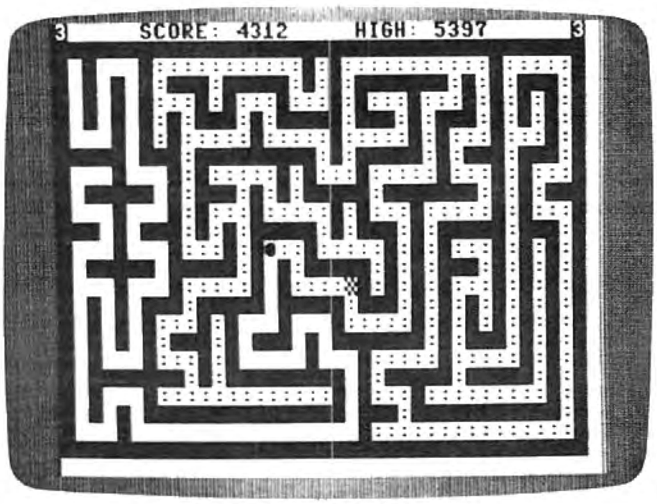
HOW TO ORDER

Order your copy of CodePro-64 today by mail or phone Send only $59.95 plus $3.00 shipping and handling to

SYSTEMS MANAGEMENT ASSOCIATES
3700 Computer Drive, Dept C
Raleigh, N C 27609

Please **specify tape or diskette** version MasterCard/VISA accepted For faster service on credit card orders call (919)-787 7703

Commodore 64 is a trademark of Commodore Business Machines, Inc
Ad no 733, Copyright 1983, SMA
Dealer inquiries invited.



A game of "Munchmaze" being played on the 64.

```
50 REM FOR THE VIC-20 (ANY MEMORY SIZE)
100 Y=PEEK(55)+256*PEEK(56)-202: X=INT(Y
    /256): Y=Y-256*X
110 POKE 55,Y: POKE 56,X: POKE 51,Y: POK
    E 52,X
120 CLR: POKE 36879,27: PRINT CHR$(142);
    : X=RND(-TI)
130 S=PEEK(55)+256*PEEK(56): A=S
140 PRINT "{CLR}{2 DOWN}LOADING..."
150 READ X$: IF X$="XXX" THEN 200
160 R=ASC(X$): Q=VAL(MID$(X$,1-(R<48)))
170 IF R<>43 THEN X=Q: GOTO 190
180 Y=S+Q: X=INT(Y/256): Y=Y-256*X: POKE
     A,Y: A=A+1
190 POKE A,X: A=A+1: GOTO 150
200 PRINT "{HOME}": IF PEEK(210)<>16 THE
    N 220
210 POKE S+17,16: POKE S+45,148: POKE S+
    48,149
220 PRINT "{DOWN}ACTIVATE WITH"
230 PRINT "{2 SPACES}{RVS}SYS"; S+8
240 PRINT "{2 DOWN}PRESS ANY KEY FOR"
250 PRINT "DEMONSTRATION MAZES."
260 PRINT "{2 DOWN}PRESS 'Q' WHEN YOU"
270 PRINT "WANT TO QUIT.": GOTO 290
280 SYS S+8: PRINT "{HOME}PRESS KEY..."
290 GET X$: IF X$="" THEN 290
300 IF X$<>"Q" THEN 280
310 PRINT "{CLR}": POKE 36879,27
320 DATA 1, 0, 234, 255, 255, 255, 22, 0
    , 169, 45
330 DATA 133, 87, 169, 22, 133, 89, 169,
     30, 133, 88
340 DATA 133, 90, 169, 25, 141, 15, 144,
     32, 95, 229
350 DATA 32, 148, 224, 165, 143, 41, 7,
    201, 2, 48
360 DATA 245, 160, 0, 153, 0, 150, 153,
    0, 151, 200
370 DATA 208, 247, 162, 0, 160, 0, 169,
    160, 145, 89
380 DATA 200, 192, 21, 208, 249, 24, 165
    , 89, 105, 22
390 DATA 133, 89, 144, 2, 230, 90, 232,
    224, 21, 208
400 DATA 229, 160, 0, 169, 4, 145, 87, 3
    2, 148, 224
410 DATA 165, 143, 41, 3, 133, 1, 170, 1
    0, 168, 24
420 DATA 185, +0, 101, 87, 133, 91, 185,
     +1, 101, 88
430 DATA 133, 92, 24, 185, +0, 101, 91,
    133, 89, 185
440 DATA +1, 101, 92, 133, 90, 160, 0, 1
    77, 89, 201
450 DATA 160, 208, 18, 138, 145, 89, 169
    , 32, 145, 91
460 DATA 165, 89, 133, 87, 165, 90, 133,
     88, 76, +87
470 DATA 232, 138, 41, 3, 197, 1, 208, 1
    89, 177, 87
480 DATA 170, 169, 32, 145, 87, 224, 4,
    240, 26, 138
490 DATA 10, 168, 162, 2, 56, 165, 87, 2
    49, +0, 133
500 DATA 87, 165, 88, 249, +1, 133, 88,
    202, 208, 238
510 DATA 76, +87, 96, XXX
```

Program 3: Maze Generator For The 64

```
10 I=49152:IFPEEK(I+2)=216THENSYS49160:E
   ND
20 READ A:IF A=256 THEN SYS 49160:END
30 POKE I,A:I=I+1:GOTO 20
49152 DATA 1,0,216,255,255,255,40
49160 DATA 0,169,81,133,251,169,40
49168 DATA 133,253,169,4,133,252,133
49176 DATA 254,169,147,32,210,255,162
49184 DATA 0,160,0,169,160,145,253
49192 DATA 200,192,39,208,249,24,165
49200 DATA 253,105,40,133,253,144,2
49208 DATA 230,254,232,224,23,208,229
49216 DATA 160,0,169,4,145,251,169
49224 DATA 255,141,15,212,169,128,141
49232 DATA 18,212,173,27,212,41,3
49240 DATA 133,173,170,10,168,24,185
49248 DATA 0,192,101,251,133,170,185
49256 DATA 1,192,101,252,133,171,24
49264 DATA 185,0,192,101,170,133,253
49272 DATA 185,1,192,101,171,133,254
49280 DATA 160,0,177,253,201,160,208
49288 DATA 18,138,145,253,169,32,145
49296 DATA 170,165,253,133,251,165,254
49304 DATA 133,252,76,62,192,232,138
49312 DATA 41,3,197,173,208,189,177
49320 DATA 251,170,169,32,145,251,224
49328 DATA 4,240,26,138,10,168,162
49336 DATA 2,56,165,251,249,0,192
49344 DATA 133,251,165,252,249,1,192
49352 DATA 133,252,202,208,238,76,62
49360 DATA 192,169,1,160,0,153,0
49368 DATA 216,153,0,217,153,0,218
49376 DATA 153,0,219,200,208,241,96,256
```

Program 4: Munchmaze For 4.0 PETs

```
5 PRINT"{CLR}PLEASE WAIT...."
10 I=12288
20 READ A:IF A=256 THEN SYS 12311
30 POKE I,A:I=I+1:GOTO 20
12288 DATA 20,4,10,0,88,178
12294 DATA 187,40,171,84,73,41
12300 DATA 58,158,49,48,52,56
```

Write For FREE Catalog

VISA

(602) 855-3357

Call your order in and get a FREE program

NEW

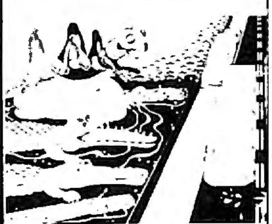
VIC SOFTWARE CBM 64

Call your order in and get a FREE program

Write For FREE Catalog

MasterCard

(602) 855-3357

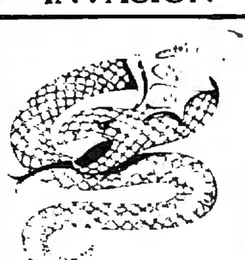


CRICKET

PARATROOPER

MOW

ALIEN INVASION



SNAKE OUT

CRICKET - From the company that brought you Asteroidz, Munchman and a host of other blockbusters. We now present CRICKET. This is a challenging game with a cast of characters you will love and hate. All you have to do is get Cherp from one side of the road to the river and then across the river. Not so fast though. First you have to figure out how to dodge the traffic and get to the center. Then how are you going to get across the river? Look here comes a log — even a turtle. Hitch a ride across the river and jump from one to the other. Keep a sharp eye out for Ade the Gator. He loves to have crickets for lunch. How many times can you get across the road and river. You will have to work as fast as you can. Time limit and bonus. You will find this game addictive and challenging and it will entertain you with hours of fun and enjoyment. **$14.95**

BUG BLAST - If you think Centipede was fun — look out for BUG BLAST. A new and fast action arcade game with realistic smooth action, quality hi-res graphics and trouble. Its very calm as the first wave attacks. Only a few bugs to kill. Just shoot thru the cactus and wipe them out. After a few attacks you feel you have everything under control. Now the attacks really start. Those protection areas have to go. Blast away. Will they ever stop? OK — the BUGS got me this time. Now its my turn, Just one more time — BUG BLAST — Now its your turn to get even. **$14.95**

BOMB'S AWAY - Can you stop him? The crazy bomber drops the bombs from the top of the screen. You get 3 buckets to catch them. Before you know it bombs are falling so fast you wonder when he will stop. Just when you think you have him under control your bucket gets smaller. Is your hand quicker than your eye? **Special $9.95**

PARATROOPER - You are the only one left to stop them. The sky is full of enemy choppers. Paratroopers keep dropping into your area with non-stop barrage of enemy troops. They are out to destroy you. This new game is an unbeatable blend of arcade action and quick thinking strategy. You must make every shot count — don't be to fast on the trigger. Every time you hit a chopper or paratrooper you get extra points. Wait until you see the climax of this game — you won't believe it! This is a multiple skill level game with razor-sharp graphics and sound. **$19.95**

MOW - Get ready for the fast and furious action of the craziest mower you have ever seen. How much grass can you cut? Joystick moves your mower around as fast as you dare. Watch out for granny's dafodils and grandpa's radio antenna. **$14.95**

COSMIC CRUZER - Bring the coin-op game into your VIC. 3 Scenarios. Your Cruzer moves over a mountainous landscape & into a tunnel of surface - to - air missle, silos and ground - to - air weapons. If you can make it in and out of the tunnel you fly into the asteroid field. Drop bombs and fire missiles at the fuel dumps to keep your fuel supply up. If you are really good you can get to the base and try, to destroy it. We don't know of any one that has hit the base yet. Maybe you will be the 1st. Cosmic Cruzer is a fun filled magnificently rendered home video game that will last for months of challenge. Highly addicting. Hi-Res Graphics, Color & Sound. SPECIAL PRICE - **$14.95**

SPACE PAK - Can you survive? 3 space games with the sights and sounds of arcade games. The excitement builds as the action is un-ending. Blast away at everything in sight. The alien attacks will stop at nothing to destroy you. Prepare for battle, there is no escape, unless you can help. Can you survive? Hi-Res, Color, Graphic & Sound. Joystick or keyboard. **$19.95**

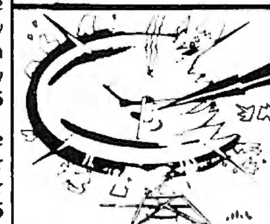
ALIEN INVASION - Invaders from space are attacking your home planet. Hurry and man your lasers and prepare your robot forces for the inevitable attack of the Alien Invaders. The excitment builds as you command a battery of missile bases in a bunker. Each invader has a laser aimed right at you. Will they ever stop. Only you can save the Galaxy. You can compete with 4 people in the solar system. There are 20 levels of play. If you destroy the Aliens in the correct order you will receive bonus points. Can you get the top score? **$14.95**

TARGET COMMAND - The whole West Coast is being bombarded and only you can save it. You are at the controls of the missile launcher and hold the destiny of our country in your hands. It takes a cool head, not hand and fast reflexes to zap those missiles right out of the air. Get ready to pulverize — atomize and vaporize them. Oh, my God, those warheads are heading right for our ammo dumps. They are everywhere. NO ONE CAN SAVE US — EXCEPT YOU. You must move your laser into position and fire as fast as you dare. Time limit with arcade style excitement. Protect your ammo at all costs. 10 levels of play. **$14.95**

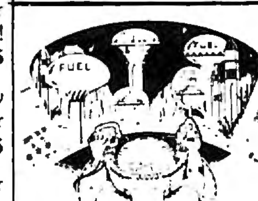
SNAKEOUT - Slip your snake into position and score by chomping the blocks. Watch the way you slither because your escape routes get smaller. 2 Bonus games included. **$14.95**

HEAD-ON - Please do not buy this game if you are the type that says "I'll play it just one more time". Players have been known to start playing HEAD ON at 8:30 p.m. and at 2 a.m., wonder where the time went? Have you ever tried to explain to someone why you played a game for five and a half hours. We know of no remedy for the addiction to HEAD ON except to beat the VIC on level 9. No one has done it, YET, will you? We think not. Move your car as fast as you can dare around the tracks. You get 3 cars and MUST avoid the computer car. Points for the most dots covered. Bonus cars, nine levels of play. **$14.95**

BUG BLAST

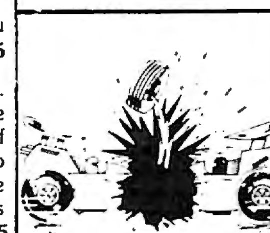


TARGET COMMAND



COSMIC CRUZER

SPACE PAK



HEAD ON

ComputerMat • Box 1664 C • Lake Havasu City, Az. 86403 *Add $1.00 For Shipping*

```
12306 DATA 0,0,0,234,234,169
12312 DATA 12,141,76,232,169,0
12318 DATA 141,96,10,141,97,10
12324 DATA 169,147,32,210,255,169
12330 DATA 44,141,153,51,169,52
12336 DATA 141,154,51,162,31,169
12342 DATA 129,32,144,51,169,0
12348 DATA 141,98,10,141,99,10
12354 DATA 133,143,24,165,142,105
12360 DATA 2,133,2,165,142,197
12366 DATA 2,208,250,169,147,32
12372 DATA 210,255,32,92,51,162
12378 DATA 39,169,160,157,39,128
12384 DATA 157,151,131,202,208,247
12390 DATA 169,80,133,84,169,128
12396 DATA 133,85,162,21,160,0
12402 DATA 169,160,145,84,160,38
12408 DATA 145,84,32,174,51,202
12414 DATA 208,240,169,43,141,153
12420 DATA 51,169,53,141,154,51
12426 DATA 162,166,169,128,32,144
12432 DATA 51,162,0,169,32,157
12438 DATA 0,11,157,0,12,157
12444 DATA 0,13,157,0,14,232
12450 DATA 208,241,169,81,133,84
12456 DATA 169,40,133,86,169,11
12462 DATA 133,85,133,87,162,0
12468 DATA 160,0,169,160,145,86
12474 DATA 200,192,39,208,249,24
12480 DATA 165,86,105,40,133,86
12486 DATA 144,2,230,87,232,224
12492 DATA 23,208,229,160,0,169
12498 DATA 4,145,84,32,41,210
12504 DATA 165,140,41,3,133,1
12510 DATA 170,10,168,24,185,255
12516 DATA 51,101,84,133,88,185
12522 DATA 0,52,101,85,133,89
12528 DATA 24,185,255,51,101,88
12534 DATA 133,86,185,0,52,101
12540 DATA 89,133,87,160,0,177
12546 DATA 86,201,160,208,18,138
12552 DATA 145,86,169,32,145,88
12558 DATA 165,86,133,84,165,87
12564 DATA 133,85,76,213,48,232
12570 DATA 138,41,3,197,1,208
12576 DATA 189,177,84,170,169,32
12582 DATA 145,84,224,4,240,26
12588 DATA 138,10,168,162,2,56
12594 DATA 165,84,249,255,51,133
12600 DATA 84,165,85,249,0,52
12606 DATA 133,85,202,208,238,76
12612 DATA 213,48,169,193,141,153
12618 DATA 51,169,53,141,154,51
12624 DATA 162,130,138,32,144,51
12630 DATA 32,228,255,208,251,32
12636 DATA 228,255,240,251,201,81
12642 DATA 208,13,169,147,32,210
12648 DATA 255,32,92,51,169,13
12654 DATA 76,210,255,201,49,48
12660 DATA 230,201,52,16,226,56
12666 DATA 233,48,133,2,169,147
12672 DATA 32,210,255,162,0,189
12678 DATA 0,11,157,0,128,189
12684 DATA 0,12,157,0,129,189
12690 DATA 0,13,157,0,130,189
12696 DATA 0,14,157,0,131,232
12702 DATA 208,229,32,92,51,24
12708 DATA 165,2,105,176,141,0
12714 DATA 128,141,38,128,24,165
12720 DATA 143,105,60,133,255,165
12726 DATA 143,197,255,208,250,169
12732 DATA 81,133,84,133,88,169
12738 DATA 128,133,85,133,89,169
12744 DATA 1,133,255,162,2,134
12750 DATA 1,160,0,169,102,145
12756 DATA 84,169,0,133,143,166
12762 DATA 1,138,10,168,24,185
12768 DATA 255,51,101,84,133,86
12774 DATA 185,0,52,101,85,133
12780 DATA 87,160,0,177,86,201
12786 DATA 160,208,9,202,138,41
12792 DATA 3,133,1,76,217,49
12798 DATA 201,81,208,3,76,184
12804 DATA 50,169,102,145,86,169
12810 DATA 58,145,84,165,86,133
12816 DATA 84,165,87,133,85,232
12822 DATA 138,41,3,133,1,165
12828 DATA 255,240,8,160,0,132
12834 DATA 255,169,81,145,88,162
12840 DATA 0,165,151,221,7,52
12846 DATA 240,8,232,224,4,208
12852 DATA 246,76,107,50,138,10
12858 DATA 168,24,185,255,51,101
12864 DATA 88,133,90,185,0,52
12870 DATA 101,89,133,91,160,0
12876 DATA 177,90,201,160,240,25
12882 DATA 201,58,208,3,32,186
12888 DATA 51,160,0,169,81,145
12894 DATA 90,169,32,145,88,165
12900 DATA 90,133,88,165,91,133
12906 DATA 89,165,143,197,2,208
12912 DATA 250,173,98,10,201,16
12918 DATA 208,61,173,99,10,201
12924 DATA 39,208,54,169,81,133
12930 DATA 84,169,128,133,85,162
12936 DATA 0,160,0,177,84,201
12942 DATA 58,208,7,32,186,51
12948 DATA 169,32,145,84,200,192
12954 DATA 37,208,238,32,174,51
12960 DATA 232,224,21,208,228,162
12966 DATA 0,189,17,54,240,6
12972 DATA 157,51,128,232,208,245
12978 DATA 76,51,51,76,213,49
12984 DATA 169,102,145,86,169,58
12990 DATA 145,84,165,88,133,84
12996 DATA 165,89,133,85,56,165
13002 DATA 84,233,41,133,86,165
13008 DATA 85,233,0,133,87,169
13014 DATA 240,133,2,169,255,133
13020 DATA 143,165,86,133,88,165
13026 DATA 87,133,89,169,0,133
13032 DATA 1,160,0,162,0,177
13038 DATA 88,221,11,52,240,5
13044 DATA 232,224,8,208,246,134
13050 DATA 254,56,169,7,229,254
13056 DATA 170,189,11,52,145,88
13062 DATA 200,192,3,208,224,24
13068 DATA 165,88,105,40,133,88
13074 DATA 144,2,230,89,230,1
13080 DATA 165,1,201,3,208,203
13086 DATA 165,143,208,252,198,2
13092 DATA 208,179,162,0,189,34
13098 DATA 52,240,6,157,55,128
13104 DATA 232,208,245,56,173,98
13110 DATA 10,237,96,10,141,100
13116 DATA 10,173,99,10,237,97
13122 DATA 10,13,100,10,240,17
13128 DATA 144,15,173,98,10,141
```

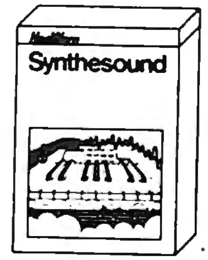
Fantastic Sale!

HES Special Programs For VIC-20 Computer

Call Today!! **Call Today!!** **Call Today!!**

SYNTHESOUND Sale $19.95

• Fantastic Sound & Music • Piano Keyboard



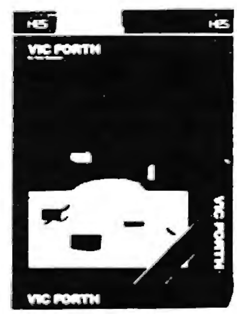
Now your VIC-20 can be used as a state-of-the-art music synthesizer and special effects generator! A very user friendly manual guides you through sessions of learning and development of your synthesizer "style"! You can program Synthesound to sound like many different instruments like bagpipes, organ, bass and many others. Synthesound brings technical musical abilities to any VIC-20 user. Prepare to step into the exciting world of the futuristic computer musician.

- VIC keyboard used as a 2 octave, fast attack keyboard.
- 3 voices with 4½ octave range.
- 4 low frequency ocillators.
- 4 envelope shaping controls (Attack, Decay, Sustain, Release).
- No additional hardware required.
- Spectacular special effects.
- 4 voice/20 note sequencer for repeating melodies or rhythms.
- High resolution piano keyboard display.

List $44.95 / **Sale $19.95**

VIC FORTH LANGUAGE

Faster than BASIC — easier than Assembly language — Language of Robots.



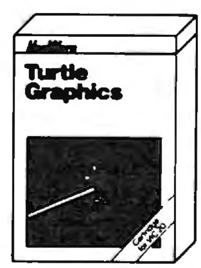
VIC FORTH is an exciting new cartridge for your VIC-20 computer. You now have a language that is more powerful than BASIC and easier to program than assembler! VIC FORTH is a nearly complete implementation of the "FORTH INTEREST GROUP" (fig.) version of FORTH. The VIC FORTH editor follows closely the standard FORTH editor described in the book, "Starting FORTH", plus it has many, many additional features.

- Full screen editor which has 16 lines of 64 characters each (standard FORTH Screen) through use of a horizontal scrolling window.
- User definable words; this means a function not already supported can be created by you and added to VIC FORTH for future use.
- Full VIC sound and color capabilities.
- Runs in standard 5k VIC.
- Output to printers and disks and tape machines.
- Up to 24k additional memory can be added.
- 31 characters for variable names.

List $59.95 / **Sale $39.95**

TURTLE GRAPHICS

• Create colorful pictures and graphics • Learn how computers compute.



Turtle Graphics is an exciting and fun way to discover computer programming concepts. Turtle Graphics is an easy-to-learn computer language. It enables the beginning programmer to master the computer in just a few minutes. By telling an imaginary turtle how to roam about on the VIC's screen, the Turtle Graphics programmer can create colorful and artistic pictures. As one becomes more and more adept at teaching the turtle to paint using the VIC's rich graphics and color palette he or she is painlessly learning all of the basics of computer programming. This not only develops computer awareness, but enriches one's math, logic, communication and artistic skills as well.

- Turtle Graphics has over 30 different commands.
- Programs may be listed on a printer and saved on or loaded from tape or disk.
- Turtle Graphics is menu driven for ease of use.
- Trace mode to help the beginning programmer follow the logic of his program one step at a time.
- Well written, easy-to-follow manual with many examples included.

List $39.95 / **Sale $29.95**

We Have All HES Products At Super Sale Prices!!

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

```
13134 DATA 96,10,173,99,10,141
13140 DATA 97,10,32,118,51,76
13146 DATA 58,48,162,0,189,19
13152 DATA 52,240,6,157,6,128
13158 DATA 232,208,245,162,0,189
13164 DATA 28,52,240,6,157,22
13170 DATA 128,232,208,245,172,96
13176 DATA 10,173,97,10,32,188
13182 DATA 196,32,147,207,162,0
13188 DATA 189,0,1,240,6,157
13194 DATA 27,128,232,208,245,96
13200 DATA 134,84,133,85,162,0
13206 DATA 160,0,189,44,52,240
13212 DATA 10,201,255,240,12,145
13218 DATA 84,200,232,208,241,32
13224 DATA 174,51,232,208,233,96
13230 DATA 24,165,84,105,40,133
13236 DATA 84,144,2,230,85,96
13242 DATA 138,72,152,72,24,173
13248 DATA 98,10,105,2,141,98
13254 DATA 10,144,3,238,99,10
13260 DATA 162,0,181,84,72,232
13266 DATA 224,8,208,248,172,98
13272 DATA 10,173,99,10,32,188
13278 DATA 196,32,147,207,162,0
13284 DATA 189,0,1,240,6,157
13290 DATA 12,128,232,208,245,162
13296 DATA 8,104,149,83,202,208
13302 DATA 250,104,168,104,170,96
13308 DATA 234,234,234,1,0,216
13314 DATA 255,255,255,40,0,54
13320 DATA 56,52,50,160,32,58
13326 DATA 102,170,186,127,255,19
13332 DATA 3,15,18,5,58,32
13338 DATA 48,0,8,9,7,8
13344 DATA 58,0,135,129,141,133
13350 DATA 160,143,150,133,146,0
13356 DATA 79,77,32,32,78,80
13362 DATA 99,80,32,79,99,79
13368 DATA 77,32,79,80,78,99
13374 DATA 99,99,77,79,80,32
13380 DATA 79,80,0,101,32,77
13386 DATA 78,32,103,32,103,32
13392 DATA 101,32,101,32,77,101
13398 DATA 103,32,32,79,76,100
13404 DATA 101,103,32,101,103,0
13410 DATA 101,32,32,32,32,103
13416 DATA 32,103,32,101,32,101
13422 DATA 32,32,32,103,32,32
13428 DATA 101,32,32,101,32,99
13434 DATA 32,103,0,101,103,77
13440 DATA 78,101,103,32,103,100
13446 DATA 101,32,101,103,77,32
13452 DATA 103,32,32,76,79,99
13458 DATA 101,103,99,101,103,0
13464 DATA 76,122,32,32,76,122
13470 DATA 77,100,100,100,78,76
13476 DATA 122,32,77,122,77,100
13482 DATA 100,100,78,76,122,32
13488 DATA 76,122,0,0,160,223
13494 DATA 32,32,233,231,32,233
13500 DATA 160,223,32,160,160,160
13506 DATA 160,231,160,160,160,160
13512 DATA 160,32,2,25,0,160
13518 DATA 160,223,233,160,231,233
13524 DATA 160,226,160,223,32,32
13530 DATA 233,160,105,160,160,0
13536 DATA 160,160,160,160,160,231
13542 DATA 160,160,98,160,231,32
13548 DATA 233,160,105,32,160,160
13554 DATA 160,160,32,32,7,1
13560 DATA 18,25,0,160,160,95
13566 DATA 105,160,231,160,160,226
13572 DATA 160,231,233,160,105,32
13578 DATA 32,160,160,0,160,160
13584 DATA 32,32,160,231,160,160
13590 DATA 32,160,231,160,160,160
13596 DATA 160,231,160,160,160,160
13602 DATA 160,32,13,1,18,19
13608 DATA 1,0,255,32,32,32
13614 DATA 184,32,32,32,32,32
13620 DATA 13,15,22,5,32,20
13626 DATA 8,5,32,34,81,34
13632 DATA 32,21,19,9,14,7
13638 DATA 0,32,32,32,30,32
13644 DATA 32,32,32,32,20,8
13650 DATA 5,32,14,21,13,2
13656 DATA 5,18,32,16,1,4
13662 DATA 46,0,32,32,32,93
13668 DATA 0,180,60,67,81,67
13674 DATA 62,182,32,32,184,32
13680 DATA 61,32,13,15,22,5
13686 DATA 32,21,16,0,32,32
13692 DATA 32,93,32,32,32,32
13698 DATA 32,180,32,61,32,13
13704 DATA 15,22,5,32,12,5
13710 DATA 6,20,0,32,32,32
13716 DATA 22,32,32,32,32,32
13722 DATA 182,32,61,32,13,15
13728 DATA 22,5,32,18,9,7
13734 DATA 8,20,0,32,32,32
13740 DATA 178,32,32,32,32,32
13746 DATA 178,32,61,32,13,15
13752 DATA 22,5,32,4,15,23
13758 DATA 14,0,255,3,8,15
13764 DATA 15,19,5,32,19,16
13770 DATA 5,5,4,32,6,1
13776 DATA 3,20,15,18,58,0
13782 DATA 0,32,32,32,177,32
13788 DATA 61,32,6,1,19,20
13794 DATA 44,32,178,32,61,32
13800 DATA 13,15,4,5,18,1
13806 DATA 20,5,44,32,179,32
13812 DATA 61,32,19,12,15,23
13818 DATA 0,0,15,18,32,16
13824 DATA 18,5,19,19,32,145
13830 DATA 32,20,15,32,17,21
13836 DATA 9,20,46,0,255,153
13842 DATA 143,149,167,146,133,160
13848 DATA 129,160,151,137,142,142
13854 DATA 133,146,161,161,0,256
```

Program 5: Munchmaze For Upgrade PETs

Use Program 4 but substitute these lines for Upgrade PET.

```
12498 DATA 4,145,84,32,127,223
13176 DATA 10,173,97,10,32,109
13182 DATA 210,32,233,220,162,0
13272 DATA 10,173,99,10,32,109
13278 DATA 210,32,233,220,162,0
13314 DATA 255,255,255,40,0,41
13320 DATA 50,42,18,160,32,58
```

Program 6: Munchmaze For The 64

```
10 I=12288 :POKE53281,1
20 READ A:IF A=256 THEN 35
30 POKE I,A:I=I+1:GOTO 20
35 SYS12311 :END
```

```
12288 DATA 20,4,10,0,88,178,187
12296 DATA 40,171,84,73,41,58,158
12304 DATA 49,48,52,56,0,0,0
12312 DATA 234,234,169,21,141,24,208
12320 DATA 169,0,141,96,10,141,97
12328 DATA 10,169,147,32,210,255,169
12336 DATA 7,162,0,157,0,216,157
12344 DATA 0,217,157,0,218,157,0
12352 DATA 219,232,208,241,169,120,141
12360 DATA 229,51,169,52,141,230,51
12368 DATA 162,31,169,5,32,220,51
12376 DATA 169,0,141,98,10,141,99
12384 DATA 10,133,162,24,165,161,105
12392 DATA 2,133,166,165,161,197,166
12400 DATA 208,250,169,147,32,210,255
12408 DATA 169,0,162,0,157,0,216
12416 DATA 157,0,217,157,0,218,157
12424 DATA 0,219,232,208,241,32,168
12432 DATA 51,162,39,169,160,157,39
12440 DATA 4,157,151,7,202,208,247
12448 DATA 169,80,133,168,169,4,133
12456 DATA 169,162,21,160,0,169,160
12464 DATA 145,168,160,38,145,168,32
12472 DATA 250,51,202,208,240,169,119
12480 DATA 141,229,51,169,53,141,230
12488 DATA 51,162,166,169,4,32,220
12496 DATA 51,162,0,169,32,157,0
12504 DATA 11,157,0,12,157,0,13
12512 DATA 157,0,14,232,208,241,169
12520 DATA 81,133,168,169,40,133,170
12528 DATA 169,11,133,169,133,171,162
12536 DATA 0,160,0,169,160,145,170
12544 DATA 200,192,39,208,249,24,165
12552 DATA 170,105,40,133,170,144,2
12560 DATA 230,171,232,224,23,208,229
12568 DATA 160,0,169,4,145,168,32
12576 DATA 151,224,165,143,41,3,133
12584 DATA 165,170,10,168,24,185,75
12592 DATA 52,101,168,133,180,185,76
12600 DATA 52,101,169,133,181,24,185
12608 DATA 75,52,101,180,133,170,185
12616 DATA 76,52,101,181,133,171,160
12624 DATA 0,177,170,201,160,208,18
12632 DATA 138,145,170,169,32,145,180
12640 DATA 165,170,133,168,165,171,133
12648 DATA 169,76,251,48,232,138,41
12656 DATA 3,197,165,208,189,177,168
12664 DATA 170,169,32,145,168,224,4
12672 DATA 240,26,138,10,168,162,2
12680 DATA 56,165,168,249,75,52,133
12688 DATA 168,165,169,249,76,52,133
12696 DATA 169,202,208,238,76,251,48
12704 DATA 169,13,141,229,51,169,54
12712 DATA 141,230,51,162,6,138,32
12720 DATA 220,51,32,228,255,208,251
12728 DATA 32,228,255,240,251,201,81
12736 DATA 208,32,169,147,32,210,255
12744 DATA 169,0,162,0,157,0,216
12752 DATA 157,0,217,157,0,218,157
12760 DATA 0,219,232,208,241,32,168
12768 DATA 51,169,13,76,210,255,201
12776 DATA 49,48,211,201,52,16,207
12784 DATA 56,233,48,133,166,169,147
12792 DATA 32,210,255,169,0,162,0
12800 DATA 157,0,216,157,0,217,157
12808 DATA 0,218,157,0,219,232,208
12816 DATA 241,162,0,189,0,11,157
12824 DATA 0,4,189,0,12,157,0
12832 DATA 5,189,0,13,157,0,6
12840 DATA 189,0,14,157,0,7,232
12848 DATA 208,229,32,168,51,24,165
12856 DATA 166,105,176,141,0,4,141
12864 DATA 38,4,24,165,162,105,60
12872 DATA 133,254,165,162,197,254,208
12880 DATA 250,169,81,133,168,133,180
12888 DATA 169,4,133,169,133,181,169
12896 DATA 1,133,254,162,2,134,165
12904 DATA 160,0,169,102,145,168,169
12912 DATA 0,133,162,166,165,138,10
12920 DATA 168,24,185,75,52,101,168
12928 DATA 133,170,185,76,52,101,169
12936 DATA 133,171,160,0,177,170,201
12944 DATA 160,208,9,202,138,41,3
12952 DATA 133,165,76,37,50,201,81
12960 DATA 208,3,76,4,51,169,102
12968 DATA 145,170,169,58,145,168,165
12976 DATA 170,133,168,165,171,133,169
12984 DATA 232,138,41,3,133,165,165
12992 DATA 254,240,8,160,0,132,254
13000 DATA 169,81,145,180,162,0,165
13008 DATA 197,221,83,52,240,8,232
13016 DATA 224,4,208,246,76,183,50
13024 DATA 138,10,168,24,185,75,52
13032 DATA 101,180,133,195,185,76,52
13040 DATA 101,181,133,196,160,0,177
13048 DATA 195,201,160,240,25,201,58
13056 DATA 208,3,32,6,52,160,0
13064 DATA 169,81,145,195,169,32,145
13072 DATA 180,165,195,133,180,165,196
13080 DATA 133,181,165,162,197,166,208
13088 DATA 250,173,98,10,201,16,208
13096 DATA 61,173,99,10,201,39,208
13104 DATA 54,169,81,133,168,169,4
13112 DATA 133,169,162,0,160,0,177
13120 DATA 168,201,58,208,7,32,6
13128 DATA 52,169,32,145,168,200,192
13136 DATA 37,208,238,32,250,51,232
13144 DATA 224,21,208,228,162,0,189
13152 DATA 93,54,240,6,157,51,4
13160 DATA 232,208,245,76,127,51,76
13168 DATA 33,50,169,102,145,170,169
13176 DATA 58,145,168,165,180,133,168
13184 DATA 165,181,133,169,56,165,168
13192 DATA 233,41,133,170,165,169,233
13200 DATA 0,133,171,169,240,133,166
13208 DATA 169,255,133,162,165,170,133
13216 DATA 180,165,171,133,181,169,0
13224 DATA 133,165,160,0,162,0,177
13232 DATA 180,221,87,52,240,5,232
13240 DATA 224,8,208,246,134,253,56
13248 DATA 169,7,229,253,170,189,87
13256 DATA 52,145,180,200,192,3,208
13264 DATA 224,24,165,180,105,40,133
13272 DATA 180,144,2,230,181,230,165
13280 DATA 165,165,201,3,208,203,165
13288 DATA 162,208,252,198,166,208,179
13296 DATA 162,0,189,110,52,240,6
13304 DATA 157,55,4,232,208,245,56
13312 DATA 173,98,10,237,96,10,141
13320 DATA 100,10,173,99,10,237,97
13328 DATA 10,13,100,10,240,17,144
13336 DATA 15,173,98,10,141,96,10
13344 DATA 173,99,10,141,97,10,32
13352 DATA 194,51,76,77,48,162,0
13360 DATA 189,95,52,240,6,157,6
13368 DATA 4,232,208,245,162,0,189
13376 DATA 104,52,240,6,157,22,4
13384 DATA 232,208,245,172,96,10,173
```

```
13392 DATA 97,10,32,145,179,32,221
13400 DATA 189,162,0,189,0,1,240
13408 DATA 6,157,27,4,232,208,245
13416 DATA 96,134,168,133,169,162,0
13424 DATA 160,0,189,13,54,240,10
13432 DATA 201,255,240,12,145,168,200
13440 DATA 232,208,241,32,250,51,232
13448 DATA 208,233,96,24,165,168,105
13456 DATA 40,133,168,144,2,230,169
13464 DATA 96,138,72,152,72,24,173
13472 DATA 98,10,105,2,141,98,10
13480 DATA 144,3,238,99,10,162,0
13488 DATA 181,168,72,232,224,8,208
13496 DATA 248,172,98,10,173,99,10
13504 DATA 32,145,179,32,221,189,162
13512 DATA 0,189,0,1,240,6,157
13520 DATA 12,4,232,208,245,162,8
13528 DATA 104,149,167,202,208,250,104
13536 DATA 168,104,170,96,234,234,234
13544 DATA 1,0,216,255,255,255,40
13552 DATA 0,37,33,34,36,160,32
13560 DATA 58,102,170,186,127,255,19
13568 DATA 3,15,18,5,58,32,48
13576 DATA 0,8,9,7,8,58,0
13584 DATA 135,129,141,133,160,143,150
13592 DATA 133,146,0,79,77,32,32
13600 DATA 78,80,99,80,32,79,99
13608 DATA 79,77,32,79,80,78,99
13616 DATA 99,99,77,79,80,32,79
13624 DATA 80,0,101,32,77,78,32
13632 DATA 103,32,103,32,101,32,101
13640 DATA 32,77,101,103,32,32,79
13648 DATA 76,100,101,103,32,101,103
13656 DATA 0,101,32,32,32,32,103
13664 DATA 32,103,32,101,32,101,32
13672 DATA 32,32,103,32,32,101,32
13680 DATA 32,101,32,99,32,103,0
13688 DATA 101,103,77,78,101,103,32
13696 DATA 103,100,101,32,101,103,77
13704 DATA 32,103,32,32,76,79,99
13712 DATA 101,103,99,101,103,0,76
13720 DATA 122,32,32,76,122,77,100
13728 DATA 100,100,78,76,122,32,77
13736 DATA 122,77,100,100,100,78,76
13744 DATA 122,32,76,122,0,0,160
13752 DATA 223,32?,32,233,231,32,233
13760 DATA 160,223,32,160,160,160,160
13768 DATA 231,160,160,160,160,160,32
13776 DATA 2,25,0,160,160,223,233
13784 DATA 160,231,233,160,226,160,223
13792 DATA 32,32,233,160,105,160,160
13800 DATA 0,160,160,160,160,160,231
13808 DATA 160,160,98,160,231,32,233
13816 DATA 160,105,32,160,160,160,160
13824 DATA 32,32,7,1,18,25,0
13832 DATA 160,160,95,105,160,231,160
13840 DATA 160,226,160,231,233,160,105
13848 DATA 32,32,160,160,0,160,160
13856 DATA 32,32,160,231,160,160,32
13864 DATA 160,231,160,160,160,160,231
13872 DATA 160,160,160,160,160,32,13
13880 DATA 1,18,19,1,0,255,32
13888 DATA 32,32,9,32,32,32,32
13896 DATA 32,13,15,22,5,32,20
13904 DATA 8,5,32,34,81,34,32
13912 DATA 21,19,9,14,7,0,32
13920 DATA 32,32,30,32,32,32,32
13928 DATA 32,20,8,5,32,12,5
13936 DATA 20,20,5,18,19,58,32
13944 DATA 32,32,0,32,32,32,93
13952 DATA 0,10,60,67,81,67,62
13960 DATA 11,32,32,9,32,61,32
13968 DATA 13,15,22,5,32,21,16
13976 DATA 0,32,32,32,93,32,32
13984 DATA 32,32,32,10,32,61,32
13992 DATA 13,15,22,5,32,12,5
14000 DATA 6,20,0,32,32,32,22
14008 DATA 32,32,32,32,32,11,32
14016 DATA 61,32,13,15,22,5,32
14024 DATA 18,9,7,8,20,0,32
14032 DATA 32,32,13,32,32,32,32
14040 DATA 32,13,32,61,32,13,15
14048 DATA 22,5,32,4,15,23,14
14056 DATA 0,255,160,32,160,32,32
14064 DATA 32,32,3,8,15,15,19
14072 DATA 5,32,19,16,5,5,4
14080 DATA 32,6,1,3,20,15,18
14088 DATA 58,0,0,160,32,160,32
14096 DATA 32,177,32,61,6,1,19
14104 DATA 20,44,32,178,32,61,32
14112 DATA 13,15,4,5,18,1,20
14120 DATA 5,44,32,179,32,61,32
14128 DATA 19,12,15,23,0,0,160
14136 DATA 32,160,32,32,32,32,32
14144 DATA 32,32,32,32,32,32,32
14152 DATA 32,16,18,5,19,19,32
14160 DATA 145,32,20,15,32,17,21
14168 DATA 9,20,46,0,255,153,143
14176 DATA 149,167,146,133,160,129,137
14184 DATA 142,142,133,146,161,161,32
14192 DATA 255,32,256
```

©

D.E.S.-SOFT ™
a division of
DES-Data Equipment Supply Corp.

ZARCON

By Steven Prentiss

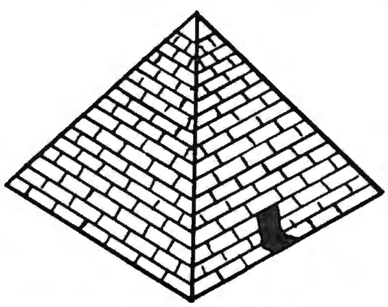
Destined to be one of the best arcade style games on commodore computers.Can you in your Zarcon Fighter shoot all the alien formations while flying through an asteriod field!

5 Levels of excitement
Joystick required
Vic-20 cassette 9.95
Commodore 64 cassette 9.95

SHROOMS

Subterranean Encounter
By Michael Koberstein

Inner Earth has invaded the surface world. The mutant spiders and the moles have taken humans captive and changed them into mus-srooms. To rescue them you must enter the caverns. You are their only hope. Good Luck!
Vic-20 & 8K Expander,-Joystick-cass 9.95



SPHINX

3-D ADVENTURE

By Cory Christensen

You the Adventurous Explorer are looking for the lost treasure of the Sphinx. It is rumored to be in the Ancient Pyramid of RA. Beware of deadly traps and hidden passage ways.
Commodore 64-cassette 9.95

You are the commander of squadron of laser ships. It is your duty to defend the cities of earth against incoming alien attack. Spectacular Hi-Res graphics and machine code for super fast action.
Joystick Vic-20 cassette 9.95

SEE YOUR LOCAL DEALER TO SEE OUR FINE PRODUCTS

Dealer inquiries invited *Software Distribution Available* **Programs wanted**

(714) 778-5455 · DES · Data Equipment Supply Corp. 8315 Firestone Blvd., Downey, CA 90241 · (213) 923-9361

ATARI® Printers/Etc.

