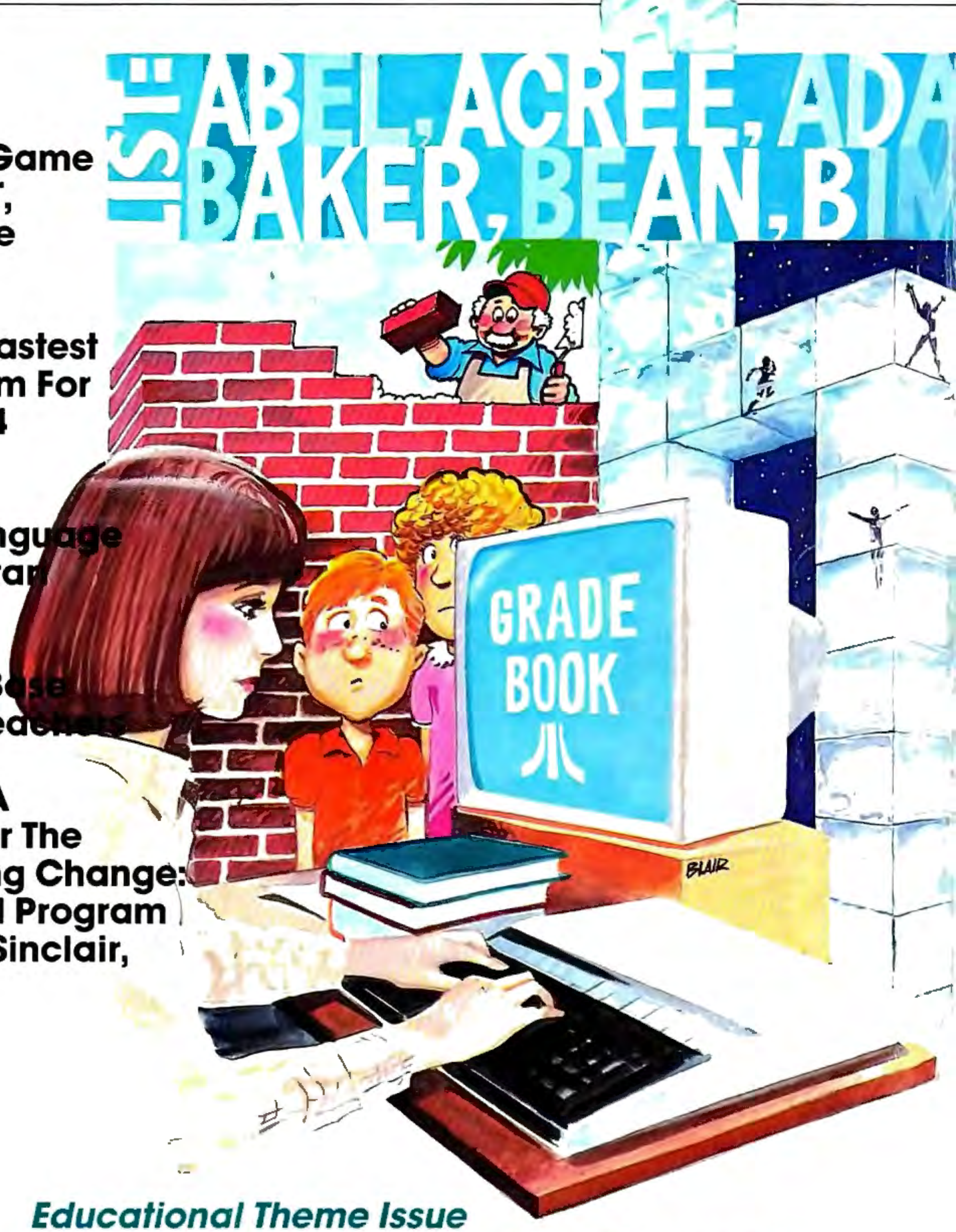
Smart Products — Computer Chips In Your Future

COMPUTE!

$2.50
September
1983
Issue 40
Vol. 5, No. 9
£1.85 UK $3.25 Canada
63379
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing



CAVES OF ICE
An Excellent 3-D Graphics Game For VIC, 64, PET, Atari And Apple

ULTRASORT
Our Best And Fastest Sorting Program For PET, VIC And 64

FONTBYTER
A Graphics Language Program For Atari

GRADEBOOK
An Atari Data Base Program For Teachers

***Plus* VIC Pilot, A Sprite Editor For The TI 99/4A, Making Change: An Educational Program For The Timex/Sinclair, And More!**

Educational Theme Issue
Games And Education, The New Classroom

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.

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

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.

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.

HEY DIDDLE DIDDLE™ Poetry in motion. Ages 3 to 10.

Kids love rhymes.

And since HEY DIDDLE DIDDLE features 30 classic rhymes with full color graphics and the neatest computer music you've ever heard, it makes rhyme games more fun than ever before.

Plus, it makes it fun for kids to read, helps them understand how words and rhymes create poetry and lets them take fragmented thoughts and rearrange them to form coherent verse.

The story of STORY MACHINE.™ Ages 5 to 9.

STORY MACHINE is like a storybook come to life. Using the keyboard, your children write their own fun little stories. The computer then takes what they've written and animates their story on the screen, complete with full color graphics and sound.

STORY MACHINE helps your children learn to write correctly, become familiar with the keyboard, and lets them have fun exercising their creativity at the same time.

KINDERCOMP.™ Numbers, shapes, letters, words and drawings make fun. Ages 3 to 8.

KINDERCOMP is a game that allows very young children to start learning on the computer. It's a collection of learning exercises that ask your children to match shapes and letters, write their names, draw pictures, or fill in missing numbers. And KINDERCOMP will delight kids with colorful rewards, as the screen comes to life when correct answers are given.

As a parent, you can enjoy the fact that your children are having fun while improving their reading readiness and counting skills.

FACEMAKER™ makes faces fun. Ages 4 to 12.

FACEMAKER lets children create their own funny faces on the screen. Once a face is completed, your children will giggle with delight as they make it do all kinds of neat things: wink, smile, wiggle its ears, or whatever their imagination desires.

Plus, FACEMAKER helps children become comfortable with computer fundamentals such as: menus, cursors, the return key, the space bar, simple programs, and graphics. FACEMAKER won't make parents frown because their children will have fun making friends with the computer.

SPINNAKER™
We make learning fun.

September 1983 Vol. 5, No. 9

GUIDE TO ARTICLES AND PROGRAMS

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 283 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 theme this issue centers on computers in education. We define education in its broadest sense – education as it permeates the home and the classroom. Several of our featured articles this month directly address this link and raise some critical questions at the same time. We look forward to your comments.

As prices continue to decline, and manufacturers begin looking forward to the Christmas season, expect some substantial purchasing opportunities. We expect to see more bundling of peripherals, software packages, and computers as vendors grow more aggressive. This lateral move will occur, in part, because basic computer prices have declined to near bottom, and future moves will have to be made through bundling and accessories.

With this issue COMPUTE!'s circulation approaches 400,000, and we expect to break the half million mark by December. Those of you who've been readers for a year or more will remember that just last October, we broke 100,000. We are proud of our leadership role in consumer computer publishing, and wish to thank you all, readers and contributors, for your support in the growth of COMPUTE!.

Gary R. Ingersoll has recently joined our staff and will be assisting in directing our future growth. Formerly president of the Chilton Company, the largest operating unit of ABC Publishing, Gary brings needed skills to our rapidly expanding division. He comes to COMPUTE! as president and publisher; I become chief executive officer and remain editor in chief.

Atari and Texas Instruments have both recently announced major revampings of the management teams responsible for their personal computer operations. Atari appears to be backing away from the \$100-\$200 price area and concentrating on building a family of systems which begins in the middle range. TI, on the other hand, appears committed to continuing to take on Commodore at the low end. A recent *Time* article indicates that IBM has now developed a 21 percent market share around the PC system. We still speculate that an IBM home PC (frequently referred to as the "Peanut") will debut soon. IBM does such a superior job of keeping the "lid" on leaks that our speculation is idle at best, but we think the middle-range market is so potentially lucrative for them that they won't stay away for long.

In a recent editorial we mentioned Adventure International in a context that was apparently misinterpreted by some readers. We want to make it clear that we respect Adventure International and their business practices, and that they have not been involved in any effort to "recruit" COMPUTE! staffers.

As our magazine and book publishing operations continue to grow, we are still looking for additional editorial support. If you're an experienced writer or journalist who has a personal computer background as well, drop us a résumé. Our growth has been consistently strong, with our staff tripling in the last year. We're located in the central Piedmont region of North Carolina in a metropolitan area selected recently as one of the three best living/working locations in the nation.