ASTRA 1620 DISK DRIVE SYSTEM

600XL .... CALL
1200XL .... CALL*

*Reflects $100 Atari Rebate

1050 DRIVE	$335	MICROBITS INFC	$78
1025 PRINTER	$399	80 COLUMN BD	$249
1020 COLOR PTR	$219	TECHNICAL NOTES	$25
1027 PRINTER	$279	REAL TIME CLK	$38
1010 RECORDER	$72	810 DRIVE	$419
850 INTERFACE	CALL		

DRIVES		MEMORIES	
RANA 1000	$319	48K RAM (INTEC)	$95
PERCOM 88-S1	$358	64K RAM (INTEC)	$119
PERCOM AT 88-S2	$539	48K RAM (MOSAIC)	$109
PERCOM 40-S1	$510	64K RAM (MOSAIC)	$145
PERCOM 40-S2	$799	128K RAM DISK	$299
PERCOM 44-S1	$639	32K RAM (MOSAIC)	$68
PERCOM 44-S2	$929		

GEMINI 10X .. $289
GORILLA $199
PROWRITER . $345
SMITH TPI ... $488

CITOH
Prowriter $345
Prowriter II $629
Starwriter $1149
Printmaster $1448

NEC
8023 A-C $409
3510 .... $1375
3530 .... $1579
3550 .... $1779
7710/7730 .... $1998

SILVER REED P $669
QUME 11/40 + ... $1299

OKI-DATA
Microline 82A $398
Microline 83A $638
Microline 84P $958
Microline 92 .... $488
Microline 93 .... $858

DIABLO
620R $939
630R $1719

MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

SPECIAL $469

MONITORS

AMDEK
Color I $289
V300 .... $139
V300A $149
Color II $449

NEC
GRN (JB1260) $115
GRN (JB1201) $155
Color Composite $298
RGB Color $598

MODEMS

HAYES
Smartmodem $209
Smartmodem 1200 .... $498
Micromodem II $259

NOVATION
J-Cat $99
Apple Cat II $259
D-Cat $149

ANCHOR AUTOMATION
Mark I or II Modem ... $78

D = DISK
R = ROM (Cartridge)
C = CASSETTE

COSMIC™

ATARI SOFTWARE

ADVENTURE INT'L
Adv. 1-12 each (C) $18
Preppie (C/D) $20
Preppie II (C/D) $23
Diskey (D) $33
Sea Dragon (C/D) $23
Bug Off! (C/D) $20
Tutti-Frutti (C/D) $17
Saga 1-3 (ea.) (D) $27
Lunar Lndr . (C) $11 (D) $15
Galactic Empire (C) .. $14
Galactic Trader (C) .. $14
Stratos (C/D) $23
Rear Guard (C) $14 (D) $17

APX
Eastern Front (C/D) .. $23
747 Land Sim. (C/D) .. $17
Fig-Forth (C) $30
Family Cash Flow (D) . $17
Downhill (C/D) $17
Avalanche (C/D) $17
Outlaw/Howitzer (C/D) $17
Salmon Run (C/D) $17
Data Management (D) .. $17
Galahad/Holy Grail (D) $21
Adv. Music System (D) $21

ATARI INC.
Microsoft Basic II (R) . $62
Mickey in Great Outdoors (C/D) $36
Paint (D) $30
Speed Reading (C) ... $54
Qix (R) $30
Dig Dug (R) $30
Atari Writer (R) $68
Time Wise (D) $23
Visicalc (D) $139
Juggles House (C/D) .. $22
Juggles Rnbw (C/D) .. $22
Pilot (Home) (R) $55
Galaxian (R) $30
Defender (R) $30
ET $34
Microsoft Basic (D) .. $62
Macro Ass. & Edit (D) $62
Assembler Editor (R) . $42
Basic Cartridge (R) ... $45
Pac Man (R) $30
Centipede (R) $30
Caverns of Mars (D) .. $28
Star Raiders (R) $30
Conv. Lang. Ea. (C) ... $42
Music Composer (R) .. $31
Super Breakout (R) .. $26
My First Alphabet (D) $26
Prog. 2 & E (ea.)(C) ... $21
Word Processor (D) . $102
Pilot (Educ.) (R) $92
Touch Typing (C) $19
Home File Mngr (D) .. $36
Bookkeeper (D) $102
Family Finance (D) ... $35
Prog. 1 (C) $18
Scram (C) $18
Asteroids (R) $27
Space Invaders (R) ... $27
Missile Command (R) . $27
Telelink (R) $21
Superman III (R) $34
Basketball (R) $24
Donkey Kong (R) $34

AVALON HILL
VC (D) $17
B-1 Nuc. Bomber (C) .. $12
Legionnaire (C) $23
Empire of Overmind .... (D) $23 (C) $20
Tanktics ... (D) $20 (C) $17
Comptr Stock & Bonds .... (D) $17 (C) $14
Dnieper River Line (D) $20
Voyager (D) $17
Controller (D) $20
GFS Sorceress .... (D) $23 (C) $20
Telegard .. (D) $19 (C) $16
Vorrak (D) $18
Lord of Karma (C) $14

AVALON HILL (Cont'd)
Moon Patrol (C) $17
Planet Miners (C) $12
Conflict 2500 (C) $12
Space Station Zulu .... (D) $17 (C) $14
Flying Ace . (D) $21 (C) $18
Gypsy ..... (D) $18 (C) $15
Galaxy ...... (D) $17 (C) $14
Guns of Fort Def. (C) . $14
Nuke War (C) $12
Andromeda Conquest .... (D) $16 (C) $13
Close Asslt (D) $23 (C) $20

BRODERBUND
Sky Blazer (D) $22
Bank St. Writer (D) ... $46
A.E. (D) $23
Arcade Machine (D) .. $39
Choplifter .. (D) $23 (R) $29
Labyrinth (C/D) $20
Serpentine . (D) $23 (R) $27
Steller Shuttle (C/D) .. $17
Apple Panic (C/D) ... $20
Genetic Drift (C/D) ... $20
David's Midnight (D) .. $23
Sea Fox (D) $20
Track Attack (D) $20
Operatn Whrlwnd (D) . $23
Match Boxes (C/D) ... $20

CDY CONSULTING
Pogoman (C/D) $27
Mad-Netter (C/D) $23
It-Is-Balloon (C/D) $23

CBS
Mountain King (R) ... $27
Boulders & Bombs (R) $27
Krazy (each) (R) $27

CONTINENTAL SOFT.
Home Accountant (D) $48
Tax Advantage (D) ... $39

DATA MOST
Pig Pen (D) $20
Mating Zone (D) $20
Roundabout (D) $20
Bilestoad (D) $20
Night Raiders (D) $20
Monster Smash (D) $20

DATASOFT
Text Wizard (D) $65
Graphic Master (D) ... $27
Micro Painter (D) $23
Lisp Interpreter (D) ... $79
Graphics Gen (D) $17
Basic Compiler (D) ... $65
Zaxxon (C/D) $27
Teletalk (D) $34
Pac. Coast Hwy (C/D) . $20
Clowns/Balloons (C/D) $20
Spell Wizard (D) $53
Canyon Climber (C/D) $20
Sands of Egypt (D) ... $27
O'Riley's Mine (C/D) .. $23
Rosen's Brigade (C/D) $23
Moon Shuttle (D) $27

DON'T ASK
Sam (D) $39
P.M. Animator (D) $23
Teletari (D) $27
Poker Sam (D) $17
Word Race . (D) $17 (C) $14

EDU-WARE
Prisoner II (D) $27
Spelling Bee (D) $27
Rendevous (D) $27
Compu/Read .... (D) $21 (C) $15
Compu/Math-FR .... (D) $27 (C) $21
Compu/Math-Dec .... (D) $27 (C) $21

EDUCATIONAL SOFT.
Tricky Tutorial 1,2,3 or 4 (C/D) $15
Tricky Tutorial 5,6 or 7 (C/D) $22

SPECIALS

Gemini 15X Printer	$379
Axiom AT-100 Printer (with interface	$229
Astra Double Density Dual Drive	$469
Rana 1000 Drive	$319
Bit-3 80 Column Board	$245
Mannesmann Talley 160L Printer	$589
Atari 400 Keyboard (In Home)	$35
Programmer Kit .. $48 Entertainer Kit	$64
Wico Joystick ... $23 Wico Trackball	$49

COSMIC COMPUTERS UNLIMITED

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789

ORDER LINES OPEN MON-SAT 8 am - 8 pm

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

(714) 594-5204

FOR TECHNICAL INFO, ORDER INQUIRIES, OR FOR CALIFORNIA ORDERS

Add $2.50 shipping per software order in continental U S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10 00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock Personal checks require 4 weeks to clear MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change

EPYX-AUTO. SIMULATION
Rescue at Rigel (C/D) . $20
Datestones/Ryn (C/D) $14
Upr Rchs Apshai (C/D) $14
Ricochet (C/D) $14
Keys/Acheron (C/D) .. $14
Curse of Ra (C/D) $14
Danger/Drindisti (C/D) $14
Jumpman (C/D) $27
Escp Vulcans Isle (D) . $20
Crypt of Undead (D) .. $20
Nightmare (D) $20
Armor Assault (D) $27
Monster Maze (R) $27
Alien Garden (R) $27
Plattermania (R) $27
Morloc's Tower (C) $14
King Arthur's Heir .... (D) $20 (C) $14
Hellfire Warrior (C/D) . $27
Invasion Orion (C/D) .. $17
Temple of Aps. (C/D) . $27
Star Warrior (C/D) $27
Dragon's Eye (D) $20
Crush Crumble (C/D) . $20

GEBELLI
Firebird (R) $27
Embargo (R) $30
Candy Factory (D) $20
Match Racers (C/D) .. $20
Dr. Goodcode (D) $20

HAYDEN
Go .... (D) $23 (C) $20
Sargon II .. (D) $23 (C) $20
Bulldog Pinball (C) ... $20

HES
Coco (C/D) $34
Coco II (C/D) $27

INFOCOM
Suspended (D) $34
Zork I, II or III (D) $27
Starcross (D) $27
Deadline (D) $34
Witness (D) $34

INHOME
Baseball .. (R) $23 (D) $23
Crypts of Terror .... (D) $23 (C) $20

INNOVATIVE DESIGN
Pool 1.5 (D) $23
Juggler (D) $20
Speedway Blast .... (D) $20 (R) $27
Pool 400 (R) $27

JV SOFTWARE
Jrny to Plnts (C/D) $20
Action Quest (C/D) ... $20
Ghost Encount. (C/D) . $20

LJK
Letter Perfect (D) ... $104
Data Perfect (D) $74
Letter Perfect (R) $137
Edit 6502 (R) $137

MONARCH DATA SYS.
ABC Compiler (D) $48

ODESTA
Chess (D) $45
Checkers (D) $34
Odin (D) $34

OPTIMIZED SYSTEMS
C-65 (D) $58
Bug-65 (D) $23
Max-65 (D) $58
Basic A + (D) $58
Action (D) $65
Speed Read Plus (D) .. $40

PHOENIX
Birth of Phoenix (D) .. $16
Adv. In Time (D) $20

QUALITY
Fastgammon (C) $14
Name That Song .... (D) $13 (C) $11
Starbase Hyperion .... (D) $17 (C) $14
Ali Baba 40 Thvs (D) .. $20
Jeeper Creepers (D) .. $20

ROKLAN
Gorf .... (D) $27 (R) $30
Wizard/Wor . (D) $27 (R) $30
Dlx Invaders (D) $23 (R) $27
Anti Sub Ptrl (D) $20 (C) $14
Telecom (D) $49
Space Journey (R) ... $30

SENTIENT
Gold Rush (D) $23
Cyborg (D) $23

SIERRA ON-LINE
Mission Asteroids (D) $17
Ulys.& G dn Fleece (D) $27
Crossfire .. (D) $20 (R) $23
Mouseattack (D) $23
Jawbreaker (D) $20 (R) $23
Threshold (D) $27
Softporn (D) $20
Ultima II (D) $39

SIERRA ON-LINE
Marauder (D) $23
Lunar Leeper (D) $20
Wiz & Princess (D) ... $22
Frogger (C/D) $23

SIRIUS
Alpha Shield (R) $27
Wavy Navy (D) $23
Bandits (D) $23
Space Eggs (D) $20
Sneakers (D) $20
Way Out (D) $27
Type Attack (D) $27
Repton (D) $27
Critical Mass (D) $27
Fast Eddy (R) $23
Worm War (R) $23

SPINNAKER
Snooper Troop 1,2 (D) . $30
Kindercomp (D) $20
Rhymes & Riddles (D) $20
Hey Diddle Diddle (D) . $20
Srch Amzng Thngs (D) $27
Story Machine (D) $23
Face Maker (D) $23

STRATEGIC SIM.
Cosmic Balance (D) .. $27
Cosmic Balance II (D) $27
Tigers In Snow (C/D) .. $27
Battle of Shiloh (C/D) . $27
Battle of Norm. (C/D) . $27
Galactic Gladiator (D) $27
Cytron Masters (D) ... $27
Knights/Desert (C/D) .. $27
Combat Leader (C/D) . $27

SUBLOGIC
Pinball (C/D) $20

SWIFTY
J. White Music Lessons .... (C/D) $20
Space Shuttle (D) $20

SYNAPSE SOFTWARE
File Mngr 800 + ... $65
Protector II (D) $23 (R) $29
Shamus ... (D) $23 (R) $29
Fort Apocalypse (C/D) $23
Shamus II (C/D) $23
Necromancer (C/D) ... $23
Pharoh's Curse (C/D) . $23
Slime (C/D) $23
Page 6 (D) $23
Reptilian (C/D) $23
Picnic Paranoia (C/D) $23
Claim Jumper (C/D) .. $23
Drelbs (C/D) $23
Shadow World (C/D) .. $23
Survivor (C/D) $23
T.N.T. (C/D) $23
N.Y.C. (C/D) $23
Slamball (C/D) $23
Blue Max (C/D) $23
Nautilus (C/D) $23
Chicken (R/D) $23
Quasimodo (C/D) $23
River Quest (C/D) $23

SYNERGISTIC
Programr Wkshp (D) .. $23
Graph Workshop (D) .. $27
Disk Workshop (D) $23
Worlock's Revenge (D) $23

THORN EMI
Soccer (R) $34
Jumbo Jet (R) $34
Submarine Comm. (R) $34
Kick Back (R) $34
Pool (C) $22
Humpty Dumpty (C) .. $22
Hckry Dckry Dck (C) .. $22
Darts (C) $22

USA
Atari World (D) $39
3-D Sprgrphcs (C/D) .. $27
Survival Adv. (C/D) ... $17

VERSA
Mind Bgglrs I (D) $14 (C) $12
Globe Master (D) $20

MISCELLANEOUS
Financial Wizard (D) .. $41
Castle Wolfenstein (D) $20
Master Type (D) $27
Astro Chase (D) $22
Miner 2049er (R) $34
Cypher Bowel (C) $33
Bug Attack (C/D) $20
Raster Blaster (D) $20
Air Strike (C/D) $27
Prism (D) $18
Megalegs (C) $23
Cap'n Cosmo (D) $19
Spy's Demise (D) $14
Galac. Chase (D) $20 (C) $18
Kid Grid (C/D) $20
Battle Trek (D) $20
Snapper (D) $19
T&F-PMP Property Management (D) .. $149
Millionaire (D) $39

VIC 20

1541 DISK DRIVE $239

1701 Color Monitor .. $255
1525 Printer $239
1520 Color Ptr $169
Hescard $23
Hes Sound Box $13
Data 20 Exp. Chassis $56
1530 Recorder $59
1600 Modem $59
1650 Auto Modem ... $158
Microtek 16K Ram $82
UMI 3K Ram $56

Large Selection of Vic 20 Software at Great Prices!

VIC 20 SOFTWARE VIC 20

APPLIED SYSTEMS
Number Gulper (C) ... $17
Number Chaser (C) ... $17
BRODERBUND
Martian Raider (C) ... $14
Multisound Synth. (C) $14
Shark Trap (C) $14
Sky Blazer (R) $27
Seafox (R) $27
AE (R) $27
CBS
K-Razy Patrol (R) $29
K-Razy Antiks (R) $29
COMM-DATA
Pakacuda (C) $13
Sketch and Paint (C) .. $13
Eight Ball (C) $13
Invader Educ. Series (each) (C) $13
COMPUTERMAT
Paratroopers (C) $15
Bug Blast (C) $12
Cricket (C) $12
CREATIVE S.W.
Black Hole (R) $36
Trashman (R) $36
Astroblitz (R) $36
City Bomber (R) $20
Apple Panic (R) $36
Choplifter (R) $36
Serpentine (R) $36
Videomania (R) $36
Terraguard (R) $36
EPYX
Monster Maze (R) $27
Sword of Fargoal (C) . $20
Ricochet (C) $14
Rescue at Rigel (C) ... $20
Temple of Apshai (D) . $27
HES
Gridrunner (R) $27
Raid of Isram (C) $14
Robot Panic (R) $27
Protector (R) $29
Shamus (R) $27
Predator (R) $27
Syn The Sound (R) ... $39
Aggressor (R) $27
Heswriter (R) $27
Turtle Graphics (R) ... $27
Hesmon (R) $27
Vic Forth (R) $39
6502 Prof.Dev.Sys. (C) $20
Torg (C) $13
Concentration (C) $12
Fuel Pirates (C) $12
Simon (C) $12
Vic Trek (C) $13
Co Co II (C/D) $27
IMAGIC
Demon Attack (R) $25
Atlantis (R) $25
LITTLE WIZARD
Pro Football (C) $15
Cosmic Crusader (C) . $12
LOGISTIC
X-Rated (C) $9
Algebra Wiz (C) $9
Profess. Journal (C) .. $36
Viccalc (C) $36
Budget & Bill Payer (C) $15
Accounts Payable (C) $36
Mad Bomber (C) $10
MIS
Vic Vango (C) $10
Galactic Crossfire (C) $11
Checkbook (C) $14
Fruitfly (C) $10
Alien Soccer (C) $11
NUFEKOP
Krazy Kong (C) $11
3-D Man (C) $17
Exterminator (C) $20
ON LINE
Crossfire (C) $20
Ultima (C) $14

QUICK BROWN FOX
Prof. Word. Proc. $48
SIRIUS
Snake Byte (R) $27
Fly Wars (R) $27
Type Attack (R) $27
Fast Eddy (R) $27
Deadly Duck (R) $27
Turmoil (R) $27
Plasmania (R) $27
Spider City (R) $27
Squish 'Em (R) $27
Final Orbit (R) $27
Bumper Bash (R) $27
SPECTRAVISION
Cave In (R) $27
Number Crunch (R) ... $27
Reaganomics (R) $27
Gold Mine (D) $27
Ape Escape (D) $27
STARTECH
Asteroids (C) $18
Backgammon (C) $18
Ski Run (C) $12
Meteor (C) $9
Vic Men (C) $18
Vic Panic (C) $18
SYNAPSE
Harrier (C) $23
Squeeze (C) $23
Astro Patrol (C) $23
TAYLORMADE
Fun Fractions (C) $18
Vic Lemonade (C) $12
Tch Typing Tutor (C) .. $13
T&F SOFTWARE
Word Search (C) $14
Sports Search (C) $14
Arcade Search (C) ... $14
TOTL
Mailing List (C) $14
Time Management (C) $22
Research Asst. (C) ... $22
THORN EMI
River Rescue (R) $29
Vic Music Comp. (R) .. $29
Submarine Comm. (R) $29
Mutant Herd (R) $29
Fourth Encounter (R) . $29
TRONIX
Galactic Blitz (C) $17
Swarm (C) $20
Sidewinder (C) $20
Scorpion (R) $27
Gold Fever (R) $27
Deadly Skies (R) $27
UMI
Video Vermin (R) $27
Amok (C/R) $20
Outworld (R) $27
Satellites & Met. (R) .. $27
Subchase (C) $17
Kosmic Kamikaze (C) . $17
Meteor Shower (C) ... $11
Super Hangman (C) ... $14
Spiders of Mars (R) .. $27
Meteor Run (R) $27
Vicalc (C) $11
Vi Term A (C) $14
Vi Cat (C) $17
Vi Check (C) $17
Alien Blitz (C/R) $20
Sky Math (C) $11
Space Division (C) ... $11
The Alien (C) $17
Grand Master (C) $27
Renaissance (R) $33
Cloud Burst (R) $20
Skibbereen (R) $20
Wordcraft 20 (R) $65
VICTORY
Adv. Pak I (C) $14
Adv. Pak II (C) $14
Annihilator (C) $18
Grave Robbers (C) ... $12
Kongo Kong (C) $18
Trek (C) $12

64 LATE ARRIVALS 64

BATTERIES INCLUDED
Delphis Oracle(D) 114
Paper Clip(D) 89

COMM—DATA
(New Versions)
Supercuda(C/D) 18
Pegassus Odyssey(C/P) 18
Escape MCP(C/D) 18
Toddler Tutor(C/D) 18
Prim. Math Tutor(C/D) . 18
Math Tutor(C/D) 18
English Invaders(C/D) . 18

DATA 20
Pro Word Proc.(C) 21
General Ledger(C) 21
Accounts Rec.(C) 21
Elec. Spreedsheet(C) .. 21

EN—TECH
Studio 64(D) 28
Sprite Fun(C) 18

EPYX
sword Fargoal(C/D) 20
Crush Crumble(D) 20
Upper Reaches APS(D) 14

HES
Synthe Sound 64(R) ... 34
64 Forth(R)............ 40
Time/Money Mgr.(D) .. 48

INFO—DESIGNS
G/L (D) 61
A/P (D) 61
A/R (D) 61

NUFEKOP
Exterminator(C) 20
3-D Man(C) 16

RAINBOW
Pers. Finance Assist.(D) 41

PSYCOM SOFTWARE
Personality Analyzer(D) 24

SIERRA ON—LINE
Crossfire(D) 20
SOUTHERN SOLUTIONS
Bill Payer(A/P) 68
Business Man(G/L) 68
Paymaster(Payroll) 68
Bill Collector(A/R) 68
Widget(Inventory) 68

T & F SOFTWARE
Word Search(C) 15
Sport Search(C) 15
Arcade Search(C) 15

TIMEWORKS
Programming Kit 1(D) . 18
Programming Kit 2(D) . 18
Programming Kit 3 (D) . 18

SPECIALS

Gemini 10X Printer $289
Prowriter Printer $345
Gorilla Banana Printer $199
Card ? Printer Interface $60
The Connection Printer Interface $85
1541 Disk Drive $239
Koala Graphics Tablet $69
Flip N' File Diskette Box $21
Elephant Disks (10) $21
The Boss Joystick (Wico) $15
Wico Joystick $23
Wico Trackball $49

COSMIC COMPUTERS
UNLIMITED

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789

ORDER LINES OPEN MON-SAT 8 am - 8 pm

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

(714) 594-5204

FOR TECHNICAL INFO, ORDER INQUIRIES, OR FOR CALIFORNIA ORDERS

COMMODORE

CBM 64 .... CALL
1541 DISK DRIVE ... $239

1701 Color Monitor .. $255
1525 Printer $239
1520 Color Ptr $169
Card ? (Infc) $60
Light Pen $29
Cassette Infc $29
Card ? Software $16
1530 Recorder $59
1600 Modem $59
1650 Auto Modem ... $158
CMB 64 Ref Guide $18
The Connection (Infc) .. $85
MSD Disk Drive $339
PTI 45 Lot Board $59

Large Selection of 64 Software at Great Prices

64 SOFTWARE 64

ACCESS SOFTWARE
Neutral Zone (C/D) ... $26
Sprite Master (C/D) ... $27
AVALON HILL
Nukewar (C) $12
Planet Miners (C) $12
Androm. Conquest (C) $14
Midway Campaign (C) $12
North Atl. Convoy (C) $12
Comp. Stcks/Bnds (C) $15
Computer Football (C) $18
Telengard (C) $16
BATTERIES INCLUDED
Paper Clip (D) $89
D.Base $114
BRODERBUND
Choplifter (R) $29
Serpentine (R) $27
Seafox (R) $27
David's Midnight (D) .. $23
COMMODORE
Easy File (D) $75
Easy Finance (D) $38
Easy Mail (D) $38
Easy Script (D) $75
Easy Schedule (D) ... $59
Logo (R) $75
Pilot (D) $75
Assembler (D) $38
Music Machine (D) ... $25
Music Composer (D) .. $25
Meza Music (D) $75
Video/Music Supt. (D) $38
Jupiter Lander (R) $25
Radar Rat Race (R) ... $25
Sea Wolf (R) $25
Kickman (R) $25
COMM-DATA
Pakacuda . (C) $14 (D) $18
Escp. MCP . (C) $14 (D) $18
Centropods (C) $14 (D) $18
COMPUTERMAT
Arcade-Pak (C) $18
Education-Pak (C) $18
CREATIVE SOFTWARE
Moondust (R) $25
Trashman (R) $25
Save New York (R) ... $25
Astroblitz (R) $25
Household Fin. (D) ... $25
DATA 20
Video Pak 80 .... $139
Z80 Video Pak $229
EN-TECH
Finance Calc 64 ..... $34
Data Base 64 .... $56
Invoice Ease 64 ..... $56
EPYX
Temple of APS (D) ... $27
Upper Reach. APS (D) $14
Jumpman (D)......... $27
HES
HES Modem $59
6502 Prof.Dev.Sys.(C) . $22
Hesmon 64 (R) $27
Turtle Grapics II (R) .. $41
Heswriter 64 (R) $32
Gridrunner (R) $27
Retroball (R) $27
INFOCOM
Zork I, II or III (D) $27
Deadline (D) $35
Starcross (D) $27
JIN SAM
Mini-Jini (R) $75
LITTLE WIZARD
Pro.Mail.List (C)$22 (D)$25
Stockmaster (Inventory) (C)$25 (D)$28
LOGISTIC
Datacalc 64 (C) $55 (D) $59
Home Journal (D) $55

MICROSPEC
Payroll System (D) ... $73
Inventory Pkg (D) $73
General Ledger (D) ... $73
Disk Data Mgr (D) $62
Mail List Mgr (D) $41
Checkbook Mgr (D) ... $39
M-SOFT
M-File (D) $89
ON-LINE
Frogger (D) $23
Jawbreaker (D) $20
PACIFIC COAST SOFT.
PCS (80 Col BD, Word Proc, D.Base, Spreadsheet) CALL
Account PAC (C/D) ... $34
File PAC (D) $30
Editor PAC (D) $39
Inquire PAC (D) $57
Happy Tutor Typng (D) $18
PROFESS. SOFTWARE
Wordpro 3 + /64 (D) ... $68
QUICK BROWN FOX
Prof.Word Proc. (R) ... $50
RAINBOW
Writers Assistant $95
Spreadsheet Assist. .. $95
File Assistant $95
SIRIUS
Blade/Blackpoodle (D) $27
Type Attack (D) $27
Repton (D) $27
Critical Mass (D) $27
Snake Byte (D) $23
Way Out (D) $27
Fast Eddie (D) $23
Turmoil (D) $23
Spider City (D) $27
Squish'Em (D) $23
Final Orbit (D) $27
Alpha Shield (D) $27
SKYLES ELEC. WORKS
Busicalc (C/D) $52
Busiwriter (D) $72
SPINNAKER
Snooper Troops 1 (D) . $29
Facemaker (D) $23
Kindercomp (D) $20
Hey Diddle (D) $20
Most Amaz. Thing (D) . $27
SYNAPSE
Fort Apocalypse (C/D) $23
Survivor (C/D) $23
Drelbs (C/D) $23
Pharoh's Curse (C/D) . $23
Protector II (D) $23
Morgal (D) $23
Shamus (D) $23
TAYLORMADE
Touch Typing Tutor 3.0 (D) $21
TIMEWORKS
Rbbrs/Lost Tomb (C/D) $21
Wall Street (C/D) $21
Money Manager (C/D) $21
Data Master (C/D) $21
Dungeons of Alg. Dragons (C/D) $21
TOTL
Text 2.6 ... (C) $32 (D) $34
Label 2.6 .. (C) $15 (D) $17
Time Manager 2.6 (C) . $24
Time Manager 2.6 (D) . $27
Resrch Assist. 2.0 (C) . $24
Resrch Assist. 2.0 (D) . $27
UMI
Motor Mania (C) $20
Renaissance (C) $27
VICTORY
Annihilator (C/D) $16
Kongo Kong (C/D) $16
Trek (C/D) $14
Adv. Pack #1 (C/D) ... $16
Adv. Pack #2 (C/D) ... $16
Grave Robbers (C/D) .. $13
Chomper Man (C/D) .. $18

Apple Sounds – From Beeps To Music

Part 1

Blaine Mathieu

In this first of a two-part series, the author takes us from the simplest possible sound on the Apple to musical notes. Several useful demonstration programs are included.

Since I first acquired an Apple II+ about a year and a half ago, I have been fascinated by the strange noises I often hear. In this first of two articles I hope to save you all the trouble I went through in learning how to use APPLE sounds. Readers who already understand how to use CTRL-G and -16336 may want to skip the next section and go on to "Paddle Sounds."

Beeps And Clicks

Before you read this section, you should enter Program 1 on your computer and save it. Now run the program. If you entered it correctly, you should see SOUND #1 at the top, a line from the program, and a small menu.

Probably the first sound that you ever heard from your Apple's speaker was the bell sound. You can reproduce this in immediate mode by holding down the control key (labeled CTRL) and pressing the G key. In SOUND #1, you see that in line 30 a CHR$(7) is being printed – 7 is the numeric code for CTRL-G. If you are in Integer BASIC, you will have to use the format shown in line 35. In this line you'll see a PRINT with two quotes. Inside these quotes is a CTRL-G. The REM statement in line 37 shows how to type line 35. As you can see, control characters don't show up in a line listing or when you type them. An interesting side effect is that when you LIST your program, you will hear all the bell sounds in your program that are printed using the method in line 35.

In Program 1, the computer waits for you to hit a key. If you hit R, it will repeat any sound that might be produced by the above program lines. If you hit C, you will proceed to the next sound in Program 1. Any other key (except RESET) will cause no change.

Clicking -16336

Now hit C to go on to SOUND #2. In this program a simple FOR/NEXT loop is set up to beep the Apple's speaker ten times. Note the semicolon at the end of line 80; this prevents the screen from scrolling. If I hadn't used the semicolon, as each CTRL-G was printed the imaginary cursor would move down the screen until the screen started to scroll upward, which is, in most cases, undesirable.

Looking at SOUND #3, you will notice the number -16336, which is the memory address of the Apple's speaker. Every time this address is accessed, the Apple gives a little push on its speaker, creating a small click. PEEKing, as I have done in line 130, is just one simple way of accessing this address. If you missed the sound the first time, press R to hear it again.

SOUND #4 includes another simple loop that will PEEK the speaker's memory address 100 times. Instead of typing -16336 every time we wanted to use it, we assigned -16336 to the variable NO (for NOise). You may use any variable you wish, of course.

In SOUND #5 you'll notice line 250, which strings a lot of "clicks" together. This produces a longer noise than in SOUND #3 and a higher-pitched noise than in SOUND #4. As a rule, the closer your PEEKs, the higher-pitched your noise is going to be. In line 250 you will notice that we PEEKed -16336 a total of 15 times, a purely arbitrary number.

Finally, SOUND #6 demonstrates most of what we've learned about clicks. It uses a FOR/NEXT loop to cause line 320 to repeat 100 times. Line 320 has an assortment of minus and plus signs to show that it rarely makes a difference what you do to this location, as long as you access it.

Now on to something a little more exciting and complicated.

Paddle Sounds

Program 2 requires paddles or joystick. It is a simple BASIC program which reads a byte from the DATA statement and POKEs it into memory locations 768 ($300) to 786 ($312). The routine begins by CALLing 768. If you entered the program correctly, you should hear a fairly high-pitched whine, and as you move the paddles or joystick, this whine will change in pitch. You may leave the program by pressing RESET or CTRL-RESET, depending on your model.

Program 3 is the source code for this little machine language routine. Here is a quick explanation of the routine:

1. Put the paddle number in the X register.
2. Jump to the PREAD subroutine (see *Apple II Reference Manual*). PREAD acts as a delay, dependent on the paddle setting.
3. Tweak the speaker by accessing -16336 ($C030).
4. Repeat for next paddle.
5. Jump to beginning.

Since the pitch of the noise depends on how close together the tweaks are, the lower the paddle setting, the higher the pitch of the noise.

Making Music

Now we'll look at a program that lets you produce notes (and thus music) on your Apple. Of course, there are some limitations. For example, you won't be playing Beethoven's Fifth Symphony in five-part harmony with snare drum accompaniment. If you want that, many peripheral boards are available for the Apple which do amazing things. However, you can do quite a lot with the hardware already in your Apple.

Program 4 is a simple BASIC program that POKEs in a machine language subroutine, sets up a few parameters, and CALLs the subroutine. The program continues running until a key is pressed. Try running it. If you've never heard notes from your Apple, you will be quite pleased.

After the program has POKEd in the subroutine, it POKEs a random number (the pitch) into location 768 ($300) and POKEs a random number (the duration) into 769 ($301). The maximum value that can be POKEd into these locations is 255.

Program 5 is the source code for the "Note Producer" program that is POKEd into memory in Program 4. In essence, the program works much like "Paddle Sounds." The main difference is that instead of the paddles controlling the pitch of the sound, locations 768 and 769 control the pitch and duration of the tone. The source code contains comments that should help you understand what is happening.

As you can see by now, whenever you want a sound routine, you're going to have to access location -16336 ($C030). Try experimenting with Program 5 by POKEing in your own note values and hearing the results.

Next month, we'll look at a program called "Apple Music Writer," which will enable you to edit and play your own song. Until then, experiment with the programs here, and you're sure to come up with some surprising results.

Program 1: Sounds And Variations

```
5  REM  PROGRAM#1
10 I = 10: HOME
20  PRINT "SOUND #1": PRINT : LIST 30,3
    7
30  PRINT  CHR$ (7)
35  PRINT "": REM  CTRL-G
37  REM  PRINT"CTRL-G"
40  GOTO 10000
50 I = 50: HOME
60  PRINT "SOUND #2": PRINT : LIST 70,9
    0
70  FOR LOOP = 1 TO 10
80  PRINT  CHR$ (7);
90  NEXT
100  GOTO 10000
110 I = 110: HOME
120  PRINT "SOUND #3": PRINT : LIST 130

130 X =  PEEK ( - 16336)
140  GOTO 10000
150 I = 150: HOME
160  PRINT "SOUND #4": PRINT : LIST 170
    ,200
170 NO =  - 16336
180  FOR LOOP = 1 TO 100
190 X =  PEEK (NO)
200  NEXT
210  GOTO 10000
220 I = 220: HOME
230  PRINT "SOUND #5": PRINT : LIST 240
    ,260
240 NO =  - 16336
250 X =  PEEK (NO) +  PEEK (NO) +  PEEK
     (NO) +  PEEK (NO) +  PEEK (NO) +  PEEK
     (NO) +  PEEK (NO) +  PEEK (NO) +  PEEK
     (NO) +  PEEK (NO) +  PEEK (NO) +  PEEK
     (NO) +  PEEK (NO) +  PEEK (NO) +  PEEK
     (NO)
260  REM  FIFTEEN TIMES
270  GOTO 10000
280 I = 280: HOME
290  PRINT "SOUND #6": PRINT : LIST 300
    ,330
300 NO =  - 16336
310  FOR LOOP = 1 TO 100
320 X =  PEEK (NO) -  PEEK (NO) +  PEEK
     (NO) -  PEEK (NO) +  PEEK (NO) -  PEEK
     (NO) +  PEEK (NO)
330  NEXT
10000  POKE  - 16368,0: VTAB 20: HTAB 1
     : CALL  - 958: PRINT "'R' FOR REPE
    AT, 'C' TO CONTINUE ";: GET A$
10010  IF A$ <  > "R" AND A$ <  > "C" THEN
     10000
10020  IF A$ = "C" THEN 10100
10030  IF I = 10 THEN 30
10040  IF I = 50 THEN 70
```

```
10050  IF I = 110 THEN X =  PEEK ( - 16
   336): GOTO 130
10060  IF I = 150 THEN 170
10070  IF I = 220 THEN 240
10080  IF I = 280 THEN 300
10100  IF I = 10 THEN 50
10110  IF I = 50 THEN 110
10120  IF I = 110 THEN 150
10130  IF I = 150 THEN 220
10140  IF I = 220 THEN 280
10150  TEXT : HTAB 1: PRINT "END OF PRO
   GRAM#1"
```

Program 2: Paddle Sounds

```
10  REM  PROGRAM#2
20  FOR LOC = 768 TO 786: READ BYTE: POKE
     LOC,BYTE: NEXT LOC
30  DATA 162,0,32,30,251,141,48,192,162
     ,1,32,30,251,141,48,192,76,0,3
40  CALL 768
```

Program 3: Source Code For Paddle Sounds

```
           ORG   $300        ;768 DECIMAL
*********************************
*                               *
*PROGRAM#3 -   PADDLE     SOUNDS *
*                               *
*********************************
PDLZERO  EQU  $00
PDLONE   EQU  $01
PREAD    EQU  $FB1E
SPEAKER  EQU  $C030
START    LDX  #PDLZERO    ;SET UP FOR PADD
                          LE ZERO
         JSR  PREAD       ;GET DELAY FROM
                          PADDLE ZERO
         STA  SPEAKER     ;TWEAK SPEAKER
         LDX  #PDLONE     ;REPEAT P
                          ROCESS FOR PADDL
                          E ONE
         JSR  PREAD
         STA  SPEAKER
         JMP  START       ;START OVER
```

Program 4: Note Producer

```
5   REM  PROGRAM#4
10  FOR LOC = 770 TO 790: READ BYTE: POKE
     LOC,BYTE: NEXT
20  POKE 768, INT ( RND (1) * 255) + 1:
      POKE 769, INT ( RND (1) * 100) +
     1: CALL 770:X =  PEEK ( - 16384): IF
     X < 127 THEN  POKE  - 16368,0: GOTO
     20
30  DATA  173,48,192,136,208,5,206,1,3,
     240,9,202,208,245,174,0,3,76,2,3,96
40  POKE  - 16368,0
```

Program 5: Source Code For Note Producer

```
            ORG   $300       ;768 DECIMAL
*********************************
*                               *
*PROGRAM#5 -  NOTE.PRODUCER     *
*                               *
*********************************
TWEAK     EQU  $C030
PITCH     EQU  $300
DURATION  EQU  $301
          DS   2             ;MAKE SPACE FOR
                             PITCH AND DURATI
                             ON
START     LDA  TWEAK         ;TWEAK THE SPEAK
                             ER
BRANCH1   DEY
          BNE  BRANCH2
          DEC  DURATION      ;DURATION = DURAT
                             ION-1
          BEQ  RETURN        ;IF DURATION=0 T
                             HEN RETURN
BRANCH2   DEX
          BNE  BRANCH1
          LDX  PITCH
          JMP  START         ;CONTINUE TO SOUN
                             D NOTE
RETURN    RTS                ;GO BACK TO OPERAT
                             ING SYSTEM
```

Extra Instructions

Joel C Shepherd

Combining machine language instructions by creating new opcodes can result in more efficient programming in certain situations. Machine language programmers can conserve memory and increase execution time with the methods discussed here. This article explores the unofficial, hidden "commands" you can give the 6502.

The 6502 microprocessor can execute 151 instructions. There are 56 different types of instructions in its library, and each can be used with at least one of the 13 addressing modes available. Each instruction performs one operation, such as loading a register, setting a flag, or rotating a byte, and deals with one byte of data or one memory location. The 6502's machine language consists of nothing more than variations on these simple instructions. It is an extremely low-level language which is both a boon and a bane to programmers. While machine language is versatile and applicable to any programming problem, it can also be tedious in a particular sense. It can require programmers to use several instructions of the same type performing similar, yet separate, operations.

Consider the LDA and LDX instructions; both load a register and set the same processor status flags. They also share four addressing modes. One, however, loads the accumulator and the other loads the X register. There is no instruction which will load both. The language is so simple that it has no instructions with more than one memory location or register as an operand.

There is no reason why the 6502's instruction set cannot be expanded to include more sophisticated instructions. Such expansion, however, would probably also drive up the processor's cost. The designers of the 6502, in an effort to keep its cost low, decided to give it a simple machine language.

Creating New Opcodes

A possible solution to the problem would be to "trick" the processor into executing instructions it didn't "know." Imagine being able to load two registers simultaneously or being able to AND two registers with one instruction. There is a simple way to do just that and more.

If you examine a list of 6502 opcodes, you'll notice that none of the codes have the form a7 or aF (a is any hexadecimal digit). Codes of this form are not "official" 6502 opcodes, yet all are executable and potentially useful.

An example is opcode AF. Opcodes AD and AE are the LDA $aaaa and LDX $aaaa instructions respectively. Opcode AF combines the two: it loads the accumulator and the X register from the same location. For instance, these two instructions:

```
AD 00  16  LDA   $1600    ;load accumulator
AA         TAX            ;and X
```

can be replaced by:

```
AF 00  16  LDAX  $1600    ;load A and X
```

(LDAX is a new mnemonic. I've included mnemonics to represent the operations performed by the new instructions.) Using LDAX saves one byte and two cycles of execution time. That might not seem like much, but in a loop routine the time savings may be significant. (The execution times given for the routines using the new opcodes are estimates. More on that later.) Additionally, with this instruction, it is possible to load the accumulator from a zero-page location indexed by the Y register; this is not possible with the LDA instruction.