Robert Lock

Coming In October

Special Games Issue

Telegaming

First-Rate Games For Several Computers – *Dragons, Moving Maze, RATS!*, And More

EXEC: A New Command For Commodore Computers

A Tutorial On TI Sprites

Atari Master Directory

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

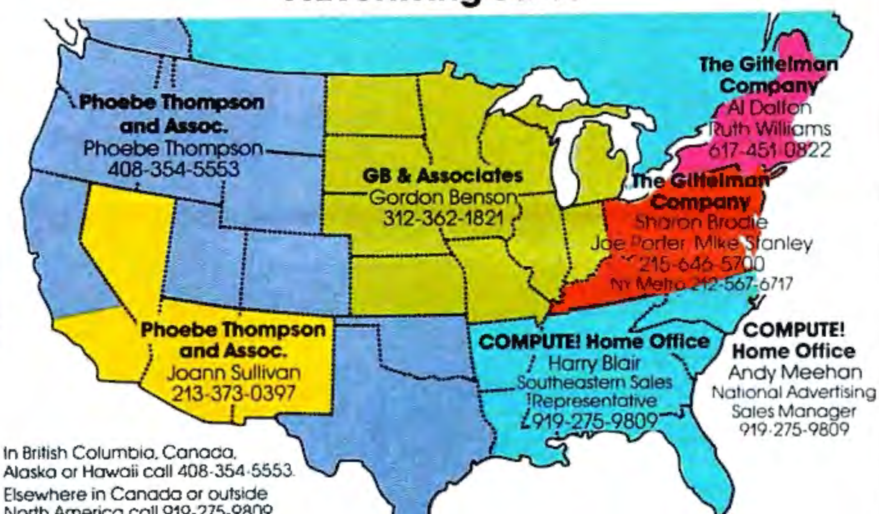
COMPUTE! Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403

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!

COMPUTE!'s Programs

I have learned more from your magazine than from textbooks on computing, but one thing puzzles me. What's your policy toward the programs you publish in the magazine? They often take a long time to type in and I usually go on to add embellishments here and there, or change them to run on other computers. Are these programs in the public domain? Could they be traded with my friends?

One note: I often type in programs and then later forget the instructions or which issue I'd gotten them from. So, I now always put REM statements into the first few lines of the program which have the date and page number where the program documentation can be found. I can't count the number of times I've been glad I do it.

Mary Howe

Programs published in COMPUTE! are in the same legal category as material published in any other magazine. They are all copyrighted; they're not in the public domain. When you buy an issue, you then have the right to make a copy of the programs therein. We realize, however, that some of the programs are long and take some time to enter into the computer. For this reason, it's permissible for you to give a copy of a COMPUTE! program to a friend or members of your user group who subscribe to the magazine.

No program in the magazine, however, may be sold, traded, or otherwise distributed for profit. Nor may any program be given to someone who does not own the issue in which the program was printed.

TI-99/4 And 4A Differences

What are the programming differences between the TI-99/4 and TI-99/4A?

The TI-99/4 has 256 more bytes of available RAM than the TI-99/4A, so a very long program may run on the TI-99/4 and not the TI-99/4A.

The TI-99/4A has lowercase capability, so some techniques are possible on the TI-99/4A that are not possible on the TI-99/4. For example, for graphics you can redefine characters using lowercase letter codes, then PRINT the letters rather than using the CALL HCHAR or CALL VCHAR statements. If you redefine the letters a and b (characters 97 and 98) to draw a car, for example, you can then PRINT ab to get a car. To convert for the TI-99/4, remember that the lowercase letters start with ASCII Code 97. The equivalent statement would be PRINT CHR$(97)&CHR$(98). A program using redefined lowercase letters that is typed on the TI-99/4A can be SAVEd then loaded onto a TI-99/4 and will work fine.

The keyboards on the two computers are different too, and several of the symbols are in different places. This change affects the CALL KEY statements. CALL KEY(0,KEY,STATUS) on the TI-99/4 is used to scan the whole keyboard, and devices 3, 4, and 5 for the first parameter are for "possible future devices." On the TI-99/4A keyboard, device 3 scans the whole keyboard in BASIC, device 4 is for Pascal, and device 5 is for BASIC using both capital and lowercase letters. Device 0 is used to indicate the same device as previously scanned. Many programs now use CALL KEY(3,KEY, STATUS) for the TI-99/4A. To write your programs compatible for both computers, use CALL KEY(0,KEY, STATUS).

The split keyboard also presents some variations. The statements are CALL KEY(1,KEY1,STATUS1) and CALL KEY(2,KEY2,STATUS2). Some of the KEY values returned are different: G, B, SHIFT, SPACE, comma, period, /, =, semicolon, and ENTER. If you use the standard arrow keys (E, S, D, X and I, J, K, M) you'll have no problem. The diagonal arrows are also the same for both keyboards. In general, avoid the middle area keys and the keys at the extreme right of the keyboard. For games written for the TI-99/4 in which you press ENTER to fire, you may need to press the period to fire on the TI-99/4A.

There may be a problem in testing for zero on the TI-99/4A when using the split keyboard scan. After the CALL KEY statement, use logic such as IF K+1<>1 rather than IF K<>0.

VIC Word Processing, Disks, And Machine Language

I am presently trying to learn machine language (ML); to this end, I bought the HES MON ML monitor. The problem is that I have been unable to use labels with it, and was wondering if it is possible to do so. (I have been unable to find a VICMON anywhere, so I have no means of comparing the two – does the latter allow the use of labels?) I would also really appreciate an expla-

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.

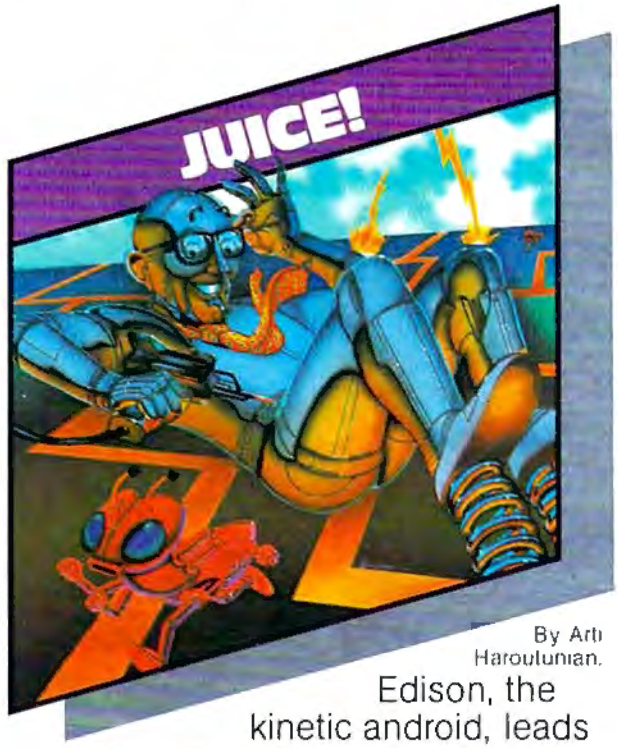
Just like VIC 20 and Atari owners. Only faster, more complex, and more challenging, too.

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.

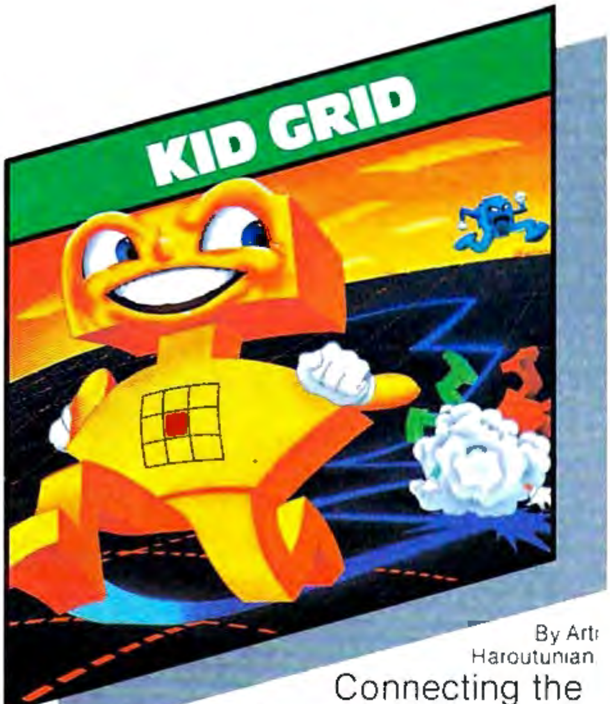
All he wants to do is build his circuit boards and go with the flow. But things keep getting in the way.

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?

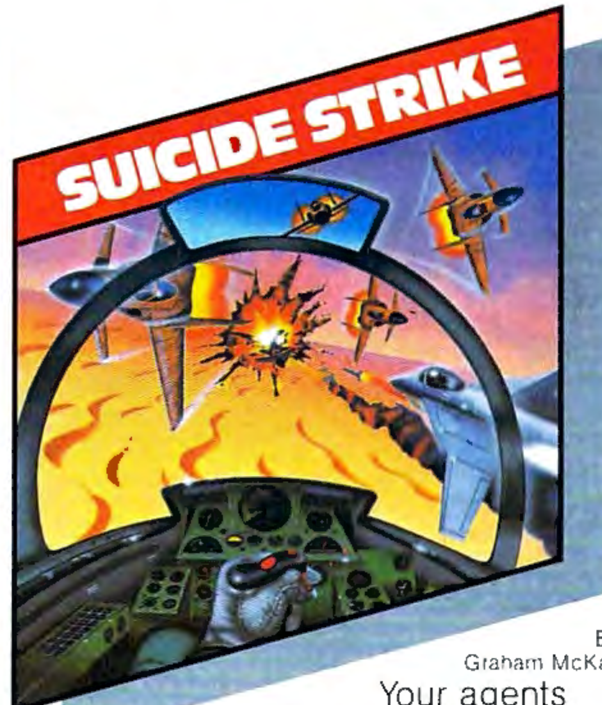
Wrong. Because the bullies are in hot pursuit!

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 nation of precisely what zero-page memory is, and which sections of it can be used by an ML program without affecting the operating system. (Leventhal's *6502 Assembly Language Programming* doesn't deal with such particulars, and it's hard to get an understanding of them by simply perusing the memory maps.)

I also want to use the VIC as a word processor. I will, of course, need a disk drive and a printer, but I'm not sure that I want to use the Commodore products in either case. In regard to the drive, it probably *would* be wisest to get the 1541, but I was wondering if I could get more for my money if I were to buy a bare Tandon or Pertec, or a used Apple drive for $200 or so and do the rest myself. In other words, would the task of interfacing and writing a DOS be excessively difficult, considering that the 1541 goes for only $340?

As for the printer, this is more difficult: I would obviously like to get a letter-quality printer, but my budget can't go much beyond $350. So I've been thinking about getting a used I/O Selectric and interfacing it to the VIC. An article in the April and May 1981 issues of *Radio-Electronics* described the general process, and it doesn't seem too difficult – besides, it looks like fun!

Peter Jeffe

Jim Butterfield replies...

1. *Most monitor systems are composites of utilities: assembler, disassembler, fill, hunt, display, etc. HES MON, VICMON, SUPERMON, and similar packages contain "nonsymbolic" assemblers; that is, you cannot use labels. These are not sold as assembler packages.*

These assemblers, I should point out, are effective in reducing transcription and lookup errors; for small programs they can be quick and useful; backward branches can be entered by inspection; forward branches can be guessed, then reentered when the actual address is established; they make no special demands for memory space or disk facilities.

But a big assembler is a whole other thing, and worth the cost when you start writing programs that are over, say, 30 instructions long. In my opinion, their major advantage is this: since you keep source code, you can make program changes without the need to type in all the coding once again. A big program will probably need several rewrites; a full (symbolic) assembler is very valuable at that time. However, I like to keep beginners closer to the machine code and encourage nonsymbolic assemblers for early learning programs.

2. *Zero-page memory is memory that extends from hex addresses 0000 to 00FF (the first two digits represent the "page"). It's important for three reasons:*

a. *(minor reason) There's an addressing mode that allows faster and more compact access to zero page than to other parts of memory. Not too important; time and space are seldom urgent machine language program considerations.*

b. *(major reason) A major method of "reaching" information anywhere in memory is indirect addressing, more specifically, indirect indexed addressing. This addressing mode needs to hold its indirect address in zero page. Zero page is in short supply; many users like to "conserve" the area for indirect address usage.*

c. *(pragmatic reason) The operating system uses zero page a good deal for BASIC and for interrupt processing. To keep the operating system healthy, you need to respect the important usage areas. Many users (who want lots of zero page) "swap out" little-used memory for their ML programs, and put it back before returning to BASIC.*

Most 6502 reference books deal with the chip "in a vacuum" – not connected to a real system. Thus, you get no hint as to where programs should be placed, how to invoke input and output, and how the monitor systems work. This makes it very difficult for the beginner – that first step is a big one.

A recent book, Machine Language for Beginners, *by Richard Mansfield (*COMPUTE! Books*), does deal with these problems on a variety of machines and may offer more help in this area. This is not said as a review or as an endorsement, but the book does approach the microprocessor as seen within its computer environment to a greater extent than previous publications I have seen.*

3. *I have a strong bias towards the manufacturer's product line on disk systems. You can go other ways; but commercial products, and club distributions, are likely to heavily favor these format disks. Building your own interface and writing your own DOS is not a trivial task; if it's a challenge you would enjoy, go for it. If your objective is to get a system up and running in reasonable time, reconsider.*

Many computer hobbyists have adapted Selectric devices; some have complained that the machines are not durable, having been designed for a lighter duty cycle than is found on computer word processors. Check with user groups for their reaction.

64 Video Glitches

I'm disappointed with the quality of the Commodore 64 video display. For example, when selecting black characters on a blue background, every other character is badly smeared. Also, when executing a program, small "birdies" appear randomly all over the screen. These are about one pixel in height, three to eight pixels in width, and appear in the same color as the characters.

Is there a fix for these problems?

Some colors don't seem to work well together on the Commodore 64; you might try combinations of foreground and background colors to see what works best on your machine.

If you are using a TV set, look for solid connections

LUXURY, FACILITY OR NECESSITY?
There are many things which make life easier for us; but everybody has his, or her own personal, individual, desires. For one it may be a luxury – for another sheer necessity.
If you use a commodore, our software-products should be a necessity.
CREATE YOUR LIFE MORE SUCCESSFUL!
WHAT'S
?
SM-CUDA User Guide
SM-TEXT
SM SOFTWARE INC.
IS A SUBSIDIARY OF
SM SOFTWARE AG, MUNICH,
ONE OF EUROPE'S TOP
MICRO SOFTWARE HOUSES.
U.S. computer technology, allied with German software engineering, has produced high quality programming aids such as:
● SM KIT – program de-bug aids
● ESH – ergonomic screen handler
● ISM – index sequential file manager
● LOS – loadable operating system
and high level application programs such as
● SM-TEXT - advanced text processor.
ASK FOR DETAILS.
SM SOFTWARE INC.
252 Bethlehem Pike
Colmar, PA 18915
SOFTWARE MADE IN MUNICH
Dealer inquiries invited.

(try wiggling things gently) and make sure your TV/COMPUTER slide switch is firmly over to the COMPUTER side. Try rearranging the cable which connects the computer to the TV set: sometimes interference is picked up along the way. Even moving the computer can often help.

On the other hand, if you are using a monitor rather than a TV set, there are other things for you to keep in mind. Commodore will soon be announcing a new interface – and a new monitor – that should significantly improve picture quality. Still on the subject of monitor interfaces, some users find that they can get better character definition by a variation in the wiring of the interface. Normally, pin 4 of the video connector is used for video out on the Commodore 64: some users like the improved contrast that may be achieved by connecting pin 1 (luminance) to pin 4 and then feeding the composite signal to their monitor.

We understand that the screen hash that you call "birdies" can be eliminated completely by the addition of a small capacitor to the video circuitry of the Commodore 64. Contact your dealer for further information.

RAMDISK

Could you tell me what a RAMDISK is?

Joe DeNicola

It is possible to use large amounts of extra RAM memory as a simulated "disk drive." In fact, the decreasing cost of RAM chips has made "memory drives" quite popular. The advantage of a memory drive is that it is extremely fast – faster than any non-solid-state peripheral. Unfortunately, most of these memory drives lose what they've stored when you turn off your computer. Battery-protected memory drives are available, but they are relatively expensive, since they require low-power special CMOS memory chips.

Specifically, the RAMDISK is 128K of "bank-selected" (you can call upon "banks" of 16K) memory. It includes software to use the extra memory as a simulated disk drive.

Finding Atari Addresses

As an owner of an Atari Assembler Editor cartridge, I'm still having trouble locating the hexadecimal address locations for BASIC commands. Any suggestions?

Eric Ermert

It is possible to call some of the ready-made routines found in the BASIC cartridge, but you must remember that they are designed to be used by BASIC itself, not external ML programs you write. You can read about the internal workings of Atari BASIC in COMPUTE!'s new Atari BASIC Sourcebook.

A better solution is to write your own routines. It isn't that difficult. For example, the SOUND command stores the values in the POKEY chip, which, among other things, is responsible for generating sound (see "Atari Sound System," COMPUTE!, January 1983). You can call any of the graphics routines BASIC uses (PLOT, DRAWTO, GRAPHICS). These are found in the Atari operating system, not the BASIC cartridge, and they are well-documented and designed to be called by your ML programs. Some references are De Re Atari, Atari Technical Notes, *and Bill Wilkinson's COMPUTE! column "Insight: Atari" (especially February 1982).*

Atari Binary LOAD/RUN From BASIC

The use of binary files in BASIC programs is increasing as Atari programmers become more sophisticated. It is not generally known that you can use the DOS command L, including the "/N" option, directly from BASIC. The necessary routines are resident in DOS itself, not DUP. They will LOAD, INIT, and RUN (or not RUN) any binary file that DOS can handle, including compound files. Control is returned to BASIC for files which ordinarily return to DOS. Here is one method. Just insert your file name in F$.

```
100 DIM F$(16):F$="D:GAME.OBJ":F$(LEN(F$)
    +1)=CHR$(155):POKE 5534,0:POKE 5535,192
110 X=ADR(F$):Y=INT(X/256):POKE 853,Y:POKE
    852,X-256*Y:X=USR(ADR("hL)U"))
```

The USR string which is not listed correctly by a printer is:

small h, capital L, inverse SHIFT 0, CONTROL U

To LOAD and INIT but not RUN, POKE 5534,192 in line 100.

The USR code, PLA, JMP $15A9 calls the resident DOS routine used by option L.

Note: Bill Wilkinson in his COMPUTE! column eloquently explains the advantages of following Atari protocol. I am embarrassed to point out that I violate that excellent advice by using a specific DOS routine which may be altered in future DOS revisions. It is safe to use this quick and dirty trick in your personal programs, but don't distribute it. Use my "Autotype" in *COMPUTE!'s Second Book of Atari* to insert the binary file directly and safely into any commercial BASIC program.

Forrest Meiere

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.

SMART PRODUCTS

Kathy Yakal, Editorial Assistant

All personal computers contain a microprocessor, the "brain" of the machine. But even if you don't own a computer, you probably have several of these tiny brains in your home, your office, or even your car. They are, in a sense, tiny computers.

Cars that act as their own mechanics. Cash registers trained to be marketing experts. Washer-dryers that take better care of your clothes than you could, and computers that guard your home. All of these "smart products" – products that can make decisions and monitor themselves – are no longer science fiction fantasy, thanks to the introduction of microprocessors in 1971.