Opcode 87 is another example. Opcode 85 is the STA $aa instruction, and 86 is the STX $aa instruction. Opcode 87 executes both 85 and 86, but with a twist. It is impossible for a location to contain two different values at one time. Opcode 87 logically ANDs the accumulator with the X register, changing neither, and stores the result in the address specified. In other words, only the bits which are set in both the accumulator and the X register are set in the byte which is stored. So, the routine:

```
A5 AD      LDA   $AD      ;get char
87 AD      ANDX  $AD      ;mask with X
```

Summary Of Extra 6502 Instructions

Name/Description	Operation	Addressing Mode	Machine Language Form	Hex Op-Code	N	#	NZCV
ANDX							
AND A with X	A&X ÷ M	Zero Page	ANDX $aa	87	3	2	----
		Zero Page,Y	ANDX $aa,Y	97	4	2	
		Absolute	ANDX $aaaa	8F	4	3	
DCMP							
Decrement M,	A-(DEC M)	Zero Page	DCMP $aa	C7	8	2	???-
Compare to A		Zero Page,X	DCMP $aa,X	D7	10	2	
		Absolute	DCMP $aaaa	CF	10	3	
		Absolute,X	DCMP $aaaa,X	DF	11*	3	
ISBC							
Increment M,	A-(INC M)-C ÷ A,C	Zero Page	ISBC $aa	E7	8	2	????
Subtract from A		Zero Page,X	ISBC $aa,X	F7	10	2	
		Absolute	ISBC $aaaa	EF	10	3	
		Absolute,X	ISBC $aaaa,X	FF	11*	3	
LDAX							
Load A and X	M ÷ A,M ÷ X	Zero Page	LDAX $aa	A7	3	2	??--
		Zero Page,Y	LDAX $aa,Y	B7	4	2	
		Absolute	LDAX $aaaa	AF	4	3	
		Absolute,Y	LDAX $aaaa,Y	BF	4*	3	
RLAN							
Rotate M left,	(ROL M) &A ÷ A	Zero Page	RLAN $aa	27	8	2	???-
AND with A		Zero Page,X	RLAN $aa,X	37	10	2	
		Absolute	RLAN $aaaa	2F	10	3	
		Absolute,X	RLAN $aaaa,X	3F	11*	3	
RRAD							
Rotate M right,	(ROR M) + A + C ÷ A,C	Zero Page	RRAD $aa	67	8	2	????
Add with carry		Zero Page,X	RRAD $aa,X	77	10	2	
		Absolute	RRAD $aaaa	6F	10	3	
		Absolute,X	RRAD $aaaa,X	7F	11*	3	
SLOR							
Shift M left,	(ASL M)V A ÷ A	Zero Page	SLOR $aa	07	8	2	???-
OR with A		Zero Page,X	SLOR $aa,X	17	10	2	
		Absolute	SLOR $aaaa	0F	10	3	
		Absolute,X	SLOR $aaaa,X	1F	11*	3	
SREO							
Shift M right,	(LSR M)⊻ A ÷ A	Zero Page	SREO $aa	47	8	2	???-
Exclusive OR with A		Zero Page,X	SREO $aa,X	57	10	2	
		Absolute	SREO $aaaa	4F	10	3	
		Absolute,X	SREO $aaaa,X	5F	11*	3	
TSTA							
Test bit 2 in A	A&#$04 ÷ A	Absolute	TSTA $aaaa	9F	4	3	----
TSTX							
Test bit 2 in X	X&#$04 ÷ A	Absolute	TSTX $aaaa	9E	4	3	----

The following notation applies to this summary:

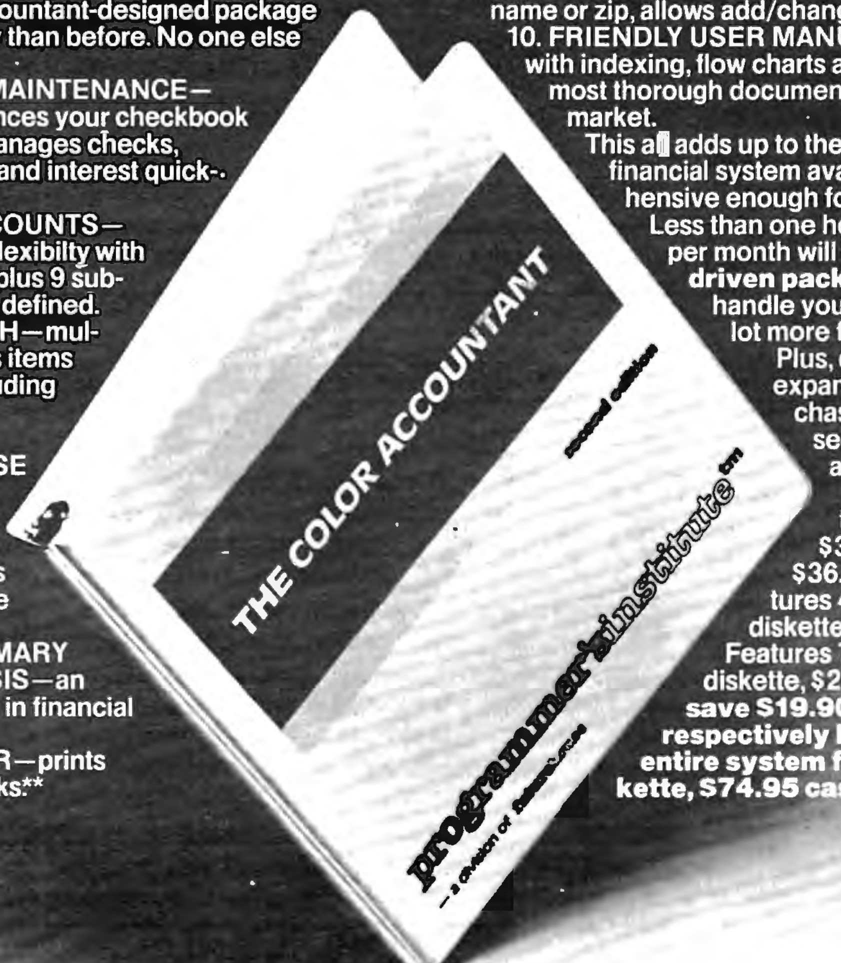
A	Accumulator	-	No Change, or Subtract	V	Logical OR
X,Y	Index Registers	+	Add	$aa	8-bit Zero-Page Address
M	Memory	&	Logical AND	$aaaa	16-bit Absolute Address
C	Carry Flag	⊻	Logical Exclusive OR	N	Number Of Clock Cycles
?	Change	÷	Transfer To	#	Number Of Storage Bytes
				*	Add 1 if page boundary is crossed

Complete Personal Accountant: we've made the best much more friendly.

If you have any doubts that we offer the best and most complete personal financial package available, look over the features listed below. Now we have the only package with **full screen editing** for Atari 400/800,* TRS-80 COLOR, Commodore 64* and VIC-20; the ability to move the cursor in any direction makes our accountant-designed package even more friendly than before. No one else offers all of these:

1. CHECKBOOK MAINTENANCE—automatically balances your checkbook with each entry; manages checks, charges, deposits, and interest quickly and accurately.
2. CHART OF ACCOUNTS—maximum of user flexibilty with up to 99 accounts plus 9 subcategories may be defined.
3. CHECK SEARCH—multi-reference; tracks items on every field including tax deductibles.
4. NET WORTH/INCOME/EXPENSE STATEMENT—know-exactly-where-you-stand program generates statements with the touch of a key.
5. DETAIL & SUMMARY BUDGET ANALYSIS—an absolute necessity in financial planning.
6. CHECK WRITER—prints personalized checks.**
7. PAYMENTS/APPOINTMENTS CALENDAR—monthly displays of up to 250 bills and 200 appointments.
8. COLOR GRAPH DESIGN PACKAGE—graphs all monthly files.
9. MAILING LIST—maintains all records, sorts by name or zip, allows add/change/delete.
10. FRIENDLY USER MANUAL—complete with indexing, flow charts and diagrams; the most thorough documentation on the market.

This all adds up to the finest personal financial system available—comprehensive enough for a small business. Less than one hour of data input per month will allow this **menu-driven package** to help you handle your finances with a lot more fun than drudgery. Plus, ours is the only expandable system; purchase the package in sections and add on as your financial needs grow. Features 1, 2, 3 and 6: $39.95 diskette, $36.95 cassette; Features 4 and 5: $29.95 diskette, $26.95 cassette; Features 7, 8 and 9: $29.95 diskette, $26.95 cassette; or **save $19.90 or $15.90 respectively by buying the entire system for $79.95 diskette, $74.95 cassette.**



*Random

Prices subject to change without notice. See your local dealer or order direct. New catalog available.
Add $3.00 for postage and handling. Credit card orders call toll free:

1-800-334-SOFT

DEALER INQUIRIES INVITED

programmer's institute

a division of **futurehouse**

p.o. box 3470, dept. C, chapel hill, north carolina 27514, 919-967-0861

(ANDX is a mnemonic for 87) loads the accumulator, ANDs it with the X register, and stores the result in location $AD. It occupies four bytes and takes six cycles to execute. Compare that to the following equivalent routine:

```
8A          TXA         ;X into A
25   AD     AND  $AD    ;AND "X" with $AD
85   AD     STA  $AD    ;store result
```

which requires five bytes and eight cycles to execute. One byte and two cycles are saved by using the first routine.

Opcode DF is an interesting instruction. Opcode DD compares the accumulator to an X indexed address, and opcode DE decrements the value in an X indexed address by one. Opcode DF first decrements the value in the address given and then compares the result to the accumulator. For example:

```
A9   A5        LDA  #$A5    ;loop terminator
DF   00   16   DCMP $1600   ;done?
D0   aa        BNE  $....   ;no, do next
```

(DCMP is a mnemonic for DF) loads the accumulator with #$A5, decrements the value in $1600 by one, compares $1600 to the accumulator, and branches if they are not equal. This could be used to control a loop.

How The New Opcodes Execute

In the table, I have included ten new instruction types, creating 41 new instructions. The operations performed by the new instructions are combinations of the "original" 6502 instructions. There are some general rules for predicting what one of the new opcodes will do.

First, all of the new instructions execute the two preceding instructions. Opcode DF executes both DD and DE, for example. The two preceding instructions we'll call the "sub-instructions" of the new instructions. So, a5 and a6 are sub-instructions of a7, and aD and aE are aF's sub-instructions.

If the sub-instructions are different types of instructions, the sub-instruction with the greatest opcode will be executed first; otherwise the sub-instructions will be executed simultaneously. For instance, since LDA and LDX are both load instructions, LDAX will execute them simultaneously. But, since ASL and ORA are different types of instructions, ASL (opcode 06) will be executed first, followed by ORA (05) when opcode 07 is executed. There is a definite inconsistency in this rule: it implies that the processor examines the current instruction and then decides to execute the sub-instructions either one at a time or simultaneously. The 6502 isn't that sophisticated. The rule, however, is the only one that explains the operations of the new opcodes. The given execution times are based on this rule.

Generally, the same operand is used for both sub-instructions. Consider:

```
07   19        SLOR  $19
```

(SLOR is a mnemonic for 07). This ASLs $19 and then ORAs $19 with the accumulator. Location $19 is the operand for both sub-instructions.

Opcodes 97, B7, and BF are special cases of the preceding rule. All three have sub-instructions which use different addressing modes, X indexed and Y indexed. The operands used by the new instructions are the given addresses indexed by the Y register. For instance, B5 is the LDA $aa,X instruction and B6 is the LDX $aa,Y instruction. Opcode B7, then, is the LDAX $aa,Y instruction.

Two Exceptions

There are two additional opcodes which are exceptions to all of the rules discussed: 9E and 9F. Opcode 9E logically ANDs the X register with the value #$04, keeping the X register intact. The result is stored in an absolute memory location. For example:

```
A6   F6        LDX   $F6
9E   40   03   TSTX  $0340      ;is bit 2 set?
```

(TSTX is a mnemonic for 9E) loads the X register, ANDs it with #$04 and stores the result in $0340. The origin of the #$04 is unknown.

Opcode 9F is similar to 9E: the accumulator is ANDed with #$04 and the result is stored in an absolute location. So, these instructions:

```
A9   06        LDA   #$06
9F   00   00   TSTA  $0000
```

(TSTA is a mnemonic for 9F) AND the accumulator with #$04 (yielding #$04) and store the result in $0000. The accumulator will contain #$06 when this routine is finished.

The execution times of the new instructions in the table are based on whether the sub-instructions are of the same type or not.

It seems that one of the major shortcomings of the new instructions is their specialization. For example:

```
4A   aa        LSR   $aa
45   aa        EOR   $aa
```

is probably not used in too many programs. The SREO (mnemonic) instruction (which replaces the above instructions) is too specialized to be of much use to anybody.

Currently, there are no assemblers which accept the new mnemonics as valid instructions. The opcodes, however, can be POKEd directly into the computer. An interesting, and useful, project would be to write an assembler which could handle the new instructions. It wouldn't be difficult to modify an assembler written in BASIC to recognize these new mnemonics. ©

Commodore DOS Wedges: An Overview

Jim Butterfield Associate Editor

Most Commodore machines have a small program called a "DOS Wedge" which allows convenient use of various disk commands. Recently, copies of this program have been in circulation without information; many users who have it don't know how to use it. Here are the details.

What It Does

DOS Wedge programs provide for three types of capability:

– the disk's error status may be checked with a simple command;

– a disk directory (or catalog) may be obtained without disturbing a program which has already been loaded;

– a number of "disk commands" may be easily transmitted to the disk.

How To Start The Wedge

Wedge programs are often self-relocating. If you LOAD the wedge program and say RUN, the program will find a spare place in memory and pack the DOS wedge system there. After this, you may type NEW and the wedge will remain in place: it's been packed into a safe part of memory.

Some wedge programs come in "pieces," or as a set of programs on the disk. The first program, called the "boot," loads in the others as necessary. This is usually the case with the C-64 Wedge, for example.

Commodore 64 and VIC-20 wedge programs often also come with a how-to program, explaining the system and how it works. Many PET/CBM wedge programs print brief instructions on the screen as they self-relocate.

For other systems, you may find it necessary to use the last resort of reading the instructions.

First Rules In Using The Wedge

DOS Wedge commands must be typed in as direct commands. You cannot include them as part of a program. This is deliberate: programs will run at full speed without the need to check for the extra wedge commands.

Wedge commands should start in column 1 of the screen. The primary wedge commands are flagged with either of two symbols:

> or @

The "greater than" sign was the original. It looks like a wedge and is partly responsible for giving the wedge program its name. On later Commodore computers, this character needed the shift key; to save finger work, the @ (at sign) character was allowed as an alternative.

A few supplementary commands which use other starting symbols are available. They tend to be less important, since their functions can be performed easily with conventional commands. The most common of these are:

/ – to load a program; and
↑ – to load and run a program.

Thus, a command such as /DOGLEG will cause a program named DOGLEG to be loaded from disk.

Pattern Matching

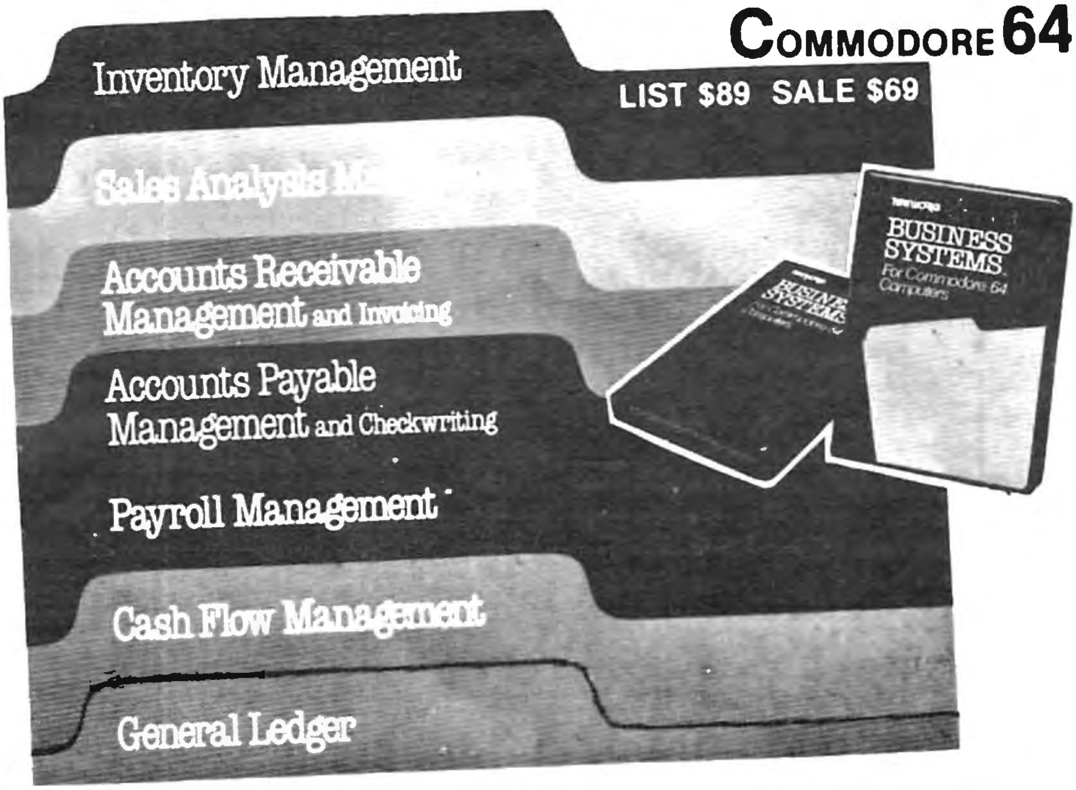
In many cases, a program name does not have to be specified exactly. We can give part of the name and use "pattern matching" to find one or more programs that match the name. The pattern matching characters are:

? – to match any single character; and
* – to match any stream of characters.

Thus, a command such as /DOG* will load the

PROFESSIONAL BUSINESS SOFTWARE SALE!

COMMODORE 64



Executive Quality by TIMEWORKS™
"Cadillac of Business Programs"

Each Program comes in a beautiful case and includes: *Disk*

- A Menu-driven program, sophisticated enough to provide complete Management and Product information, yet requires no prior computer or accounting knowledge to operate.
- A Manual written in easy-to-understand, people-friendly English, abundantly illustrated to provide further clarity and eliminate guesswork.
- A unique method of creating your own, unlimited array of reports—easily and quickly. YOU select the information you want, and YOU determine the sequence of the report column headings.
- Password Protection, to prevent unauthorized access to confidential data.
- A program wnich can be used by itself (stand-alone), or can be interfaced, one at a time, with other TIMEWORKS MANAGEMENT INFORMATION programs into a fully integrated accounting system.

PLUS: THESE PROFESSIONAL BUSINESS PROGRAMS -

- ***Script 64 - No. 1 Executive Word Processor - Disk*** — ***List $99 - Sale $59***
- ***Complete Data Base - Tape-Disk*** — ***List $89 - Sale $59***
- ***Electronic Spread Sheet (like Visicalc) - Tape-Disk*** — ***List $89 - Sale $59***

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

first program it comes to whose name starts with the characters DOG. This might include DOGLEG, DOGHOUSE, DOG, and DOGOOD. Similarly, a command such as /D?G will load a three-character name such as DIG, DOG, or DRG.

Pattern matching can be used for many DOS Wedge commands. You must not use it, of course, when you SAVE a program, since the program must have an exact name.

Disk Status

This is perhaps the most useful (and the briefest) DOS wedge command of all. Simply type the › or @ character, followed by the RETURN key, and you'll get the status of the disk.

Most of the time, the disk will not report an error, but will give you a message like: 00,OK,00,00. If there's an error, or immediately following a SCRATCH command (more on this later), you'll get a different message. Following a SCRATCH, you'll be told how many files have been removed from disk. If the disk is signalling an error condition, you'll be told what the error is.

Most of the time, the error will be an obvious one, and you'll spot the difficulty right away. You might have attempted to write to a protected disk, or one that is full. You might have tried to create a new file giving a name that already exists. You might be trying to read or load a file that isn't there. You won't need to translate the number if the disk reports 62,FILE NOT FOUND,00,00. Sometimes, however, the disk error number can give useful information; look it up in the manual.

4.0 Equivalent: Users with a 4.0 BASIC system can type PRINT DS$ and receive the equivalent information.

Non-wedge Equivalent: If you don't have a wedge, you'd need to write a program to open secondary address 15 and INPUT from that channel. (INPUT works only from within a program.) Get a wedge.

Directory Or Catalog

You may obtain a catalog from disk, without disturbing a program already loaded, by typing ›$ or @$. If you have a dual disk (two drives ganged together in a single housing), you may specify the drive by adding a number, yielding commands such as ›$0 or @$1. Many experienced users suggest that you'll do well to specify drive 0 for single disks: the zero in @$0 won't hurt and might help.

You may take a "specific" catalog by using a filename with pattern matching. For example, ›$0:D* will give only programs whose names start with the letter D. @$0:??? will report programs with exactly three characters in their names.

4.0 Equivalent: The command CATALOG or DIRECTORY will produce a directory. There's no simple way to get a pattern match with this.

Non-wedge Equivalent: You'll need to get a directory using a LOAD "$0",8 command. This destroys the program you have in place. (Get a wedge.)

Other Disk Commands

By following the › or @ sign with command characters, you may send a number of special commands to the disk. We'll list them below.

4.0 Equivalent: Specific commands are often available, such as SCRATCH, COLLECT, or HEADER. We'll deal with them individually below.

Non-wedge Equivalent: You must do this by opening secondary address 15, and then sending the command as part of a PRINT# sequence. For example, ›I0 could be matched with OPEN 15,8,15:PRINT#15,"I0". This is no hardship since it can be done with direct commands. The wedge is slightly more convenient.

Initialize

The command ›I0 or @I1 causes the disk electronics to "shake hands" with the appropriate disk drive for a smoother transfer of data. It shouldn't be needed, but can be most useful if you start encountering DRIVE NOT READY errors. On single disks, @I0 is recommended.

4.0 Equivalent: None. You'd have to use the secondary address 15 equivalent.

Header (Or New)

Watch this one: It's powerful, and will wipe the previous contents of a disk. @N0:DISKNAME,JB will format the drive over all its tracks, using the ID of "JB", and will prepare an empty directory with the title "DISKNAME". This job will take a few minutes.

@N0:ANOTHER will not format the disk. Instead, it will just wipe the directory and put the new title in place, in this case "ANOTHER". It only takes a couple of seconds to do this, but all your disk information will immediately disappear. The disk must have previously been formatted before this version of the command can be used.

You cannot use a disk until it has been formatted. Be very careful that all your working disks have different IDs (the two letter identifiers); this will give your disk-stored programs and files important protection against harm.

4.0 Equivalent: HEADER. The 4.0 system asks ARE YOU SURE? before it dives in and wipes the data from the disk.

Scratch

Removes one or more files from a disk and frees the directory and disk space. A command of @S0: TURNPIKE removes the file called TURNPIKE, if there is one by that name on the disk.

DISK CLEANER

Everyone Needs A

Disk Drive Doctor

Sale $19.95

FACTS

- It has been estimated that 60% of all drive downtime is directly related to poorly maintained drives.
- Drives should be cleaned each week regardless of use.
- Drives are sensitive to smoke, dust and all micro particles.
- Systematic operator performed maintenance is the best way of ensuring error free use of your computer system.

The Cheetah disk drive cleaner can be used with single or double sided 5½" disk drives. The Cheetah is an easy to use fast method of maintaining efficient floppy diskette drive operation.
The Cheetah cleaner comes with 2 disks and is packed in a protective plastic folder to prevent contamination. List $29.95 / Sale $19.95

Features	Cheetah	3M	Discwasher	Verbatim
Cleanings Per Kit	40 +	25	30	2 original - 10 refill
Price	$19.95	$34.95	$24.95	$12.95
Abrasivity	None	Minimum	High	Moderate
Programming Required	No	No	Yes	No

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

You may use pattern matching, but watch it! A command of ›S0:C* will remove all files which start with the letter C. A command of @ S0:* will remove all files, giving the same effect as New, but slower.

4.0 Equivalent: SCRATCH.

Verify (Validate, Collect)

A widely misunderstood command. This causes the disk unit to try to rebuild its map of free blocks, removing incomplete files as it does so. It does not report errors; it just reconstructs the Block Availability Map (BAM), using information from the directory.

If you have somehow created an incomplete file, it will appear in the directory with an asterisk beside the file type (e.g., *SEQ). You should immediately remove it with @ V0 in order to avoid complications. (*Do not, repeat do not, try to scratch such a file.*) You'll probably need to wait awhile: rebuilding the BAM often takes time.

4.0 Equivalent: COLLECT.

Rename (Name)

Lets you change the name of a file. To change WATER to ICE, you type @ R0:ICE=0:WATER. Think of the syntax as similar to BASIC: the value to the left of the equals sign indicates what has been changed.

4.0 Equivalent: RENAME ... TO ...

Copy

Lets you make a new copy of a program on disk. Most useful on dual disk units, but you can make an extra copy even when you have a single disk. A command such as C0:DOG=0:CAT will make an extra copy of file CAT, naming the new copy DOG.

A lesser-known feature of the Copy command is that it allows two or more files to be put together (or "concatenated") into a single file. This can be done with a command such as @ C0:FRACAS=0:DOG,0:CAT which puts files DOG and CAT together into a single new file called FRACAS. This feature is most useful for data files, by the way; you *can* stick programs together, but the result won't be very useful.

4.0 Equivalent: COPY ... TO ...; or CONCAT ... TO ...

Duplicate (Backup)

Only for dual drives. Allows a disk to be copied, block for block, from one drive to the other. Dangerous! Be sure you specify the correct direction, or you're in big trouble. So @ D1=0 will make an exact copy of the information on drive 0 placing it onto drive 1. If you intended to go the other way, it's too late.

4.0 Equivalent: BACKUP D0 to D1. The system does not ask ARE YOU SURE? It just goes ahead.

Other Commands

There are other commands that may be sent down the command channel, but I don't recommend their use with a wedge. In fact, I recommend careful study of the manual or sample programs before using them in any form.

There are several DOS wedges; it's impossible to cover details on each of them. But they have similar patterns.

Learning to use the DOS wedge effectively is almost the same as learning to use the disk effectively. More correctly, it's learning how to use the disk – easily and effectively. ©

Protector For VIC-20

George Trepal and Doug Smoak

Disabling certain commands on the VIC can be a very useful way to protect your programs. This article demonstrates how to disable the SAVE, RESTORE, STOP, and LIST commands with a few simple POKEs. There are some techniques here for the 64, too.

Sometimes it's nice to have a VIC that isn't fully functional. Maybe you're a teacher and you don't want your program listed by students. Or perhaps you've written a game you don't want someone to save. Let's look at a few ways to block your VIC's SAVE, RESTORE, STOP, and LIST.

Obviously, it's nice to have a program that can't be SAVEd (that is, copied). When you SAVE a program, the VIC goes to a *table* (a list of memory addresses in RAM) to get the address of the SAVE routine. Once it has the address, it jumps to the routine and puts the program on tape. Guess what happens if it gets the wrong address.

If you POKE the two values under the protection method for SAVE, the address now points to the LOAD routine. When you tell the VIC to SAVE, it tries to LOAD instead. It would have been just as easy to give the address of the NEW or LIST routine or anything else. In fact, as long as the value of 818 is not 113, or the value of 819 is not 246, the machine will not SAVE. You can try any numbers you want and see what happens. To reactivate the SAVE routine, POKE in the normal values in the table. Or you can press the RUN/STOP and RESTORE keys at the same time, and that will reassign the correct addresses.

Now let's get the RESTORE key out of action so that it can't correct the wrong address. Check the table and use the two POKEs. Notice that the table doesn't give the normal values for the RESTORE key. The normal values are obtained by PEEKing the table values used before POKEing in your values. With the RESTORE, POKEing in the normal values after the new values are in will not reactivate the RESTORE key.

The STOP key is easy to do away with. If you're using the VIC for a demonstration, it's nice to have it unstoppable.

LISTing is a bit more complicated to disable. One way to do it is to embed a control character that will cause the listed characters to be the same color as the screen. For example, white letters on a white screen result in a blank screen. The table gives the method. Put a REM line in the program and follow the instructions. Everything after the REM " will be invisible.

The super POKE is POKE 808,100. STOP, RESTORE, and LIST are now all dead. LIST isn't fully dead, but it might as well be since it prints only nonsense.

Before you use any of these commands, make sure that your program is running the way you want it to. Now add the lines that you want from the table. Usually, adding them as the first line of the program is best. Do *not* RUN the program yet, however. Double-check to make sure all the numbers are right, and SAVE the program. The table POKEs won't take effect until the program is RUN. The program is now safe on tape. When it is LOADed and RUN, the POKEs will take effect.

VIC Memory Addresses In RAM
(Some 64 addresses also given)

Function	Protection Method	Normal Values
STOP, RESTORE, and LIST	(VIC) POKE 808,100 (64) POKE 808,225	(VIC) POKE 808,112 (64) POKE 808,237
STOP	(VIC) POKE 808,127 (64) POKE 788,52	(VIC) POKE 808,112 (64) POKE 788,49
RESTORE	POKE 792,90 POKE 793,203	——
SAVE	POKE 818,73 POKE 819,245	POKE 818,113 POKE 819,246
LIST (VIC or 64)	1. REM " " 2. press CONTROL and 9 3. move cursor to second quote mark 4. press SHIFT and INST DEL 5. press SHIFT and M 6. press SHIFT and INST DEL 7. press CONTROL and screen color 8. press RETURN	

USR Sort

Walter D. Thompson, Jr

One way to create your own operands (commands) is to write subroutines that can be called from a BASIC program. You can, for example, use the Atari's USR function to create a routine with the speed of machine language. The time-saving routine can be especially important in sorting programs.

How many times have you created a simple file to which you have made several additions? Later, you discover that you want to list the records in the file in some order other than the sequentially entered order. Or perhaps you would even like to rewrite the file in a new order.

A sort utility in DOS could very well handle all possible demands. However, a simple array sort can handle many smaller applications by reading the fields or records into an array in memory and sorting the array. The only limitation is available memory.

The Sort Routine

A sorting routine is a method by which related elements are compared and ranked. In the Atari, this ranking is accomplished by comparing the ATASCII values. Thus, if we had an array with 26 elements, each one byte long, we could load the array with each letter of the alphabet, and after sorting the array, we would have the letters in order from A to Z.

The sort routine that we want would need to compare a string of letters to another string, perhaps of unequal length. The sort order is left justified so that an element beginning with a D would come before an element starting with an N. However, we also want DUSTY to come after DUST and before DUTY.

Our sort routine must compare each element byte to the corresponding byte of the elements before and after.

Some languages have built-in operands which accomplish this in some manner. For example, IBM BASIC operands AIDX and DIDX return a second array which has sorted pointers to the first array.

Atari Restrictions And Solutions

When Atari BASIC was written, several operands were omitted simply because there was not sufficient internal memory for all of the special operands available. However, BASIC does allow the user to build assembler subroutines, which can be called from the BASIC program. The utility of the Atari is greatly enhanced by the user's ability to write many special applications. These applications can then use the far greater speed of machine language.

Speed is particularly important in sort functions. Anyone who has written a BASIC program to do a bubble sort on a string array, and has then waited for many minutes for strings of moderate length to process, can attest to the importance of speed in a sort. A string of file-size proportions could very well take hours. This is because a sort, of any type, requires many repetitive compares to accomplish its final result. These are time-consuming operations in BASIC, but they are extremely fast with machine language.

Locating The Sort

So it seems that we must simply load a machine language sort module and call it from a BASIC program. But first we must consider where to locate the machine language module. It could be in the same place for any program which would load and access it, yet it must not interfere with the operating system or with our BASIC program which has called it. We would not accomplish very much if the sort module were located in the very middle of the array to be sorted.

The first sort utility I wrote worked extremely well and fit entirely into page six. So what more could I want from a sort that handled a large (32,000) element array and executed in mere seconds?

The answer is *safety*. Upon closer examination, I realized that a simple error in programming could result in the programmer sorting all of his or her program and memory above the array. This could include the display list, screen memory, and even BASIC itself. The routine needed a method of validating the total length of the sort array. We could have in memory an array of 100 elements of 127 bytes each, but through error, pass a value of 1000 elements to the sort routine. And the sort routine would, of course, attempt to sort 124K of memory, being forced to stop only when it attempted to sort ROM.

After developing and adding an error-check to make the program safe, I discovered that my original memory location was no longer able to hold the improved version. After some more juggling, I decided the solution was to make the object code relocatable. This means that the program permitted no jumps to a specified address; it allowed only relative branches.

Although the sort routine is relocatable, it does make use of a number of page zero and page six memory locations when executed. During the course of a sort, the program will alter the contents of locations $CB-$D1 and $6F8-$6FF.

Building The Array

We want to sort a group of unequal fields in a string array of as yet unspecified size. We must work with elements of a specified length, padding with blanks to the right for alphanumeric fields and zeros to the left for numeric fields. For example, a name field to be sorted may have a maximum length of 25 characters. There are 100 names or elements to be sorted. And so:

```
LET MLENGTH=25
LET ELEMENTS=100
LET X=MLENGTH*ELEMENTS
DIM SARRAY$(X)
```

In order to load the array so that the sort routine has equivalent elements to compare, we assign each field the maximum length using the following technique:

```
DIM FIELD$(MLENGTH)
  FOR I=1 TO MLENGTH:LET FIELD$(I)=" ":
    NEXT I
  LET L=LEN(NAME$)
  LET FIELD$(1,L)=NAME$
  LET P=ELEMENT*MLENGTH
  LET SARRAY$(P,P+MLENGTH-1)=FIELD$
```

If numeric fields are to be loaded into a string array for sorting, the following technique will build equal length elements with leading zeros:

```
DIM FIELD$(MLENGTH),TEST$(MLENGTH)
  FOR I=1 TO MLENGTH:LET FIELD$(I)="0":
    NEXT I
  LET TEST$=STR$(ZIPCODE)
  LET L=LEN(TEST$)
  LET FIELD$(MLENGTH+1-L,MLENGTH)=
    TEST$
```

We have now built an array for sorting, retaining the number of elements and the length of each element.

The USR Call

Upon calling the sort routine, we specify the number of elements to be sorted, the length of each element, the sort order, and the relative variable number of our array.

We call the sort routine by executing the following USR call:

```
LET EFLAG=USR(STARTADDR,ELEMENTS,
  MLENGTH,ORDER,RELVARNUM)
```

As long as the program does not change graphics modes, at least prior to all sorting, a very handy place to locate the sort routine object code is just below RAMTOP. The STARTADDR= PEEK(741)+256*PEEK(742) -339. Or in BASIC A+, STARTADDR=DPEEK(741) -339. ELEMENTS should be the actual number of elements loaded into the array. The additional routine which validates the length of the array is based on the LEN value of the variable, not its DIMed length. This helps to prevent garbage from being mixed in accidentally with good data. In our case, MLENGTH is 25, although the range of values for length is from 1 to 255 bytes. ORDER determines whether we will sort the array elements in ascending or descending order; 1 represents ascending order, and 255 indicates descending order. RELVARNUM, or the relative variable number, is the relative appearance of our string array in the BASIC program. Looking back to our first example, we see that MLENGTH was the first variable in our example, ELEMENTS was second, X third, and SARRAY$ fourth. The first variable has a relative number of zero, so RELVARNUM=4 -1 or RELVARNUM=3.

When a USR routine is called, a value may be returned to the BASIC program. This value is used to return error conditions. Those error conditions are represented by values of EFLAG, which are described as follows:

A Value of	Indicates
0	An error-free sort
1	ELEMENTS*MLENGTH > ARRAY LENGTH
2	Sort ORDER <> 1 or 255
3	MLENGTH=0 or >255
4	ELEMENTS=0
5	Variable not a DIMed string array
or	relative variable number exceeds 31
6	Invalid number of parameters passed

The Sort Utility

Our sort utility works by comparisons made in a loop, from the most significant byte to the least significant byte. Comparisons are made until two compared bytes are found out of order or until the element is fully tested. The order of comparison reduces the number of passes through the loop to a minimum.

To further increase efficiency, only one pass is made through the array. When one element is exchanged with a lower one, it is tested against the next lower element, until it is no longer exchanged or until it is placed at the bottom of the array. After placing that element, the routine returns to the element's original position and moves to the next element. In this way, each element is ranked among all previously sorted elements.

Of course this is not a DOS utility sort, and it is limited by available memory. However, the USR function allows us to use machine language speeds in our routine. And with a little programming and thought, most files can be sorted on any field by placing the sort field first in the element string, followed by the entire record.

For convenience, the machine language sort routine is provided as a BASIC loader program. For anyone who would prefer a copy on diskette or tape, please send a self-addressed, stamped mailer, a blank tape or diskette, and $3 to:

Walter D. Thompson, Jr.
P.O. Box 6602
Greensboro, NC 27405

All diskettes will be written in DOS 2.0 format. In addition, all copies of the BASIC program listed here will be accompanied by an Assembler version and the original version of the object code which fits in page six.

USR Sort

```
290 PRINT "{CLEAR}":PRINT "ARRAY SOR
    T PROGRAM":PRINT
300 TRAP 300:PRINT "INPUT # OF ELEME
    NTS TO SORT";:INPUT ELEMENTS
310 TRAP 310:PRINT "INPUT MAXIMUM EL
    EMENT LENGTH";:INPUT LENGTH
320 TRAP 320:PRINT "INPUT SORT ORDER
     - ENTER 1 OR 255";:INPUT ORDER
330 PRINT :PRINT "INITIALIZING..."
340 DIM ARRAY$(LENGTH*ELEMENTS),ELEM
    ENT$(LENGTH),BLANK$(LENGTH),WORK
    $(LENGTH):GOSUB 700
350 PRINT "{CLEAR}":REM ****  CLEAR
    SCREEN
360 PRINT "ARRAY SORT ENTRY":PRINT
370 VNUM=3:REM ****  ARRAY$ IS THE 4
    TH VARIABLE TO APPEAR IN PROGRAM
380 FOR X=1 TO LENGTH:BLANK$(X)=" ":
    NEXT X
390 FOR I=0 TO ELEMENTS-1
400 WORK$=BLANK$
410 TRAP 410:PRINT "INPUT UP TO ";LE
    NGTH;" CHARACTERS":INPUT ELEMENT
    $
420 L=LEN(ELEMENT$):IF L=0 THEN 450
430 S=LENGTH*I+1:E=LENGTH*(I+1):WORK
    $(1,L)=ELEMENT$:ARRAY$(S,E)=WORK
    $:NEXT I
440 I=ELEMENTS:REM ***  ESTABLISH I
    IF ALL ELEMENTS FILLED
450 EFLAG=USR(START,I,LENGTH,ORDER,V
    NUM)
460 ON EFLAG GOTO 470,480,490,500,51
    0,520:GOTO 530
470 PRINT "**ERROR S/";EFLAG;" - ARR
    AY LENGTH EXCEEDED":STOP
480 PRINT "**ERROR S/";EFLAG;" - INV
    ALID SORT ORDER":STOP
490 PRINT "**ERROR S/";EFLAG;" - ELE
    MENT LENGTH INVALID":STOP
500 PRINT "**ERROR S/";EFLAG;" - ELE
    MENT COUNT = 0":STOP
510 PRINT "**ERROR S/";EFLAG;" - INV
    ALID VARIABLE TYPE":STOP
520 PRINT "**ERROR S/";EFLAG;" - INV
    ALID # PARAMETERS":STOP
530 FOR J=0 TO I-1
540 S=LENGTH*J+1:E=LENGTH*(J+1):PRIN
    T ARRAY$(S,E):NEXT J
550 STOP
700 REM **  LOCATE OBJECT BELOW OS M
    EMTOP
710 START=PEEK(741)+256*PEEK(742)-33
    9
720 TRAP 725:FOR I=0 TO 338:READ A:P
    OKE (START+I),A:NEXT I
730 RETURN
740 PRINT "ERROR DURING OBJECT LOAD"
    :STOP
750 DATA 169, 0, 133, 212, 133, 213,
     104, 168
760 DATA 240, 109, 201, 4, 240, 7, 1
    04, 104
770 DATA 136, 208, 251, 240, 98, 104
    , 141, 251
780 DATA 6, 133, 207, 104, 141, 250,
     6, 133
790 DATA 206, 104, 240, 4, 169, 1, 1
    33, 213
800 DATA 104, 133, 205, 133, 203, 10
    4, 240, 4
810 DATA 169, 2, 133, 213, 104, 133,
     204, 104
820 DATA 240, 4, 169, 4, 133, 213, 1
    04, 10
830 DATA 176, 55, 10, 176, 52, 10, 1
    76, 49
840 DATA 168, 165, 213, 240, 8, 201,
     2, 48
850 DATA 44, 240, 44, 16, 36, 177, 1
    34, 201
860 DATA 129, 208, 30, 165, 140, 200
    , 200, 24
870 DATA 113, 134, 141, 252, 6, 141,
     254, 6
880 DATA 165, 141, 105, 0, 200, 113,
     134, 141
890 DATA 253, 6, 141, 255, 6, 208, 1
    7, 230
900 DATA 212, 230, 212, 230, 212, 23
    0, 212, 230
910 DATA 212, 230, 212, 169, 0, 133,
     213, 96
920 DATA 200, 200, 169, 0, 24, 141,
    249, 6
930 DATA 141, 248, 6, 162, 8, 102, 2
    03, 144
```

```
940 DATA 26, 24, 173, 248, 6, 101, 2
    06, 141
950 DATA 248, 6, 173, 249, 6, 101, 2
    07, 141
960 DATA 249, 6, 176, 213, 165, 205,
     56, 229
970 DATA 203, 144, 206, 6, 206, 38,
     207, 202
980 DATA 208, 219, 177, 134, 205, 24
    9, 6, 48
990 DATA 192, 208, 8, 136, 177, 134,
     205, 248
1000 DATA 6, 48, 182, 165, 205, 240,
      174, 165
1010 DATA 204, 201, 1, 240, 4, 201,
     255, 208
1020 DATA 166, 173, 250, 6, 13, 251,
      6, 240
1030 DATA 154, 173, 254, 6, 133, 208
     , 24, 101
1040 DATA 205, 141, 254, 6, 173, 255
     , 6, 133
1050 DATA 209, 105, 0, 141, 255, 6,
     206, 250
1060 DATA 6, 208, 6, 206, 251, 6, 16
     , 1
1070 DATA 96, 160, 0, 24, 165, 208,
     101, 205
1080 DATA 133, 206, 165, 209, 105, 0
     , 133, 207
1090 DATA 177, 206, 209, 208, 240, 8
     , 8, 104
1100 DATA 69, 204, 48, 9, 16, 195, 2
     00, 196
1110 DATA 205, 240, 190, 208, 235, 1
     60, 0, 177
1120 DATA 206, 170, 177, 208, 145, 2
     06, 138, 145
1130 DATA 208, 200, 196, 205, 208, 2
     41, 165, 208
1140 DATA 56, 229, 205, 176, 2, 198,
      209, 133
1150 DATA 208, 165, 209, 205, 253, 6
     , 48, 153
1160 DATA 208, 183, 165, 208, 205, 2
     52, 6, 16
1170 DATA 176, 48, 142
```

©

Working With SID

Jerry M Jaco

In this unique approach to the Commodore 64's SID chip, the author discusses the SID chip's anatomy and capabilities in the context of its essential similarity to the design of music synthesizers.

If you've decided you want to make music on your Commodore 64, and you've read all the literature on the subject and still don't know where to begin, perhaps a look at how an analog synthesizer is used in an electronic music studio will clarify many aspects of the 64's amazing sound capabilities. Once we have covered the physical aspects of a synthesizer, we can begin to understand some of the techniques used to create sounds artificially.

Electronic music studios usually include at least one analog synthesizer. Most synthesizers have a modular design which allows the synthesizer to be built and expanded according to the dictates of budget, space, and ability. Each module on the synthesizer has a different function, and the builder-user is free to duplicate or omit these functions in any way he sees fit.

Each module on the synthesizer is independent of all the others. The only way to connect them is either by a panel of fancy selector switches or via the more common *patch cords*. Patch cords are simply pieces of electrical cable of varying lengths which have standard plugs attached on each end. Plugging one end of a patch cord into the output socket of one module and the other end of the patch cord into the input socket of another module creates an electrical pathway called a *patch*.

If a patch leads from a Source module, such as an oscillator, to an Output module, such as a mixer, the resulting sound will be audible to the outside world. (See Figure 1.) The term *signal* is used to describe the electrical current being passed from one module to another. A *source signal* is one that will eventually be heard as a real sound. A *control signal* is a varying voltage used to electronically control another module. It does not contain sound information per se.

A "Patch" For The SID Chip

In Figure 1, there is only one source signal being processed by the mixer. A mixer can handle up to three source signals on our hypothetical system, which it combines into one composite signal that gets sent on to the speakers, and to your ears. (See Figure 2.) On the 64, Program 1 accomplishes exactly what the analog synthesizer does in Figure 2.

Program 1: Three Voices – Or A Chord

```
10 FORI=0TO24:POKE54272+I,0:NEXT
20 POKE54272,37:POKE54273,17:REM OSC1

30 POKE54279,229:POKE54280,22:REM OSC2

40 POKE54286,214:POKE54287,28:REM OSC3

50 POKE54276,17:POKE54283,17:POKE54290,17
   : REM TRIANGLE WAVE FOR ALL OSC'S

60 POKE54278,245:POKE54285,245:POKE54292,
   245:REM SUS/REL VALUES FOR ALL OSC'S

70 POKE54296,15:REM MASTER VOLUME ON

75 FORT=1TO500:NEXT:REM CHORD DURATION

80 POKE54276,16:POKE54283,16:POKE54290,16

90 FORT=1TO250:NEXT:REM RELEASE DURATION

95 POKE54296,0:REM TURN OFF VOLUME:rem 29
96 END
```

This is a very basic "patch" for the Sound Interface Device (SID) chip on the 64. Lines 20, 30, and 40 set the frequencies of the three oscillators. Line 20 POKEs the values for middle C into Voice 1. Line 30 POKEs the values for F into Voice 2, and line 40 POKEs the values for A into Voice 3. This gives us a "chord," which is simply three notes (voices) sounding simultaneously. Line 50 selects a triangle wave output for all three voices. Line 70 is the mixer volume control. When the value 15 is POKEd into this location, the master volume control is turned all the way up. When 0 is POKEd, the volume control is turned off, as in line 55. The other lines will become clearer as we go along.

Figure 1: Processing A Single Source Signal

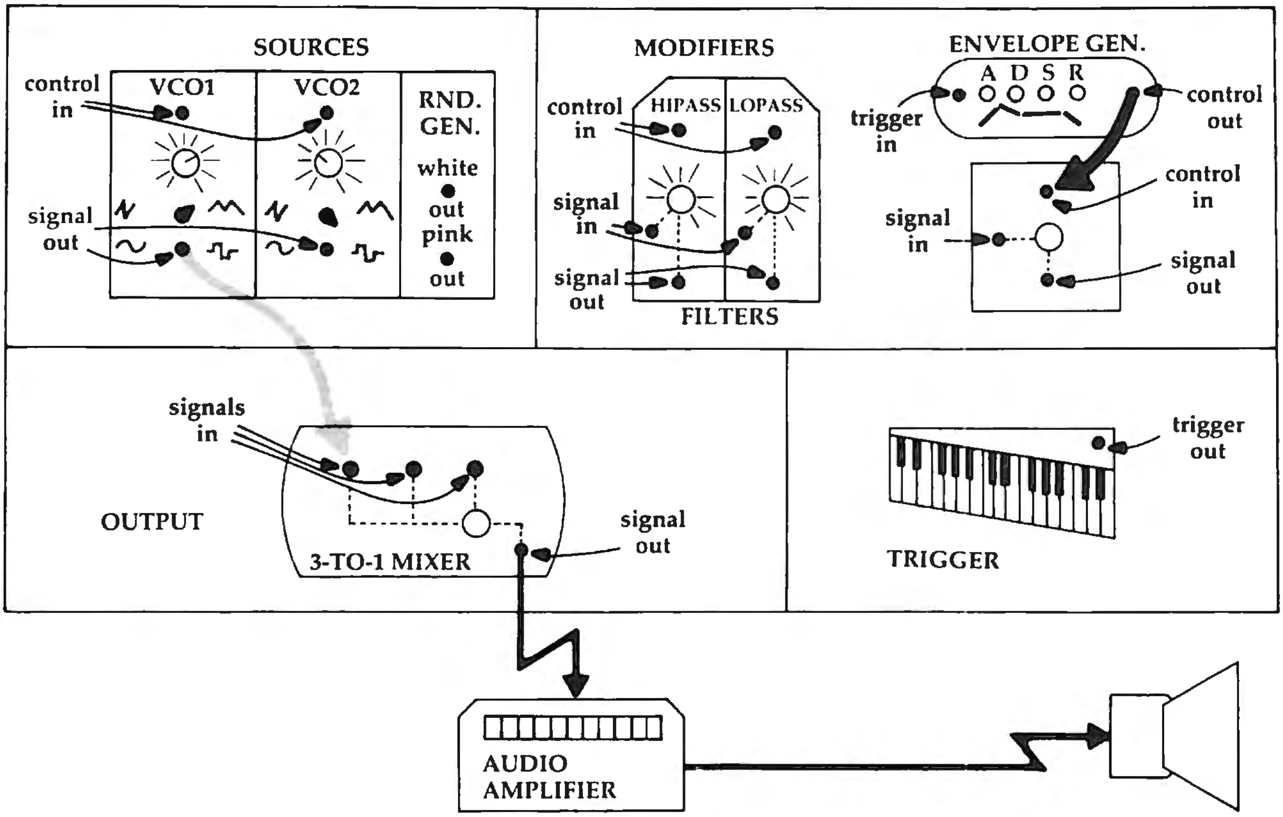
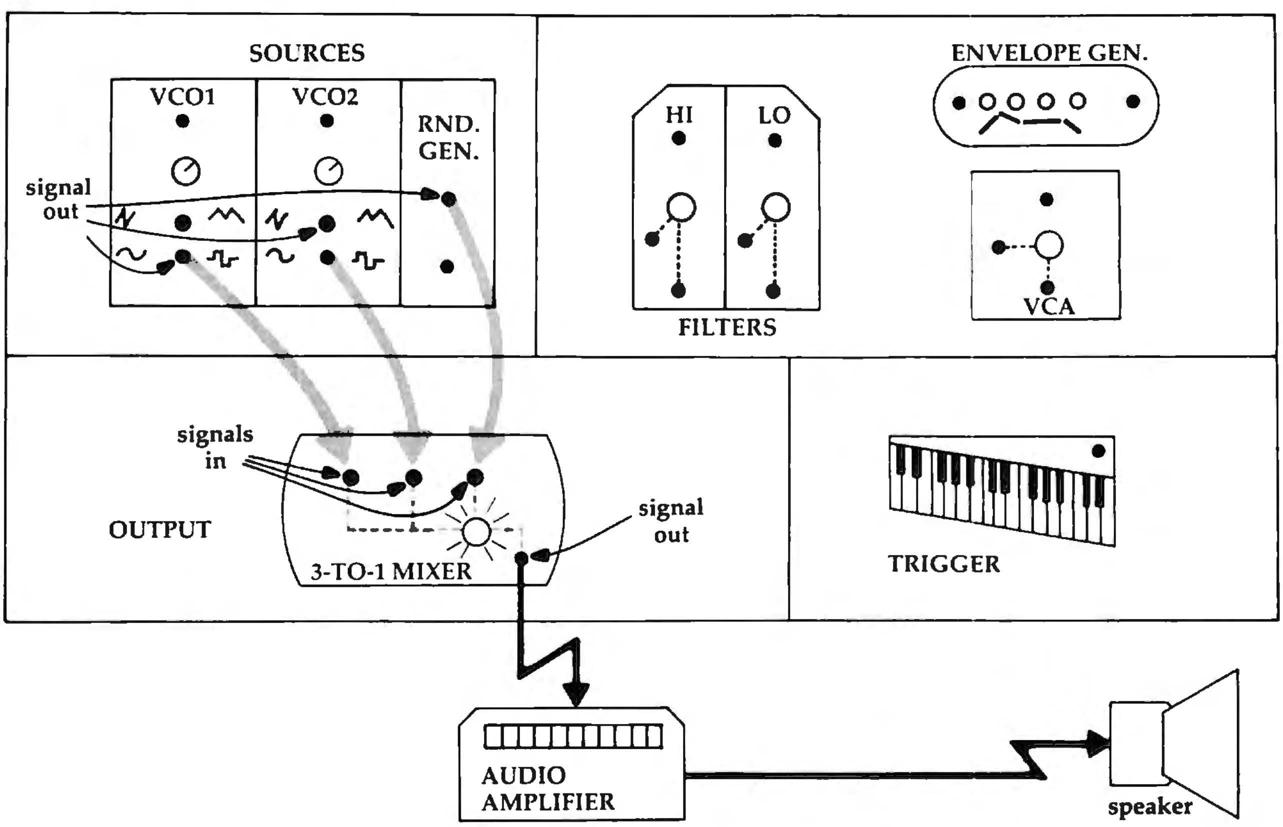


Figure 2: Processing Three Source Signals



On an analog synthesizer, *pots* (potentiometers) are controls that do things such as raise and lower the volume of a sound signal or change the frequency (pitch) of an oscillator. Pots are also the main components of game paddles and TV volume controls. To make new sounds on an analog synthesizer, the user will twist pots on each module and listen for the resulting effect. When he finds one he likes, he either records the sound on tape or writes down the patch on a patch chart, marking the pathways made by the patch cords and the positions of the pots for future reference. Analog synthesizers are very useful in this way because drastic changes in a sound can be quickly made by simply twisting a knob or plugging a patch cord into something else.

Turning Knobs With POKEs

A digitally-controlled synthesizer, such as our SID chip, uses numbers POKEd into control registers to accomplish the same things that knob-twisting and patch-cord-plugging do on an analog synthesizer. For example, if you POKE a 16-bit value into the first two registers of the SID chip (54272 and 54273), you've set the frequency value for Oscillator 1. POKE a four-bit number into the high nybble of the sixth register on the chip, and you've set the Attack value of the envelope for Oscillator 1. POKEing different values into other registers will activate them in the same way that turning the pots or setting switches will activate the analog synthesizer modules.

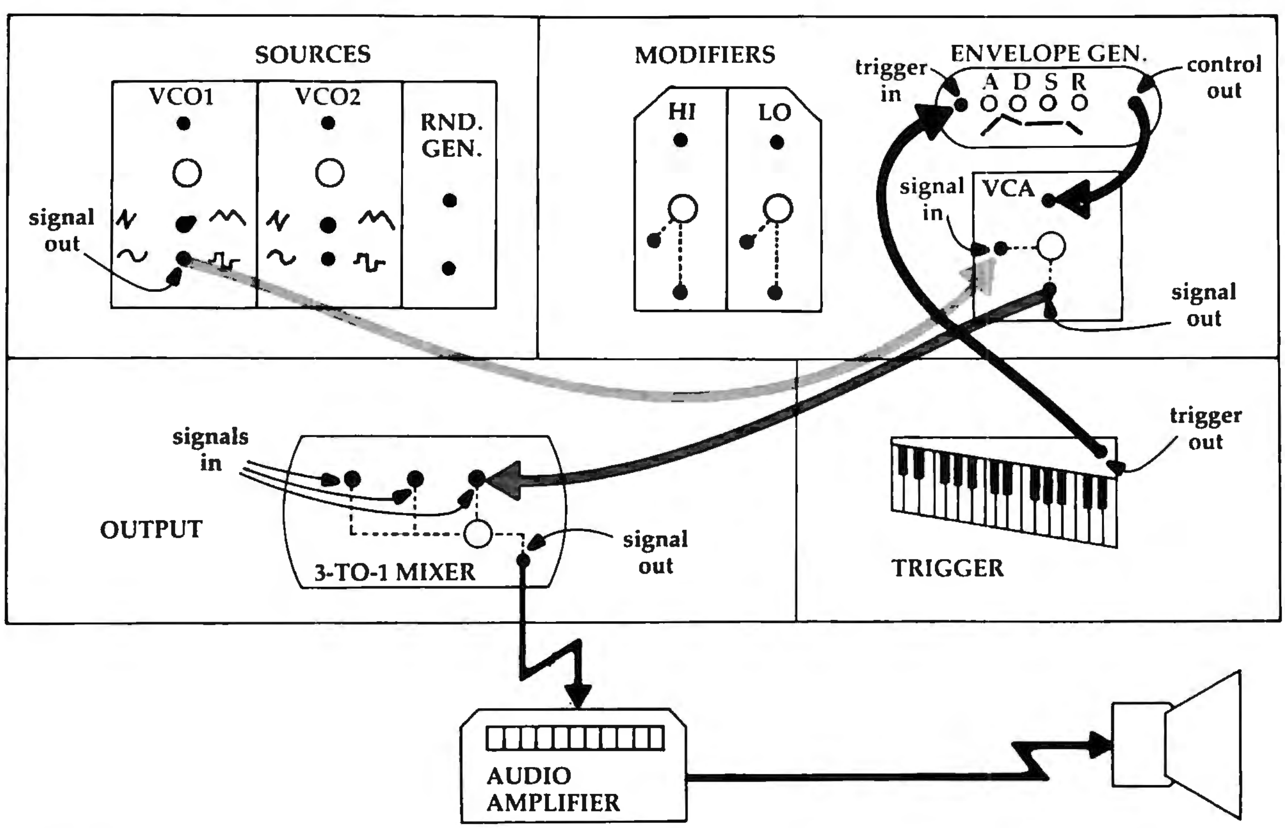
Envelope Generation

Look at Figure 1 again. It shows a direct path from an oscillator (VCO1) to the mixer. If we were to break that path, sending the output of VCO1 to the input of the amplifier module (VCA), we would then need to send the output of VCA to the mixer so that the sound from VC01 could still be heard. The patch shown in Figure 3 would be the result. Now we can make VCO1's signal even louder by adjusting the pot on VCA or on the mixer. The real reason for taking this route is that the envelope generator can be brought into play, since it directly controls the VCA.

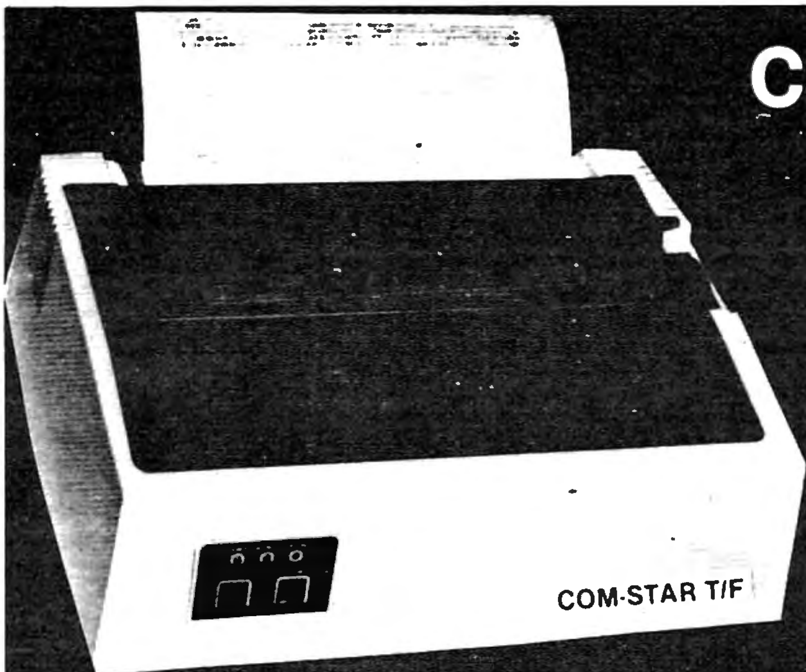
There are four pots on the envelope generator module. The first controls the Attack time; the second, the Decay time. The third sets the Sustain level, and the fourth controls the Release time. On the SID, two registers in high-low nybble format control these functions. The most important function is perhaps the Sustain level. It is not a timing value, but is rather the level at which the amplifier's volume control is set while the note is being sounded. If the Sustain level is zero, no sound will be heard after the Attack and Decay phases have ended.

The envelope generator puts out an electrical signal which tells the amplifier when to turn up

Figure 3: Using The VCA



80 COLUMN PRINTER SALE—$149.00*



COM-STAR T/F

Tractor
Friction
Printer

only **$219**

(Parallel or Serial)

- Lowest price quality tractor friction printer in the U.S.A. • Fast 80 characters per second
- 40, 46, 66, 80, 96, or 132 characters per line spacing • Prints labels, letters, graphs, and tables
- List your programs • Print out data from modem services

*STX-80 COLUMN PRINTER—$149.00

Prints full 80 columns. Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal roll paper!

DELUXE COMSTAR T/F PRINTER—$219.00

The Comstar T/F is an excellent addition to any micro-computer system. (Interfaces are available for Apple, VIC-20, Commodore-64, Pet, Atari 400 and 800, and Hewlett Packard). At only $219 the Comstar gives you print quality and features found only on printers costing twice as much. Compare these features.

- **BI-DIRECTIONAL PRINTING** with a **LOGIC SEEKING CARRIAGE CONTROL** for higher through-put in actual text printing. 80 characters per second.
- **PRINTING VERSATILITY:** standard 96 ASCII character set plus block graphics and international scripts. An EPROM character generator includes up to 224 characters.
- **INTERFACE FLEXIBILITY:** Centronics is standard. Options include EIA RS232C, 20mA Current Loop.
- **LONG LIFE PRINT HEAD:** 100 million character life expectancy.
- **THREE SELECTABLE LINE SPACINGS:** 6, 8 or 12 lines per inch.
- **THREE SELECTABLE CHARACTER PITCHES:** • 10, 12 or 16.5 characters per inch. 132 columns maximum. Double-width font also is standard for each character pitch.
- **PROGRAMMABLE LINE FEED:** programmable length from 1/144 to 255/144 inches.
- **VERTICAL FORMAT CONTROL:** programmable form length up to 127 lines, useful for short or over-sized preprinted forms.
- **FRICTION AND TRACTOR FEED:** will accept single sheet paper.
- **224 TOTAL CHARACTERS**
- **USES STANDARD SIZE PAPER**

If you want more try —

Premium Quality COMSTAR T/F SUPER-10X PRINTER—$299.00
More Features Than RX-80

For $299 you get all of the features of the Comstar T/F plus 10" carriage 120 cps, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix. High resolution bit image (120 x 144 dot matrix), underlining, backspacing, left and right margin settings, true lower descenders, with super and subscripts, and prints standard, Italic, Block Graphics, special characters, plus 2K of user definable characters. For the ultimate in price performance the Comstar T/F Super 10" leads the pack!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge via United Parcel Service, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

15 DAY FREE TRIAL

OTHER OPTIONS

Extra Ribbons	$ 5.95
Roll Paper Holder	32.95
Roll Paper	4.95
5000 Labels	19.95
1100 Sheets Fan Fold Paper	13.95

Add $17.50 shipping, handling and insurance. Illinois residents please add 6% tax. Add $40.00 for CANADA, PUERTO RICO, HAWAII, ALASKA orders. WE DO NOT EXPORT TO OTHER COUNTRIES. Enclose cashiers check, money order or personal check. Allow 14 days for delivery, 2 to 7 days for phone orders. 1 day express mail available!! Canada orders must be in U.S. dollars.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

GRAFSTAR (Apple interface card · "Like Grappler Plus") SALE $69.00.

the volume and how long it should take, as well as how high to set the volume, and when and how long to turn it all the way off again. This is why the amplifier module in the diagrams is called "VCA." This stands for Voltage Controlled Amplifier and means that the amplifier can be controlled by an incoming variable voltage, such as the one supplied by the envelope generator.

ADSR Values

On the SID chip, each voice has its own envelope generator. Within the group of seven registers (0-6) that control the three oscillators, register 5 contains the Attack and Decay values in high-low nybble format, and register 6 contains the Sustain/Release values. All values are four-bit numbers (nybbles). The Attack value determines how long the amplifier should take to reach peak amplitude (maximum volume).

The Decay value determines how long the amplifier should take to go from peak amplitude to the level specified by the Sustain value. The Release value is the time the amplifier will use to return to the lowest amplitude level ("off") from the Sustain level.

Remember, though, that on the ânalog synthesizer as well as on the SID chip, the envelope will not go into effect until it is "triggered." The lowest order bit (bit 0), the Gate bit, triggers each envelope on the SID chip. On the analog synthesizer, triggering of the envelope is accomplished through the use of an attached keyboard module. When a key is pushed down (and as long as it is held down), the Attack, Decay, and Sustain values will go into effect in order. When the key is released, the Release phase is triggered, and the VCA will close down the volume of the signal it is operating on over the length of time specified by the Release value.

Program 2 demonstrates the effect of the various ADSR values:

Program 2: Effects Of The ADSR Values

```
100 FORI=0TO24:POKE54272+I,0:NEXT
110 POKE54272,37:POKE54273,17:REM OSC1

120 POKE54276,129:REM NOISE WAVE OSC 1

130 POKE54277,240:REM SLOW ATTACK/FASTEST
     DECAY RATE
140 POKE54278,240:REM HIGHEST SUSTAIN LEV
    EL/FASTEST RELEASE RATE
150 POKE54296,15:REM FULL VOL. AT MIXER

160 FORT=1TO4500:NEXT:REM DURATION FOR AT
    TACK,DECAY, AND SUSTAIN
170 POKE54276,128:REM BEGIN RELEASE CYCLE

180 FORT=1TO4500:NEXT:REM REL. DURATION

190 POKE54296,0:REM TURN OFF VOLUME

191 END
```

In line 130, the Attack value is all the way on, and the Decay value is all the way off. In line 140, the Sustain value is all the way on and the Release value is off. Each value is a four-bit number, 0 to 15. With the Attack and Sustain setting, the actual POKE value is shifted to the high nybble; thus, 240 is actually the Attack value equal to 15 (for slowest Attack) multiplied by 16. The sound generated is a random noise that gradually gets louder and then stops suddenly. It stops suddenly because we have set the Release value to 0, allowing no time for a gradual decrease in volume.

Change the value 240 in line 140 to 255 and RUN the program again. The sound should slowly fade away. The high nybble of 54278 (Sustain) is now 240 and the low nybble (Release) 15, making a total of 255, the value we just POKEd into 54278. Try lowering the Sustain value by two or three (2*16 or 3*16); that is, POKE 54278 with either 223 or 207 and see what happens. The sound should build up as before but should then fall off markedly. Change the Decay value from 0 in line 130 to about 8 (POKE 54277,248) and hear how the drop-off is now smoothed out. Similarly, shorten the Attack time to vary the start of the sound the same way the Sustain value was altered. The results should be vastly different from those we started with, and we've been working with only two registers!

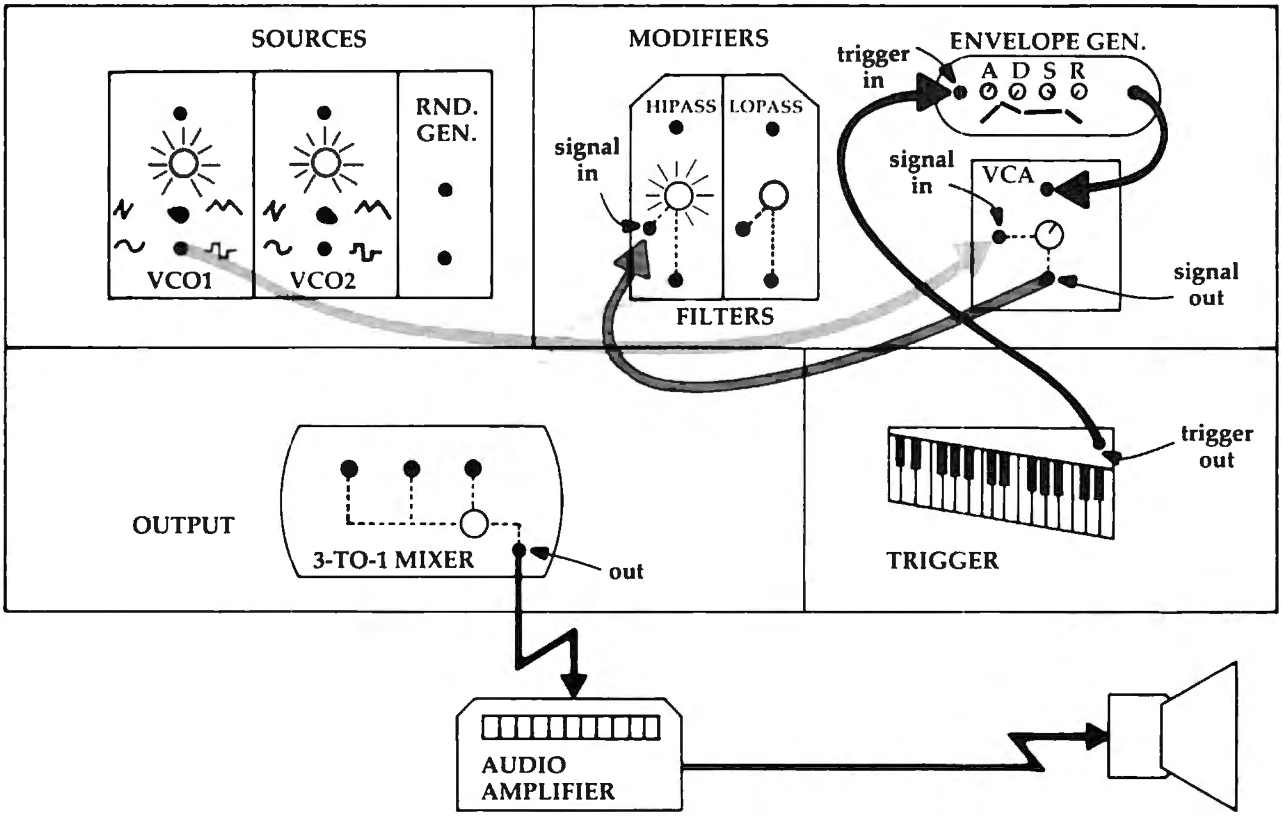
Look now at line 170. Notice that we subtracted one from the value we originally POKEd into 54276 in line 120. This zeros the Gate bit in 54276, and it is the same as taking your finger off the keyboard on the analog synthesizer: the Release cycle gets triggered. Of course, it works only if the VCA Sustain level has been previously raised high enough to hear the tone. The delay loop in line 180 is also necessary to allow the Release cycle to reach its lowest level.

Using Filters To Color Sound

Let's add a filter to the path in Figure 3. The path from the VCA to the mixer is broken so that filtering the modulated signal will be more easily heard. In our diagram, we have a choice of a high-pass or low-pass filter. On the SID chip, we can also utilize a Band-Pass filter.

The pot on each filter is used to adjust the cutoff frequency, which is the frequency above which a high-pass filter allows frequencies in the sound spectrum to be heard and below which the filter suppresses them. The low-pass filter is the opposite of the high-pass filter in that it suppresses the frequencies above the cutoff value and allows those below it to sound. A Band-Pass filter allows frequencies to be heard within a narrow band surrounding the cutoff frequency (called a center frequency in this case), while suppressing all the rest. Use of filters constitutes a technique called

Figure 4: **Using A High-Pass Filter**



subtractive synthesis, which selectively eliminates available frequencies of the sound spectrum, producing widely varying sound colors.

Figure 4 indicates that we've decided to filter VCO1 through a high-pass filter. VCO1 is set to produce a sawtooth wave. The path of the patch runs out of VCO1 into the VCA, and from the VCA into the HIPASS filter. From there, the signal heads to the mixer and out to the speaker. Program 3 is a routine that does the same thing.

Program 3: Filtered Sound

```
200 FORI=0TO24:POKE54272+I,0:NEXT
210 POKE54272,37:POKE54273,17:REM OSC1

220 POKE54276,33:REM SAWTOOTH WAVE OSC1

230 POKE54277,120:REM MED. ATTACK/MED. DE
    CAY
240 POKE54278,245:REM HIGHEST SUSTAIN/MED
    . RELEASE
245 POKE54293,40:POKE54294,5:REM CUTOFF F
    REQUENCY FOR HIGH-PASS FILTER
250 POKE54295,129:REM MED RES'NCE AND OSC
    1 TO BE FILTERED
255 POKE54296,79:REM FULL VOL. AND CHOOSE
     HIGH-PASS FILTER
260 FORJ=1TO250:POKE54294,J:NEXT:REM SWEE
    P CUTOFF FREQ. UPWARDS

270 POKE54276,32:REM BEGIN RELEASE CYCLE

280 FORT=1TO500:NEXT:REM REL. DURATION

290 POKE54296,0:REM TURN OFF VOLUME

295 END
```

To hear the effect of the filter, in line 260 we will sweep the value of the cutoff frequency from low to high. This will allow less and less of the available sound spectrum to be passed by the filter. Listen carefully to the richness of the tone as it is diminished. Switch the wave form to Noise in line 220 by POKEing 129, instead of 33, into 54276 to hear a different version of the effect. Many effects are possible using filters.

Frequency Modulation

Figure 5 introduces another technique called Frequency Modulation. Notice now that the signal from VCO1 is entering the control input of VCO2, and that the signal from VCO2 is going through the VCA and on to the mixer. The frequency of VCO2 is now being controlled automatically by the output voltage of VCO1 instead of manually by the pot. This is another example of voltage control. The envelope generator controlled the VCA before and an oscillator now controls a VCO (Voltage-Controlled Oscillator).

Frequency Modulation (FM), along with filtering and envelope control, is one of the most significant techniques of sound synthesis. Using one signal source to alter the sound quality of

EPYX
TOP 10 GAME SALE
* BUY TWO — GET ONE FREE *

SPECIAL FREE GAME
Buy Two Epyx Games From Protecto And Get A Free Game

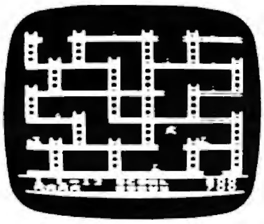
The Thinking Man's Paradise

(Disk / Cassette)

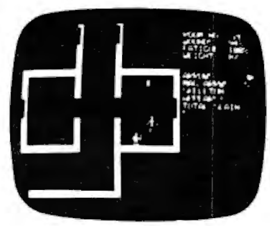
- **Award Winning Games**
- **Fantastic Graphics**
- **Skill (not luck) Needed**
- **Already in top 10 charts**

COMMODORE - 64 / VIC-20

Buy Any Two Epyx Games From Protecto And Send The Proof Of Purchase Seals To Epyx And Epyx Will Send You A Free Game.

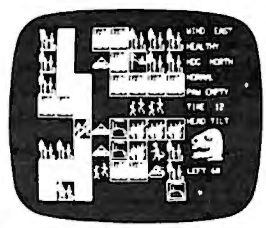


JUMPMAN
If you like Donkey Kong, you'll love Jumpman. Over 30 different screens with 8 speeds and 5 skill levels make this the fastest action game in the country. You must leap girders, climb ropes, and scale ladders to reach and diffuse bombs while avoiding robots, birds, bullets, explosives, crumbling walls, vanishing escape routes, and many other obstacles. (Truly a fantastic game!)
List $39.95 **Sale $27.95**

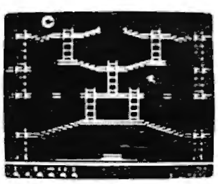


TEMPLE OF APSHAI (computer game of the year)
This is the standard by which other adventure games are judged. Full color graphics portray the temple and all its contents - magic, monsters, doomed cities and damsels in distress. Do battle in real time with over 20 monsters, expansion modules will keep your adventure alive in the future.
List $39.95 **Sale $29.95**

SWORD OF FARGOAL
Search for the wondrous sword in the depths of an ever changing dungeon. Make yourself invisible, teleport to a new location, drink a healing potion or use enchanted treasures, but watch out for traps and hideous creatures who will try to stop you. (Fantastic dungeon adventure)
List $29.95 **Sale $21.95**



CRUSH CRUMBLE & CHOMP (Computer game of the year nominee)
Choose one of six monsters or create your own, and use your monster to destroy one of four unsuspecting cities. The cities aren't totally defenseless, they call on police, national guard, and even a mad scientist, complete with helicopter, to save humanity from the relentless threat.
List $29.95 **Sale $21.95**

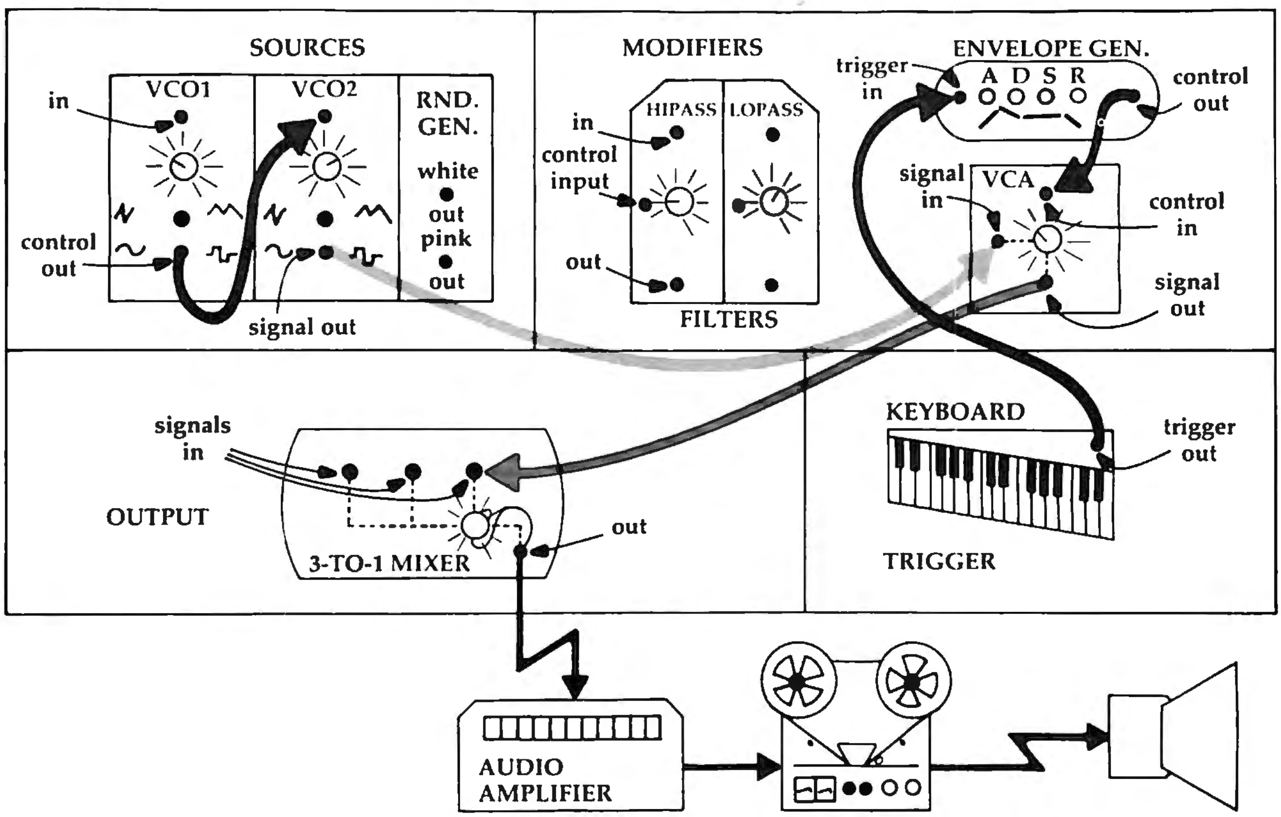


JUMPMAN JUNIOR
The devilish ALIENATORS are back! And they have overrun the Jupiter Command Substation. In this cartridge format sequel to the best-selling Jumpman, players must leap through 12 all new screens featuring electrocution traps, moving walls, hellstones, and dangers much too bizarre to be believed. How many screens can you master? Twelve different screens, 8 speeds.
List $39.95 **Sale $27.95**

We have all Epyx games

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Figure 5: Frequency Modulation



another provides incredibly powerful and varied tools for sound manipulation. Program 4 is one simple example of the FM technique.

Program 4: Siren

```
300 FORI=0TO24:POKE54272+I,0:NEXT
310 POKE54276,33:REM SAWTOOTH WAVE OSC1

320 POKE54286,3:REM CONTROL FREQ. OSC3

330 POKE54290,16:REM TRIANGLE WAVE OSC3

340 POKE54296,175:REM FULL VOL.& SELECT B
    AND-PASS & DISC. OSC3 FROM AUDIO

350 POKE54295,1:REM NO RES'NCE & CHOOSE O
    SC1 FOR FILTER
360 POKE54293,255:POKE54294,78:REM CUTOFF
     FREQUENCY
370 POKE54278,240:REM FULL SUSTAIN/FASTES
    T RELEASE RATE
375 FORT=1TO300
380 F=20000+PEEK(54299)*20:REM ADD OSC3 O
    UTPUT TO BASE FREQUENCY
390 HF=INT(F/256):LF=F-256*HF:REM SPLIT N
    EW FREQUENCY INTO HIGH/LOW BYTES

400 POKE54272,LF:POKE54273,HF:REM SET NEW
     OSC1 FREQUENCY
410 NEXT:POKE54276,32:POKE54296,0
420 END
```

The third oscillator on the SID chip is our control oscillator, as VCO1 is in Figure 5. We get access to a value corresponding to the wave shape of Oscillator 3 in register 27 (54299). If Oscillator 3 is set to a triangle wave, the values in register 27 will go up from 0 to 255 and then down from 255 to 0 in a symmetrical rhythm.

This is a nice shape for a siren sound, which is what Program 4 creates. Notice that the frequency of Oscillator 3 in line 520 is very low. This value allows the tracing of the waveform to be heard as a siren. The range of frequencies under approximately 32Hz is called the sub-audio range and refers to the fact that the actual waveform at these frequencies is discernible as individual pulses instead of as a continuous tone. When Oscillator 3's frequency is increased into the audio range (above about 29), the quality of the resulting tone becomes enjoyably less predictable.

Try POKEing 220 into 54286 at line 320 and running the routine. Note how the information in register 27 (54299) is utilized in line 380. It is increased by a factor of 20 and then added to the base frequency of 20000. Program 4 also uses a Band Pass filter, but for no particular reason other than simply to stick one in. Try a different value for the waveform in line 330. If you use 64 as your value, be sure to add a line to set Oscillator 3's pulse width.

The techniques of sound manipulation described above as used with an analog synthesizer have perhaps given you a better picture of the

workings of the SID chip. As you learn more about the internal registers which control other functions, you will discover others just as interesting as those discussed here.

Get a copy of the *Commodore 64 Programmer's Reference Guide* and read about Ring Modulation, Filtering, and other advanced techniques. Sound effects are the most directly useful sound patches to work with at the start. Program 5 is an example of one I used for a Hangman program: it's the sound of nails being driven into wood. Imagine the other sound effects you can create for new game ideas.

The *User's Guide* and *Programmer's Reference Guide* have suggested patches for you to try out. Put some FOR..NEXT loops in, as we did in line 260 of Program 3, to have the computer, in effect, "adjust the pots" for you, as it alters individual registers. Once you've found a patch you like, save the register values for future reference. As you become more acquainted with the ways that sounds can be altered, you will find yourself noticing the subtler shades of sound color. You'll also begin to really know how the sounds on a TV commercial, videogame, or science fiction movie are created.

Program 5: Driving Nails Into Wood

```
700 FORI=ØTO24:POKE54272+I,Ø:NEXT
710 CT=Ø
720 POKE54278,5:REM SUSTAIN/RELEASE
730 POKE54277,5:REM ATTACK/DECAY
740 POKE54276,129:REM NOISE WAVEFORM
750 POKE54295,241:REM RES'NCE & VOICE
760 POKE54293,54:POKE54294,28:REM CUTOFF
770 READA:REM INPUT HI BYTE FREQ. VALUE
780 READB:REM INPUT LO BYTE FREQ. VALUE
790 IFB=-1THEN9ØØ:REM BRANCH ON END-OF-DA
    TA
800 POKE54273,A:POKE54272,B:REM SET FREQ
810 FORT=1TO35:POKE54296,79:NEXT:REM TURN
     ON VOLUME & FILTER
820 POKE54276,128:REM RELEASE CYCLE
830 GOTO73Ø:REM GET NEW NOTE
840 DATA17,37,19,63,21,154,22,227,25,177,
    28,214,32,94,34,175,34,255
845 DATA-1,-1
900 CT=CT+1:IFCT+1<6THENRESTORE:FORT=1TO1
    ØØ*CT:NEXT:GOTO77Ø
910 POKE54296,Ø:REM TURN OFF VOLUME
```
©

Atari Safe RAM

E H Foerster

Are you tired of waiting while your BASIC program initializes? Would you like to save two-thirds of the memory required to store the DATA statements for your USR routines, character sets, or player/missile data? Would you like to save data generated by your program along with the program using just SAVE commands? This article will show you how to create protected and SAVEable RAM within the BASIC token program. Several USR routines are also included to allow you to add or delete safe RAM, move data in memory, and point a string to any place in memory. Also, the game "SKI!" illustrates these techniques in a practical, executable program. And each of these utilities does its job in a fraction of a second.

The Atari 400/800 computers provide page six as free RAM to be used by BASIC. However, to use this RAM, the information must be stored in DATA statements in the BASIC program and then transferred to RAM using READ and POKE loops. This often delays the start of a program while the initialization routine is processed. You are also limited to 256 bytes when using page six.

A technique commonly used to circumvent this problem is to put the desired file in a string at the beginning of the string array storage area (STARP). The pointer to STARP is then moved past the desired file. When the program is SAVEd, the file is SAVEd along with the program. This technique requires special programming not only to save the file, but also to move the pointer back after the program is LOADed into the computer. You must insure that the pointers are not moved a second time when a program is executed following an initial run.

But there is another location within the BASIC token program that can be used to save and store data. The procedure we'll discuss will provide all the RAM you need at this location and protect it from any additions, deletions, or changes in the program. This protection, of course, does not extend to direct POKEs to this area.

No special routine will be needed to protect this RAM area once it has been inserted in the program. The RAM area and the information stored in it will remain there even when the program is SAVEd. The information is available immediately after the program is loaded and can be moved rapidly from this location to anywhere in memory using the MOVE DATA routine provided in this article.

Using The Pointers

Before explaining the technique, let's briefly review the BASIC token program. There are several sections to every BASIC program, each with a two-byte pointer to a particular memory address. The address of the location is obtained by multiplying the second byte by 256 and adding the first byte. The location, name, and purpose of the pointers for the Atari BASIC token program are:

128,129 LOMEM: A 256-byte section used by BASIC for temporary storage.

130,131 VNTP: A table containing a list of all the variable names.

132,133 VNTD: Ending address of variable name table, plus one, one byte containing a zero when fewer than 128 variables have been used.

134,135 VVTP: Variable value table containing values for scalar variables and offset dimension, and length for arrays and strings.

136,137 STMTAB: Statement table containing the tokenized version of program statements.

138,139 STMCUR: Current statement being executed.

140,141 STARP: String array storage area.

When a program is SAVEd, LOMEM is subtracted from each of the pointers, and these values are SAVEd. Along with these pointers, all the information from VNTP up to STARP is SAVEd. When a program is LOADed, the reverse process occurs. This process is necessary because the addition of peripheral devices such as disk drives changes the value of LOMEM and thus the location of the BASIC program.

These pointers and the contents of the various sections are not static, but are moved around in memory as additions and deletions are made. The use of a new variable name – even in direct mode (no line number) – creates changes in the program even though no changes in the listed program occur. The new name is added to the variable name table. The pointers from VNTD to end are incremented by the length of the name and the contents of VNTD to end-of-program are moved up in memory by the length of the name. Then eight bytes are reserved at the end of the variable value table, and the contents and pointers for STMTAB to end-of-program are similarly moved.

RAM Utilities

If we want to create RAM space for our own use within the BASIC program, several requirements must be satisfied. The area must be located in a place where it will be SAVEd when the program is SAVEd. The area must be in a location where any changes in the program (either in the direct mode or program mode) will not only leave the contents of the area intact, but also move the contents when the adjoining area is moved. Since the RAM area is not static, we also must have a pointer to this area. An area created between VNTD and VVTP is the only area that meets these requirements. Because VNTD is always one byte long, the location of safe RAM can be calculated from the formula:

```
PEEK(132)+1+256*PEEK(133)
```

Enter and run Program 1, and the routine for adding RAM along with four other utilities will be installed in safe RAM at location VNTD+1. You can now delete all program lines and save these utilities in a blank program for future use. You may want to leave a REM statement just to remind you what is in the program. The utilities and the calls for these utilities are as follows:

Add RAM: A=USR(VNTD+1, bytes of RAM)

Delete RAM: A=USR(VNTD+12, bytes of RAM)

Move Data: A=USR(VNTD+23,destination, source,bytes)

String Assign: A=USR(VNTD+90,ADR (string),dimension,length,desired location) where VNTD=PEEK(132)+256xPEEK(133)

You need to be careful when using these and other machine language routines. A mistake in any of the USR statements will more than likely lock up your computer. So before executing one of these calls, double-check your typing. If it is part of a program, SAVE the program before running it. Then if the computer locks up, shut it off, LOAD your program back in, and correct the mistake. No test is included in these routines to check for the correct number of arguments in the USR statement. If only one or some of the routines are needed for your program, then only those portions can be retained. We'll talk more about this later.

If you look at Program 1, you will notice that the Add RAM and Delete RAM routines are each only 11 bytes long. How do we go about moving all those pointers and memory with just 11 bytes of code? Why not let BASIC do it for us? The memory manager routines are located in the Atari BASIC cartridge at location 43135-43358 ($A87F-$A95E). All our USR routine does is load the 6502 registers with the appropriate values and then jump to the memory manager routines.

Routine Features

The Add RAM routine cannot create RAM and is therefore limited by the amount of memory in your computer. If you attempt to add more bytes than are available, an "ERROR 2 AT LINE X" message is returned. Memory is always added at the current location of VVTP. Thus, any additional RAM is added at the end of current safe RAM.

The Delete RAM routine similarly deletes RAM at the end of current safe RAM. However, if you try to delete more bytes than were previously added, you might as well start all over.

The Move Data routine lets you move a block of data rapidly from one place to another in memory, including safe RAM. For example, moving 1K of data takes about 13 seconds using a PEEK and POKE routine. Using the Move Data routine, this same move takes a fraction of a second.

The move routine can even move a section of memory into an adjacent overlapping area without erasing any part of it. To demonstrate this point, execute the following line:

```
DIM A$(9):A$="ABCDEF":A$(4)=A$:?A$
```

The result: A$="ABCABCABC". What happens is that the origination string is changed by the destination string during the transfer of data. However, the Move Data routine moves bytes starting with either the first or last byte of the section of code to be moved, depending on whether the movement is forward or backward in memory. If you execute the following line:

```
DIM A$(9):A$"ABCDEF":A=USR(PEEK(132)+23
  +256xPEEK(133),ADR(A$)+3,ADR(A$),6):?A$
```

the resulting A$ equals "ABCABCDEF". So A$(4) actually is equal to the original A$.