A microprocessor is organized, compressed electronic circuitry which can execute programs and respond to changing conditions. It's about the size of your little fingernail and consists of a small silicon "chip" with complex patterns of lines etched on it.

Microprocessors replace circuitry many times their own size. For instance, if you take the back off a transistor radio, most of the components you see inside could be replaced with a single, small microprocessor.

There are several advantages to using this new technology in the production of consumer products. For one thing, because they are so much smaller than discrete (singular) circuits, products which house them can be much more compact. This is especially true of microcomputers, which wouldn't exist without them.

Microprocessors perform functions fast and precisely. They are easier to produce than discrete electronics and they don't wear out as quickly, since they have no moving parts. Most important, they are intelligent. They can be programmed to make decisions based on predefined conditions.

The Consumer Market

You can identify a consumer product that contains a microprocessor. There are generally no dials to turn or buttons to push or timers to set. Quite often there will be a flat membrane-type control panel that is responsive to the slightest touch. And you may be able to tell if the product contains a microprocessor by the type of input required from you (for example, instead of indicating how long you want your clothes to dry, you would only need to indicate the fabric type – the microprocessor would know how long and how hot to run).

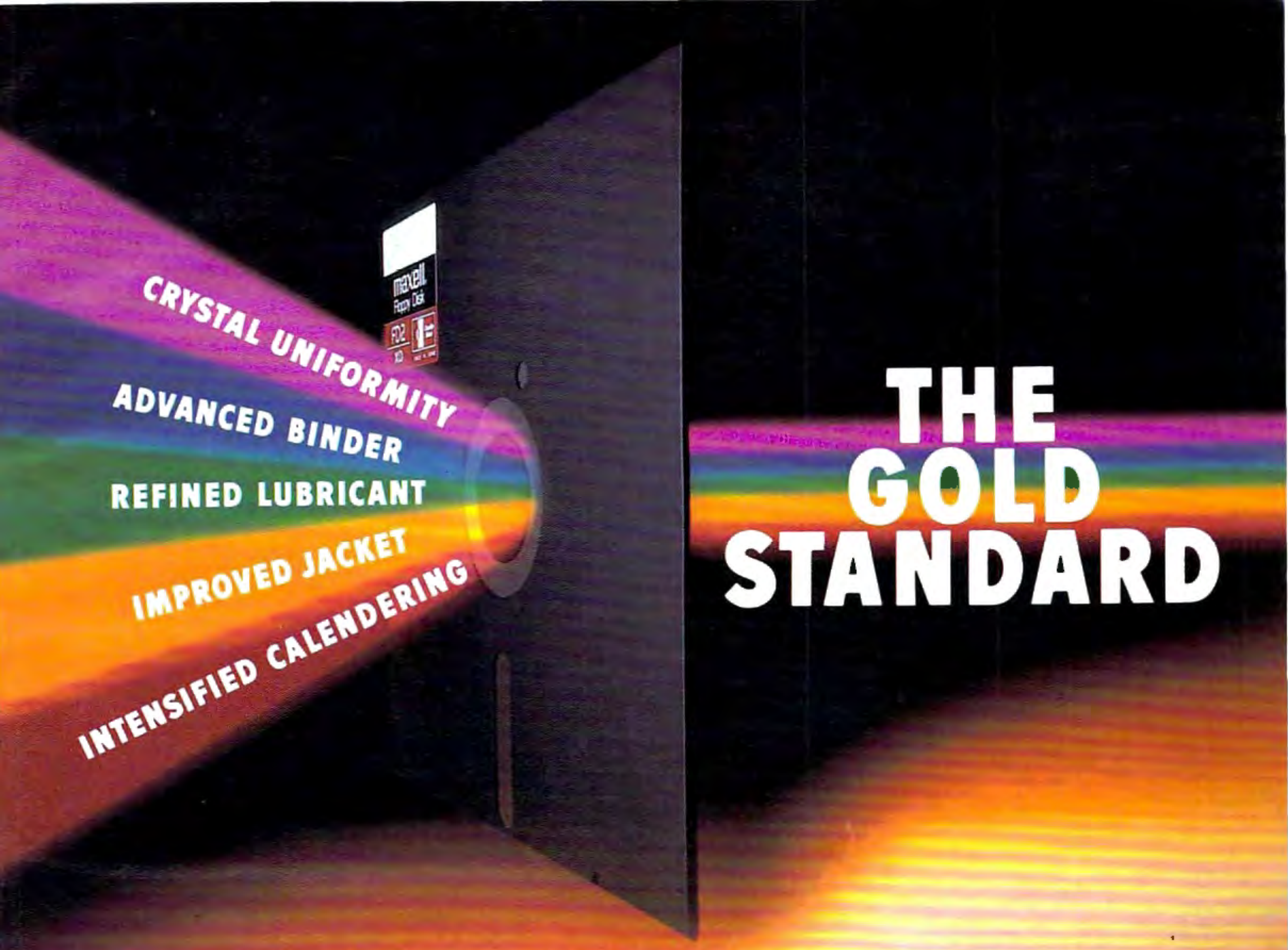
Manufacturers of consumer products don't automatically use microprocessors in every product they could. The same technology that brought microprocessors into being also facilitated better design of microelectronic circuits; each has its own place. Further, some manufacturers are holding back to gauge public acceptance of the new breed of consumer products. This is critical: it's possible to make a washing machine that talks, but do people want that?

In the following product descriptions, we have chosen a few manufacturers which are representative of several industries. These companies are not the only ones using microprocessors. Also, these companies do not use microprocessors in all of their product lines. What we are looking at is still a state-of-the-art technology.

Self-monitoring

First let's explore some home appliances that use microprocessors. Refrigerators have been able to monitor themselves for a long time. You set the temperature level desired, and the unit shuts off upon reaching it.

The Whirlpool Corporation makes refrigerators that c even more. They beep if the



You can wait for industry standards to mandate improved performance. Or you can have it now on Maxell. The Gold Standard.

The refinements of The Gold Standard, from oxide particles to lubricant to jacket, are uniquely Maxell. And therefore, so are the benefits.

Our unique, uniform crystals assure dense oxide packing. So you begin with an original signal of extraordinary fidelity. A signal we safeguard in ways that leave industry standards in our wake.

An advanced binder bonds oxides to the base material preventing time and money-wasting dropouts. Calendering then smooths the surface for a read/write signal that stays clear and accurate. And lubricants reduce friction between head and disk for a longer media and head life. To house it, we then constructed a new jacket heat-resistant to 140° F to withstand drive heat without warp or wear. And created the floppy disk that leads the industry in error-free performance and durability.

All industry standards exist to assure reliable performance. The Gold Standard expresses a higher aim: perfection.

maxell®

IT'S WORTH IT.

Computer Products Division, Maxell Corporation of America, 60 Oxford Drive, Moonachie, N.J. 07074 201-440-8020

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 beeping button tells you your write protect feature is keeping your information safe.

The remaining buttons beep when touched, and provide readouts on density storage, error status, and drive number.

This button beeps when you touch it, and the LED readout 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 and costs ⅓ less than their disk drive. 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 actually beep and 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. It lets you switch from a single density of 90,000 letters to a double density of 180,000 letters, on a single diskette. 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.

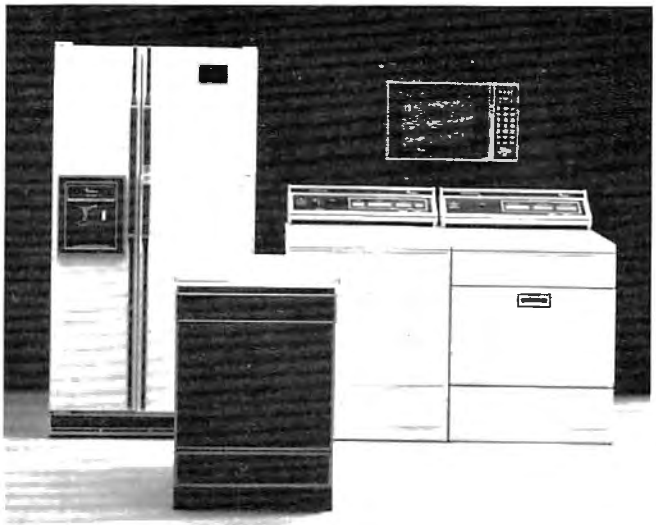
Just common sense.

RanaSystems *Always a step ahead of the originals.*

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

Available at all participating Computerland stores and other fine computer dealers.

* Apple is a registered trademark of Apple Computer, Inc. * Atari is a registered trademark of Atari, Inc., a Warner Communications Company. See us at the West Coast Computer Show. Circle 44



Whirlpool uses microprocessors in the design of many home appliances for more efficient operation.

door is left ajar. They let you know when the coils need dusting, or if there is a problem with the temperature inside, or if there is anything going on which will keep the machine from doing its best job of keeping food fresh.

Some microwave and toaster ovens use microprocessors. The purpose of these appliances is to cook food, and microprocessors are employed to do that better and faster. Whirlpool has some microwaves that will monitor food temperature so that it won't be overcooked, and will time several dishes so they are ready at the same time. Digital clock functions are built in.