It is possible to use this routine for many purposes including player/missile movement. I have not tested to see what would happen if the routine were called from a Vertical Blank Interrupt. Since the routine uses part of the memory manager routine and some of BASIC's zero page, it might interfere with the BASIC program. If this occurs, then the Vertical Blank Interrupt must first save locations $99-9C and then restore them at the end of the routine.

Safe Strings

Safe RAM may also be used for storing strings using string manipulation techniques. The String Assign routine readily performs that task for us. Usually strings are manipulated by changing values in the variable value table. Bytes two and three must be changed to contain the low and high bytes of the offset of the location of the address of the string from STARP. Safe RAM, however, is located before STARP, and the offset from STARP is negative. To get a positive offset for a string located before STARP, we must add 65536 to the location before subtracting STARP and converting to a low and high byte offset. We are simply assuming that memory wraps around to zero when top-of-memory is reached.

Before we get too involved in how to calculate offset and where to POKE the values, let's look at the String Assign routine. This routine will calculate the offset for you and POKE it, the current length, and dimension values into the corresponding bytes in the variable value table for the variable identified by ADR(string). All you need to do is provide the address where the routine is located, the dimension, the length, and where you want the string to be located.

Before going to the String Assign routine, you must DIM the string to avoid getting an ERROR 9 message. But to save memory in the string array area, DIMension the string to length one. Then call the routine and change to the desired dimension. The term ADR(string) in the USR call is not used directly, but is used by the routine to identify the string to be changed. If you have ever tried string manipulation by calculating and POKEing string offsets, then you will certainly appreciate this routine.

String Assign is best used in a program. If used in direct mode, it is best done with all statements in a single, direct mode line. If, for example, you locate A$ at location 1536, the address for A$ will remain at 1536 while the direct line is being executed or while a program is running. But as soon as the program returns with a READY, the address for A$ will change.

There is a good reason for this. Any line that is typed in direct mode is entered as the last line in the statement table. STARP is moved backwards or forwards in memory, depending on the length of the line. Along with STARP, all pointers (offsets) for strings and arrays are changed accordingly. A string residing in the normal string array area can still be printed. However, a string moved outside that area will have only its offset changed, but not its location.

This routine can be used to clear memory for player/missile graphics. Point string A$ to the location of the player/missile area and set length and dimension to the number of bytes to be cleared with the String Assign routine. Then follow with the statement:

```
A$(1,1)=CHR$(0):A$(2)=A$
```

Deleting The Utilities

If, after finishing a program, you want to delete part or all of the utility routines, you must perform several steps:

1. Move the utilities to location 1536+1 (page six). Then use the USR routines in page six to perform the remaining steps. Substitute 1536 for VNTD in the USR calls.

2. Move the routines you want to keep to the beginning of safe RAM.

3. Delete the remaining safe RAM.

4. Change all lines in the BASIC program that contain references to safe RAM to the new location within safe RAM (e.g., the first argument of the USR routines).

5. SAVE the program.

Safe RAM can be used for USR routines, player/missile data, custom character sets, and strings. But certain rules must be followed. For example, only relocatable USR routines can be left in safe RAM. Others must be moved to their designated location. If you want to change a program containing a USR routine, try the program by leaving it in safe RAM. If it does not work there, transfer it to the prescribed location using the Move Data routine. Character sets must be located on a 1K or 1/2K memory boundary and must be moved from safe RAM to their correct location.

Even though data must be moved from safe RAM to another location, you will still save more memory than when using DATA statements. For example, the 133 bytes of data in Program 1 occupy a total of 555 bytes when stored in the DATA statements. This is because each digit of each number and each comma occupies one byte of memory in the BASIC token program.

A Practical Application For Safe RAM

As an example of the improvements which can be made using this technique, I have converted

"SKI!" (Atari version of "Slalom") from February COMPUTE! to initialize from safe RAM (see Program 2). The 15-second initialization has been cut to just a fraction of a second. And preloading the data into safe RAM saves 930 bytes more than the original initialization routine.

Other additions include a USR routine to generate the course, with the option of viewing the course before running it. Control of the skier's horizontal motion has been added to the Vertical Blank routine. This makes horizontal motion proportional to the scrolling rate. These changes do not affect the nearly instantaneous start of the program.

To enter the modified version, first type in the BASIC loader for SKI!. The DATA statements contain the safe RAM utilities and initialization data for the game. Each DATA line includes a checksum, which should greatly reduce the chance of errors. When you RUN the loader, it will POKE the data into safe RAM, then erase most of itself. Delete the remaining lines and, for safety, SAVE a copy of the safe RAM portion. Then type in modified SKI!. You must SAVE, not LIST, the combined program to tape or disk, since a LISTed version will not include the safe RAM portion. If you typed in the original SKI!, you'll be amazed at the increase in speed when you RUN the new version.

Using safe RAM and the utilities given in this article, you should be able to write programs that do not start with the message "Just a Moment" or "Initializing." The uses for safe RAM are not limited, of course, to the examples we've discussed. There's a lot of room for you to develop your own applications for safe RAM.

Explanation Of USR Routines

Add RAM (11 bytes):

The routine uses a JMP $A881 to the memory manager routine for moving pointers and contents of token program to a higher location in memory. Before jumping, the following registers are loaded: X = token file pointer, which in our case is $86(VVTP); A = MSB(length); Y = LSB(length). Jumping to $A871 automatically loads A = 0.

Delete RAM (11 bytes):

This routine is identical to the above, except for JMP $A8FD.

Move DATA (67 bytes):

The routine stores destination in $9B,9C and source in $99,9A. It then determines if the move is positive or negative. For positive direction, a routine at $A8E3 is used which requires that X = MSB (length) + 1, Y = LSB (length), $9A = $9A + MSB (length), and $9C = $9C + MSB (length). For a negative move, the routine at $A94C is used which requires X = MSB (length) + 1, Y = complement of LSB (length) and ($99),Y and ($9B),Y point to first byte of source and destination, respectively.

String Assign (44 bytes):

This routine first obtains the variable number for the desired string from the statement table and loads it into the accumulator. A JSR $AC28 returns with the address of the desired string in the variable value table in $9D,9E. The dimension and length are then pulled off the stack and stored in ($9D),7 through ($9D),4. The desired location is then pulled off, and the offset calculated and stored in ($9D),2 and ($9D),3.

Program 1: Safe RAM

```
10 FOR A=1 TO 133:READ B:POKE 1536+A
   ,B:C=C+B:NEXT A:IF C<>18631 THEN
   PRINT "CHECK DATA STATEMENTS":END

20 REM ADD RAM
30 DATA 104,104,170,104,168,138,162,
   134,76,129,168
40 REM DELETE RAM
50 DATA 104,104,170,104,168,138,162,
   134,76,253,168
60 REM MOVE DATA
70 DATA 104,162,3,104,149,153,202,16
   ,250,56,165,155,229,153,165,156,2
   29,154,104,170,144,16,24,101,154,
   133,154,138
80 DATA 101,156,133,156,232,104,168,
   76,227,168,232,104,168,101,153,13
   3,153,176,2,198,154,152,24,101,15
   5,133,155
90 DATA 176,2,198,156,152,73,255,168
   ,200,76,76,169
100 REM STRING ASSIGN
110 DATA 104,104,104,160,4,200,177,1
    38,201,60,208,249,200,200,200,17
    7,138,32,40,172,160,7,104,145,15
    7,136,192,2
120 DATA 208,248,56,170,104,229,140,
    145,157,200,138,229,141,145,157,
    96
130 A=USR(1536+1,133)
140 A=USR(1536+23,PEEK(132)+1+256*PE
    EK(133),1537,133)
```

Program 2: Safe RAM Application

BASIC Loader For SKI!

```
0 REM LOADER FOR 'SKI'
10 ? "JUST A MOMENT":DIM A$(746):A=1
   :B=0:C=20:FOR D=0 TO 36:GOSUB 70:
   NEXT D:C=6:GOSUB 70
20 IF B<>73882 THEN ? "CHECK ALL DAT
   A LINES":END
30 VNTD=PEEK(132)+256*PEEK(133)
40 A=USR(ADR(A$),746)
50 A=USR(ADR(A$)+22,VNTD+1,ADR(A$),7
   46)
60 GOTO 1000
70 E=0:FOR F=1 TO C:READ G:E=E+G:B=B
   +G:A$(A,A)=CHR$(G):A=A+1:NEXT F
80 READ F:IF F<>E THEN ? "CHECK DATA
    STATEMENTS AT LINE ";100+D*10:EN
   D
90 RETURN
```

```
100 DATA 104,104,170,104,168,138,162
    ,134,76,129,168,104,104,170,104,
    168,138,162,134,76,2617
110 DATA 253,168,104,162,3,104,149,1
    53,202,16,250,56,165,155,229,153
    ,165,156,229,154,3026
120 DATA 104,170,144,16,24,101,154,1
    33,154,138,101,156,133,156,232,1
    04,168,76,227,168,2659
130 DATA 232,104,168,101,153,133,153
    ,176,2,198,154,152,24,101,155,13
    3,155,176,2,198,2670
140 DATA 156,152,73,255,168,200,76,7
    6,169,104,104,104,160,4,200,177,
    138,201,60,208,2785
150 DATA 249,200,200,200,177,138,32,
    40,172,160,7,104,145,157,136,192
    ,2,208,248,56,2823
160 DATA 170,104,229,140,145,157,200
    ,138,229,141,145,157,96,112,112,
    112,70,155,34,102,2748
170 DATA 20,144,38,38,38,38,38,38,38
    ,38,38,38,38,38,38,38,38,38,38,3
    8,848
180 DATA 38,38,6,65,130,9,0,0,0,21,0
    ,0,0,0,0,0,0,0,0,0,307
190 DATA 0,0,6,14,28,24,32,0,128,0,0
    ,0,0,0,0,0,0,0,0,0,232
200 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
210 DATA 0,192,192,220,20,28,7,5,7,0
    ,0,24,52,44,60,24,0,16,56,56,100
    3
220 DATA 124,124,254,16,16,8,28,62,6
    2,62,8,8,0,0,56,94,106,94,116,56
    ,1294
230 DATA 0,0,119,69,117,21,119,0,0,8
    ,24,56,120,8,8,8,8,0,0,0,685
240 DATA 48,88,56,16,186,254,89,24,1
    56,82,33,16,8,0,0,0,12,26,28,8,1
    130
250 DATA 93,127,154,24,57,74,132,8,1
    6,0,0,24,60,60,24,24,60,186,89,2
    4,1236
260 DATA 154,170,198,65,65,1,18,36,7
    4,161,18,156,77,10,24,24,0,0,0,0
    ,1251
270 DATA 0,169,0,133,0,169,1,141,99,
    6,169,8,141,98,6,104,104,133,7,1
    04,1592
280 DATA 133,6,104,104,133,1,162,10,
    160,117,169,7,32,92,228,96,216,1
    73,4,208,2155
290 DATA 240,4,169,0,133,0,165,0,240
    ,85,165,1,240,81,206,99,6,173,99
    ,6,2112
300 DATA 208,73,198,203,208,26,169,2
    ,133,203,173,124,2,208,3,206,100
    ,6,173,125,2543
310 DATA 2,208,3,238,100,6,173,100,6
    ,141,0,208,165,0,141,99,6,206,98
    ,6,1906
320 DATA 174,98,6,142,5,212,208,27,1
    60,0,56,177,6,233,20,145,6,160,1
    ,177,2013
330 DATA 6,233,0,145,6,169,7,141,98,
    6,141,5,212,198,1,76,98,228,0,0,
    1770
340 DATA 0,0,104,162,228,160,98,169,
    7,32,92,228,96,173,10,210,41,3,2
    01,3,2017
350 DATA 176,247,96,170,169,72,224,1
    ,240,6,169,73,144,2,169,0,145,20
    3,96,165,2567
360 DATA 206,133,205,173,10,210,41,7
    ,24,105,6,168,169,134,145,203,17
    3,10,210,48,2380
370 DATA 14,169,23,133,207,169,18,13
    3,208,169,22,133,209,208,12,169,
    17,133,207,169,2522
380 DATA 22,133,208,169,18,133,209,5
    6,152,229,209,168,162,3,152,24,1
    01,207,168,169,2692
390 DATA 204,145,203,152,24,101,208,
    168,169,204,145,203,202,208,235,
    96,104,104,133,204,3212
400 DATA 104,133,203,169,0,133,205,1
    69,24,133,206,32,113,6,168,32,11
    3,6,32,123,2104
410 DATA 6,136,16,247,160,17,32,113,
    6,201,1,240,4,200,144,1,200,32,1
    13,6,1875
420 DATA 32,123,6,200,192,20,208,245
    ,173,10,210,201,13,176,18,24,165
    ,205,105,10,2336
430 DATA 197,206,176,9,32,139,6,240,
    52,208,2,208,194,173,10,210,201,
    25,176,4,2468
440 DATA 169,7,208,20,173,10,210,201
    ,25,176,4,169,10,208,9,173,10,21
    0,201,25,2218
450 DATA 176,19,169,139,170,173,10,2
    10,41,15,201,12,176,247,24,105,3
    ,168,138,145,2341
460 DATA 203,230,206,165,206,201,175
    ,208,1,96,24,169,20,101,203,133,
    203,165,204,105,3018
470 DATA 0,133,204,208,182,0,727
1000 A=0:FOR B=1 TO 5
1010 GRAPHICS 0:POSITION 2,4
1020 FOR C=1 TO 10:? A*10:A=A+1:NEXT
     C
1030 ? "CONT":POSITION 2,0:POKE 842,
     13:STOP
1040 POKE 842,12
1050 NEXT B
1060 ?" SAFE RAM IS LOADED. DELETE
     REMAINING LINES AND ENTER MODI
     FIED SKI!"
```

Modified SKI!

```
100 DIM SCREEN$(1),PM$(1)
101 DIM LEFT$(20),CENTER$(20),RIGHT$
    (20),CURR$(20),CRASH$(20),ERASE$
    (20),DIR(8),SCR(4),DLIST$(1)
102 DIM T$(20),TOPLINE$(20):GOTO 130
110 REM * SKI * LINE 100 MUST BE TYP
    EDIN FIRST!!!
120 HI=INT(A/256):LO=A-HI*256:RETURN
125 POKE 66,1:FOR W=1 TO 10:POKE 532
    79,0:POKE 53279,8:NEXT W:POKE 66
    ,0:RETURN
130 GOSUB 790:REM Initialization rou
    tines
140 REM PLAYER ROUTINE
150 POKE 559,62:POKE 54279,PMBASE
160 POKE 53277,3:POKE 704,2*16+6
170 PO=1024:YP=180:XP=128
180 PM$(PO)=CHR$(0):PM$(PO+254)=CHR$
    (0):PM$(PO+1)=PM$(PO)
195 SCR(0)=0:SCR(1)=10:SCR(2)=4:SCR(
    3)=2:SCR(4)=1
200 ERASE$=CHR$(0):ERASE$(20)=CHR$(0
    ):ERASE$(2)=ERASE$
```

```
210 LEFT$=ERASE$:CENTER$=ERASE$:RIGHT$=ERASE$:CRASH$=ERASE$
220 FOR I=0 TO 15
230 LEFT$(I+2,I+2)=CHR$(PEEK(CHSET+208+I))
240 CENTER$(I+2,I+2)=CHR$(PEEK(CHSET+224+I))
250 RIGHT$(I+2,I+2)=CHR$(PEEK(CHSET+104+I))
260 CRASH$(I+2,I+2)=CHR$(PEEK(CHSET+240+I))
270 NEXT I
280 DIR(0)=0:DIR(1)=20:DIR(2)=19:DIR(3)=21:DIR(4)=1:FOR I=0 TO 3:DIR(I+5)=-DIR(I):NEXT I:DIR(5)=-1
290 CURR$=CENTER$
295 POKE 1636,XP:POKE 203,2:SCR=0
300 PM$(PO+YP,PO+YP+20)=CURR$
310 SCR=SCR+SCR(PEEK(0)):POKE 77,0
320 POSITION 2,0:? #6;SCR;" ";:POSITION 15,0:IF PEEK(0)<>0 THEN ? #6;(5-PEEK(0))*100
330 IF PEEK(1)<3 THEN POKE 0,0:GOTO 740
340 ST=STICK(0)
350 LEFT= NOT PTRIG(1):RIGHT= NOT PTRIG(0):LR=LEFT+2*RIGHT
360 CURR$=CENTER$:XP=PEEK(1636)
370 IF LEFT THEN CURR$=LEFT$:IF LR<>OLR THEN SV=2:TI=5
380 IF RIGHT THEN CURR$=RIGHT$:IF LR<>OLR THEN SV=4:TI=5
390 IF TI>O THEN TI=TI-1:SOUND 0,SV,0,TI
400 IF LR=0 THEN SOUND 0,0,0,0:TI=0
410 OLR=L
420 UP=(ST=14 OR ST=10 OR ST=6):DOWN=(ST=5 OR ST=9 OR ST=13)
430 YP=YP-2*UP+2*DOWN:IF YP>200 THEN YP=200
440 IF YP<40 THEN YP=40
450 POKE 0,1+(YP>130)+(YP>160)+(YP>185)
460 IF PEEK(P0PF)=0 THEN 300
470 WHICH=INT(LOG(PEEK(P0PF))/LOG(2)+0.1):TEMP=PEEK(0):POKE 0,0
480 PM$(PO+YP,PO+YP+20)=ERASE$
490 POKE HITCLR,1:IF WHICH<>2 THEN 620
500 REM POINTS
510 PTR=ASC(DLIST$(8))+256*ASC(DLIST$(9))
520 LINE=INT((YP-39)/8)+1
530 COL=INT((XP-49)/8)+1
540 LOC=PTR+LINE*20+COL:SOUND 0,0,0,0
550 FOR I=0 TO 8:P=PEEK(LOC+DIR(I))
560 IF P<128 OR P>192 THEN 590
570 POKE LOC+DIR(I),O
580 SCR=SCR+(P=139)*50+(P=134)*100*(5-TEMP):I=11:NEXT I:GOTO 600
590 NEXT I:GOTO 610
600 FOR W=15 TO 0 STEP -1:SOUND 0,20,10,W:NEXT W
610 POKE 0,TEMP:POKE HITCLR,1:GOTO 300
620 REM  CRASH! 
630 SOUND 0,0,0,0
640 PM$(PO+YP,PO+YP+20)=CRASH$
650 FOR W=100 TO 150 STEP 2:SOUND 0,W,12,10:NEXT W
660 PM$(PO+YP,PO+YP+20)=ERASE$
670 YP=200
680 PM$(PO+YP,PO+YP+20)=CURR$
690 POKE O,1:SOUND 0,0,0,0
700 XP=INT(72+90*RND(0)):POKE 53248,XP:POKE 1636,XP
710 IF PEEK(P0PF)<>0 THEN POKE HITCLR,1:GOTO 700
720 POKE HITCLR,0:SCR=SCR-50:IF SCR<0 THEN SCR=0
730 GOTO 300
740 IF SCR>HSCR THEN HSCR=SCR
745 POSITION 8,0:? #6;"  HIGH ";HSCR
750 SOUND 0,0,0,0
760 SCREEN$(326,336)="press{,}start"
770 IF STRIG(0) THEN 770
780 GOTO 130
790 REM INITIALIZATION
800 GRAPHICS 17:HILO=120:POKE 53248,0:POKE 0,0
810 SETCOLOR 4,0,12:SETCOLOR 1,12,8:SETCOLOR 2,9,6:SETCOLOR 0,15,4
820 P0PF=53252:HITCLR=53278:POKE HITCLR,0
830 SCRBASE=PEEK(106)-16:REM 4K BOUNDARY
840 PMBASE=SCRBASE-8:REM 2K BOUNDARY,DOUBLE-LINE RES
850 CHBASE=PMBASE:REM FILL UP OFFSET WITH CHARACTERS
870 VNTD=PEEK(132)+256*PEEK(133)
880 A=USR(VNTD+90,ADR(SCREEN$),4097,4097,SCRBASE*256)
890 A=USR(VNTD+90,ADR(PM$),2049,2049,PMBASE*256)
900 A=USR(VNTD+90,ADR(DLIST$),40,40,VNTD+134)
910 CHSET=CHBASE*256
920 A=USR(VNTD+23,CHSET,VNTD+174,120):A=USR(VNTD+23,CHSET+128,57472,344):A=USR(VNTD+23,CHSET+208,VNTD+294,48)
930 A=VNTD+377:GOSUB 120:POKE VNTD+368,HI:POKE VNTD+370,LO
940 A=USR(VNTD+23,1649,VNTD+494,103)
950 Z=USR(VNTD+483):REM DISABLE VBLANK
960 POKE 756,CHBASE:RESTORE 990
980 A=ADR(DLIST$):GOSUB HILO:POKE 561,HI:POKE 560,LO
1020 DLIST$(32)=CHR$(PEEK(560)):DLIST$(33)=CHR$(PEEK(561))
1030 SCREEN$(1)=CHR$(0):SCREEN$(4095)=CHR$(0):SCREEN$(2)=SCREEN$
1040 TOPLINE$=SCREEN$
1050 A=ADR(TOPLINE$):GOSUB HILO
1060 DLIST$(5,5)=CHR$(LO):DLIST$(6,6)=CHR$(HI)
1070 POKE 88,LO:POKE 89,HI
1080 POSITION 8,0:? #6;"SKI!";
1082 SCREEN$(121,139)="press{,}button{,}to{,}ski"
1083 SCREEN$(163,178)="pull{,}joystick{,}to":SCREEN$(185,195)="view{,}course"
1085 SCREEN$(403,419)="f{2 ,}i{2 ,}n{2 ,}i{2 ,}s{2 ,}h"
1090 A=USR(VNTD+597,ADR(SCREEN$)+480)
1410 A=SCRBASE*256
1420 A=A-20*(STICK(0)=14)+20*(STICK(0)=13)
1430 IF A>SCRBASE*256+3480 THEN A=A-20
1440 IF A<SCRBASE*256 THEN A=A+20
```

```
1450 GOSUB HILO:T$=CHR$(LO):T$(2)=CH
     R$(HI):DLIST$(8,9)=T$
1460 IF STRIG(0)=1 THEN 1420
1470 A=SCRBASE*256+3480:GOSUB HILO
1480 T$=CHR$(LO):T$(2)=CHR$(HI):DLIS
     T$(8,9)=T$
1481 GOSUB 125:IF STRIG(0) THEN 1481
1490 A=USR(VNTD+342,ADR(DLIST$(8)),1
     76)
1495 SCREEN$(121,195)=SCREEN$(120)
1500 RETURN
2000 VNTD=PEEK(132)+256*PEEK(133)
2010 POKE VNTD+342+26,78
2020 POKE VNTD+342+28,67
2030 POKE VNTD+342+39,85
2040 POKE VNTD+342+43,81
2050 POKE VNTD+342+51,73
2060 A=USR(VNTD+23,20000,VNTD+342,36
     5)
2070 A=USR(VNTD+23,20000+52+30,20000
     +52,365)
2075 A=USR(VNTD+23,20000+36+9,20000+
     36,400)
2080 RESTORE 2000
2085 FOR A=0 TO 8:READ B:POKE 20036+
     A,B:NEXT A
2086 DATA 173,4,208,240,4,169,0,133,
     0
2090 FOR A=0 TO 29:READ B:POKE 20000
     +61+A,B:NEXT A
2095 DATA 198,203,208,26,169,2,133,2
     03
2100 DATA 173,124,2,208,3,206,100,6
2110 DATA 173,125,2,208,3,238,100,6
2112 DATA 173,100,6,141,0,208
2200 FOR A=0 TO 103:? A,PEEK(VNTD+36
     0+A),PEEK(20000+A):NEXT A
```

©

VIC Scaling Bar Graphs

James P McCallister

Bar graphs often have more visual impact than figures or statistics. This utility program gives you multicolor bar graph capability and demonstrates its practical uses. There is a discussion of modification of control variables to suit a variety of applications.

As you learn to solve problems in BASIC on your VIC, you realize that calculations are fairly easy to program no matter how complex the math. However, you sometimes want to display your answers in a way that highlights their meaning. One very effective method is to construct a bar graph.

This article describes a utility bar graph subroutine you can add to your own programs. It's very easy to use, and it's suitable for many applications. Once you have it saved on tape or disk, you can quickly turn a modest calculation into an analysis of options or a graphic history of trends.

One Practical Application

Here's an example that demonstrates a practical use of the bar graph subroutine. Suppose you want to borrow money to buy a car. Checking a reference book on business math, you find that the monthly payment calculation for an installment loan is:

$$\text{payment} = \text{amount of loan} * r/(1-(1+n)^{-r})$$

where n is the number of months and r is the monthly interest rate, expressed as a decimal fraction. Also,

$$\text{total finance cost} = (n * \text{payment}) - \text{amount of loan}$$

So, as a first step, you come up with this BASIC program to do the fundamental calculation:

```
100 INPUT"{2 DOWN}AMNT,APR";AM,AR
110 MR=AR/1200
120 INPUT"MONTHS";N
130 MP=AM*MR/(1-(1+MR)↑-N)
140 FC=N*MP-AM
150 DEFFNR(X)=-INT(-X*100)/100
160 PRINT"PAYMNT=$";FNR(MP),"FIN COST=$"
    ;FNR(FC)
170 GOTO100
```

In this program, AM is the amount of the loan, AR is the annual percentage rate, and N is the term, in months. MR is the monthly rate expressed as a decimal fraction. The results are MP, the monthly payment, and FC, the total finance cost; both are based on the formulas you started with. The function FNR(X) defined in line 150 rounds to the next highest penny, and the results are printed in line 160.

Naturally, increasing the number of months (or term) of the loan makes the payment lower, but increases the finance cost. With this program you could experiment with various amounts, APRs and terms, and arrive at a decision about which loan suits you best.

Instead, let's modify the program to *plot* the effect of various term lengths of the loan, using the bar graph subroutine. We'll still use the INPUT statement for amount and APR; but we'll use the FOR/NEXT statements to automatically vary N and take care of storing the answers. Now the program looks like this:

```
100 INPUT"{CLR}AMNT,APR";AM,AR
110 MR=AR/1200
120 FORI=0TO8:N=6*(I+2):REM N=12TO60BY6'
    S
130 Y(0,I)=AM*MR/(1-(1+MR)↑-N)
140 Y(1,I)=N*Y(0,I)-AM
150 NEXTI:QB=8
160 QL$="{12 SPACES}1 2 3 4 5{15 SPACES}
    YEARS"
170 QT$="{2 SPACES}MONTHLY PAYMENT":Z=0:
    GOSUB9900
180 QT$="{2 SPACES}TOTAL FINANCE COST":Z
    =1:GOSUB9900
190 GOTO100
```

Add lines 9900 to 9999 (see Program 2).

The resulting graphs are in Figures 1 and 2 for a loan of $5000 at 12.8 percent.

We'll deal with the new variables in detail later. But you can already see that plotting the graphs requires these steps:

1. The BASIC instructions for the subroutine are added to the program, beginning in line 9900. (Later, we'll discuss the most efficient ways to do this.)

2. The numbers to be plotted are put into two lists, one for each graph. At line 130, the list for the monthly payment graph is put in Y(0,I), and at line 140 the list for the finance cost graph is put in Y(1,I). The first subscript for each Y variable identifies the list. The second subscript is a label for each number in the list.

3. After the lists are completed, GOSUB9900 is "called" once for each graph (line 240).

4. Before calling GOSUB9900 the first time, QB is given a value of 8 so the subroutine would know how long the lists were and how many bars to plot. Also, the legend to appear at the bottom of the screen is put in QL$.

5. Before calling GOSUB9900 each time, title information is put in QT$. This is different for each graph.

6. Before calling GOSUB9900 each time, Z is given a value corresponding to the list to be plotted.

The graphs in Figures 1 and 2 can be displayed using just the lines listed so far. The subroutine does all the layout work. The vertical scale and labeling are automatic. The example could have included billions of dollars or 10^{-9} seconds of time – the scale would still be worked out and labeled correctly.

Program Features

Now we're ready to explore the features of the subroutine in greater depth. The primary features are:

- one to 21 vertical bars per display (number specified by user).
- automatic ranging and scaling for each display, with no restrictions on the signs or magnitudes of the values to be plotted. The scale is labeled.
- operates from a two-dimensional array, so that several lists can be stored before plotting the first one.
- positive bars go up; negative bars go down.
- built-in "hold" of display, released by touching any key; cursor prompt in lower-right corner.
- universal memory configuration. Adds 2300 bytes to the calling program.
- scale lines in contrasting color (green).

The subroutine also has a number of *optional* features which are *transparent* to the user. (Transparent features are built-in program features that you can ignore without consequence – they take care of themselves.) The features are controlled by giving values to certain control variables. In BASIC, all numeric variables are initially set to zero, and the subroutine is designed so that a zero signifies the standard condition for each option. That's why the standard choices are transparent. The standard choice for an option is often called the "default" option, because if you don't specify an option, you'll get the standard choice by default. The transparent features are:

- The top four lines of the display can be printed with title or explanatory information. (Null strings are standard.) Also, the two bottom lines can be printed with legend information.
- Choice of bar color (black is standard); or contrasting colors for "up" and "down" bars; or alternating bar colors; or contrasting colored bars grouped in pairs.
- Close-spaced (standard) or double-spaced bars.
- Graph positioned against right edge of screen (standard), or moved left a chosen number of spaces.
- Automatic ranging and scaling, with all bars starting at zero (standard); or expanded, offset scale; or preassigned scale. Also, prescaling of multiple lists before plotting.
- Vertical scale labeled (standard), or unlabeled. Labels are printed to the left of the scale, if there's room. Otherwise, they will move automatically to just above or below the scale.

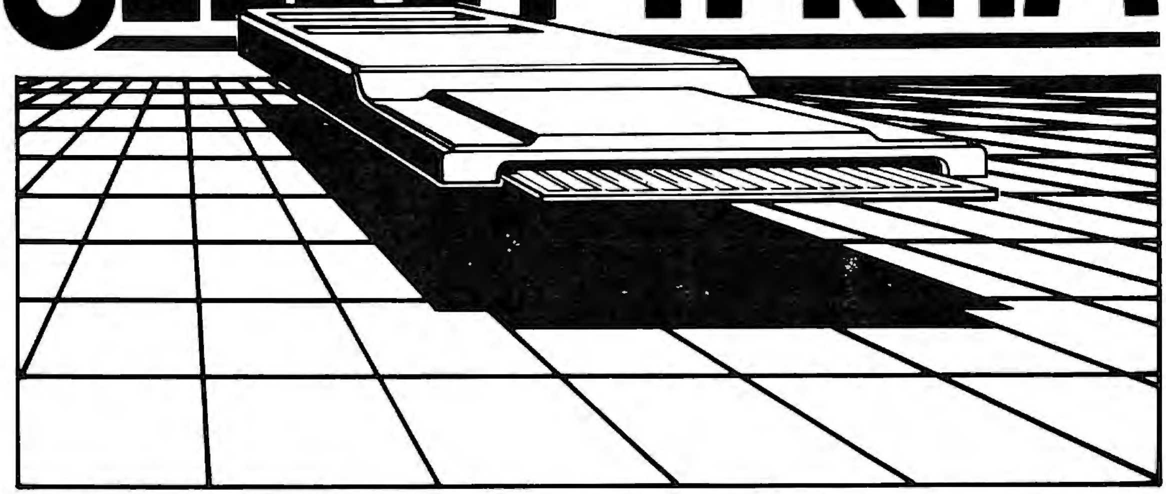
How To Use The Subroutine

To create a bar graph plot, insert GOSUB9900 into your program after assigning the value or values to be plotted into the Y array. There are two fundamental restrictions in your main program. First, you cannot use line numbers from 9900 to 9999. Second, you shouldn't use variable names starting with Q, unless you're willing to share them with the subroutine.

The result of the GOSUB9900 (if no control variables are set) will be a single bar, representing the value stored in Y(0,0). Going beyond this simple graph requires using control variables – but only those you want to change from zero. Most likely you'll find that several of the controls are quickly mastered. You can then add more to your repertoire as you gain experience. All control variables must be given any new values before GOSUB9900.

64K for VIC 20™

SELECT·A·RAM™



SELECT·A·RAM

STANDARD FEATURES

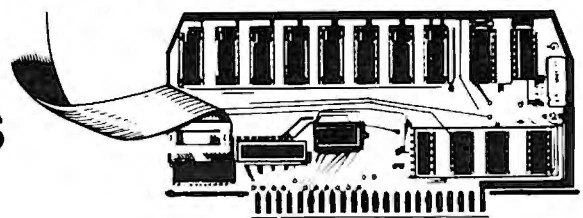
- 8K BLOCKS SELECTABLE FROM THE KEYBOARD OR BY SOFTWARE COMMAND
- TWO EXPANSION SLOTS
- WRITE PROTECTION
- RESET SWITCH
- EXPANDABLE TO 192K WITH ADDITION OF 64K EXPANSION MODULES
- COMPATIBLE WITH ROM CARTRIDGES
- ONE YEAR WARRANTY ON PARTS AND LABOR
- 15 DAY MONEY BACK GUARANTEE

SELECT-A-RAM ....$169.

64K EXPANSION MODULE ....$149.

TRADE-INS ACCEPTED
3K $5 8K$10 16K-$20

VIC 20 IS A TRADEMARK
OF COMMODORE ELECTRONICS LIMITED



APS-52A

52K MEMORY FOR ATARI 400/800 .... $119.00

The APS-52A memory expansion boards come assembled and tested with a 90 day warranty covering materials and workmanship. Boards come with complete documentation including operation and installation instructions and a memory test program.

16K TRADE-INS ACCEPTED $15.00

ATARI 400 AND 800 ARE TRADEMARKS
OF ATARI,INC.

512-441-3202 PO BOX 43006 Austin, Tx. 78745-0001

ADVANCED·PROCESSOR·SYSTEMS

MasterCard

VISA

Optional Feature Controls

These are the rules for the control variables:

1. List identification. The variable Z becomes the first subscript of the Y values, thereby controlling which list is to be plotted. If there is only one list, then Z is always zero and never needs to be changed from its initial zero value.

2. Number of bars. The number of bars plotted is controlled by the variable QB. QB = 0 plots one bar; QB = 1 plots two bars, etc. The maximum value of QB is 20, which will plot 21 bars scaled to the values stored in Y(Z,0), Y(Z,1), Y(Z,2) – up to Y(Z,20). If either Z or QB is greater than ten, it is mandatory to use a DIM statement at the beginning of the program to DIMension the Y array. Even if they are both less than ten, it's still good programming to DIMension to save memory.

3. Title and legend. Up to 88 characters of title can be printed on the top four lines by creating a string variable QT$. Likewise, up to 44 characters of legend can be printed on the bottom two lines with QL$. [*Note:* The 44th character of QL$ can be used only with a trick. For example, if the last word is MONTH, end QL$ with the sequence "...MONH{LEFT}{INST}T". Otherwise, the top of the display will roll up and off. This sequence is only needed for exactly 44 characters.]

4. Bar color options. Bar colors are controlled with the standard VIC POKE codes for colors (which are one number lower than that color key on the keyboard). For example, the code for black is zero, and black is the standard color for the bars. The color control variables are Q0 and Q1. By assigning nonzero values to one or both, you can do the following:

- All bars one color: assign the same color to Q0 and Q1. Example: Q0 = 6: Q1 = 6 will result in all blue bars.
- Up bars one color, down bars another: assign the color for the up bars to Q0, and the color for the down bars to Q1. Example: Q0 = 0: Q1 = 4 will result in black bars pointing up and purple bars pointing down.
- Bars alternating colors: if either Q0, Q1, or both are given a minus sign, then ABS(Q0) is the color of the even bars and ABS(Q1) is the color of the odd bars. Example: Q0 = 6: Q1 = -7 will result in bars alternating between blue and yellow.

5. Bar spacing. The standard option provides bars which are closely spaced. However, if QB is assigned a negative value, the bars will be plotted with a space between them. The maximum negative value allowed is ten (QB = -10), which will plot 11 bars separated by 10 spaces. In addition, if the alternating bar colors option is chosen at the same time that QB is negative, the bars will be plotted in closely spaced pairs separated by open spaces. Under these conditions, the maximum negative value allowed is 13 (QB = -13), which will plot seven pairs (14 bars) separated by six spaces.

6. Graph centering. The graph is normally positioned against the right edge of the screen (standard). You can move it to the left n spaces by setting QC = n.

7. Scale factor options. The options for the automatic ranging and scaling feature are controlled by the variable QS. The standard choice, QS = 0, will always produce a useful graph of the data, and in many cases the result can't be improved upon. However, sometimes a bar graph can be done in a different way that makes it more informative. After all, effective chartmaking will always consider the reader, the data being compared, the most significant facts, and so on. The optional choices for QS allow you to have a scale offset from zero, or a prespecified scale. You can also prescale lists of values before doing any plots, which is desirable for merged graphs or for finding a common scale for several graphs in sequence.

Figures 3 and 4 illustrate the scale offset from zero. The graph in Figure 3 results from QS = 0. Depending on the circumstances, this may be the best graph for this data. One problem, however, is that the variations are small compared to the length of the bars. If the *variations* are the most important characteristic of the data, then Figure 4 is a better display. This is achieved by making QS = 1. The scale is expanded as much as possible – times ten, in this case. It must also be offset so the bottoms of the bars don't reach zero.

If you were the chairman of a charity sale and this was a graph of yearly results, you'd probably use Figure 4 when talking to your committee because it's easy to see the changes. But you might use Figure 3 in the final report because it doesn't exaggerate the bad news for '83. In Figure 4, the bar for '83 is only 50 percent as high as '82, but Figure 3 shows the true proportion – 93 percent. An offset scale isn't always possible, as, for example, when the list includes both positive and negative numbers. In such cases, QS = 1 will not have any special effect on the graph. Figure 5 is a graph of such data.

The QS = 2 option allows you to second-guess the Automatic Ranging and Scaling Rules by giving starting values to QX and QN. These two variables store the maximum and minimum values found in scanning the list of numbers to be plotted. This option, in effect, allows you to preassign a particular scale or a particular offset. Study the rules under "Automatic Ranging And Scaling," and then experiment to master this option.

The prescaling options, QS = 3, 4, and 5, are

COMPUTER BOOKS
★ ★ SALE ★ ★

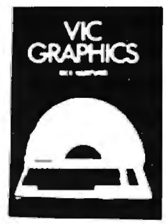
VIC-20 and COMMODORE-64

• Learn Programming • Learn Graphics • Ready Made Programs
Now You Can Explore The Inner Workings of your Computer

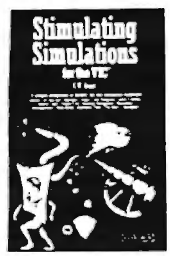
Beginning Assembly Language Programming For Your VIC or 64
Now you can learn the powerful assembly language, to get the most out of your computer. Easy to use and simple to understand, comes with machine language monitor. Perfect for the beginner — specify VIC or 64.
List $29.95 / **Sale $24.95**

I Speak BASIC To My VIC
This is the first book you should read when you get your VIC-20. Everything you need to know about BASIC programming, cassette operation, and even file handling is explained in an easy to read, easy to use format.
List $10.95 / **Sale $7.95**

VIC Revealed
The most comprehensive explanation of the hardware capabilities of your VIC-20. All the chips and corresponding instruction sets are included *plus* BASIC ROM variables and sound effects programming. (A must for Assembly language programming.)
List $14.95 / **Sale $11.95**



VIC Graphics
Finally you can learn how to do color plotting, drawing maps, pie charts, line graphs and even moving shapes on the screen. All in high-resolution with easy to understand instructions. (This is a must for game designers) requires Super Expander.
List $14.95 / **Sale $11.95**



Stimulating Simulations
Twelve simulation programs. Each program is presented with a listing, sample run, instructions, and ideas for variations. Original program listings are provided in VIC BASIC. (This book is a good starting point for the computer hobbyist who wishes to explore the use of the small computer in simulating real events.)
List $8.95 / **Sale $5.95**

Elementary Commodore-64
Now you can learn how to do more than play games. Even if you're already writing programs, this book has all the information you need and will satisfy the rest of the family's desire to participate. Step by step chapters lead you through the process of hooking up the computer, loading and saving programs on either disk or tape and using the powerful cursor commands for easy correcting. The sample programs are fun and useful too. You will learn how to create music, graphics and all sorts of handy utilities like checkbook balancers and file systems.
List $16.95 / **Sale $13.95**

Commodore-64 Reference Guide — List $19.95 / **Sale $16.95**

VIC-20 Reference Guide — List $16.95 / **Sale $13.95**

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and Master-Card.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

the same as QS=0,1, and 2, except nothing is put on the screen. But the maximum value (QX), minimum value (QN), scale factor (QZ), upper-scale label (QU), and lower-scale label (QD) are all computed and made available to the main program. Our final program uses this capability to combine Figures 1 and 2 into one graph. It's possible to make some very effective displays with these options, without a lot of programming.

8. Scale labels. You can suppress the printing of the scale labels within the subroutine by making QS negative, or if QS=0, by giving it a value of -.1.

Putting It All Together

Now we're ready to demonstrate some of these optional features using the car loan example. Program 1 merges the two previous bar graphs (Figures 1 and 2) into one display using two bar colors (see Figure 6).

Line 90 dimensions the Y array to conserve memory. If we don't do this, the BASIC interpreter will DIMension it Y(10,10) by default and tie up 455 extra bytes of memory.

We discussed lines 100-150 in the car loan example. Lines 160-190 prescale the Z=0 and Z=1 lists and then create a merged list suitable for graphing on a common scale. The merged list, in Z=2, contains *both* monthly payment and finance cost data, but at twelve-month intervals, instead of six. After prescaling each old list, the values are divided by their respective scale factors for combining into the new list. As a result, the numbers in the new list are all in the range of 0 to 5, instead of their true values. Therefore, we suppress the printing of the automatic scale labels and overprint with the labels (QU) obtained during the prescaling calls. As you can see, line 160 prescales the Z=0 list, and line 170 prescales Z=1. Line 180 puts five Z=0 values into the Z=2 list, with subscripts (2,0), (2,2), (2,4), (2,6), and (2,8). Line 190 puts five Z=1 values into subscripts (2,1), (2,3), (2,5), (2,7), and (2,9).

Line 200 establishes the optional features for the bar graph by assigning values to control variables Z, Q1, QB, QS, and QC. You should be able to match up the values in the program with the features in the graph.

Lines 210-230 create the string variables for the title and legend. Notice that part of the title uses the input variables for the amount and APR of the loan.

The scale labels we need for this graph are special, so the subroutine labels were turned off. In line 9988, you can GOSUB to your own subroutine and overprint the graph with anything you wish. In our program, we GOSUB 800-830.

Adding The Subroutine To Your Program

There are several ways to use this program with your own programs, but the easiest is to plan ahead. If you want bar graph displays, load this program into memory before you start to type in your main program, and you've got it. However, if you want to combine the bar graph subroutine with a program that's already on tape or disk, without retyping the program, here's a technique to do this which should work every time. It's a slight embellishment of Mark Niggemann's method (COMPUTE!, March 1983, p. 210).

Let's assume that you have a copy of the bar graph subroutine (Program 2) saved on tape or disk with a filename "BARSUB", and that you have a main program we'll call PROGA also saved. You want to improve PROGA by adding bar graph capability. First of all, the program cannot have any line numbers as high as BARSUB's line numbers. That is, they must be below 9900. In the case of Program 1 in this article, you must be sure to delete line 9988 before going further. LOAD PROGA and observe the line number restriction. Then clear the screen and type:

```
?PEEK(43),PEEK(44)
```

RETURN and write down the two answers, which make up the start-of-BASIC memory pointer. On an unexpanded VIC, they'll be 1 and 16 (location 1 on page 16). The trick is to change the Start pointer to be two less than the current value of the End pointer in 45 and 46. A reliable way to do this is to type in the following in direct mode (no line number) and hit RETURN:

```
A=PEEK(45)+256*PEEK(46)-2:B=INT(A/256):
  POKE43,A-256*B:POKE44,B
```

Then proceed:

```
LOAD"BARSUB"
PRESS PLAY ON TAPE
OK
SEARCHING FOR BARSUB
FOUND BARSUB
LOADING
READY.
```

Finally, POKE the numbers you wrote down back into memory. For the unexpanded VIC, for example:

```
POKE43,1:POKE44,16
```

At this point you have combined the two programs and are ready to proceed with debugging.