Washing machines and dryers are more efficient when microprocessors are monitoring their functions. Some of the newer Whirlpool washers do not require you to choose a setting; if you just turn them on, they will go into a standard warm wash-cold rinse cycle. If you want to give additional information about the fabric you're washing, there is a flat membrane panel about a foot long listing your choices. If you make a choice that could damage your clothes, like selecting "knits" along with "hot water," the washer will let you know that this could be a mistake. It will make a low "boop" sound when something's wrong, a high "beep" when everything's all right.

Dryers also use microprocessors to determine the safest way for fabrics to dry. You select the fabric type, and the machine decides how hot the dryer should be and even how long it should dry (unless you set the manual timer). It also automatically fluffs the clothes at five-minute intervals to keep them from wrinkling.

Sony uses microprocessors in virtually all of its home entertainment components. The most common use in television sets is in the channel changer. Instead of a standard dial, many television sets now have a push-button control with a digital display; you can turn the television off or on, choose the channel (either by entering the number or scanning up and down), and adjust the picture by pressing some buttons. Increasingly popular remote control features are also made possible by microprocessors.

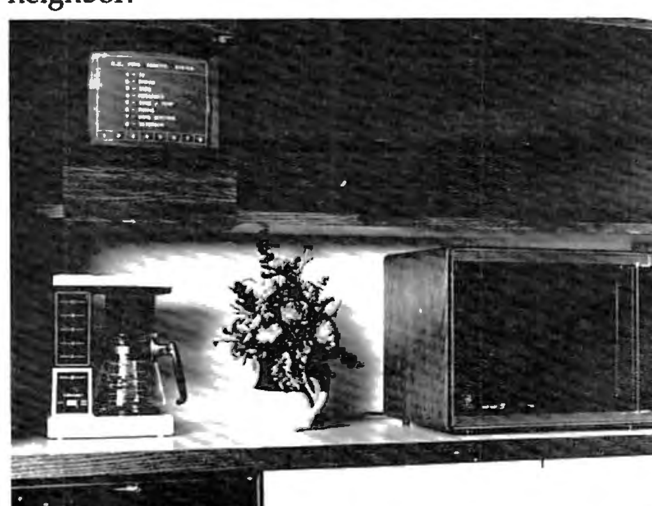
Tape players, stereo receivers, and turntables are also making increasing use of microprocessors. A variety of search, memory, and automatic play features can now be controlled by them.

Possibly the most sophisticated use of microprocessors in home entertainment systems is to be found in video cassette recorders. Virtually all of their working parts use them because the requirement for accuracy is so great.

Streamlining Daily Routines

Running a household may be simplified in the future because of these advances in electronic technology. General Electric has found a way to make it even easier with their "Homenet," a computer-based home automation system. Controlled by a video screen and keypad, the system allows control and monitoring of heating and air conditioning, security and fire systems, lights, appliances, and entertainment components. It uses existing house wiring and electrical current to send signals to appliances. Consequently, the system is compatible with any brand of home appliance.

Built-in telephone circuits allow complete access to the system by phone, so you can call your "Homenet" and tell it what time to start dinner or the washing machine, or to change any earlier instructions. The phone capabilities also enable a home security system, so that if your smoke or burglar alarm goes off, the computer is alerted to call the police, the fire department, or a neighbor.



The GE "Homenet," a home automation system, lets you control household appliances, entertainment components, and security systems through one central keypad and video screen.

The "Homenet," of course, uses microprocessors in its computer unit. It is possible to use your own home computer to set up such a system, but this can require a fairly sophisticated understanding of computers and interfacing.

Electronic Motoring

The use of microprocessors in automotive electronics is expanding rapidly, according to a representative of the Nissan Corporation. Some of Nissan's top-of-the-line cars (for example, the turbocharged ZX) use microprocessors, especially in engine control. All fuel injection functions are digitalized; that is, everything necessary for getting gasoline into the engine efficiently is computed. Fuel economy is constantly monitored based on current conditions, so if you're going into a headwind, the computer tells you how long your fuel will last if the wind keeps up.

Microprocessors measure and correct the EGR standard. The audio warning system – the ability of the car to say "Your door is open" or "Fuel is low" – is prioritized. The microprocessor decides which condition is more critical, and warns you of that one first. This is where microprocessors are clearly revealed as a technological leap: they can make *intelligent decisions.* The most visible use of microprocessors in Nissan cars, however, may be in the scanning and memory capabilities of the digitally tuned AM/FM radio.

The digital instrument display of this 1983 Datsun 280ZX illustrates the numerous features now controlled by microprocessors in cars.

Nissan has some more exotic uses planned for microprocessors in its 1984 models. One of these, the "knock sensor," will adjust the spark advance to help prevent the "pinging" created by low-octane fuel.

The Ford Motor Company says that its 1983 Lincoln Continental is the best example of the new electronic technology in the company. Microprocessors are used for five different functions in this car: in the radio-electronic cassette; the electronic "instrument cluster"; the "trip minder" (a trip computer that calculates time, engine functions, etc., when you're driving a long way); a keyless entry system (a panel of five push buttons on the outside of the driver's door that requires certain entry codes to lock and unlock the car doors and trunk); and the EEC IV Electronic Engine Control System, a fourth-generation engine-control system developed jointly by Ford and Intel using a 16-bit microprocessor.

Additional computerized functions you may see on the 1984 Ford models include a digital thermometer (for outside temperature); digital temperature control; a digital fuel gauge; and electronic air suspension (springs replaced by air bags using a height sensor).

Increased Business Efficiency

All of these products utilize technologies that now exist (or soon will) in your home or garage. But businessmen have not ignored microprocessor-based technology either. The chips will also have a dramatic effect on ordinary commercial transactions.

It's becoming common these days to go to the grocery store and have your purchases rung up by a clerk who barely touches any keys on the cash register. This "price look-up file" goes one step further: the item being scanned shows up on a digital display with its name and price. You then get a printout of what you bought and what it cost. This is all accomplished by National Cash Register (NCR) through the use of microprocessors.

Information gathered by such accounting is not just useful to the customer and a time-saver for the clerk. Marketing experts can use the data to tell if, for instance, a person who buys a certain brand of toothpaste also buys baby food or exotic hors d'oeuvres or cigarettes. In that way, they can get a better idea of what market should be targeted for their advertising. Store managers also use the information to gauge the effectiveness of store displays, shelf height, or their own advertising.

Replacing People

You may already be accustomed to banking at an "instant cash machine." Though used to a degree for about ten years, the machines have gained real public acceptance only in the last couple of years, says a representative of NCR. The same kind of microprocessor-based technology found in these machines may also put computer terminals in places where you're accustomed to seeing people: at gas stations, in hotel lobbies, and at airports.

This is not to say that computers will completely replace clerks within the decade. But NCR will be introducing self-service terminals to streamline certain businesses. At a gas station, you may be able to put your credit card in a slot, enter your secret code and the amount and kind of gasoline you want, and the computer inside

Introducing the KoalaPad™ Touch Tablet for less than $125.00. It's the friendliest innovation in personal computing.

With a touch of your finger, the KoalaPad tablet takes control of your computer. Faster than a paddle controller. More versatile than a joystick. And much friendlier than a keyboard.

The KoalaPad tablet is compatible with most game software. And most KoalaPad sets come with a KoalaWare™ software program—Micro-Illustrator.™ It's the best way yet to create beautiful, high-resolution computer graphics.

And there are lots more KoalaWare programs for computer fans of all ages. Like Dancing Bear,™ the funny, furry computer cabaret. Spellicopter,™ the fast action spelling game. And Spider Eater,™ the lively music learning adventure.

Just try the KoalaPad Touch Tablet. You'll discover a whole new dimension in home computing.

To locate the dealer near you, call toll-free 800-227-6703 (in California, 800-632-7979).

Koala Technologies Corporation

We make computing more personal.™

Trademarks: KoalaPad, KoalaWare, Spider Eater and Dancing Bear are trademarks of Koala Technologies Corp. Micro-Illustrator is a trademark of Island Graphics, and Spellicopter is a trademark of DesignWare Inc.

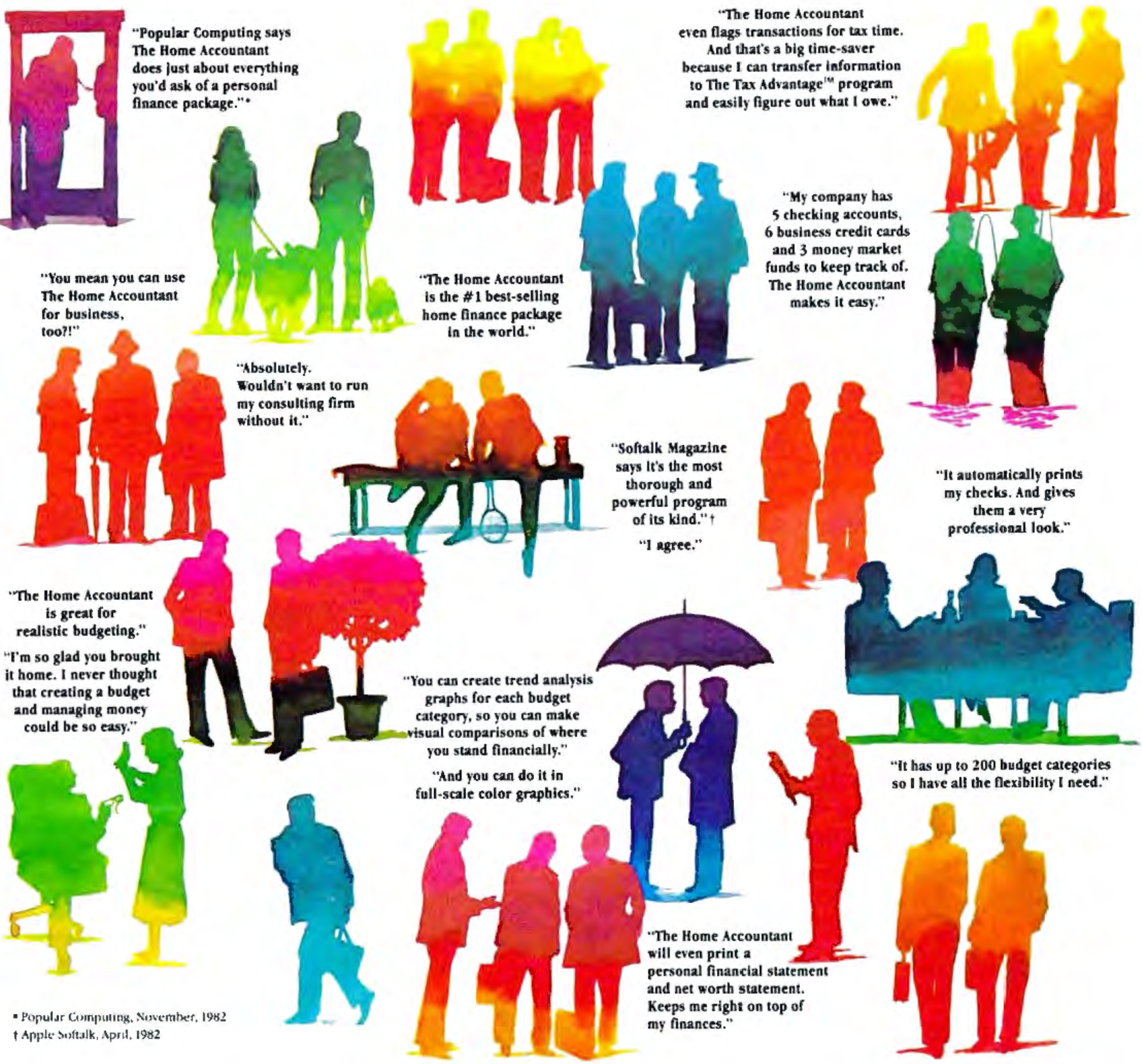
would dispense the gas and charge it to your account.

In a hotel lobby, you could have the option of confirming your reservation and getting your room assignment from a terminal in the lobby. This terminal would interface with the guest accounting system that is already in use in many hotels, to provide you with a computerized bill at check-out time.

Terminals programmed with flight information may begin to appear in shopping malls and other convenient locations soon. You will be able to get flight information and make reservations on these; then, when you arrive at the airport for your flight, another terminal will check you in and give you your ticket and boarding pass.

The Invasion Of Intelligence

However, for all of the seeming inevitability of the invasion of microprocessor intelligence into our daily lives, it's not entirely beyond question or modification. Manufacturers of consumer products are watching public reaction to these new inventions closely. The technology is there. What remains to be seen is how people will feel about the new smart machines. ©



* Popular Computing, November, 1982
† Apple Softalk, April, 1982

Everyone's talking about The Home Accountant.™

Is it because it's the #1 bestselling home finance package in the world? Or because it's extremely thorough and powerful and easy to use? Or because it's great for home and business use? Or because it has up to 200 budget categories and handles up to 5 checking accounts?

Yes. But there are a lot more reasons why people buy The Home Accountant. And why you will, too.

Because The Home Accountant can literally save you hours of time. And take the headache out of handling your finances. Whether it's setting up a budget, cataloging your expenses, balancing your checkbooks or handling your credit cards and money market funds. For personal or business use.

The Home Accountant will even print net worth and financial statements. Not to mention being a lifesaver at tax time. Especially when you're able to transfer information onto Continental's The Tax Advantage™ program and figure out what you owe. Quickly.

In short, The Home Accountant is the most effective software program there is for managing your money. And managing it easily.

Stop by your Continental Software dealer today and pick up The Home Accountant. You'll see what everyone's talking about.

The Home Accountant is available for Apple II/IIe, IBM PC/XT, Atari 400/800/1200XL, Osborne,® TRS-80 Models III/4, Commodore 64, Texas Instruments Professional, Zenith Z-100/110, Compaq and KayPro computers. Actual budget capacities will vary with each computer.

For your free 48 page booklet, "Tips For Buying Software," please write Continental Software, 11223 South Hindry Avenue, Los Angeles, CA 90045, 213/417-8031, 213/417-3003.

Continental
Software
A Division of Arrays, Inc.

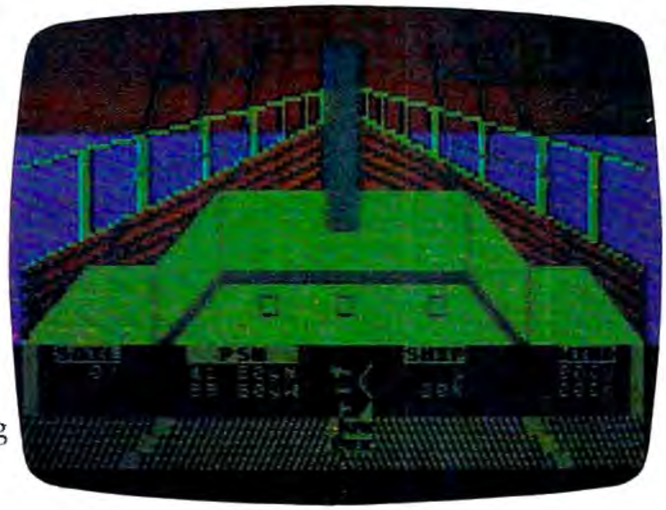
Games That Teach

John Blackford, Assistant Features Editor

Ever since computers were first pieced together out of radio tubes, their potential as teaching machines has fascinated educators. One of the first ideas was to let computers drill students on important skills. Later, programmers enlivened these practice sessions with games and arcade-type action. Today, innovative teachers and game designers are working to create new software and develop teaching methods that make learning itself part of the game. Some of these new products and ideas are finding their way into the home – via the personal computer.

Walk into any video arcade and you'll see teenagers in a twilight world, hunched over machines about the size of small refrigerators. Seemingly oblivious to the beeps and whistles that fill the air, they concentrate on the small screen. They can play for hours. This scene is noticed by some educators, who would like to tap that intensity for the learning process.

Some of them have. Educational games have acquired a reputation for being rather dull, and many are. But that is changing. Publishers of educational material are developing computer programs; makers of computer games are diversifying into educational products; and fledgling school computer ventures are maturing into active resource centers and using the best software available (see "Computers In School: New Approaches," in this issue).



The title screen of the educational game "Clipper."

What Makes It Educational?

Almost any activity involves some learning. It wouldn't hold anyone's interest long if it didn't. For an arcade classic to enthrall a player for hours at a time, perhaps week after week, there must be a progressive mastery of the game's secrets. Experts at a game like *Pac-Man* say they've memorized several complex patterns of movement in order to "beat" the game. But you wouldn't expect to find "Theory of *Pac-Man*" being taught in schools. So what does set an educational game apart from any other kind?

First, it must have a clear educational goal. John Victor, whose Program Design, Inc. (PDI) produces such programs as *Clipper: Around the Horn in 1890*, says, "When we do an educational product, we sit down and define a set of educational parameters with measurable results." Then, a program can be tested in the classroom to see how well it meets its objectives. Before they started on software, PDI designed programmed instructions for educational groups, but Victor believes that market pressures are going to encourage firms new to the field to introduce educational games. He feels it's important for the purchaser to consider the educational value of a product.

Furthermore, just the educational value of software has itself become a selling point for computer manufacturers and retailers, according to Doug Carlston, president of Brøderbund Software.

Spinnaker presents Delta Drawing.™

An introduction to the modern art of programming using the ancient art of drawing.

Although educational software still sells more slowly than games, Carlston says that if the purchaser becomes convinced of a computer's educational potential, a sale is much more likely.

Another important factor in determining a game's educational value is whether the game is related to the educational goal. Some games take what might be termed the Mary Poppins approach to learning, making everything fun for children.

A logbook for the clipper ship The Andrew Jackson.

Games of this type start with an educational goal, perhaps a multiplication drill. To make the work seem more palatable to children, the programmer may add a game aspect. For example, if a child gets a correct answer, he or she can then shoot an approaching alien. But here the game is unrelated to whatever is being taught. Dr. Thomas Malone, who has studied several computer games at the Xerox Palo Alto Research Center, suggests this possibly negative effect. In an interview in the April 1983 *Classroom Computer News*, Malone suggests that if the game aspect is available only as a reward for getting a correct answer, the student's motivation can actually decrease. He is learning that arithmetic is an unenjoyable activity.

According to Malone, games that make the *learning* aspect part of the fun are more effective. He calls these intrinsically motivating games. In one that he's studied, *Darts*, you try to pop balloons by guessing where they are located on the screen. If you guess too high, an arrow shoots above the balloon. Too low, and it shoots below – so the game reinforces the concept being taught. Such games impose an extra burden on the game designer. While various number concepts – addition, subtraction, fractions – can easily be incorporated, a subject such as spelling or word use would impose different requirements.

Preschoolers

Surprisingly, preschoolers seem not to need motivation from game playing during learning. Their enthusiasm for computers runs high, and a well-produced program for drill and practice meets the youngsters' approval all by itself, according to programmer Bruce Mitchell. He created a line of such programs for preschool through second grade, distributed by the Programmer's Institute. Mitchell is not sold on the fun-and-games approach to programming: "I am an absolute firm believer in the idea that educational programs should be educational – not games."