A Look At The Program

The bar graph subroutine adds 2300 bytes to the main program. That's not including the Y array and the part of the main program that gets ready to call the subroutine. Even so, a worthwhile program will still fit in a 5K unexpanded VIC. Our

NEW

MPI INTRODUCES SUPER ACTION MEMORY EXPANDER BOARD FOR VIC 20*

Adds 24K and 3 Expansion Slots With Switches and Fuse $109.95

(Expandable To 35K By Simply Adding Memory Chips and Switches)

INTRODUCTORY PRICE

$109.95

Transfer Cartridges To Tape

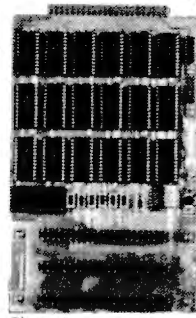
Evade Self Destruct Code With Write Protect Switch

All Boards Factory Tested

Complete Instructions And Illustrations Included.

Fuse Protected

Reset Switch



Pictured Above — Action Board with all options

24K BOARD FEATURES:

- *Adds 24K Memory (29K with VIC* 5K).
- *Upgrade Board to 35K by adding IC's and switches.
- *Memory switchable in 8K sections. (No need to remove memory board to run your other programs).
- *3 expansion slots with switches (for game or extra utility cartridges).
- *Reset button allows restarting computer without turning power off.
- *.5 amp fuse protected.
- *Switch relocates expansion cartridges in memory so that it can be saved on tape as a backup for your valuable programs. (The unexpanded VIC will not allow cartridges to be saved on tape).
- *Write protect switches allow programs stored in RAM at ROM location or entire Memory to be protected against accidental write.
- *Switch allows memory to be moved between RAM and ROM location. (Useful for developing your own games and saving on tape).
- *Gold plated card edge connector.
- *No other memory expansion needed.
- *Easily plugs into your VIC, no modifications necessary. Saves wear on your VIC 20 since board never needs to be removed or power turned off and on to run other tapes or cartridges.
- **Optional 35K memory (40K with VIC* 5K).

24K memory, 3 expansion slots, switches assembled and tested $109.95

Same as above with sockets that allow you to later add your own memory chips to bring memory from 24K to 35K $124.95

Full 35K memory, 3 expansion slots, 3K expander mode, eprom socket (switch selectable between BLK 3 & BLK 5) and all switches assembled and tested (eprom not included) $149.95

Bare 35K board with complete instruction and parts list $39.95

COD ORDERS
816 444 4651
MIDWEST PERIPHERAL INDUSTRIES

Add $5 for shipping and handling
Mo. residents add 5½% sales tax
Send check or money order to

MPI
Box 8123-B
Kansas City, MO 64112

VIC 20* is a registered trademark of Commodore.

NEW NEW NEW

"The Original"

THE FINEST
EXPANSION CHASSIS
for the
VIC-20*

"Since 1981"

15-Day Money Back Trial

6-Month Warranty

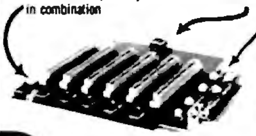
The "GAME MACHINE"®

$85

- Fully buffered Electronics.

6-slots
Plug in up to 6 GAMES or MEMORY PACKS then Switch Select each separately or in combination

RESET



Plug in up to 40K RAM and all other PACKS that are available. (Can be daisy chained.)

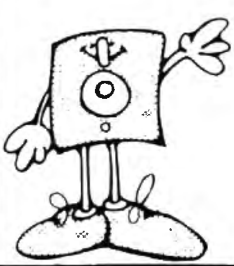
- Memory Protect included
- Fully Buffered (prevent memory dropouts)
- Fuse Protection
- Large switches
- Rigid support
- Also other prod. avail.

IN STOCK Immediate delivery.
Send Money Order and we pay the shipping.
—ORDER TODAY—

(MC & VISA accepted)
Call: (415) 651-3160
PARSEC RESEARCH
Drawer 1766-C
Fremont, CA 94538
• Dealer inquiries invited •

® PARSEC RESEARCH (Established 1976)
* VIC 20 is a TM of Commodore

Verbatim
flexible disks



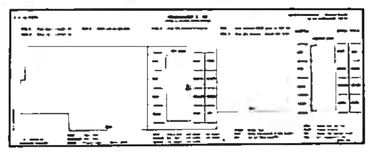
Call Free **(800) 235-4137** for prices and information. Dealer inquiries invited. C.O.D. and charge cards accepted.

PACIFIC EXCHANGES
100 Foothill Blvd
San Luis Obispo, CA
93401. In Cal. call
(800) 592-5935 or
(805) 543-1037

VISA

NEW

LEROY'S CHEATSHEET™ KEYBOARD OVERLAYS FOR YOUR VIC-20™



AT LAST! The information you need, at your fingertips, without going back to the manual. No need to memorize commands. These durable plastic coated overlays are designed for use by first time users, amateur enthusiasts, and trained professionals. Each overlay contains program starting locations, function key labeling, commands and warnings in bold letters, and additional helpful aids in center cutout.

DON'T DELAY - MAIL TODAY

Please send me the following Leroy's Cheatsheet™ keyboard overlays

- ☐ Programmer's Aid¹
- ☐ Vicmon¹
- ☐ Super Expander¹
- ☐ Vic Typewriter¹
- ☐ Victerm 1¹
- ☐ Graphic Printer (1515 & 1525)¹
- ☐ UMI Wordcraft 20²
- ☐ HES Writer³
- ☐ HES Vic Forth³

ONLY $3.95 EACH

Send check or money order plus $2.00 (postage and handling)
PA residents add 6% sales tax.

Name____________________

Address____________________

City__________ State__________ Zip__________

(1) Products of Commodore Business Machines Inc (2) Product of United Microware Industries, Inc
(3) Products of Human Engineered Software VIC-20 is a Trademark of Commodore Business Machines, Inc

CHEATSHEET PRODUCTS™ CP1083
P.O. Box 8299 Pittsburgh PA. 15218

Figure 1:
Car Loan Analysis – Monthly Payment

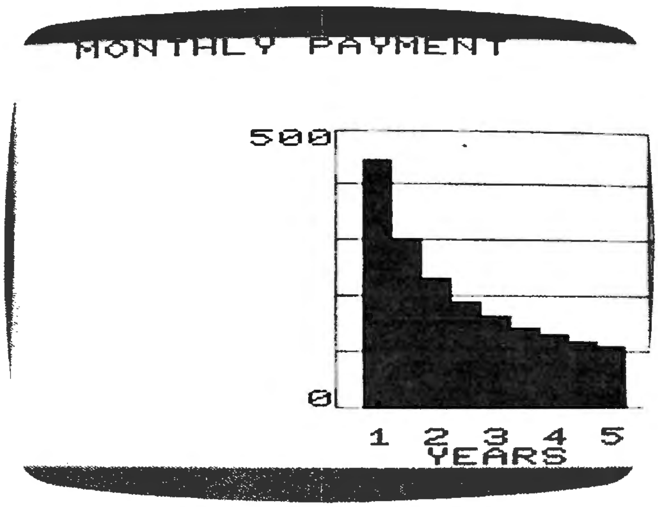


Figure 2:
Car Loan Analysis – Total Finance Cost

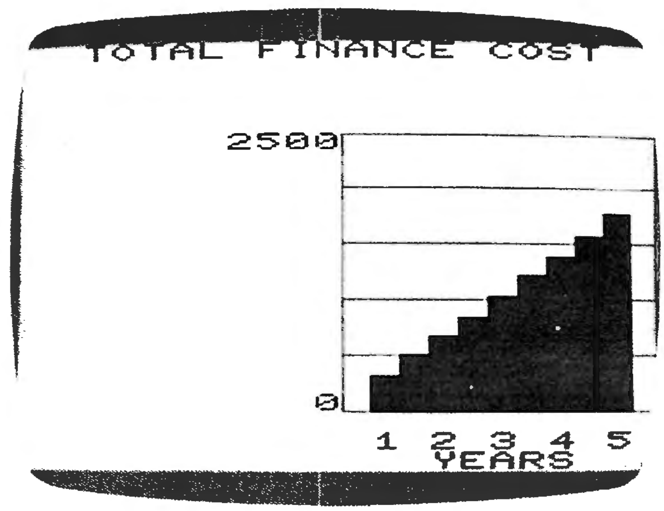


Figure 3: **Standard Scale**

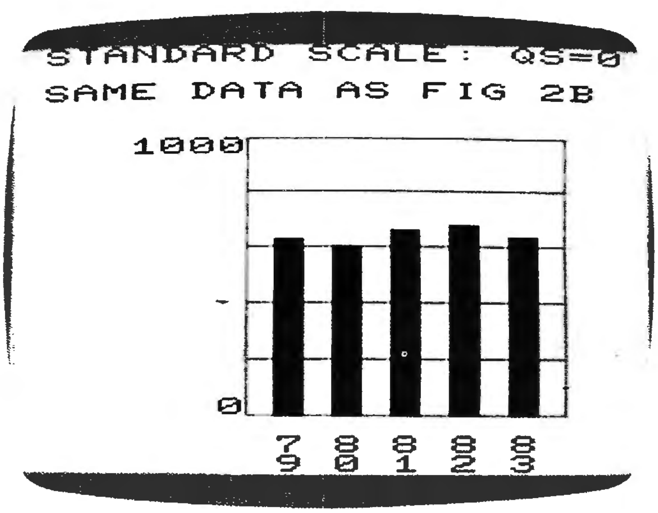


Figure 4: **Expanded Scale With Offset**

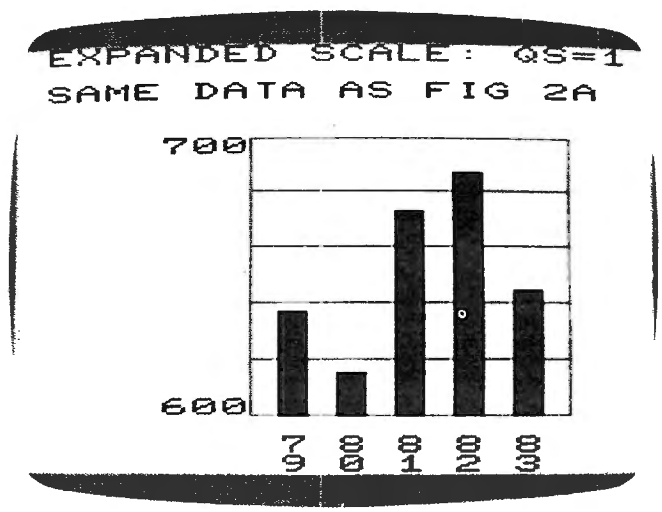


Figure 5:
Scale Including Both Positive And Negative Numbers

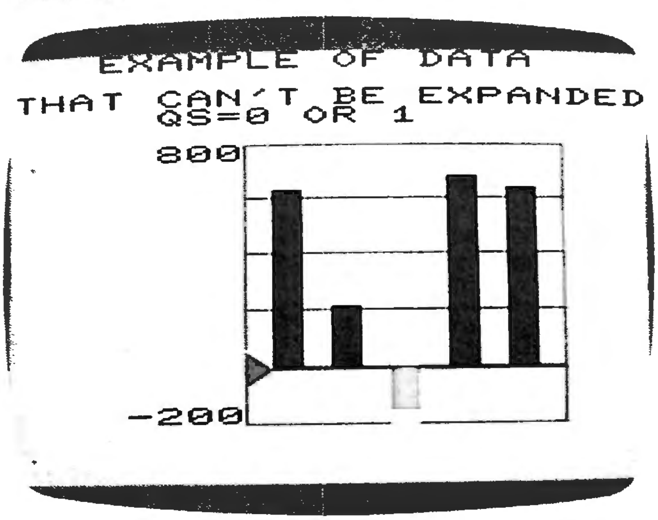
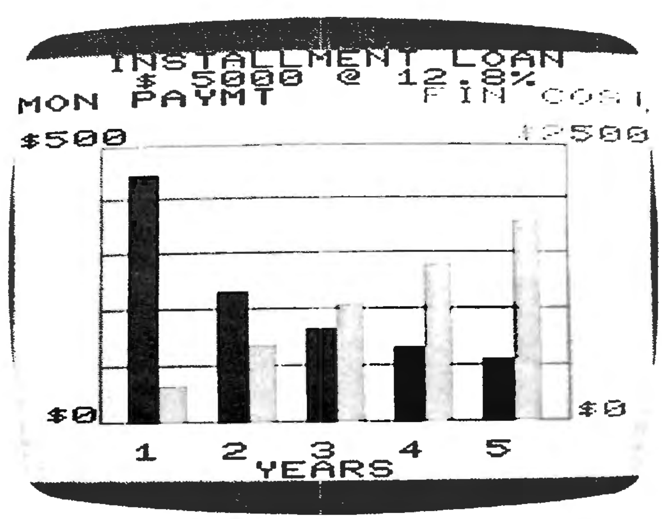
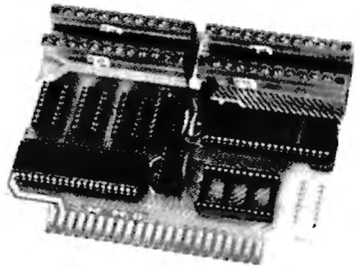


Figure 6: **Car Loan Analysis – Total Cost (Combines graphs from Figures 1 and 2.)**



NEW!
Universal Input/Output Board for VIC-20/64

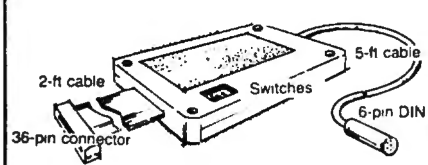


16-channel A/D converter & 1-channel D/A output; 16 high-current discrete outputs.

• University physics & electrical engineering labs and hospital applications.

VIC-20 uses MW-311V $205.00
CBM-64 uses MW-311C $225.00

MW-302: VIC-20/64 Parallel Printer Interface.



Works with all centronics type parallel matrix & letter printers and plotters—Epson, C.Itoh, Okidata, Nec, Gemini 10, TP-I Smith Corona, and most others. Hardware driven; works off the serial port. Quality construction: Steel DIN connectors & shielded cables. Has these switch selectable options: Device 4, 5, 6 or 7; ASCII or PET ASCII; 7-bit or 8-bit output; upper & lower case or upper only. Recommended by PROFESSIONAL SOFTWARE for WordPro 3 Plus for the 64, and by City Software for PaperClip.

MW-302 .... $119.95

Dealer inquiries invited.

VISA MasterCard

Micro World Electronix, Inc.
3333 S. Wadsworth Blvd. #C105,
Lakewood, CO 80227
(303) 987-9532 or 987-2671

FREE

WRITE FOR FREE CSI CATALOG OF VIC 20 and CMB 64 PRODUCTS

- SOFTWARE
- HARDWARE
- PROGRAMMING AIDS
- OPERATIONAL AIDS
- SUPPLIES
- MEDIA
- BOOKS

COMPATIBLE SYSTEMS INCORPORATED
P.O. Box 2070 • Dept. C
Saratoga, CA 95070
(408) 255-2024

64K MEMORY FOR THE *VIC 20?
YES, IT'S HERE!

LETCO

ANNOUNCES
THE ULTIMATE *VIC 20 MEMORY!

THE 64KV MEMORY EXPANSION MODULE W/24K OF NORMAL EXPANSION + 40K ADD'L FOR PROGRAM OR DATA STORAGE. BOOSTS MEMORY TO ALMOST 70,000 BYTES! THIS 40K IS PAGED IN OR OUT IN 8K BLOCKS WITH A SINGLE POKE INSTR. THE 64KV USES STATE OF THE ART DYNAMIC RAMS WITH TRANSPARENT REFRESH. ALL 8K BLOCKS ARE SWITCH SELECTABLE AND WRITE PROTECTABLE. THE ENTIRE UNIT DRAWS ONLY 250 MA. THE 64 KV HAS A GOLD PLATED EDGE CONNECTOR AND IS HOUSED IN A DISTINCTIVE BLACK CASE. PLUGS DIRECTLY INTO EXP PORT. ALL THIS AT A PRICE YOU CAN AFFORD.

ONLY $149.95
(SHIPPING AND HANDLING INCLUDED)
90 DAY LIMITED WARRANTY
OHIO RESIDENTS ADD 5% SALES TAX

LETCO
7310 WELLS RD
PLAIN CITY, OH. 43064 DEPT. CM
1-614-873-4410

NAME ____
ADDRESS ____
CITY, STATE ____
ZIP CODE ____

WE ACCEPT VISA OR MASTERCARD. PLEASE INCLUDE CARD NO. EXP. DATE AND SIGN

*VIC 20 IS A REG. TM. OF C.B.M. INC

VIController
HOME CONTROL SYSTEM FOR THE VIC 20 & CBM 64

- Control up to 256 lights & appliances
- 9 levels of brightness
- Manual & time control software
- Uses BSR remote switches
- Plugs into user port

ONLY $59.95

EXTENDED TIME CONTROL PROGRAM FOR CBM 64 . . . $9.95

COMvoice
NEW SPEECH SYNTHESIZER FOR VIC 20 & CBM 64

- Unlimited Vocabulary
- Automatic English to Speech Conversion
- New BASIC Command "SPEAK"
- Will Speak String Variables
- Speaks with Four Levels of Inflection
- Plugs into Expansion Port
- Built in Audio Jack

ONLY $149.95

AS EASY TO USE AS A PRINT STATEMENT!

Call (215) 861-0850 to order
MasterCard or Visa Accepted
1444 Linden Street, Bethlehem, PA 18018

GENESIS COMPUTER CORPORATION

Automatic Ranging And Scaling Rules

The bar graph subroutine follows a set of automatic ranging and scaling rules in the process of making the graph. The entire scale consists of six lines outlining five intervals. The value represented by one interval is called the scale factor. For example, in Figure 4 there are six lines representing 600, 620, 640, 660, 680, and 700. Each interval between lines represents 20, which is the scale factor.

These are the rules the subroutine follows to decide on the scale factor and the values for the scale lines:

1. The maximum value, QX, is the most positive, or failing that, the least negative value in the list of bars. Likewise, the minimum value, QN, is the most negative, or, failing that, the least positive.

2. The scale factor will be 1, 2, or 5 times 10^n where *n* is a positive or negative integer, or 0. (If *n* is 0, 10^n is 1.) Typical scale factors would be 5, .02, 1000, 1E-6.

3. The values of the scale lines must be multiples of the scale factor, or zero.

4. The scale factor will be the *smallest* number possible, so that the full scale will include the maximum and minimum values, QX and QN. For the standard scale option, QS=0 or 3, the full scale must include zero. Therefore, QX and QN are given starting values of zero before scanning the list of Y values.

5. For the QS=1 or 4 option, QX and QN will be given a starting value equal to Y(Z,0) before the maximum/minimum scan. This will result in an expanded scale, with three exceptions:

- If all Y values are identical, an expanded scale is possible, but the automatic rules can't decide on one.
- If the Y list contains both positive and negative values, the scale cannot be expanded.
- If the maximum and minimum values are already spread out over the normal scale, the scale cannot be expanded.

In these cases, the subroutine will work as if QS=0 or 3.

6. If QS=2 or 5, the values of QX and QN at the time of GOSUB9900 are carried into the subroutine, and the rules are followed on that basis. This allows you to choose your own scale factor or offset. However, if the data in the list won't fit your scale, the scale will automatically be changed to fit the data.

final example, which generated Figure 6, left 205 bytes free after RUNning. To keep the subroutine memory size to 2300 bytes, be sure to omit all REMarks and spaces (except inside quotes) when typing the program.

Program 1:
Car Loan Application For Bar Graph

```
90 DIMY(2,9)
100 INPUT"{CLR}AMNT,APR";AM,AR
110 MR=AR/1200
120 FORI=0TO8:N=6*(I+2):REM N=12 TO 60 BY
     6'S
130 Y(0,I)=AM*MR/(1-(1+MR)↑-N)
140 Y(1,I)=N*Y(0,I)-AM
150 NEXTI:QB=8
160 QS=3:Z=0:GOSUB9900:U0=QU:Z0=QZ
170 Z=1:GOSUB9900:U1=QU:Z1=QZ
180 FORI=0TO4:Y(2,2*I)=Y(0,2*I)/Z0
190 Y(2,2*I+1)=Y(1,2*I)/Z1:NEXT
200 Z=2:QB=-9:Q1=-4:QS=-.1:QC=3
210 T1$="{3 SPACES}INSTALLMENT LOAN
     {3 SPACES}"
220 QT$=T1$+"{4 SPACES}$"+STR$(AM)+"@"+ST
     R$(AR)+"%"
230 QL$="{4 SPACES}1{2 SPACES}2{2 SPACES}
     3{2 SPACES}4{2 SPACES}5{13 SPACES}YEA
     RS"
240 GOSUB9900:GOTO100
800 T2$="{BLK}MON PAYMT{5 SPACES}{PUR}FIN
     COST"
810 T3$="{BLK}$"+MID$(STR$(U0),2):T4$="
     {PUR}$"+MID$(STR$(U1),2)
820 PRINT"{HOME}"SPC(44)T2$SPC(22)T3$TAB(
     23-LEN(T4$))T4$SPC(255)SPC(54)"{BLK}$
     0"SPC(16)"{PUR}$0{BLU}"
830 RETURN
840 REM ADD LINE 9988 AFTER PROGRAM 2 IS
     {SPACE}IN MEMORY!
9988 GOSUB800
```

Program 2: Bar Graph Subroutine

```
9899 REM HI-PRECISION NORMALIZE
9900 DEF FNQ(X)=SGN(X)*INT(ABS(X)/Q9*(1E5
     +1E-3))/1E5
9901 REM COLOR AND QB CONTROL
9902 Q2=-(Q1<0)OR-(Q0<0):Q5=1:IFQB<0THENQ
     5=2-.5*Q2
9903 Q8=INT(20/Q5):IFABS(QB)<Q8THENQ8=ABS
     (QB)
9904 REM MAX/MIN SCAN
9905 QD=0:IF(ABS(QS)=1ORABS(QS)=4)THENQX=
     Y(Z,0):QN=Y(Z,0)
9906 IF(ABS(QS)<1ORABS(QS)=3ORQD=9)THENQX
     =0:QN=0
9908 FORQ=0TOQ8:IFY(Z,Q)>QXTHENQX=Y(Z,Q)
9910 IFY(Z,Q)<QNTHENQN=Y(Z,Q)
9912 NEXT
9913 REM AUTO RANGE/SCALE
9914 QF=QX-QN:IFQF>0THEN9918
```

```
9915 IFQX=ØTHENQU=5:QD=Ø:QR=Ø:QZ=1:GOTO99
     42
9916 QD=9:GOTO9906
9918 QL=INT(LOG(QF)/LOG(1Ø)):GOSUB9996:QP
     =FNQ(QF):QY=.5:IFQP>2.5THENQY=1
992Ø IFQP>5THENQY=2
9922 IFQP=1THENQY=.2
9924 GOSUB9996:Q3=FNQ(QX):Q4=FNQ(QN)
9925 QU=-INT(-Q3/QY):QD=INT(Q4/QY)
9926 IFQD>=ØTHENQR=Ø:QU=QD+5:GOTO9934
9928 IFQU<=ØTHENQR=5:QD=QU-5:GOTO9934
993Ø IFQ3+Q4>ØTHENQU=QD+5:QR=-QD:GOTO9934
9932 QD=QU-5:QR=-QD
9934 IFQ3/QY<=QUANDQ4/QY>=QDTHEN9938
9935 QY=QY*2+3.5*(QY=2)-.1*(QY=.2):IFQY=.
     5ANDQP>1THENQL=QL+1
9936 GOTO9924
9938 QZ=QY*Q9:QS$=""
994Ø QO=24*QD*(QD>Ø)-24*QU*(QU<Ø):QU=QU*Q
     Z:QD=QD*QZ
9941 IFQS=>ØTHENQS$=MID$(STR$(QU),1-(QU=>
     Ø))
9942 IFABS(QS)>2THENRETURN:REM{4 SPACES}I
     FTRUE, NO PLOT
9943 REM COLOR FUNCTIONS
9944 DEF FNQT(X)=ABS(Q1)*(Q2ANDXAND1)-ABS
     (QØ)*NOT-(Q2ANDXAND1)
9945 DEF FNQB(X)=ABS(QØ)*(Q2ANDNOT(XAND1)
     )-ABS(Q1)*NOT-(Q2ANDNOT(XAND1))
9946 PRINT"{CLR}";QT$;"{HOME}";:Q3=INT(Q5
     *Q8):IFQ3>19THENQ3=19
9947 REM BUILD SCALE LINES
9948 QA$="O":QB$="L":FORQ=ØTOQ3:QA$=QA$+"
     {T}":QB$=QB$+"{@}":NEXT:Q3=Q:QW=
     2Ø-Q3-ABS(QC)
995Ø QW=-QW*(QW>Ø):QL=2Ø-QW-Q3:QA$=QA$+"P
     ":QB$=QB$+"@":Q4=QW-LEN(QS$):IFQ4>=Ø
     THEN9954
9951 REM PRINT SCALE & LABELS
9952 PRINTSPC(88)QS$:PRINTSPC(QW):QF=PEEK
     (646):GOTO9955
9954 PRINTSPC(11Ø+Q4)QS$;:QF=PEEK(646)
9955 PRINT"{GRN}"QA$SPC(QL):FORQ=ØTO12:PR
     INTSPC(QW)"{G}"SPC(Q3)"{M}"SPC(Q
     L):NEXT
9956 PRINT"{HOME}"SPC(154):FORQ=4TO1STEP-
     1:IFQR>QTHENPRINTSPC(22):GOTO996Ø
9958 PRINTSPC(QW)QB$SPC(QL):IFQR<QTHENPRI
     NTSPC(44):GOTO9962
996Ø PRINTSPC(QW)QA$SPC(22+QL)
9962 NEXT:IFQS=>ØTHENQS$=MID$(STR$(QD),1-
     (QD=>Ø))
9964 PRINTSPC(QW)QB$;:POKE646,QF:PRINTSPC
     (QL):Q4=QW-LEN(QS$):IFQ4<ØTHENPRINTQ
     S$:GOTO9967
9965 PRINT"{UP}"SPC(Q4)QS$SPC(Q3+24+QL)
9966 REM Q3=START SCREEN RAM{12 SPACES}Q4
     =COLOR RAM ADDER
9967 Q3=256*PEEK(648):Q4=37888+4*(PEEK(36
     866)AND128)-Q3
9968 REM ZERO MARKER
9969 IFQR>ØANDQR<5THENQP=Q3+418+QW-66*QR:
     POKEQP,223:POKEQP+22,105
997Ø REM COMPUTE BAR LENGTH IN PIXELS; CO
     MPUTE POKE ORIGIN
9971 FORQ=ØTOQ8:QY=INT(24*ABS(Y(Z,Q))/QZ)
     +QO
9972 QF=INT(QY/8):QL=QY-8*QF:IFQY=ØTHENQL
     =1
9973 QP=Q3+419+QW-66*QR+INT(Q*Q5):IF(Y(Z,
     Q)<ØORQR=5)THEN9982
9974 REM POKE "UP" BARS
9975 IFQF>ØTHENFORQP=QPTOQP-22*(QF-1)STEP
     -22:POKEQP,224:POKEQP+Q4,FNQT(Q):NEX
     T
9976 REM Q%=CODES FOR UP SEGMNTS
9977 Q%(1)=1ØØ:Q%(2)=111:Q%(3)=121:Q%(4)=
     98:Q%(5)=248:Q%(6)=247:Q%(7)=227
9978 IFQL>ØTHENPOKEQP,Q%(QL):POKEQP+Q4,FN
     QT(Q)
998Ø GOTO9987
9981 REM POKE "DOWN" BARS
9982 QP=QP+22:IFQF>ØTHENFORQP=QPTOQP+22*(
     QF-1)STEP22:POKEQP,224:POKEQP+Q4,FNQ
     B(Q):NEXT
9983 REM Q%=CODES FOR DWN SEGMNTS
9984 Q%(1)=99:Q%(2)=119:Q%(3)=12Ø:Q%(4)=2
     26:Q%(5)=249:Q%(6)=239:Q%(7)=228
9985 IFQL>ØTHENPOKEQP,Q%(QL):POKEQP+Q4,FN
     QB(Q)
9986 REM END BAR LOOP;PRINT LEGEND
9987 NEXT:PRINTQL$TAB(43)
9988 REM GOSUB OVERPRINT HERE,{9 SPACES}I
     F DESIRED
9989 REM HOLD DISPLAY; TURN ON CURSOR
999Ø PRINT"{HOME}"SPC(255)SPC(25Ø):POKE2Ø
     4,Ø
9992 GETQ$:IFQ$=""THEN9992
9994 POKE2Ø7,Ø:POKE2Ø4,1:RETURN
9995 REM HI-PRECISION EXPONENT SUBROUTINE
9996 Q9=1:ONSGN(QL)+1GOTO9999,9998
9997 FORQ=1TO-QL:Q9=Q9/1Ø:NEXT:RETURN
9998 FORQ=1TOQL:Q9=Q9*1Ø:NEXT
9999 RETURN
```

©

64 CHARACTER CREATOR

Al J. Ockert

This menu-driven utility makes it easy to create, modify, and save custom characters. Also, the program calculates DATA statements for your new characters and allows a review of your creation in any 64 color.

This character generator program moves the complete character set (256 characters) into RAM and sets up the screen so you can easily redefine characters. Up to 128 characters can be redefined at any one time. However, the reversed (inverse video) characters cannot be redefined without modifying the program.

An 8x8 grid will be drawn on the screen. By using the cursor keys, you can draw any character you wish to create.

With single-key commands, all of which are listed next to the grid on screen, characters can be manipulated for a variety of purposes. When you have the character the way you want it, the D command will display the DATA statement containing your new character.

After you have redefined as many characters as you wish (up to 128), you can save them in a data file by using the S command. Preceding all the character data will be the total number of characters saved in that data file. And preceding each character's data will be the key code of that character.

The next time you run the program, you can load all the characters saved in a data file by using the L command. The program is set up for tape files. If you are a disk drive owner, simply substitute the following lines:

```
640 IF A$="L"THEN OPEN1,8,0,N$:GOTO900
645 OPEN1,8,1,N$:GOTO800
```

Easy Single-Key Access

When the program is first run after the ROM character set is transferred to RAM and the screen set up, the program initializes the C (select a character) command. If you want to load previously defined characters first, select any character to get back to the main menu. Then select the L (load) command.

While running the program, you can view the character that you are creating in the Multi-Color Mode (MCM) by selecting the M (Multi-Color ON) command. All the characters displayed on the screen will be in multicolor. However, only the character that you are creating, which is displayed at screen location 1731, will change when you change the color with F7. This is because only Color Memory location 56003, which contains the color information for screen location 1731, is changed with the F7 function key.

Also, while in MCM, the redefined character has to have a color between 8 and 15 in Color RAM (CRAM) to be in MCM. That is, if the color code is between 8 and 15 (orange through gray 3) in the CRAM, then the bit pairs will change colors as the background colors BKG #0 through BKG #2 are changed. However, if the color code in CRAM is from 0 to 7 (black through yellow), then the redefined character will be displayed only in that color.

You can also see what different color combinations look like by changing the screen background color BKG #0 using the F1 function key and changing the character color by using the F7 function key. When the three background colors are changed by using F1, F3, and F5, respectively, all the screen characters change.

The bit pair/color register format is displayed in the upper-right corner of the screen. This is to assist you in determining what color the bit pair will have while in Multi-Color Mode. If both pixels of the bit pair are off (00), the screen (BKG #0) register color will be displayed there.

If only the second pixel of the bit pair is on (01), background #1 (BKG #1) register color will be displayed for the bit pair.

If only the first pixel of the bit pair is on (10), background #2 (BKG #2) register color will be displayed for the bit pair.

If both pixels of the pair are on (11), the Color Memory (CRAM) color will be displayed for the bit pair — but only in screen location 1731.

After you have created a few characters, by using the left arrow (←) command you can break from the program to see what your characters look like on the screen in different combinations. The screen can be cleared and your characters typed onto the screen. In this mode the program can even be listed. However, if you modify the program now, you will lose your redefined character data in the arrays since all variables are zeroed when you modify or edit a program.

If you stay in the immediate mode, you will not lose your characters, and you can return to the main program by typing in GOTO 12 and RETURN.

64 Character Creator

```
5 POKE52,56:POKE56,56:CLR:PRINTCHR$(142)
10 DIMA(8,8),D%(127,8),C$(15),K%(127):L=0
   :CH=0:F=0
12 X=0:Y=0:R=0:C=O:CM=14336:RA=56334:V=53
   248:S=1367
15 IFF=0THENPRINT"{CLR}{DOWN}IT WILL TAKE
    ABOUT 40 SECONDS TO SET UP"
20 RESTORE:FORX=0TO15:READC$(X):NEXT
21 FORR=1TO8:FORC=1TO8:A(R,C)=46:NEXTC,R
25 C0=PEEK(V+33)AND15:C1=PEEK(V+34)AND15
27 CL=PEEK(V+2755)AND15:C2=PEEK(V+35)AND1
   5
29 IFL>0ORF=1THEN39
31 POKERA,(PEEK(RA)AND254):POKE1,PEEK(1)A
   ND251
34 FORI=0TO255:FORJ=0TO7:POKECM+I*8+J,PEE
   K(V+I*8+J):NEXTJ,I
36 POKE1,PEEK(1)OR4:POKERA,PEEK(RA)OR1
37 POKEV+24,(PEEK(V+24)AND240)OR14
39 PRINT"{CLR}"SPC(20)"IN MULTI-COLOR MOD
   E"
40 PRINT" CHARACTER GENERATOR{2 SPACES}BI
   T PAIR/COLOR REG"
48 PRINT"{UP} *{2 SPACES}TURN ON PIXEL
   {7 SPACES}00{7 SPACES}BKG#0"
50 PRINT" -{2 SPACES}ERASE PIXEL
   {9 SPACES}01{7 SPACES}BKG#1"
51 PRINT" F1 BKG#0-"C$(C0)TAB(24)"10
   {7 SPACES}BKG#2"
52 PRINT" F3 BKG#1-"C$(C1)TAB(24)"11
   {7 SPACES}CRAM "
54 PRINT" F5 BKG#2-"C$(C2)
56 PRINT" F7 CRAM -"C$(CL)
58 PRINT" ={2 SPACES}COMPUTE CHARACTER"
60 PRINT" D{2 SPACES}DATA"
61 PRINT" O{2 SPACES}MULTI-COLOR OFF"
62 PRINT" N{2 SPACES}NEW SCREEN"
63 PRINT" M{2 SPACES}MULTI-COLOR ON"
64 PRINT" S{2 SPACES}SAVE CHARACTERS"
66 PRINT" L{2 SPACES}LOAD CHARACTERS"
68 PRINT" C{2 SPACES}NEW CHARACTER"
69 PRINT" ←{2 SPACES}BREAK PROGRAM"
70 PRINT" Q{2 SPACES}QUIT"
80 POKES-40+1,76:POKES-40+2,122:POKES-40+
   3,76:POKES-40+4,122:POKES-40+5,76:POKE
   S-40+6,122
81 POKES-40+7,76:POKES-40+8,122
82 FORR=0TO7:POKES+C+R*40,84:NEXT
84 FORC=C-1TO1STEP-1:POKES+C+R*40,69:NEXT
86 FORR=R-1TO0STEP-1:POKES+C+R*40,89:NEXT
90 Y=0:FORR=1TO8:FORC=1TO8:Y=Y+1:POKES+Y,
   A(R,C):NEXT:Y=Y+32:NEXT
95 Y=1:X=1:GOTO300
100 GETA$:IFA$=""THEN100
110 R=S+X+(Y-1)*40:C=A(Y,X):POKER,C
120 IFA$="{DOWN}"THENY=Y+1:IFY>8THENY=1
130 IFA$="{UP}"THENY=Y-1:IFY<1THENY=8
140 IFA$="{RIGHT}"THENX=X+1:IFX>8THENX=1
150 IFA$="{LEFT}"THENX=X-1:IFX<1THENX=8
160 IFA$="-"THENA(Y,X)=46
170 IFA$="*"THENR=42:A(Y,X)=R
180 IFA$="="THEN400
190 IFA$="D"THEN500
200 IFA$="N"ORA$="Q"ORA$="←"ORA$="L"ORA$
    ="S"THEN600
205 IFA$="M"THEN580
207 IFA$="O"THEN590
210 IFA$="C"THEN700
220 IFA$="{F1}"THENR=33:GOTO520
230 IFA$="{F3}"THENR=34:GOTO520
240 IFA$="{F5}"THENR=35:GOTO520
250 IFA$="{F7}"THENR=2755:GOTO520
300 R=S+X+(Y-1)*40:C=A(Y,X)+128:POKER,C:I
    FF=0THEN700
305 GOTO100
400 FORY=0TO8:D%(L,Y)=0:NEXT:CH=1
401 FORY=1TO8:FORX=1TO8:PIX=A(Y,X)-42:IFP
    IX>0THENPIX=0:GOTO420
410 PIX=1
420 D%(L,Y)=D%(L,Y)+PIX*2↑(8-X):NEXTX,Y
430 FORY=0TO7:POKECM+8*K%(L)+Y,D%(L,Y+1):
    POKE1731,K%(L):NEXT:GOTO95
450 PRINT"{HOME}{20 DOWN}{RVS}"A$": YES O
    R NO {OFF}{22 SPACES}"
460 FORX=1TO10:GETN$:NEXT
470 GETN$:IFN$=""THEN470
480 PRINT"{UP}{16 SPACES}{UP}":IFA$="N"OR
    A$="Q"ORA$="←"ORA$="L"ORA$="S"THENRE
    TURN
481 IFN$="Y"THEN12
482 IFN$="N"THENPRINT"{DOWN}TO RETURN TO
    {SPACE}THE PROGRAM TYPE GOTO 12":STOP
490 PRINT"{RVS}{HOME}{6 DOWN}CONTINUE":GO
    TO460
500 PRINT"{CLR}{DOWN}{3 RIGHT}DATA";:FORY
    =1TO8:PRINTD%(L,Y)"{LEFT},";:NEXT:PRI
    NT"{LEFT} "
510 PRINT"{DOWN}FOR KEY CODE="K%(L):GOTO4
    90
520 C=PEEK(V+R)AND15:C=C+1:IFC>15THENC=0
530 POKEV+R,C:PRINT"{HOME}{4 DOWN}";:IFR=
    33THENC0=C:GOTO570
540 PRINT"{DOWN}";:IFR=34THENC1=C:GOTO570
550 PRINT"{DOWN}";:IFR=35THENC2=C:GOTO570
560 PRINT"{DOWN}";:CL=C
570 PRINT"{10 RIGHT}"C$(C)"{2 SPACES}":GO
    TO300
580 POKEV+22,PEEK(V+22)OR16:GOTO300
590 POKEV+22,PEEK(V+22)AND 239:GOTO300
600 GOSUB450:IFN$<>"Y"THEN95
610 GETN$:GETN$:IFA$="N"THEN12
620 IFA$="Q"THENPOKEV+24,21:POKEV+22,PEEK
    (V+22)AND239:END
625 IFA$="←"THENPRINT"{DOWN}TO RETURN TO
     THE PROGRAM TYPE GOTO 12":STOP
630 PRINT"{CLR}":INPUT"NAME OF CHARACTER
    {SPACE}FILE";N$:PRINT
640 IFA$="L"THENOPEN1,1,0,N$:GOTO900
645 OPEN1,1,1,N$:GOTO800
700 PRINT"{HOME}{20 DOWN}{RVS}PRESS KEY T
    HAT YOU WANT TO CHANGE{OFF}":F=1
705 IFCH=1THENL=L+1:CH=0
710 GETA$:IFA$=""THEN710
720 POKE16385,ASC(A$):K%(L)=PEEK(16385)AN
    D191:IFK%(L)>64THENK%(L)=K%(L)-64
722 POKE1731,K%(L)
725 PRINT"{2 UP}"SPC(24)"KEY CODE="K%(L)"
    {LEFT}{2 SPACES}"
730 PRINT"{33 SPACES}":IFL=0THEN100
735 FORY=0TO8:FORX=0TO8:A(Y,X)=46:NEXTX,Y
    :GOTO90
800 PRINT#1,L
802 FORY=0TOL:PRINT#1,K%(Y):FORX=1TO8:PRI
    NT#1,D%(Y,X):NEXTX,Y:CLOSE1:GOTO12
900 INPUT#1,L:FORY=0TOL:INPUT#1,K%(Y):FOR
    X=1TO8:INPUT#1,D%(Y,X):NEXTX,Y:CLOSE1
910 PRINT"COMPUTING CHARACTERS"
920 FORY=0TOL:FORX=1TO8:POKECM+8*K%(Y)+(X
    -1),D%(Y,X):NEXTX,Y:GOTO12
1000 DATABLACK,WHITE,RED,CYAN,PURPLE,GREE
     N,BLUE,YELLOW
1010 DATAORANGE,BROWN,LT RED,GREY1,GREY2,
     LT GREEN,LT BLUE,GREY3
```

©

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Retirement Planning

In Craig Cole's suggested improvements to the Atari version of Retirement Planner (COMPUTE!, April 1983) which appeared in the July "Readers' Feedback" section (p. 16), line 70 should have read:

```
70 Y=Y+1
```

TI Goblin

As this game from the July issue (p. 72) was presented, your goblin could not reach the faces in the rightmost column of the screen. This can be corrected by changing line 760 to read:

```
760 COL=COL+SGN(31-COL)
```

To make the game display the proper high score, delete lines 270, 280, and 290, and add the following line:

```
1125 IF S>HS THEN 1130 ELSE 1140
```

Thanks to Canadian reader Luc Cousineau and others who pointed out this problem.

Fortress Of Adnil

In the program for this Timex/Sinclair game (July 1983, p. 92), the GOTO statements were missing from the following two lines:

```
8040 IF INKEY$="" THEN GOTO 8020
8209 IF INKEY$="" THEN GOTO 8207
```

Roadblock

Readers who have had trouble typing in this long machine language program for the Atari may want to add the following lines to the BASIC loader (July 1983, p. 108) to help check for typing errors in their DATA statements:

```
15 LN=1010
20 FOR L=0 TO 16
25 FOR C=0 TO 69
30 D=PEEK(L*70+13824+C):IF D>15010 T
   HEN 40
35 T=T+D:NEXT C
40 READ CK:IF CK<>T THEN PRINT "ERRO
   R IN LINES ";LN;"-";LN+90:STOP
45 PRINT "LINES ";LN;"-";LN+90;" OK"
50 T=0:LN=LN+100:NEXT L
55 A=USR(14788)
3000 DATA 3797,8372,7521,6620,6660,8
     073
3010 DATA 7319,7083,6853,6626,6631,8
     059
3020 DATA 7685,7702,7951,7720,7014
```

Circles

For the machine language circle-drawing routine presented in this Atari graphics article from the July issue to work properly, the following lines must be added to Program 7 (p. 168), the BASIC loader for the routine:

```
28004 RESTORE 29500
28005 FOR I=1577 TO 1584:READ A:POKE
       I,A:NEXT I
29500 DATA 128,64,32,16,8,4,2,1
```

Timex/Sinclair Screenscrolls

Reader Daniel Froats notes that the screen scrolling programs from the July issue (p. 216) will work with the basic 2K of memory if the following two lines are entered before running the program:

```
POKE 16389,n
CLS
```

Replace n with any number greater than 90.

Commodore 64 Video Tour

Jim Butterfield writes that readers whose 64s have the newer ROM sets may experience problems with the BASIC programs from Parts VI (July 1983 p. 218) and VII (August 1983, p. 182) of his series on the 64's video capabilities. This is because the ROM routines leave the raster interrupt in a nonstandard state. To make the programs work, add the following line:

```
90 POKE 53265,27
```

VIC Bitmapping

Line 100 of Program 1 from this article on VIC high resolution graphics (July 1983, p. 248) should read:

```
100 POKEV+15,17*T-9:FORI=.TO255:POKEW+I,I
    :POKER+I,C-1:NEXT:FORI=.TOL*2*↑STEPB
```

The "shifted up-arrow" key combination on the VIC gives the symbol for pi, which acts as a constant with the value of pi when used in calculations.

First Math

A bug in the VIC, 64, and Apple versions of this educational game from the August issue (p. 92) sometimes causes a "division by zero" error when playing in the addition, subtraction, or multiplication modes. To prevent this, add the following line:

VIC or 64 Version (Programs 1 or 2)

```
123 IF A$<>CHR$(47) THEN 140
```

Apple Version (Program 5)

```
395 IF A$ <> ("/") THEN 430
```

©

A Beginner's Guide To Typing In Programs

What Is A Program?

A computer cannot perform any task by itself. Like a car without gas, a computer has *potential*, but without a program, it isn't going anywhere. Most of the programs published in **COMPUTE!** are written in a computer language called BASIC. BASIC is easy to learn and is built into most computers (on some computers, you have to purchase an optional BASIC cartridge).

BASIC Programs

Each month, **COMPUTE!** publishes programs for many machines. To start out, type in only programs written for your machine, e.g., "TI Version" if you have a TI-99/4. Later, when you gain experience with your computer's BASIC, you can try typing in and converting certain programs from one computer to yours.

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one "right way" of stating something. Every letter, character, or number is significant. A common mistake is substituting a letter such as "O" for the numeral "0", a lowercase "l" for the numeral "1", or an uppercase "B" for the numeral "8". Also, you must enter all punctuation such as colons and commas just as they appear in the magazine. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Brackets And Special Characters

The exception to this typing rule is when you see the curved bracket, such as "{DOWN}". Anything within a set of brackets is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to the appropriate key for your computer. For example, if you have an Atari, refer to the "Atari" section in "How to Type COMPUTE!'s Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could "lock up," or "crash." The keyboard, break key, and RESET (or STOP) keys may all seem "dead," and the screen may go blank. Don't panic – no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, so always SAVE a copy of your program before you RUN it. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get To Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter inverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1) Type in the program a line at a time, in order. Press RETURN or ENTER at the end of each line. Use backspace or the back arrow to correct mistakes.

2) Check the line you've typed against the line in the magazine. You can check the entire program again if you get an error when you RUN the program.

3) Make sure you've entered statements in brackets as the appropriate control key (see "How To Type COMPUTE!'s Programs" elsewhere in the magazine.)

We regret that we are no longer able to respond to individual inquiries about programs, products, or services appearing in **COMPUTE!** *due to increasing publication activity. On those infrequent occasions when a published program contains a typo, the correction will appear on the CAPUTE! page, usually within eight weeks. If you have specific questions about items or programs which you've seen in* **COMPUTE!**, *please send them to Readers Feedback, P.O. Box 5406, Greensboro, NC 27403.* ©

COMMODORE 64
PROFESSIONAL BUSINESS PROGRAMS

- **INTRODUCTORY OFFER ½ PRICE!**
- **15 DAY FREE TRIAL!**
- **LIFETIME GUARANTEE!**
- **FILE GUARD (TM)**

PROFESSIONAL BUSINESS PROGRAMS! *Disk)*

You take no risk! We are so sure these professional business programs will meet and exceed your highest expectations, we are willing to allow you to try these programs at our expense! These are designed and produced by **Southern Solutions** who produces professional business programs for **Commodores** most expensive computers. These business program will convert your **Commodore 64** into a Professional Business Machine!!!

INTRODUCTORY OFFER! (Expires 12-25-83)

	List	½ Price
• General Ledger & Cash Flow	$119.00	$59.00
• Accounts Payable Plus Check Writing	$119.00	$59.00
• Accounts Receivable	$119.00	$59.00
• Payroll	$119.00	$59.00
• Inventory	$119.00	$59.00
• Data Base Manager	$119.00	$59.00

15 DAY FREE TRIAL!

We give you 15 days at your business for you to try out these programs! Should they not meet your requirements just send them back prepaid and we'll refund your purchase price!

LIFETIME GUARANTEE!

If a program fails due to faulty workmanship or material anytime you personally own and use the program we will replace it at no charge!

FILE GUARD (TM)

Prevents loss of data and confidential files due to power failure — a Southern Solutions exclusive!

PLUS: THESE PROFESSIONAL BUSINESS PROGRAMS -

- ***Script 64 - No. 1 Executive Word Processor - Disk*** ***List $99 - Sale $59***
- ***Complete Data Base - Tape-Disk*** ***List $89 - Sale $59***
- ***Electronic Spread Sheet (like Visicalc) - Tape-Disk*** ***List $89 - Sale $59***

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $3.00 for postage. Add $6.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

NEWS&PRODUCTS

80-Column Commodore 64 Word Processor

Mirage Concepts has produced an 80-column word processor for the Commodore 64. The machine language program produces the expanded display without hardware modifications.

The *Word Processor* allows text to be formatted on the screen exactly as it will appear on the printed page. Other features of the program include word wrap, search and replace, block functions, and more than 70 single-keystroke commands. The $99.95 word processor is designed to work with a variety of printers.

Another Mirage product for the 64 is *Database Manager*, which features free-form design and input. It includes the ability to sort any field and to calculate fields. It can handle up to 65,535 records per file, 2,000 characters per record, 200 fields per record, and 250 characters per field.

Database Manager sells for $99.95.

Word Processor *and* Database Manager *are available from Mirage Concepts for the Commodore 64.*

Mirage Concepts, Inc.
2519 W. Shaw, No. 106
Fresno, CA 93711
(209) 227-8369

64 Math Drill

The *Math Teacher* is a math tutorial program that drills students in the basics of addition, subtraction, multiplication, and division. The program, which is available from CompuTech, is designed for use with students from first grade through junior high school.

The game, which makes use of the Commodore 64's graphics capabilities, is designed to work with either a color or black-and-white monitor.

The *Math Teacher* is available on tape for $39.95.

CompuTech
P.O. Box 7000-309
Redondo Beach, CA 90277

Apple, Atari, 64 Games

Avalon Hill has introduced a handful of new games for the Apple, Atari, and Commodore 64 computers. Here is what the company has to offer.

Paris in Danger is a simulation of Napoleon's 1814 campaign in France. The player can assume the role of Napoleon, Commander Schwarzenberg, or can play both sides. The $35 game is available for 48K Atari 800 and 1200XL computers.

T.G.I.F. is a party game for up to four players. The game recreates a typical working-class workweek. Can you survive until payday? The game is available on cassette for the Commodore 64, and on cassette or disk for 40K Atari machines.

Parthian Kings is an Apple game for up to four players. Feudal civil war is coming and the struggle is on for who will rule the land. The winner will be the best strategist and leader.

S.C.I.M.M.A.R.'S, which stands for Surface Contra-Gravity Individual Manned Mobile Anti-Vehicle Raiders, is a game of gladiatorial armor combat set in the far future. The $30 game is available for Apple computers.

Computer Football Strategy is a detailed statistical treatment of 18 great teams. You have 20 offensive and 10 defensive plays to choose from as the animated action unfolds on the scrolling football field. The game is available on cassette for the Commodore 64 and Atari computers, and on disk for Atari. Price is $16 for tape, $21 for disk.

T.A.C., an acronym for Tactical Armor Command, is a simulation of armored combat during World War II. Five different scenarios are available in this $40 Apple game.

The Avalon Hill Game Company
4517 Harford Road
Baltimore, MD 21214
(301) 254-5300

Automodem With Parallel Printer Port

The AutoPrint Microconnection, an auto-answer, auto-dial modem for the VIC-20, Commodore 64, and Atari computers, has been introduced by Microperipheral.

The modem, which operates at 300 baud, includes a Centronics compatible parallel printer port.

The two-pound unit plugs directly into the computer. With the modem connected to the phone line, an attached printer will provide a hard copy of whatever appears on the screen.

The AutoPrint Microconnection is available for $149.95.

The Microperipheral Corp.
2565 152nd Ave. N.E.
Redmond, WA 98052
(206) 881-7544

Strategy Wargame

Combat Leader is a battle-of-the-tanks game that combines realistic rules, historical accuracy, and speed.

The game, available for Atari computers with a version forthcoming for the Commodore 64, gives you control of more than 70 tanks on a scrolling battlefield. Each tank is historically rated for armor thickness, strength, speed, and accuracy.

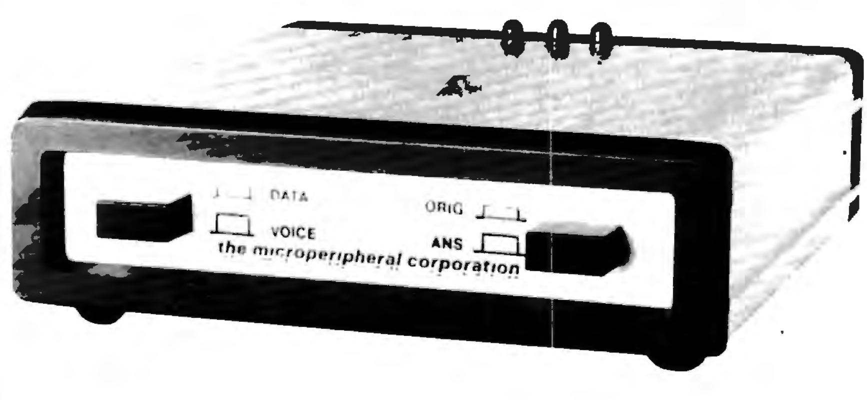
Combat Leader, which includes eight speed levels and a realtime option, sells for $39.95.

Strategic Simulations Inc.
883 Stierlin Road
Building A-200
Mountain View, CA 94043
(415) 964-1353

Internal VIC Expansion

Legend Valley Computer Systems has produced a 27K internal expansion board for the VIC-20.

The board fits under the keyboard and is piggybacked to the original 6502 microprocessor. The additional memory is selectable in 8K blocks via a switch on



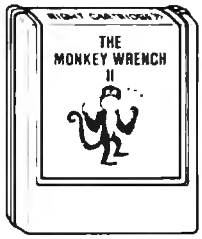
The AutoPrint Microconnection is a combination modem-parallel printer interface for the Atari, Commodore 64, or VIC-20.

Products for Commodore, Atari, Apple, and others!

NEW

THE MONKEY WRENCH II

A PROGRAMMERS AID FOR ATARI 800
NEW AND IMPROVED — 18 COMMANDS
PLUGS INTO RIGHT CARTRIDGE SLOT



$59.95

If you are a person who likes to monkey around with the ATARI 800, then THE MONKEY WRENCH II is for you!! Make your programming tasks easier, less time-consuming and more fun. Why spend extra hours working on a BASIC program when the MONKEY WRENCH can do it for you in seconds. It can also make backup copies of boot type cassette programs. Plugs into the right slot and works with ATARI BASIC cartridge.

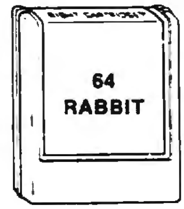
The MONKEY WRENCH provides 18 direct mode commands. They are: AUTO LINE NUMBERING — Provides new line numbers when entering BASIC program lines. RENUMBER — Renumbers BASIC's line numbers including internal references. DELETE LINE NUMBERS — Removes a range BASIC line numbers.

VARIABLES — Display all BASIC variables and their current value. Scrolling — Use the START & SELECT keys to display BASIC lines automatically. Scroll up or down BASIC program. FIND STRING — Find every occurrence of a string. XCHANGE STRING — Find every occurrence of a string and replace it with another string. MOVE LINES — Move lines from one part of program to another part of program. COPY LINES — Copy lines from one part of program to another part of program. FORMATTED LIST — Print BASIC program in special line format and automatic page numbering. DISK DIRECTORY — Display Disk Directory. CHANGE MARGINS — Provides the capability to easily change the screen margins. MEMORY TEST — Provides the capability to test RAM memory. CURSOR EXCHANGE — Allows usage of the cursor keys without holding down the CTRL key. UPPER CASE LOCK — Keeps the computer in the upper case character set. HEX CONVERSION — Converts a hexadecimal number to a decimal number. DECIMAL CONVERSION — Converts a decimal number to a hexadecimal number. MONITOR — Enter the machine language monitor.

In addition to the BASIC commands, the Monkey Wrench also contains a machine language monitor with 16 commands used to interact with the powerful features of the 6502 microprocessor.

NEW

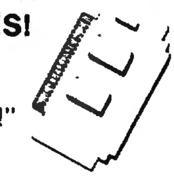
VIC RABBIT CARTRIDGE AND CBM 64 RABBIT CARTRIDGE



NEW FEATURE!
DATA FILES!

"High Speed Cassette Load and Save!"

$39.95
(includes cartridge and manual)



for VIC

Don't waste your Life away waiting to LOAD and SAVE programs on Cassette Deck.

Load or Save 8K in approximately 30 seconds! Try it—your Un-Rabbitized VIC or 64 takes almost 3 minutes. It's not only fast but VERY RELIABLE.

Almost as fast as 1541 Disk Drive! Don't be foolish — Why buy the disk when you can get the Rabbit for much, much less!

Allows one to APPEND Basic Programs!
Easy to install — just plugs in.
Expansion Connector on rear of the VIC Rabbit.
Works with or without Expansion Memory.
Works with VIC or 64 Cassette Deck.
12 Commands provide other neat features.
Fast Data Files - two data file modes.
Also Available for 2001, 4001, and 8032.

Now for the "64"

STCP — 300/1200 Baud

Standard Terminal Communications Package

`PFO` IOD OOA CP<D1>D2 BELL 12 30 00 10 14 36

Don't settle for non-standard Communications Protocol! Access Micro Net, Source, Bulletin Boards, Local Main-frame, etc.

- Complete Package — Includes RS232 Interface Board and software (does not include modem)
- Communicates in Industry Standard ASCII
- Upload/Download to/from Disk
- Automatic File Translation
- Can be controlled from keyboard or user supplied basic or machine language program

Specify 3.0 or 4.0 ROMS or 8032 Commodore Computer 4040 or 8050 or PEDISK II Disk or CBM64 on 1541

Price: $129.95

PET BASIC SCROLL PROGRAM

Scroll thru Basic Programs using cursor up/down keys. Specify computer. $6.00 on cassette, $9.00 on diskette.

65C02 MAE

Same as our MAE but enhanced for the new 65C02 Opcodes. Turns your computer into a development system for the new ROCKWELL 65C02 Microprocessor. $200.00 — Specify Computer.

6800 CROSS ASSEMBLER

A Cross Assembler based on the MAE that runs on the PET, Apple, or Atari but assembles opcodes for the Motorola 6800 microprocessor. Turns your computer into a development system for the Motorola 6800 Microprocessor. $200.00 — Specify Computer.

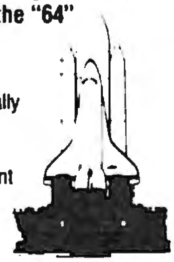
ATARI and VIC Cartridges

EHS can supply large quantities of ATARI and VIC Cartridges for software developers. If you need cartridges, call for pricing.

More than just an Assembler/Editor!
Now for the "64"

MAE

It's a Professionally Designed Software Development System



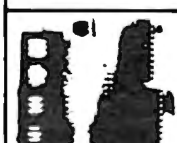
for
PET
APPLE
ATARI
~~$169.95~~
New Price
$99.95

Blast off with the software used on the space shuttle project!

- Designed to improve Programmer Productivity
- Similar syntax and commands — No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI
- Coresident Assembler/Editor — No need to load the Editor then the Assembler then the Editor, etc
- Also includes Word Processor, Relocating Loader, and much more
- Options: EPROM Programmer, unimplemented opcode circuitry
- STILL NOT CONVINCED: Send for free spec sheet!

ATARI AND PET EPROM PROGRAMMER

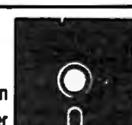
Programs 2716 and 2532 EPROMs. Includes hardware and software. PET = $75.00 — ATARI (includes sophisticated machine language monitor) = $119.95



TRAP 65

TRAP 65 is a hardware device that plugs into your 6502's socket. Prevents execution of unimplemented opcodes and provides capability to extend the machines' instruction set.
For PET/APPLE/SYM
Reduced from $149.95 to $69.95

5¼ INCH SOFT SECTORED DISKETTES



Highest quality. We use them on our PETs, APPLEs, ATARIs, and other computers. $22.50/10 or $44.50/20

Prowriter Printer - Excellent dot matrix print parallel = call
Serial = Call IEEE = Call

DC Hayes Smart Modem = $235.00
DC Hayes Micro Modem II = $289.00

Rana Disk Drive - 375
4 Drive Controller - 114

EPROMS 2716 = $4.50 2532 = $7.50
Over 40 Commodore Programs by Baker (on 4040) = $25.00

Eastern House

3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA
MasterCard

the rear of the VIC-20 cabinet.

The expansion board is available for $129.95.

Legend Valley Computer Systems
1474 Naughtingham
Newark, OH 43055

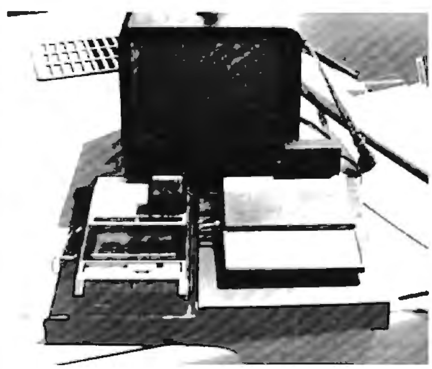
Traveling Timex

Car Comp is an accessory for the Timex/Sinclair computer that turns it into a traveling companion.

Car Comp is a rigid platform to which the computer and cassette recorder can be attached. It can be used anywhere with AC power or a 12-volt cigarette-lighter outlet.

Car Comp, which is produced by L&G Enterprises, sells for $59.95.

L&G Enterprises
Box 6854
Silver Spring, MD 20906



Car Comp secures a Timex/Sinclair and cassette deck as well as allowing power to be drawn from a standard auto cigarette-lighter outlet.

TI-99/4A Assortment

Western Properties Investment Company has produced a line of products designed for the TI-99/4A computer. The programs include a word processor, a data base, and a spreadsheet.

The word processor, *Printer Book*, is designed to handle up to two pages of text. Control of the printer is achieved through use of the CTRL key. The program can be used in conjunction with one of Western Properties' *File Book* programs to merge records from a data base with text. *Printer Book*'s 14 menu options include record merging, merge to screen or printer, and automatic multi-letter printing.

File-Book III handles up to 100 records with 6 items per record. The program includes full editing, search and sort capabilities as well as output to screen, printer or tape.

Income and Expense Spreadsheet IV is an accounting spreadsheet composed of 2 income and 50 expense categories. The program produces monthly charts of each of the 52 categories and an annual chart. Data is saved to tape with a cassette routine that is four times normal speed.

Printer Book and *File-Book III* are available for $39.95. *Income and Expense Spreadsheet* sells for $43.95. The programs run on Extended BASIC, but require no memory expansion. A printer is optional for the data base and spreadsheet programs.

Western Properties Investment Company
Software Division
P.O. Box 9602
Marina Del Rey, CA 90295

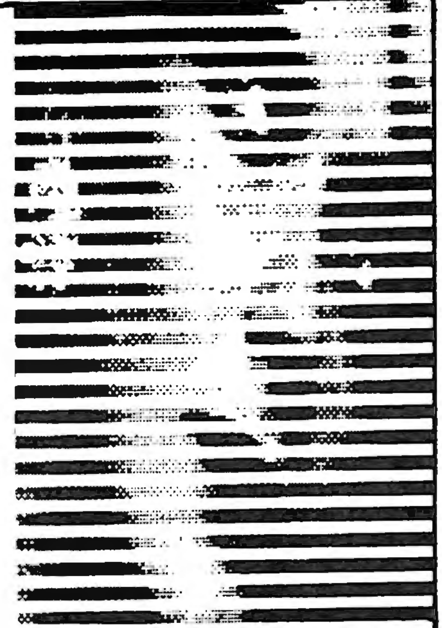
VCR Interface For VIC And 64

The Videobook Corporation has introduced Prometheus 1, an interface for the Commodore 64 and VIC-20 computers and home video cassette recorders.

Prometheus 1 allows you to produce interactive videotape courseware. It will connect a VIC-20 or Commodore 64 to various Panasonic, Magnavox, Canon, and Hitachi video cassette recorders.

The interface, which sells for $49.95, is being marketed in conjunction with Videobook's *Comp-U-Tutor Computer/VCR*

Will This Happen to You?



NOT IF YOU BUY YOUR PRINTER FROM...

THE PRINTER STORE

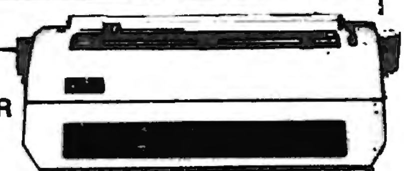
After reading this issue of ***Compute,*** you are now fully aware of the many outstanding printers on the market today. But what you might not be aware of is that the printer you like best, might not be compatible with your computer and software. (The picture above is good example of printer incompatibility). At the **Printer Store,**we specialize in printers, so our experienced professional staff can help you choose the right printer for your personal and business needs. If you want the **Best Value, Low Price, Product Availability,** and **Support,** call The Printer Store and ask us about:

- **FREE TECHNICAL CONSULTATION**
- **FULL AFTER SALE SUPPORT**
- **FULL FACTORY AUTHORIZED SERVICE**

C. ITOH 8510 Prowriter

- 120 CPS - 1.3K Buffer - 144x60 dots 1 inch Nx9 dot matrix - Proportional Spacing - 8 Character sizes - 5 unique alphabets - Greek character set - Graphic symbols - bi-directional, logic-seeking - Adjustable tractors - Single-sheet friction feed - Vertical & horizontal tabbing.

C. ITOH 8510 Prowriter ..... List $795 **$395**



BROTHER HR-1

- 16 CPS - Bi-directional - Super and Subscript
- IBM Selectric type ribbon - Prints up to 6 copies
- Ideal for word processing

Parallel .... List 1100 **$ Call**

Serial .... List 1200 **$ Call**

DOT MATRIX PRINTERS

EPSON SERIES
- FX 80 .... $ CALL
- FX 100 .... $ CALL

OKIDATA SERIES
- 82A .... $ CALL
- 83A .... $ CALL
- 92A .... $ CALL
- 93A .... $ CALL
- 84 (parallel) .... $ CALL

C. ITOH SERIES
- 8510 Prowriter .... $ 395
- Prowriter II .... $ CALL
- New! Banana .... $ 239

IDS SERIES
- Microprism 480 .... $ CALL
- Prism 80 .... $ CALL
- Prism 132 .... $ CALL

GEMINI SERIES
- Gemini 10X .... $ CALL
- Gemini 15 .... $ CALL

NEC 8023 .... $ 419

Toshiba P 1350 .... $ 1750

LETTER QUALITY PRINTERS

BROTHER SERIES
- HR-1 (parallel) .... $ CALL
- HR-1 (serial) .... $ CALL
- HR-15 .... $ CALL

COMREX SERIES
- CR-1 (parallel) .... $ 795
- CR-1 (serial) .... $ 865
- CR-2 .... $ CALL

C.ITOH SERIES
- F-10 40 CPS .... $ CALL
- F10 55 CPS .... $ CALL

Daisywriter 48K .... $ CALL

NEC SERIES
- 3510 .... $ CALL
- 3530 .... $ CALL
- 3550 .... $ CALL
- 7710 .... $ 2425
- 7720 .... $ 2900
- 7730 .... $2400

NEC Accessories .... $ CALL

INTERFACE EQUIPMENT

- Apple Dumpling $145
- Grappler Plus .... $ CALL
- Complete stock of Epson Accessories $ CALL
- Custom Printer Cables for Apple, Atari, IBM TRS-80 $ CALL
- Printer Stands: large .. $95
- Printers Stands: small . $25
- Printer Ribbons ... $ CALL
- Printer Sound Enclosure $ CALL
- Vic 20/Comm. 64 Interface ... $ CALL

In order to serve you better, The Printer Store will be moving to larger facilities. The move should be completed by Aug 25.

We carry a full line of Cables and Accessories

Call (714) 241-0701 and ask us about...

1) LOW PRICES
2) SAME DAY SHIPPING
3) FREE TECHNICAL SUPPORT
4) FULL SERVICE OPTION

PHONE REBATE We are so confident of our LOW PRICES and SUPPORT that we are going to sk you to make the initial investment by calling us. In return, when you buy your printer from us, we will rebate the cost of your call and deduct it from your invoice.

HOW TO ORDER: Our phone lines are open from 8 a.m. to 6 p.m. PST, Monday-Friday. We accept VISA, MASTERCHARGE (at no extra charge), personal checks take two weeks to clear. COD's accepted. Same-day shipment on orders placed before 1 p.m. Manufacturer's warranty applicable on all equipment. Prices subject to change.

THE Printer STORE — 2720 S. Harbor Blvd., Suite E, Santa Ana, CA 92704

Interactive Authoring System. The program, which sells for $149.95, permits the playback of packaged commercial programs, or will allow the user to write custom programs.

Videobook Corporation
P.O. Box 19597
Seattle, WA 98109

Ergonomic Joysticks

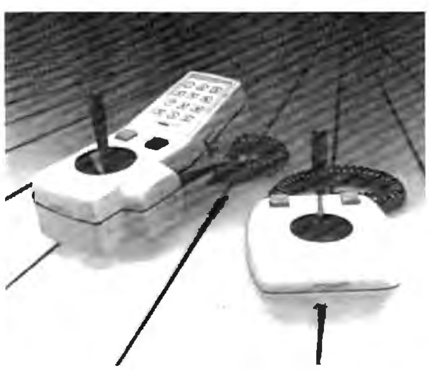
Comrex has released a pair of joysticks described as "electronically and ergonomically" designed to allow game players to improve their skills with less fatigue.

The deluxe ComMander, which sells for $44.95, is shaped like a hand-held telephone. It features a precision control, self-centering handle, and four firing buttons that have user-definable functions.

The standard ComMander, which is priced at $19.95, is designed with rounded corners and two fire buttons.

Both joysticks, which are equipped with coiled cords, are compatible with Atari 400/800, Commodore 64, and VIC-20 computers. The deluxe model will work with Apple computers.

Comrex International, Inc.
3701 Skypark Drive
Torrance, CA 90505
(213) 373-0280



The Comrex deluxe ComMander joystick is shaped like a hand-held telephone. The standard model has rounded, smooth corners to fit large and small hands.

Apple Space Game

Pentapus is a high-resolution space survival game for Apple II computers from Turning Point Software.

You begin the game in an uncharted region of another universe. To return home, you must defeat the Devilish Mutant Army for control of the stargates between universes. You must use your speed and intelligence to defeat waves of Drangels, Eagulls, and the Pentapus.

Pentapus, which sells for $29.95, incorporates graphics, sound, high-score editing, and four skill levels, including a children's level.

Turning Point Software, Inc.
11A Main Street
Watertown, MA 02172
(617) 923-4441

64K VIC Memory Expansion

Advanced Processor Systems has introduced SELECT-A-RAM, a 64K memory expansion cartridge for the VIC-20. The cartridge provides two expansion slots for program or game cartridges or additional memory expansion up to 192K.

The unit plugs into the VIC memory expansion slot and draws its power from the VIC-20. The memory expander, which sells for $169, features write protection, a reset switch, and optional use of external power.

Advanced Processor Systems
P.O. Box 43006
Austin, TX 78745
(512) 441-3202

64 Spreadsheet

Multiplan, a multiple-page financial spreadsheet program produced by Microsoft, will be mar-

keted by Commodore for the Commodore 64 computer.

The program, which can handle home budgets, financial planning, small business accounting, educational projects, and statistics, will be available on disk for less than $100.

Commodore
Computer Systems Division
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

TI Joystick

The Prostick 2002 is a direct replacement joystick for the TI-99/4 and TI-99/4A computers.

The joystick, which sells for $29.95, requires no additional interface for connection with the TI. It includes a 4-way/8-way switchable gateplate that allows 8-way action to be disabled when playing games that are limited to vertical and horizontal movement.

The Prostick 2002 has two firing buttons located on the top end of the base, allowing both right- and left-handed play. The fire buttons are designed to be controlled by the index finger for faster response and decreased fatigue during play.

Newport Controls
15425 Los Gatos Boulevard
Los Gatos, CA 95030
(408) 358-3439

Data Base Manager For VIC And 64

Jini Micro Systems, creator of several data base managers for Commodore computers, has released *Mini Jini*, a record keeper for the VIC and 64.

The program is available in cartridge format, and files can be saved to either tape or disk. It will handle between 35 and 500 records, depending on available memory.

Mini Jini will accept up to 10 fields of information per record, can sort by any field, and can search by record number, name, or phrase. When used with a printer, the program can generate reports and mailing labels.

Mini Jini includes a math function to perform calculations on file data, and it can be used in conjunction with word processing programs to produce personalized letters and custom reports.

The program sells for $89.95. Data files with sample records set up for dozens of applications are available on disk for $14.95, or tape for $9.95.

Jini Micro Systems, Inc.
Box 274
Riverdale, NY 10463

TI-99/4A Cartridge Connection

Romox is producing the GamePort expansion module for the TI-99/4A. The module, which sells for $39.95, plugs into the computer's I/O port and accesses the 9900 CPU directly.

The GamePort is designed to circumvent the reported plan of Texas Instruments to modify its internal software to accept only cartridges programmed in TI's patented GROM format.

The GamePort module, which accepts both GROM and standard ROM cartridges, includes an 8K ROM, plus RAM memory.

Romox, Inc.
501 Vandell Way
Campbell, CA

Grade And Attendance Manager

GradeCalc is a grade and attendance management package designed to relieve teachers of time-consuming, record-keeping tasks.

The grade section can average grades using a variety of methods, including percentage

APPLE COUNTRY LTD.®

YOU'RE GONNA LOVE THESE ROCK BOTTOM PRICES!

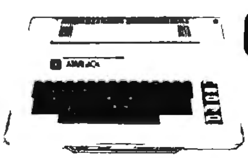
ACTION	99.00	68.95
ADVANCED MUSIC SYSTEM	29.95	21.95
AE	34.95	24.49
ANTI-SUB PATROL	29.95	21.95
APPLE CIDER SPIDER	39.95	27.95
ARCADE MACHINE	59.95	41.49
ARMOR ASSAULT	39.95	27.95
ATARI BOOKKEEPER KIT	249.95	179.95
ATARI MACRO ASSEM/TEXT ED	89.95	64.49
ATARI MICROSOFT BASIC II	89.95	64.49
ATARI MUSIC COMPOSER-ROM	39.95	29.95
ATARI PILOT EDUCATION-ROM	129.95	94.95
ATARI PROGRAMMER KIT	59.95	45.95
ATARI SPEED READING	74.95	54.95
ATARI TECH USER NOTES	29.95	21.95
ATARI TOUCH TYPING	24.95	17.95
ATARI WRITER	79.95	56.95
BANDITS	34.95	24.95
BANK STREET WRITER	69.95	49.95
BASIC COMPILER	99.95	68.95
BATTLE FOR NORMANDY	39.95	27.95
BATTLE OF SHILOH	39.95	27.95
BILESTOAD	29.95	21.95
THE BLADE OF BLACKPOOL	39.95	27.95
BOOK OF ATARI SOFTWARE'83	19.95	14.95
BUG/65	34.95	24.49
C/65	80.00	54.95
CAP'N COSMO	34.95	18.95
CASTLE WOLFENSTEIN	29.95	21.95
CENTIPEDE-ROM	44.95	31.95
CHOPLIFTER - ROM	44.95	31.95
COCO	49.95	34.49
COCO II	39.95	27.95
COLOR PRINT	39.99	27.95
COMMUNICATOR KIT	279.95	214.95
CONVERSATIONAL FRENCH	59.95	42.95
CONVERSATIONAL SPANISH	59.95	42.95
THE COSMIC BALANCE	39.95	27.95
CROSSFIRE-ROM	44.95	29.95
CYTRON MASTERS	39.95	27.95
DA FUZZ	44.95	31.95
DATA PERFECT	99.95	74.95
DATALINK	39.95	27.95
DAVID'S MIDNIGHT MAGIC	34.95	24.49
DEADLINE	49.95	34.49
DEFENDER	44.95	31.95
DE RE ATARI	19.95	14.49
DIG DUG	44.95	31.95
DISK MANAGER	29.95	21.95
DISK WORKSHOP	34.95	24.49
DISKETTE INVENTORY SYSTEM	24.95	17.49
DISKEY	49.95	34.49
DISKSCAN	40.00	28.00
DISKWIZ	29.95	21.95
DIVISION I	44.95	31.95
DNIEPER RIVER LINE	30.00	21.95
DODGE RACER	34.95	24.49
EASTERN FRONT (1941)	29.95	21.95
EDIT 6502-ROM	199.95	144.95
THE EDUCATOR KIT	164.95	109.95
FACEMAKER	34.95	24.49
FANTASTIC VOYAGE-ROM	34.95	26.49
FILE MANAGER +	99.95	68.95
FINANCIAL WIZARD	59.95	39.95
FLAME LORDS	34.95	24.95
FLASH GORDON-ROM	34.95	26.95
FLIP OUT	29.95	21.95
FIREBIRD-ROM	39.95	27.95
FORT APOCALYPSE	34.95	24.49
FROGGER	34.95	24.49
GALAXIAN	44.95	31.95
GHOST ENCOUNTERS	29.95	21.95
GHOSTLY MANOR	24.95	17.95
GLOBE MASTER	29.99	21.95
GORF-ROM	44.95	28.49
GRAPHIC GENERATOR	24.95	17.49
GRAPHIC MASTER	39.95	27.95
THE HOME ACCOUNTANT	74.95	54.95

JOURNEY TO THE PLANETS	29.95	21.95
JUMPMAN	39.95	27.95
KIDS AND THE ATARI-BOOK	19.95	13.95
KINDERCOMP	29.95	21.95
K-RAZY SHOOTOUT-ROM	49.95	34.49
KING ARTHUR'S HEIR	29.95	21.95
LEGIONNAIRE	35.00	25.00
LETTER PERFECT (40/80)	149.95	109.95
LETTER PERFECT UTILITY	29.95	21.95
LUNAR LEEPER	29.95	21.95
MASTER MEMORY MAP	6.95	4.95
MASTER TYPE	39.95	27.95
MATING ZONE	29.95	21.95
MATCH BOXES	29.95	21.95
MAURAUDER	34.95	24.49
MAX/65 (WITH OS/A+)	80.00	54.95
MICKEY IN GREAT OUTDOORS	49.95	35.95
MINER 2049'ER-ROM	49.95	34.49
MISSILE COMMAND-ROM	34.95	26.49
MONSTER SMASH	29.95	21.95
MORL	44.95	31.95
NAUTILUS	34.95	24.49
NECROMANCER	34.95	24.49
THE NIGHTMARE	29.95	21.95
OPERATION WHIRLWIND	34.95	24.49
OS-A+ & BASIC A+	80.00	54.95
PAINT	39.95	29.95
PINBALL	29.95	21.95
P.M.ANIMATOR	34.95	24.49
POKER-S.A.M.	24.95	17.95
PREPPIE II	34.95	24.49
PYRAMID PUZZLER	44.95	31.95
QIX	44.95	31.95
RASTER BLASTER	29.95	21.95
READING FLIGHT	44.95	31.95
ROUNDABOUT	29.95	21.95
SAMMY LIGHTFOOT	34.95	24.49
SAMMY THE SEA SERPENT	23.95	16.95
SEA DRAGON	34.95	24.49
SEA FOX	29.95	21.95
747 LANDING SIMULATOR	22.95	16.95
SHADOW WORLD	34.95	24.49
SHAMUS-ROM	44.95	31.95
SNEAKERS	29.95	21.95
SOFTWARE AUTO-MOUTH (SAM)	59.95	41.49
SPEED READ PLUS	59.95	41.49
SPEEDWAY BLAST	29.95	21.95
SPELLING BEE GAMES	39.95	27.95
SPELL WIZARD	79.95	54.95
STARCROSS	39.95	27.95
STAR MAZE	44.95	31.95
STAR RAIDERS-ROM	44.95	31.95
STAR WARRIOR	39.95	27.95
STORY MACHINE	34.95	24.49
SUPERMAN III	49.95	35.95
SURVIVOR	34.95	24.49
SWIFTY TACH MASTER	29.95	21.95
SYN ASSMBLER	49.95	34.49
TELECOM	69.95	49.95
TELETALK	49.95	36.95
TELETARI	39.95	27.95
TEMPLE OF APSHAI	39.95	27.95
TEXT WIZARD I	99.95	68.95
TIGERS IN THE SNOW	39.95	27.95
TIME WISE	29.95	21.95
TYPE ATTACK	39.95	27.95
TUTTI FRUTTI	24.95	17.95
VC	25.00	17.49
VISICALC	199.00	149.95
WARLOCK'S REVENGE	34.95	24.49
WAY OUT	39.95	27.95
WIZARD&PRINCESS HIRES ADV	32.95	22.95
WIZARD OF WOR - ROM	44.95	31.95
YOUR ATARI COMPUTER-BOOK	16.95	12.95
ZAXXON	39.95	27.95
ZORK I II OR III	39.95	27.95

AMDEK COLOR I PLUS MONITOR		319.00
APPLE EMULATOR	CBM64	79.50
CARDBOARD (6 CART + RESET)	VIC-20	70.00
THE CARD? (FOR PARALLEL PRTR)	CBM64&VIC-20	58.00
CARDETTE (CASSETTE INTFCE)	CBM64&VIC-20	22.00
CARDAPTER/1 ATARI 2600 INTFCE	VIC-20	50.00
CARDWRITER/1 LIGHT PEN W/CASS	CBM64&VIC-20	16.50
COMMODORE 64 HOME COMPUTER	CBM64	369.50
COMMODORECOLOR PLOTTER	CBM64&VIC-20	179.50
COMMODORE 1530 DATASETTE	CBM64&VIC-20	65.00
COMMODORE 1541 DISK DRIVE	CBM64&VIC-20	329.50
COMMODORE 1525 PRINTER	CBM64&VIC-20	329.50
COMMODORE 1600 MODEM	CBM64&VIC-20	95.50
COMMODORE COLOR MONITOR	CBM64&VIC-20	269.50
COMMODORE 1650 AUTODIAL MODEM	CBM64&VIC-20	169.50
MICRO EXPANSION CHASSIS	CBM64	35.75
VIDEO PAK 80 (80 COLUMN)	CBM64	129.00
VIDEO PAK 80 WITH CP/M	CBM64	219.00
ATARI 800 48K	$489-$100 REBATE=	389.00
ATARI 1200 64K	$519-$100 REBATE=	419.00
ATARI NUMERIC KEYPAD		94.95
ATARI PROGRAMMER KIT		49.95
410 RECORDER		72.95
810 DISK DRIVE		424.95
850 INTERFACE MODULE		169.95
C. ITOH PROWRITER I		394.95
C. ITOH PROWRITER II		649.95
C. ITOH STARWRITER		1325.00
CASSETTE 'N CARTRIDGE FILE		21.95
ELEPHANT SS/SD DISK		10/16.95
ELEPHANT SS/DD DISK		10/19.95
ELEPHANT DS/DD DISK		10/25.95
EPSON FX-80 W/TRACTOR		LOW!!
EPSON MX-100 F/T		LOW!!
FLIP'N-FILE		21.95
GEMINI-10 PRINTER		LOW!!
GEMINI-15 PRINTER		LOW!!
IN HOME 400 KEYBOARD		94.95
INTEC 32K RAM		69.95
INTEC 48K RAM		119.95
MOSAIC 64K RAM SELECT		149.95
HAYES SMARTMODEM 300 BD		194.95
HAYES SMARTMODEM 1200 BD		499.95
NEC 8023 PRINTER		459.95
NEC 12" HIRES GREEN SCRN		149.95
NEC 12" ECONO GREEN SCRN		79.95
NOVATION J-CAT MODEM		109.95
NOVATION SMART-CAT 103		179.95
NOVATION SMART-CAT 212		424.95
PERCOM SS/SD/1DR (88K)		419.95
PERCOM SS/DD/1DR (176K)		539.95
PERCOM SS/DD/2DRS (352K)		859.95
PERCOM DS/DD/1DR (352K)		649.95
PERCOM DS/DD/2DRS (704K)		939.95
SIGNALMAN MK II MODEM		79.95
USI 12" AMBER MONITOR		159.95
VERSAWRITER GRAPH TABLET		239.95
WICO JOYSTICK		21.95
WICO REDBALL JOYSTICK		24.95
WICO DELUXE JOYSTICK		29.95
WICO TRACKBALL		49.95
WICO 12 FT EXTENSION CORD		6.95

commodore

AGGRESSOR	CART/VIC20	27.50
ALIEN SOCCER	CASS/VIC20	10.25
ANDROMEDA CONQUEST	CASS/C64	12.50
APE ESCAPE	DISK/VIC20	27.50
BLADE OF BLACKPOOLE	DISK/C64	27.50
CAVE-IN	CART/VIC20	27.50
CHECKBOOK	CASS/VIC20	13.75
COMPUTER FOOTBALL STRATEGY	CASS/C64	11.00
COMPUTER STOCKS & BONDS	CASS/C64	13.75
CONCENTRATION	CASS/VIC20	11.00
CRITICAL MASS	DISK/C64	27.50
CROSSFIRE	CASS/VIC20	21.00
DEADLINE	DISK/C64	34.50
DEADLY DUCK	CART/VIC20	24.00
DEADLY SKIES	CART/VIC20	27.50
DEMON ATTACK	CART/VIC20	27.50
DRELBS	CASS&DISK/64	24.00
FACEMAKER	DISK/C64	24.00
FAST EDDY	CART/VIC20	27.50
FINAL ORBIT	CART/C64	24.00
FT. APOCALYPSE	CASS&DISK/64	24.00
FROGGER	DISK/C64	24.00
FRUIT FLY	CASS/VIC20	9.00
FUEL PIRATES	CASS/VIC20	11.00
GALACTIC BLITZ	CASS/VIC20	17.25
GALACTIC CROSSFIRE	CASS/VIC20	10.25
GOLD FEVER	CART/VIC20	27.50
GOLD MINE	DISK/VIC20	27.50
GRIDRUNNER	CART/64/VIC20	27.50
HANGMAN	8KCASS/VIC20	6.25
HARRIER	CASS/VIC20	24.00
HESMON MONITOR	CART/64&VIC20	27.50
HESWRITER W/P	CART/VIC20	27.50
HESWRITER 64	CART/C64	31.00
HEY DIDDLE DIDDLE	DISK/C64	21.00
IN SEARCH OF THE MOST AMAZING THING	DISK/C64	27.50
JUMP MAN	DISK/C64	27.50
KINDERCOMP	DISK/C64	21.00
MARTIAN RAIDER	DISK/VIC20	13.75
METEOR	8KCASS/VIC20	7.50
MONSTER MAZE	CART/VIC20	27.50
MULTISOUND SYNTH.	DISK/VIC20	13.75
NUMBERCHASER	16KCASS/VIC20	17.25
NUMBER CRUNCH	CART/VIC20	27.50
NUMBER GULPER	8K CASS/VIC20	17.25
PHAROH'S CURSE	CASS&DISK/64	24.00
PREDATOR	CART/VIC20	27.50
PROTECTOR	CART/VIC20	30.25
QUICK BROWN FOX W/P	CART/VIC20/64	44.75
RAID ON ISRAM	CASS/VIC20	13.75
REAGANOMICS	CART/VIC20	27.50
REPTON	DISK/C64	27.50
RESCUE AT RIGEL	16KCASS/VIC20	21.00
RETRO BALL	CART/C64	27.50
RICOCHET	8K CASS/VIC20	13.75
ROBOT PANIC	CART/VIC20	27.50
SCORPION	CART/VIC20	27.50
SHAMUS	CART/VIC20	27.50
SHARK TRAP	DISK/VIC20	13.75
SIDEWINDER	8K CASS/VIC20	21.00
SIMON	CASS/VIC20	11.00
6502 PROF DEV SYS	CASS/64/VIC20	21.00
SKI RUN	8KCASS/VIC20	12.50
SNAKE BYTE	CART/C64	24.00
SNOOPER TROOPS #1	DISK/C64	27.50
SPACE ATTACK	8K CASS/VIC20	6.25
SPIDER CITY	CART/64/VIC20	27.50
SPORTS SEARCH	CASS/VIC20	13.75
SQUISH'EM	CART/64/VIC20	24.00
STARCROSS	DISK/C64	27.50
SURVIVOR	CASS&DISK/C64	24.00
SWARM!	CASS/VIC20	21.00
SWORD OF FARGOAL	16KCASS/VIC20	21.00
SYN THE SOUND/MUSIC	CART/VIC20	41.00
TELENGARD	CASS/C64	15.75
TORG	CASS/VIC20	12.25
TURMOIL	CART/VIC20	27.50
TURTLE GRAPHICS	CART/VIC20	27.50
TURTLE GRAPHICS II	CART/C64	41.00
TYPE ATTACK	CART/64/VIC20	27.50
THE VEIN GAME	DISK/VIC20	24.00
VIC FORTH	CART/VIC20	41.25
VIC-MEN	8KCASS/VIC20	15.00
VIC MUSIC COMPOSER	CART/VIC20	30.00
VICTREK	CASS/VIC20	12.25
VIC VANGO	CASS/VIC20	9.00
VIDEO SEARCH	CASS/VIC20	13.75
WORD SEARCH	CASS/VIC20	13.75
ZORK I II OR III	DISK/C64	27.50



ATARI PERCOM RanaSystems Hayes MOSAIC 64K RAM SELECT EAGLE COLUMBIA DATA PRODUCTS INC FRANKLIN NEC IBM TEXAS INSTRUMENTS OKIDATA

FOR FASTEST DELIVERY: CASHIER'S CHECK OR VISA/MASTERCARD (NO EXTRA CHARGE FOR CARDS. INCLUDE NUMBER, EXPIRATION DATE, NAME, ADDRESS & PHONE). PERSONAL CHECK ALLOW 2 WEEKS TO CLEAR. PURCHASE ORDER MUST INCLUDE CHECK. SHIPPING & HANDLING: CONTINENTAL U.S. 5% ($5 MIN), U.P.S. STREET ADDRESS REQUIRED; APO FPO ALASKA HAWAII & MONITORS 5% ($10 MIN); FOREIGN 15% ($15 MIN). INCLUDE PHONE NUMBER WITH ALL ORDERS. ALL ITEMS ARE NEW WITH MANUFACTURER'S WARRANTY. APPLE COUNTRY, LTD. CANNOT GUARANTEE THE MERCHANTABILITY OF ANY PRODUCT. PRICES ARE SUBJECT TO AVAILABILITY & CHANGE WITHOUT NOTICE. DUE TO OUR LOW PRICES, ALL SALES ARE FINAL. RETURNED MERCHANDISE IS SUBJECT TO A RESTOCKING FEE & MUST COME IN ORIGINAL UNDAMAGED CARTON WITH RMA NUMBER. NO SOFTWARE EXCHANGES. CALIFORNIA RESIDENTS ADD 6% TAX. SEND $1 FOR NEW FALL CATALOG (GOOD TOWARD FIRST PURCHASE). APPLE COUNTRY, LTD. IS A DISCOUNT MAIL ORDER HOUSE FOR THE MICROCOMPUTER INDUSTRY & IS A CALIFORNIA CORPORATION NOT AFFILIATED WITH APPLE COMPUTER INC. APPLE IS TRADEMARK OF APPLE COMPUTER INC. 6.1.83

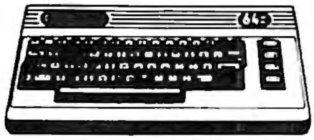
Call us . . . we can help! (800) 222-2602

P.O. Box 1099, Julian, Calif. 92036

Tech. Info., Order Status Call (619) 765-0239

Apple Country, Ltd. is a DISCOUNT MAIL ORDER HOUSE for the micro computer industry and is a California corporation not affiliated with Apple Computer Inc. Apple is a trademark of Apple Computer Inc.