What he strives for is user-friendliness – anticipating any problem the user could have and building the solution into the program. Another thing important to Mitchell is consistency. Every program he's written uses identical command procedures, so a child moving from one to another doesn't have to learn new instructions.

And finally, Mitchell feels that good documentation is important. That means that the child – or the teacher – should be able to learn about the program by reading the literature that accompanies the product. If these criteria are met, and the skills being taught are appropriate to the child's level of development, then the program will be useful.

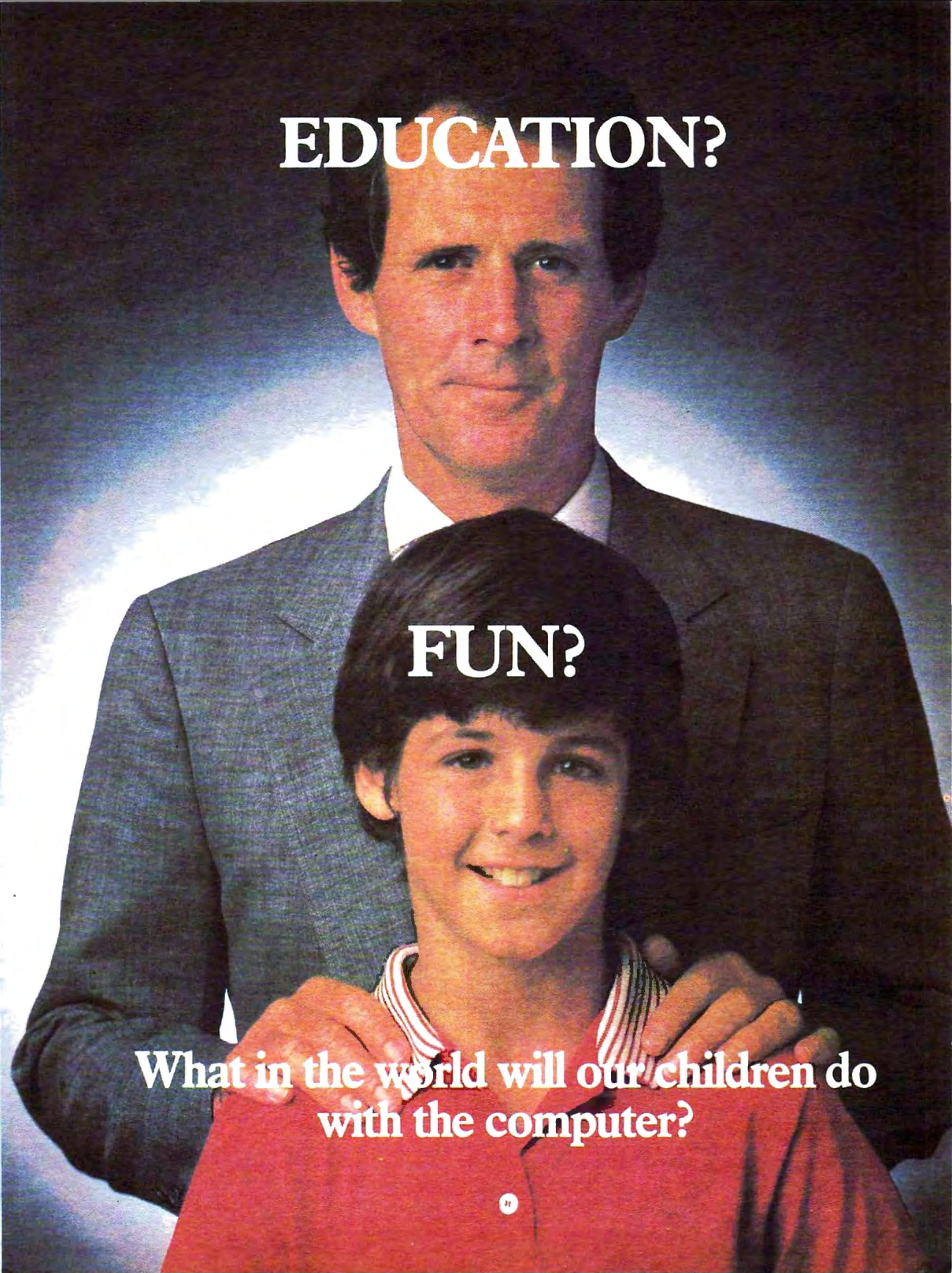
"But I don't believe the computer should be the primary tool to teach a concept," adds Mitchell. "It should be used to help teach the idea, rather than do it all."

For older kids, Mitchell sees computer literacy and programming skills as fundamental. "The creative thinking that permits you to write programs is something you can use throughout life," he says.

Simulations

At the other end of the spectrum from drill and practice are simulation games. They engage the user in a real-life situation – whether it's trying to cross the United States in a covered wagon or running a profitable lemonade stand. Such games are powerful because they not only teach, but also allow the user to make choices within a complex, detailed scenario while observing the consequences of various decisions.

For years educators have used such games as *Oregon*, *Lemonade*, and *Cartels and Cutthroats* to provide an extra dimension to their computer instruction. In *Oregon*, you join the westward migration, avoiding Indians, thirst, and starvation in an effort to reach new territory. *Lemonade* simulates a small business and the player can see the effects of reducing an advertising budget, raising prices, and fighting the weather while trying to stay in the black in the business of lemonade sales. *Cartels* is about business on a grand scale; the


EDUCATION?
FUN?
What in the world will our children do
with the computer?

4. WHAT WILL THE FIRST YEAR OF MICROZINE BRING?

An ever-expanding library of fun and learning!
Every eight weeks there are four full-length programs and the variety is incredible:

Who's that trying to solve the *Mystery at Pinecrest Manor*, lost in the *Northwoods Adventure* and challenging the far reaches of space on *Mission to Planet XII*? Could it be someone we know? (With Microzine, *anything* is possible.) All set for takeoff on the *Space Shuttle*— or maybe a hot air balloon race? Wow! (Will Microzine *ever* get down to Earth?)

Back safe on Planet Earth, visit the *Pet Store*. "A flamingo for your bathtub? Or maybe a rare three-humped camel?"

Too wild? *Gift Store* offers practical suggestions for everyone from Great Uncle Oscar to your 16-year-old twin sisters. Or explore your creative options by composing a song, writing and illustrating a book, or inventing your own game.

Microzine is *always* full of questions. If you've ever dreamed of talking to a computer game designer, *Ask Me* gives you the chance. You won't believe all the current big stars waiting to ask you questions, too.

JUST TURN THE PAGE TO FILL OUT YOUR MICROZINE REPLY CARD NOW!

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 2316 HICKSVILLE, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

Microzine™

Scholastic, Inc.
P.O. Box 947
Education Plaza
Hicksville, NY 11802

5. BECOME A CHARTER SUBSCRIBER TODAY AND YOUR FIRST MICROZINE IS JUST $7.95. (A $39.95 VALUE: YOU SAVE $32.00!)

Thereafter, pay only $24.95, plus shipping and handling, for each Microzine you accept. *You save nearly 40% off the retail price by subscribing!*

Each bimonthly Microzine Package contains; a Microzine Floppy Disk with Four Complete Programs.

On your first disk, discover:

1. POSTER. What's two spaces wide, slow, green and silent—and changes to three spaces wide, fast, pink and squeaky? It's an imaginary paintbrush that lets you create as many colorful posters as you want!

2. SECRET FILES. If your children want to remember who starred in their favorite horse movie of 1982, they need Microzine's electronic filing system. It works just like the electronic filing systems adults use.

3. ASK ME. Robert Macnaughton of "ET" fame is standing by to accept questions—and ask a few of your children in return!

4. HAUNTED HOUSE. There's never been a haunted house so funny—or one so willing to let you plan your own visit! Like all Twistaplots,® this one ends differently every time you venture inside.

NOTE: The first Microzine Package also contains a bonus: *a separate data disk that can be used to save original posters and other personal creations!* (A $4.95 value.)

The Microzine Handbook

After consulting this monthly step-by-step guide written in plain English for users 9-13, your children will be able to show *you* how Microzine activities work. Easy-to-follow instructions encourage independent work habits, creative thinking and follow-through.

Microzine™

THE SCHOLASTIC CHILDREN'S MAGAZINE ON A MICROCOMPUTER DISK.

FOR APPLE COMPUTERS.

O.K., Scholastic.
We're ready to enter a fantastic world of fun and learning.
Start our Microzines coming!

☐ Please send us Microzine #1 (featuring Poster, Secret Files, Ask Me and Haunted House) and enter a Charter Subscription to Microzine for the child named below. Bill me the special introductory price of just $7.95 (plus shipping, handling and any applicable sales tax). We will be billed $24.95 (plus delivery) for every Microzine we keep. After purchasing Microzine #1, we are never obligated to buy even one additional Microzine, and we may cancel our subscription at any time. If we are not delighted with the introductory Microzine, we may return it within 10 days and pay nothing.

☐ I've already purchased Microzine #1. Please start our subscription with Microzine #2, for which I will be charged just $7.95 (plus shipping, handling and any applicable sales tax). 61010 08

Child's Name________ (please print) Age____ Grade____

Address________ Apt.____

City______ State______ Zip______

Telephone No. (____)________

Parent's Signature________

SEND NO MONEY;

JUST FILL OUT AND MAIL THIS MICROZINE REPLY CARD NOW!

player is in charge of a multimillion-dollar manufacturing plant.