Prices the Compe



Commodore 64	$219
VIC 20	$ 99
1525 Printer	$229
1530 Datasette	$ 64
1541 Disk Drive	$249
1600 Modem	$ 89
1701 Commodore Monitor	$289
VIC 1311 Joystick	$ 8
VIC 1312 Game Paddles	$ 16

VIC 1210 3K Memory Expander	$34
VIC 1110 8K Memory Expander	$52
VIC 1111 16K Memory Expander	$89
VIC 1011 RS 232 Terminal Interface	$43
VIC 1211 Super Expander	$59
VIC 1213 Vicmon Machine Language Monitor	$45
VT 106A Recreation Pack	$45
VT 107A Home Calculation Pack	$45
VIC 1600 Vicmodem	$89
VM Programmer's Reference Guide	$14

Commodore 64

Avalon Hill Game Company

180-701 B-1 Nuclear Bomber (C)	$12
180-702 Midway Campaign (C)	$12
180-703 North Atlantic Convoy Raider (C)	$12
180-704 Nukewar (C)	$12
180-706 Planet Miners (C)	$12
180-712 Computer Stocks & Bonds (C)	$15
180-719 Andromeda Conquest (C)	$14
181-721 Computer Football Strategy (C)	$12
181-732 Telengard (C)	$16

Broderbund

David's Midnight Magic	$23
Choplifter (CT)	$34
Serpentine (CT)	$27
Sea Fox (CT)	$27

Datamost

Roundabout	$20
Bilestoad	$20
Mating Zone	$20

EPYX/Automated Simulations

14E-036 Jump Man (D)	$27

Human Engineered Software (HES)

HEE-307 6502 Professional Dev. System (C)	$23
HEE-400 Retro Ball (Crt)	$27
HEE-401 Hesmon (Crt)	$27
HEE-402 Turtle Graphics II (Crt)	$45
HEE-404 Heswriter 64 (Crt)	$35
HEE-412 Gridrunner (Crt)	$27

Infocom

63E-001 Zork (D)	$ 27
63E-002 Zork II (D)	$ 27
63E-003 Deadline (D)	$ 34
63E-004 Starcross (D)	$ 27
63E-005 Zork III (D)	$ 27

Sierra On-Line

54E-048 Frogger (D)	$23
Jaw Breaker	$20

Sirius Software

70E-036 Blade of Blackpoole (D)	$27
70E-037 Type Attack (Crt)	$27
70E-043 Repton (D)	$27
70E-046 Critical Mass (D)	$27
70E-424 Snake Byte (Crt)	$23
70E-445 Spider City (Crt)	$27
70E-447 Squish 'em (Crt)	$23
70E-448 Final Orbit (Crt)	$23

Spinnaker

SKE-001 Snooper Troops #1 (D)	$30
SKE-004 Facemaker (D)	$ 23
SKE-006 Kindercomp (D)	$ 20
SKE-008 Hey Diddle Diddle (D)	$ 20
SKE-009 In Search of the Most Amazing Thing	$ 27
Fraction Fever (CT)	$ 20
Alphabet Zoo (CT)	$ 20
Delta Drawing (CT)	$ 23

Snyapse Software

SSE-011 Ft. Apocalypse (D)	$ 23
SSE-016 Drelbs (D)	$ 23
SSE-019 Survivor (D)	$ 23
SSE-020 Pharoah's Curse (D)	$ 23
SSE-311 Ft. Apocalypse (C)	$ 23
SSE-316 Drelbs (C)	$ 23
SSE-319 Survivor (C)	$23
SSE-320 Pharoah's Curse (C)	$ 23

United Microwave Industries

92E-302 Renaissance (C)	$ 20
92E-331 Motor Mania (C)	$ 20

Educational

Books

Kids and the Vic	$ 18
Programmer's Reference Guide (Vic)	$ 14
Programmer's Reference Guide (64)	$ 18

Language Arts

Super Hangman (C)	$ 14
Simon/Hess (C)	$ 13
Concentration (C)	$ 13
Home Babysitting	$ 23

Math

Sky Math (C)	$ 12
Space Division	$ 12
Bingo Speed Math (CT)	$ 23
Number Crunch (CT)	$ 23
Number Chaser	$ 17
Number Gulper	$ 17

Music

Vic Music Composer (CT)	$ 42
HES Synthesound (CT)	$ 49

Reading and Language Arts

Finger Spelling (D, C)	$ 12
My Spelling Easel (Ages 3-10)	$ 26

Programming Techniques

Intro to Basic Prog. I	$ 22
Intro to Basic Prog. II	$ 22
Programmer's Aid Cart	$ 22
Turtle Graphics/Hess (CT)	$ 29
Gorteck & the Microchip	$ 23

Social Studies/Science

Visible Solar System	$ 23
Reaganomics (CT)	$ 27

Commodore Software

Avenger	$ 23
Superslot	$ 23
Super Alien	$ 23
Jupiter Lander	$ 23
Draw Poker	$ 23
Midnight Drive	$ 23
Radar Rat Race	$ 23
Raid on Fort Knox	$ 23
Sargon II Chess	$ 29
Cosmic Cruncher	$ 23
Gorf	$ 29
Omega Race	$ 29
Sea Wolf	$ 23
Adventureland	$ 29
Pirate Cove	$ 29
Mission Impossible	$ 29
The Count	$ 29
Voodoo Castle	$ 29
The Sky is Falling	$ 23
Mole Attack	$ 23
Bingo Speed Math	$ 23
Home Babysitter	$ 23
Visible Solar System	$ 23
Personal Finance	$ 29
Quick Brown Fox	$ 65

Vic 20

Avalon Hill

Nuke War	$12

Automated Simulations

Rescue at Rigel (C)	$20
Ricochet (C)	$15
Monster Maze (CT)	$27
Sword of Fargoal	$27

Broderbund

Martian Raider	$15
Multisound Synthesizer	$15
Shark Trap	$15
Sky Blazer (CT)	$27
Sea Fox (CT)	$27
A E (CT)	$27

Creative Software

Black Hole (CT)	$36
Trashman (CT)	$36
Astroblitz (CT)	$36
City Bomber & Minefield (CT)	$20
Apple Panic (CT)	$36
Choplifter (CT)	$36
Serpentine (CT)	$36
Videomania (CT)	$36
Terraguard (CT)	$36

HES Software

HES Mon (CT)	$ 29
HES Writer (CT)	$ 29
Synthesound Music Synthesizer (CT)	$ 49
Turtle Graphics (CT)	$ 29
VIC Forth (CT)	$ 45
Victrek (C)	$ 15
Predator (CT)	$ 27

Sirius

Type Attack	$27
Snake Byte	$27

Thorn EMI

River Rescue (CT)	$ 29
Mutant Herd (CT)	$ 29

Tronix

Galactic Blitz (C)	$17
Swarm (C)	$20
Sidewinder (C)	$20
Gold Fever (CT)	$27
Deadly Skies (CT)	$27

United Microware

Spiders of Mars (CT)	$ 34
Meteor Run (CT)	$ 34
Amok	(C) $ 15 (CT) $ 27
Alien Blitz	(C) $ 17 (CT) $ 27
Skymath (C)	$ 12
Space Division (C)	$ 12
Super Hangman (C)	$ 14
The Alien (C)	$ 17
3D Maze (C)	$ 12
Kosmic Kamikaze (C)	$ 17
Sub Chase (C)	$ 15
Renaissance	$ 20
Cloud Burst (CT)	$ 27
Satellites & Meteorites (CT)	$ 34
Outworld (CT)	$ 34
Wordcraft	$ 65

Call Toll Free

1-800-634-6766

Order Line Only

1095 E. Twain (702) 796-0296
Las Vegas, Nevada 89109

Information & Order Inquiries
(702) 369-5523
We accept VISA and MasterCard
Mon. — Fri. 8 A.M. to 6 P.M.
Sat. 9 A.M. to 6 P.M.
Dealers' Inquiries Invited

COMPUTER OUTLET

scores and letter grades. Reports on missing assignments, grade totals, averages, and assignment summaries are available.

The attendance records can generate cumulative totals of all attendance records, or can produce reports on excessive absences or other problems.

GradeCalc is available for the Commodore 64, PET, and Apple computers for $29.95. An Atari version sells for $34.95.

Tamarack Software, Inc.
Water Street
Darby, MT 59829
(406) 821-4596

Wizware For Children

A new line of software called Wizware, designed for children between 6 and 16, is available from Scholastic. The programs are available on disk and tape for a variety of computers including Apple II, Atari, TI-99/4A, and VIC-20.

Microzine is a children's magazine in computer format. The articles in the magazine are designed to be worked with, rather than simply read. *Microzine* sells for $39.95 per issue, or $149 for six bimonthly issues. The programs are compatible with Apple II and Atari 800 computers.

Turtle Tracks teaches children the fundamentals of programming by allowing them to experiment with colors and shapes as they control the movements of a "turtle."

Square Pairs teaches youngsters number matching and how to recognize words and patterns. As the children learn more, they can use the software to design their own games.

Your VIC-20 is a hands-on introduction to the VIC for first-time users. The program explains input/output devices, graphics, music, and BASIC programming.

Electronic Party allows children to create electronic greeting cards, and lets them move surprise packages through a maze. Inside each of the packages is a surprise activity that can be acted out.

Scholastic, Inc.
730 Broadway
New York, NY 10003
(212) 505-3000

Murder Mystery – Computer Style

Infocom's *The Witness* is a 1930s whodunit that is available for most popular microcomputers.

The game immerses players in 1930s detail. In addition to the game diskette, the package includes a replica of a 30s detective magazine that serves as a users

manual, a copy of a 1930s newspaper page, a suicide note, Western Union telegram, and a matchbook.

Clues necessary to unravel this adventure mystery are hidden in this paraphernalia. The game sells for $49.95 to $59.95 depending on the version.

Infocom, Inc.
55 Wheeler St.
Cambridge, MA 02138

Thinking Game For Apple

Pensate is a thinking game of evasion for Apple computers.

The object of the game is to get to the top of an 8x8 playing grid while avoiding other playing pieces. One- and two-player options are available, and a game can be instantly replayed once it is over.

The game, which includes hundreds of possible setups, is available from Penguin Software for $19.95.

Penguin Software
830 4th Ave.
Geneva, IL 60134
(312) 232-1984

Computer Aided Instruction Programs

Random House has announced a new series of Computer Aided Instruction programs called *Custom Ware.*

The programs, described as a "mini authoring series," make it possible for the user to create and edit his own CAI lessons.

The initial *Custom Ware* offering includes lessons for grades three through six on topics including multiple-meaning words, figurative language, synonyms and antonyms, prefixes and suffixes, dictionary skills, and spelling demons.

Instructions provided with the package allow the user to customize the programs by creating additional lessons following Random House's formats.

Custom Ware is available for Apple II and IBM computers.

Random House
7307 S. Yale Ave. 103
Tulsa, OK 74136

64K RAM For Timex/Sinclair

Gladstone Electronics has introduced a 64K RAM expansion for the Timex/Sinclair computer.

The RAM pack, which sells for under $100, is enclosed in a molded plastic case.

Gladstone Electronics
1585 Kenmore Ave.
Buffalo, NY 14217
(716) 874-5510

Data Base For Commodore 64

Melcomp has produced *Database 64*, a data management system for the Commodore 64.

With the program, the user can define the number and length of fields, and records can be displayed on the screen or printer in a variety of formats. *Database 64* is menu-driven and is written in BASIC for easy modification. The program sells for $39.

Melcomp
P.O. Box 1085
Melbourne, FL 32901

Weather Sensor For VIC And 64

The Home Automatic Weather Station is an external sensing device that allows a Commodore 64 or VIC-20 to monitor weather conditions indoors or outdoors. The package includes software to analyze conditions and make weather predictions.

The Home Automatic Weather Station can be connected to a VIC or 64 to monitor changing weather conditions.

The sensor is designed to be used as an educational tool or as a monitoring tool for sensitive environments such as greenhouses. The package is available for $199.95. The software comes in both tape and disk versions.

Vaisala Inc.
2 Tower Office Park
Woburn, MA 01801
(617) 933-4500

DOS Enhancer For VIC And 64

Softron has introduced *EZ/DOS*, a program that simplifies the use of the Commodore 1540/1541 disk drives.

The machine language program is loaded into memory and then is invisible until you need it. Call the menu and execute one of the options. They include LOADing or SAVEing a program, replacing a program, VERIFYing a program, reading a disk directory, formatting a new disk, renaming or erasing a file, or renaming a disk.

The program, which is available on disk, sells for $29.95.

Other new products available from Softron include:

The Do It Yourself Book, a book and software package for the VIC-20. The disk version is available for $29.95, and the tape version is available for $27.94.

Pizza Time! is an 8K arcade game for the VIC-20. The disk version sells for $24.95, and the tape version sells for $22.95.

Softron's Keyboard Soft/Lay is a keyboard overlay that includes BASIC commands, memory locations, and DOS commands. It is available for both the VIC and 64 for $8.95.

Softron, Inc.
2067 Broadway, Suite 27
New York, NY 10023
(212) 490-0077

The Family Tree

A pair of genealogical programs for the TRS-80 Color Computer are available from The Word Merchant.

The programs are *Family*, which maintains data on 255 ancestors in eight generations, and *Famprint*, which uses the information in the *Family* data base to print out three genealogical forms. They are Pedigree, which provides five generations of information for any name; Family Group Chart, which provides information on each husband-wife pair; and Reference Number Index, which lists all names with their corresponding reference numbers.

The programs, which require Extended BASIC and 32K, are available as a package on cassette for $9.95.

The Word Merchant
P.O. Box 232
Lititz, PA 17543

Optical Scanner System

Databar has introduced an optical scanner, called OSCAR, designed to read specially coded programs into most popular microcomputers.

The $79.95 scanner includes an issue of a monthly magazine which contains BASIC programs coded for scanning. Subsequent issues of the magazine will sell for $10.

The OSCAR system is expected to be available in early 1984.

Databar Corporation
10202 Crosstown Circle
Eden Prairie, MN 55344
(612) 944-5700

Color Computer Spreadsheet

Elite-Calc is a machine language electronic spreadsheet for the TRS-80 Color Computer. The program, a product of Elite Software, adjusts itself to any memory size, but requires Extended BASIC to run.

With *Elite-Calc*, spreadsheets of 255 rows by 255 columns can be created. Manipulation of data is accomplished through a variety of functions, including LOG, SQR, ABS, SUM, AVERAGE, MIN, and MAX. Relational operators and the trigonometric functions also are supported.

The program includes HELP displays, rapid entry modes for text and data, and an option for automatic cursor movement. *Elite-Calc* is available on tape or disk for $44.95.

Elite Software
Box 11224
Pittsburgh, PA 15238
(412) 795-8492

CALENDAR

October 19-21, Pinehurst Hotel and Country Club, Chapel Hill, NC. "Introduction to Computers and Their Applications." A three-day seminar sponsored by LEARNING AT PINEHURST. Hands-on instruction and learning sessions on software, hardware, languages, applications, and the history and future

of computers. Course fee is $550 per person; group discounts available. For more information, write LEARNING AT PINEHURST, P.O. Box 2328, Chapel Hill, NC 27514.

October 20-21, Mission Valley Inn and Convention Center, Raleigh, NC. Carolina Micro-Expo. Over 50 seminars on microcomputers and their use will be presented. Local and national manufacturers and suppliers will demonstrate equipment. For more information, contact Richard Gardner, Show Manager, HMM Associates, Inc., 8400 Glenwood Avenue, Raleigh, NC 27612. (919) 782-0824.

October 26-28, Red Lion Inn, San Jose, CA. Annual Conference of the California Educational Data Processing Association (CEDPA): "Update '83: Are We Ready For 1984?" Sessions will focus on the spectrum of computing power available to data processing and instructional professionals and explore the impact of high technology on education. For information and registration, contact Jane Householder, Office of the Los Angeles County Superintendent of Schools, Room 226, 9300 East Imperial Highway, Downey, CA 90242. (213) 922-6141.

November 3-6, New York Coliseum, New York, NY. Electronic Fun Expo. New York's first state-of-the-art consumer electronics show. Open to the public. For more information, write to Bill O'Brien, Electronic Fun, 350 East 81st Street, New York, NY 10028. (212) 734-4440.

November 5-6, Scottish Rite Center, San Diego, CA. 4th Annual San Diego Computer Fair. The fair will feature short technical sessions, programming and computer game contests, door prizes, commercial displays, and displays by computer user groups. Registration is $5, Saturday night banquet fee is $12. Reservations can be made by mailing a check or money order to: San Diego Computer Society, P.O. Box 81537, San Diego, CA 92138. (619) 565-8720.

November 16-18, Pinehurst Hotel and Country Club, Chapel Hill, NC. "Introduction to Microcomputers and Their Applications." A three-day seminar featuring hands-on instruction and learning sessions on software, hardware, languages, applications, and the history and future of microcomputers. Course fee is $550; group discounts are available. For more information, write to LEARNING AT PINEHURST, P.O. Box 2328, Chapel Hill, NC 27514.

November 17-19, Hynes Auditorium, Boston, MA. Fifth Annual Northeast Computer Show and Software Exposition. Features nearly 500 displays and exhibits of microcomputers, accessories, peripherals, and software. Show hours are 10:30 a.m. to 5:30 p.m. daily. Admission is $7.50. For more information, call or write Northeast Expositions, 822 Boylston St., Chestnut Hill, MA 02167. (800) 841-7000 or (617) 739-2000 (in Massachusetts).

New Product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

COMPUTE! *welcomes notices of upcoming events and requests that the sponsors send a short description, the.r name and phone number, and an address to which interested readers may write for further information. Please send notices at least three months before the date of the event, to: Calendar, P.O. Box 5406, Greensboro, NC 27403.*

MEMORY EXPANSION BOARDS for ATARI* COMPUTERS

Tiny Tek Inc Memory Boards are fully assembled tested and guaranteed

48K/52K Memory Board	$114.95
For ATARI* 400 52K Addressable Memory Easy to Install	
32K Memory Board	$79.95
For ATARI* 400 or 800 No Modifications Required	
16K Memory Board	$49.95
For ATARI* 800 No Modifications Required	

We Will Meet Advertised Prices

BUILD YOUR OWN MEMORY

48K/52K Board (No Components)	$35.00
32K Board (No Components)	$30.00
16K Board (No Components)	$10.00
48K/52K Complete Kit	$90.00
32K Complete Kit	$60.00
16K Complete Kit	$35.00

Add $2 Shipping & Handling
Visa & MasterCard Accepted
*ATARI is a trademark of Atari Inc
Dealer Inquiries Welcome

Tiny Tek, Inc.
P O. Box 12609 • Dallas, TX 75225
214-373-8926

NEW! ! FACTORY DIRECT

SPACE SAVING
C.R.T./T.V. STAND
IN DURABLE FORMICA
(Oak-Beech-Beige)
For Most Popular Computers
VIC-ATARI-T I -TIMEX-64-etc
NOW ONLY $47.00*
(* shipping included)

EYE LEVEL VUEING - COVERS CABLE JUMBLE

VIC-20™ AFFORDABLE SOFTWARE

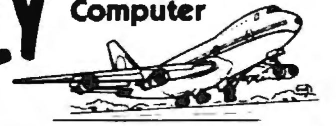
• BILOW'S CAR RACE • BILOW'S BOMBERS •
• BILOW'S CHOPPERS • DENVER SKIERS •
• BILOW'S SUBS • BILOW'S MISSILES •
Hi-Resolution Graphics, Sound & Colors.

Cassette Tapes	$ 6.00*
6-Pac *Special*	$30.00*

ORDER TODAY
CHECK, MONEY ORDER

"BILOW'S" ENTERPRISES
BOX 661655
MIAMI SPRINGS FLORIDA 33166

FLY Your Computer



FLIGHT SIMULATOR GAMES

Sky Pilot (8K VIC-20)	**$18.00**
Runway 20 (16K VIC-20)	**$25.00**
Runway 64	**$25.00**
Micro-Pilot (EPSON HX-20)	**$18.00**

ADD $2.00 FOR DISK VERSION

TORPEDO!

Submarine Battle Games
8K VIC-20 or Commodore 64 **$25.00**

SUSIE SOFTWARE
709 Wilshire Dr. Mt. Prospect, IL 60056
(312) 394-5165

Wasting Money? We Have the World's Most Cost Effective Development System.

- Includes Hexkit 1.0 a powerful 100' machine code editor debugger utility program that makes coding for 8-bit Micros a snap
- Program from Commodore VIC-20 keyboard into built-in 4K ROM emulator
- Jumper to target ROM socket
- Test programs in circuit
- Built-in EPROM programmer and power supply
- Burns & runs EPROMS for the Commodore VIC-20 too
- Comprehensive manuals
- Fits EXPANSION PORT

Send for Free Brochure

VISA AND MASTERCARD ACCEPTED

PROMQUEEN CARTRIDGE
ONLY $199 U.S. $269 CAN.

+ Prices	U.S.	Can.
Promqueen 64	$299.00	$399.00
8K board w 1 Eprom	$ 29.95	$ 39.95
16K board w 1 Eprom	$ 39.95	$ 49.95
8K ROM board w 1 Eprom-C64	$ 39.95	$ 49.95

Distributed in U S by **Arbutus Total Soft, Inc.**
4202 Meridian, Suite 214
Bellingham, WA 98226
Ph 800-426-1253, in Washington 206-733-0404

Distributed in Canada by
IBC/Distribution Canada
4047 Cambie St., Vancouver, BC V5Z 2X9
Ph 604-879-7812

HAVING LOADING PROBLEMS ?

GET A Z-DUBBER

of the great variation in cassette recorders used with the TS1000 some in very poor shape you may be having a problem loading programs. The Z-Dubber connects between the cassette recorder and the computer

CHARGE OR COD ORDERS CALL
1-800-227-3800
ASK FOR OPERATOR 225

BYTESIZE MICRO TECHNOLOGY
PO BOX 12309 DEPT DM
SEATTLE, WA 98111
(206) 236-BYTE

CALL OR WRITE FOR DEALER INFORMATION

SM-ESH

Screen displays are the interfaces between micros and human beings.

Good screen displays are the hallmark of top rank application programs — but they are expensive.

The ESH (ergonomic screen handler) from SM allows you to produce superb screen displays with an astonishing economy of effort.

for cbm 8032 or 8096

FREE CATALOG!

JEWEL THIEF
ADVENTURE GAME - $14.95
CAN YOU BREAK INTO THE CRYSTAL CITY MUSEUM AND STEAL THE 'BLUE WALLABY' DIAMOND WITHOUT BEING CAUGHT?
FOR: VIC20 - REQUIRES 8K EXPANDER
ZX81/TIMEX1000 - REQUIRES 16K
COMMODORE 64 ($19.95)
KAYPRO II ($19.95)

ANDROID
STRATAGY GAME - $18.95
RETREIVE THE S.S.R. FROM THE ALIEN ANDROIDS. GET PAST INTERCEPTERS, DESTROYERS, AND THEN GEORGE.
FOR: VIC20 - REQUIRES 16K EXPANDER
ZX81/TIMEX1000 - REQUIRES 16K
COMMODORE 64

AVAILABLE ON CASSETTE OR DISK.
ADD $1.00 FOR EACH DISK ORDERED.
CHECK OR MONEY ORDER (NO C.O.D.).
FORIEGN ORDERS ADD $2.00.
ORDERS PROCESSED IN 1-3 DAYS.
• DEALER INQUIRIES INVITED. •

MORE PROGRAMS AVAILABLE.
SEND FOR FREE CATALOG.

CRYSTAL MICROSOFT LTD.
P.O. BOX 440852
HOUSTON, TX.
77244

Makes checks as welcome as cash.

VIC-20 COMMODORE 64

THE RECIPE BOX

Now you can easily store and recall your favorite recipes on your Commodore computer. THE RECIPE BOX is a complete menu-driven disk system that comes with these additional features:

SEARCH BY INGREDIENT — Only have a pound of hamburger in the freezer? Let THE RECIPE BOX show you all the recipes that you have on file that use hamburger, or any other ingredient you choose

SEARCH BY CATEGORY — Code your recipes as to breakfast, lunch, dinner, snacks, etc

SEARCH BY CATEGORY/INGREDIENT — Any combination of the above

AUTOMATIC MEASUREMENT — THE RECIPE BOX will automatically scale up or down the amount of ingredients you need according to how many servings you want

SCREEN OR PRINTED OUTPUT — Have printed copies to use in the kitchen or give to friends

THE RECIPE BOX requires one disk drive and will run on a 5K VIC-20, Commodore 64. Please specify Send check or money order for $21.95 to:

Aries Marketing Co.
P.O. Box 4196
4200 Shannon Drive
Baltimore, Md. 21205
Md. residents add 5% sales tax

COMMODOR 64

"ABC" THE ALPHABET FOR PRESCHOOERS
TEACH YOUR CHILDREN THE ALPHABET EARLY WITH GRAPHICS AND SOUND
ONLY $18.95 ON CASSETTE
OR $21.95 ON DISK

VIC-20

(CASSETTES ONLY)
THE ALPHABET GAME ....$ 9.95(ea)
LEARNING NUMBERS 0-50 PART I....$ 9.95(ea)
LEARNING NUMBERS 51-100 PART II....$ 9.95(ea)
PART I&II ARE AVAIL. IN: ENGLISH; SPANISH; FRENCH; AND GERMAN VERSIONS PLEASE STATE LANG
ANY 3 VIC-20 CASSETTES FOR ONLY $ 25.-
Check or Money order+$1.50 Post. N.Y.Res+7¼tx.

THOR SOFTWARE
47 CHERNUCHA AVE.
MERRICK, N.Y. 11566

SOFTWARE SUPER SAVINGS

		TCE	LIST
12	A.E.	23.99	34.95
12	BANKSTREET WR.	49.99	69.95
12	CASTLE WOLFEN.	19.99	29.95
2	DONKEY KONG	36.99	49.95
1234	FROGGER	22.99	34.95
124	HOME ACCOUNTANT	47.99	74.99
24	JUMPMAN	27.99	39.95
12	KNIGHTS DESERT	27.99	39.95
1	LEGACY LLYLGAMYN	28.99	39.95
4	QUICK BROWN FOX	45.99	65.00
12	RANA DISK DRIVES	CALL	
12	SARGON II	23.99	34.95
1234	SNOOPER TROOPS	31.99	44.95
1234	SUSPENDED	33.99	49.95
1234	TEMPLE APSHAI	27.99	39.95
12	ULTIMA II	39.99	59.95
12	ZAXXON	26.99	39.95
1234	ZORK 1,2,3 EA.	26.99	39.95

	DISKS	ELEPHANT	MAXELL
12	SSSD	$17.99/10	—
23	SSDD	$21.99/10	$28.99/10
3	DSDD	$26.99/10	$40.99/10

(313) 528-1554
The Computer Express
P.O. BOX 569
DEPT. 8
TROY, MI 48099

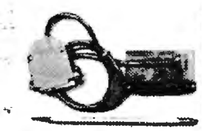
MASTER CARD/VISA/CHECKS/MONEY ORDERS/COD ACCEPTED
SHIPPING: US $2 CANADA $5 FOREIGN $10 MICH ADD 4% TAX

GET THE MOST FROM YOUR VIC-20/C64

CASSETTE INTERFACE

- USE ANY STANDARD CASSETTE RECORDER
- CONNECTS TO THE CASSETTE PORT
- [illegible] THE CASSETTE MOTOR
- [illegible] BATTERIES
- [illegible]
- THIS IS THE BEST ONE FOR LESS $$$
- [illegible] POSTPAID. ADD $2.50 FOR SHIPPING OUTSIDE US CANADA MEXICO

THE MODEM INTERFACE



- [illegible]
- [illegible]
- [illegible] ANY STAND[illegible]
- [illegible]
- [illegible] TYPE IN [illegible] GRAM
- [illegible] CANADA [illegible]

CHARGE OR COD ORDERS CALL
1-800-227-3800
ASK FOR OPERATOR 225

BYTESIZE MICRO TECHNOLOGY
PO BOX 12309 DEPT DM
SEATTLE, WA 98111
(206) 236-BYTE

CALL OR WRITE FOR DEALER INFORMATION

'PUBLIC DOMAIN'™ — SOFTWARE —

Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★
Over 1300 programs and growing.

VIC 20™

VIC collection #1 – 70+ programs – Tape/Disk – $10.00
VIC collection #2 – 70+ programs – Tape/Disk – $10.00
VIC collection #3 – 70+ programs – Tape/Disk – $10.00
VIC collection #4 – 50+ programs – Tape/Disk – $10.00

COMMODORE 64™

COMMODORE 64 #1 – 25+ programs – Tape/Disk – $10.00
COMMODORE 64 #2 – 25+ programs – Tape/Disk – $10.00
COMMODORE 64 #3 – 25+ programs – Tape/Disk – $10.00

PET® / CBM®

PET/CBM – 5 Utility – Tapes/Disks – $10.00 each
PET/CBM – 11 Game – Tapes/Disks – $10.00 each
PET/CBM – 6 Educational – Tapes/Disks – $10.00 each

DINSET: Reset Switch

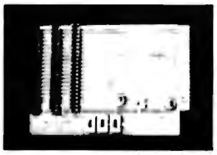
Works on Vic 20 or Commodore 64 – $5.00
All prices include shipping and handling.

We are YOUR world wide user software connection.
An alternative to the high cost of software.

CHECK, MONEY ORDERS, VISA and MASTERCARD accepted.

For A Free Flyer Write:

Public Domain, Inc.
5025 S. Rangeline Rd., W. Milton, OH 45383
Phone (513) 698-5638



VIC-20

QUALITY EXPANSION BOARD

ONLY $31.95

- UP TO 3 GAMES OR UTILITY CARTRIDGES.
- 3 LARGE SLIDE SWITCHES, NOT SMALL DIP SWITCHES
- RESET BUTTON, NO NEED TO TURN COMPUTER OFF BETWEEN GAMES.
- FUSED TO PROTECT VIC.
- HIGH QUALITY GOLD PLATED EDGE CONNECTOR AND NICKEL PLATED BOARD.
- SOLID RUBBER FEET SUPPORTS BOARD WHEN INSERTING CARTRIDGES.
- INSTRUCTIONS AND HINTS.

TO ORDER SEND CHECK (ALLOW 3 WEEKS) OR MONEY ORDER

MESI
P.O. BOX 51544
NEW ORLEANS, LA 70151

ADD $2.00 POSTAGE & HANDLING
LA. RESIDENTS ADD 3% TAX

VIC-20 IS A TRADEMARK OF CBM, INC.

VIC & 64

BE A COPY C.A.D.
(CASSETTE AIDED DUPLICATOR) NOW YOU CAN MAKE BACKUP COPIES OF ALL THE COSTLY, NON-SAVEABLE CASSETTE PROGRAMS YOU BOUGHT.

OUR BACKUP V1.0 UTILITY PROGRAM WILL LET YOU MAKE DUPLICATES THAT RUN.

BACKUP V1.0 WILL WORK WITH A STANDARD 5K UNEXPANDED VIC. MEMORY EXPANSION IS REQUIRED TO COPY PROGRAMS LONGER THAN 3K BYTES.

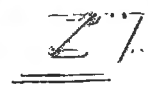
$24.95
PLUS $2.00 SHIPPING & HANDLING

SOFTWARE PLUS
6201 SUITE C
GREENBACK LANE
CITRUS HEIGHTS, CA 95610

VISA, MASTERCARD, AND MONEY ORDERS
CA. RESIDENTS ADD 6% SALES TAX.

VIC IS A TRADEMARK OF COMMODORE
916-726-8793

The UnKeyboard
a friendly alternative



★DRAW ★TRACE

★USE IT AS A GAME CONTROLLER
★USE IT AS A "MOUSE"

ATARI PADDLE COMPATIBLE
FREE PROGRAMS
STURDY LUCITE DRAWING ARM
2 PROGRAMMABLE FUNCTION KEYS

"THE HELPING HAND"
DRAWING BOARD DIGITIZER

$49.95

PERSIMMON PERIPHERALS
RTE. 2 BOX 2306A
CLAYTON GA 30525

VIC-20 C-64

VIC 20 POSTAGE PAID SOFTWARE VIC 20
100% MACHINE CODE
MONEY BACK GUARANTEE

Q. WARRIOR $13.95 U.S.

On a routine patrol of your defences, you discover that the Zoronian mutants are breaking through in huge numbers, you engage them in battle, almost at once you realize your laser snares, your main defences are malfunctioning, they are designed to detect and destroy mutant life forms. Instead they are hunting you. The mutants are bringing up their Z-rammer laser bolts. You summon help, none comes. You are alone, but you are a Q WARRIOR. Joystick required, fire in any direction

PLAGUE $13.95 U.S.

Use your super energy bolt to destroy the advancing Plague-Pests, but watch out for their guardians patrolling the screen. If anything touches you - Plague Pests, their stun bombs, eggs or patrolling guardians, you will crash to the bottom of the playing area while the pests will advance

CHRISTMAS SPECIAL -- BUY ANY TWO GAMES AND GET ONE FREE! Send cheque or money order to:

CHADWELL'S SOFTWARE
DEPT TS, 230, 4144A - 97 Street
Edmonton, Alberta, Canada T6E 5Y6

U.S. ORDERS WELCOME
DEALER INQUIRIES WELCOME

SM-TEXT + CUDA

A powerful combination of superb text processing and a name and address data base

maths and multi-search facilities make this pair into virtually a complete commercial administration system.

for cbm 8032 or 8096

Epson® MX70/MX80 & MX100 Replacement Ribbons

Manufactured by Aspen Ribbons, Inc. Buy direct from manufacturer & save. Standard ink color is black. Red, green, blue, brown, & purple colors are available for $2.00 extra per ribbon.

PRICES

Epson® MX70/80	$2.75 to $6.50 ea.
Epson® MX100	$4.75 to $9.75 ea.
Price depends on quantity ordered.	

CALL FOR FREE CATALOGUE.

*Aspen Ribbons, Inc. is not affiliated with any company mentioned in this ad.

Aspen Ribbons, Inc. 1700 N. 55th St.
Boulder, CO 80301-2796 (303)444-4054
Telex: 45-0055 End User: 800-525-0646
Wholesale: 800-525-9966

Advertisers Index

IF PERSONAL COMPUTERS ARE FOR EVERYBODY, HOW COME THEY'RE PRICED FOR NOBODY?

$1395* APPLE® IIe 64K

$999* TRS-80® III 16K

$1355* IBM® PC 64K

A personal computer is supposed to be a computer for persons. Not just wealthy persons. Or whiz-kid persons. Or privileged persons.

But person persons.

In other words, all the persons whom Apple, IBM, and Radio Shack seem to have forgotten about (including, most likely, you).

But that's okay. Because now you can get a high-powered home computer without taking out a second mortgage on your home.

It's the Commodore 64. We're not talking about a low-priced computer that can barely retain a phone number. We're talking about a memory of 64K. Which means it can perform tasks most other home computers can't. Including some of those that cost a lot more. (Take another look at the three computers above.)

By itself, the Commodore 64 is all the computer you'll ever need. Yet, if you do want to expand its capabilities some day, you can do so by adding a full complement of Commodore peripherals. Such as disk drives. Modems. And printers.

You can also play terrific games on the Commodore 64. Many of which will be far more challenging than those you could ever play on a game machine alone.

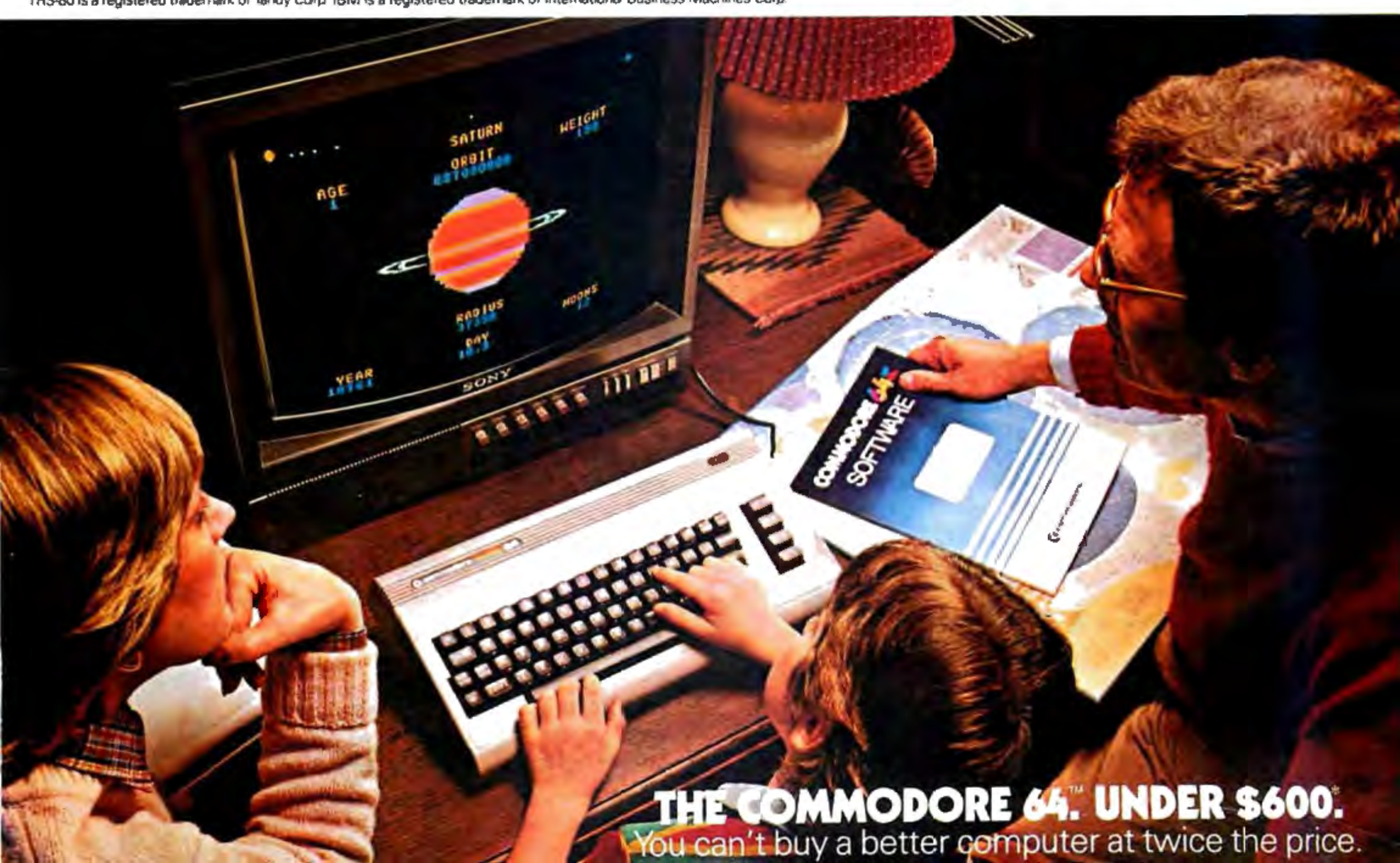
And as great as all this sounds, what's even greater-sounding is the price. It's hundreds of dollars less than that of our nearest competitor.

So while other companies are trying to take advantage of the computer revolution, it seems to us they're really taking advantage of something else: Their customers.

*Manufacturers' suggested list prices as of March 20, 1983. Monitor included with TRS-80 III only. Commodore Business Machines, P.O. Box 500R, Conshohocken, PA 19428; Canada–3370 Pharmacy Avenue, Agincourt, Ont., Can. M1W 2K4.

commodore COMPUTER

Apple is a registered trademark of Apple Computer, Inc.
TRS-80 is a registered trademark of Tandy Corp. IBM is a registered trademark of International Business Machines Corp.



THE COMMODORE 64.™ UNDER $600.*

You can't buy a better computer at twice the price.