One professional software house which has incorporated educational values into games is Spinnaker Software. For preschoolers, games such as *Story Machine* and *Facemaker* support creative efforts by the child. Adventure games for older children encourage problem solving. In *Snooper Troops*, for instance, the user tries to solve a mystery. To be successful, you must learn as you proceed, experimenting with the Snoop-Mobile, a wrist radio, and a camera to discover and use clues.

Another adventure, *In Search of the Most Amazing Thing*, involves traveling through 20 different countries. When you pass through one, you must learn the language, the money system, and some of the local customs. "It's learning without making you feel you are being educated," says Nancy Evans, an editor at Spinnaker. "If you get something wrong, the game is not over."

The adventures hone problem-solving skills indirectly, making them part of the excitement of playing. Learning becomes a natural part of the action. At first, children don't take any notes when playing a game like *Snooper Troops*, according to Spinnaker's chairman, Bill Bowman. Then they begin to jot things down in haphazard fashion. Finally, he says, "kids begin taking notes in a structured way." No one tells them to; it just makes the player more successful.

Into The Home

These games are attractive to many educators, but school budgets are tight. In many cases, school administrators are unsure of what to buy. New software must first be reviewed, sometimes at the state level, and approval can take months, even years.

Often, the only way a teacher can get a computer venture under way is to rely on individual initiative. It's not uncommon for a teacher to use personal funds to purchase a computer for students. In fact, says Bowman, nearly all of Spinnaker's sales to educational groups are paid for by individuals – evidence to him that teachers are buying the products with their own money.

"They realize what the computer can do," he adds. "But schools are too slow and too bureaucratic. We feel that the revolution in educational computing will occur in the home."

To tap this market, Spinnaker is emphasizing cartridge software. People who don't have disk drives can acquire the games without having to make a substantial investment. Other manufacturers are undertaking similar efforts. They are stressing both educational quality and fun. And lest parents forget, producers are reminding them that students who learn at home have an edge at school. "When a child is exposed to software that teaches at home, chances for high success in school are greatly improved," according to Dr. Larry Lowery.

Lowery, who lectures on courseware evaluation at the University of California, Berkeley, created an extensive manual that is used by Soft-Kat's Educational Computer Centers. To help potential purchasers examine educational software before they buy, Soft-Kat has established over 300 centers where parents, teachers, and children can select programs and try them out.

Computers Vs. Game Machines

Activity such as this suggests that there is real interest in the home educational market on the part of software producers. In fact, both PDI's John Victor and Broderbund's Doug Carlston see a rapidly growing market there. Part of the reason is that home users are beginning to purchase almost as many computers as video games. Experts in the computer industry had thought that it would be years before home computers began to sell in such quantities. However, intense price cutting among manufacturers has dropped the price of some computers below that of video games. It is now quite possible that computers will begin outselling game machines as early as next year.

This could make educational games the next growth area in the computer business. As more manufacturers get into "eduware" and computer users look for software variety, the field could blossom. Competition may be tough, though. As Victor notes, "Parents don't like wasting money. The people who put up the bucks really want to be sure they get results." ©

Guest Commentary

COMPUTERS IN EDUCATION

Robert Nielsen

King Solomon, writing about the futility of various pursuits in life as ends in themselves, did not neglect learning. He wrote: "Of making many books there is no end, and much study is a weariness of the flesh" (Eccl. 12:12b). Children today must feel similarly: a bachelor's degree comes at the end of *seventeen years* of education for most people. Consequently, educators are always seeking more effective ways to accomplish their task of imparting knowledge and training minds. One useful means to this end is the computer.

Responsive Pacing

In its pure form, programmed instruction involves the presentation of new material step-by-step. Additionally, learners work individually at their own speed, and there are frequent examinations followed by immediate correction. Usually the learner is given a short piece of material followed by a fill-in-the-blank, multiple-choice, or other question for which the answer can be mechanically graded.

Traditional books and teaching machines, however, do not accommodate differences between fast and slow learners. Although the students work individually, at their own pace, all students must go through the same syllabus in exactly the same way. There is little flexibility for the weak student who needs extra drill and practice or for the advanced student who needs greater challenges.

Fortunately, the computer is able to handle what is called a *branching program*. In such a program there is no *one* correct way for the learner to move through the material. Instead, material is presented based on the learner's past performance. Thus, if there is evidence that a student already knows some of the material, then future reference to that topic may never be presented. The student who does slightly substandard work can be given extra drill and practice, while the one who does very poorly can be given a different, expanded explanation. In each case the computer can offer an individualized learning program to the student.

Whatever the technology – books, teaching machine, or computer – programmed instruction is limited to subjects which can be quantified. Therefore, such subjects as mathematics, chemistry, and physics can easily be adapted to programmed learning. Much more difficult to adapt are fields such as art appreciation, philosophy, and literature, since these subjects often require treatment which cannot be mechanically scored. The question "Why is *Moby Dick* a great piece of literature?" requires a type of answer radically different from the physics problem concerning the acceleration of an object dropped near the surface of the earth.

Not too long ago (only a few years) it might have been easily assumed that computer-assisted instruction would continue closely allied with the application of behavioral science to learning theory. This has not been the case, in part because not all educators are behaviorists. Instead, instructors use the computer when it can present material in a way which is consistent with their educational philosophy.

Firing Ranges, Foreign Languages

This wider use of the computer encourages new speculation as to how CAI (Computer Assisted

Instruction) works. The behaviorist sees the computer as a giver of rewards to the successful learner, thus increasing learning. However, behaviorism is only one branch of education and one which is limited, by definition, since it is concerned only with behaviors and nothing else. One explanation for the effectiveness of CAI comes from two disparate realms of education: the firing range and the foreign language classroom.

The largest educational organization in the United States is the military. Much time, energy, and money is spent training personnel. Consequently, the military is always looking for faster, cheaper ways to teach. One improvement that it discovered was in teaching marksmanship.

The computer is silent, which allows the student time to think and, consequently, learn.

The old method of teaching soldiers to shoot accurately was to let the soldiers take shots at a target. Then the sergeant would come over and tell the recruit how well he did. Informed of his results, the soldier would have another try.

As an experiment, the army tried placing targets which fell over when accurately hit, but which did nothing when missed. There was no instructor to tell the soldiers how they did: they could see for themselves.

The result was that soldiers learned faster and used fewer bullets. Interestingly enough, when people are told that they have done a job poorly, they say to themselves, "I'm no good." That is, they take an evaluation of their *performance* and apply it to *themselves*. While it is very easy intellectually to separate performance from person, emotionally it is very difficult. Furthermore, people who think they have been (or actually have been) judged negatively as a person tend to do worse in performance. In short, a person who tells you that you did a job poorly is not helping you to do the job better next time.

The second example comes from the foreign language class. Here, just as in the army, much time and effort is spent to teach students. Consequently, there is a proliferation of methods to teach foreign languages, each method trying to do the job better than previous ones. One surprising way that works well involves a mostly silent teacher.

The teacher rarely speaks even when students make errors. For example, when a student makes a pronunciation error, most traditional teachers would say something like "No, the correct pronunciation is ...." The silent teacher, however, would point to the part of the word where the error occurred. Students would then guess new pronunciations until hitting on the correct one (usually rather quickly). The correct answer is met with a slight nod of the head. In spite of the apparent paradox or apparent inefficiencies of a silent language teacher, students not only learn but seem to thrive under this system.

The point of the above examples is that the computer provides similar feedback to student responses. Because it is a machine rather than a person which gives the feedback to students, their egos are not as threatened. Additionally, the computer is usually silent, which allows the students time to think and, consequently, learn.

Inexpensive, Safe, Holistic

Another, well-established application of CAI is in the field of simulations. Simulations are used in education to provide a substitute for the real thing. Sometimes a substitute is preferred because it is less costly – learning to fly an aircraft, for example. A mock-up of an airplane cockpit connected to a high-speed computer can give every effect of flying an airplane, yet never leave the ground.

Moreover, simulations can provide learners with experience that would be too dangerous in real life. For example, pilots need to practice emergency situations, such as landing with one inoperative engine. Done with actual aircraft, this procedure may result in disaster. Simulated with the help of a computer, such an "emergency" gives pilots invaluable experience for a genuine emergency, should one ever happen.

Finally, simulations provide a holistic view – an appreciation for how everything works together. It has been said that scholars today know more and more about less and less. The knowledge that a simulation provides is just the opposite: a view of the whole instead of a focus on the details. The world of the classroom is one where details can be examined at length and at leisure. Outside the classroom, things are important not only for what they are in themselves, but for how they fit in with everything else that is happening. A simulation can provide this insight.

There are clearly several significant uses for computers in education. We've only described pacing, efficiency, and simulation. There are also strong arguments for using computers in educational management (grading, attendance) and in games which teach. The pessimism expressed by King Solomon may not apply to the learners and teachers of the future. ©

Ed·u·ware

(ej' o͝o wâr) ***n.***

The first name in learning.

THE SCIENCE OF LEARNING™

Professionally designed instruction in specific academic skills for ages 4 through adult.

Elementary Mathematics

Introduction to Counting presents young children with the concepts of numbers, addition, subtraction, shape discrimination, weight, height, and measurement. *Apple, Atari disk $39.95. Atari tape $29.95.*

Arithmetic Skills teaches basic number skills. High resolution graphics, animation, and sound effects make counting, addition, subtraction, multiplication, and division interesting. *Apple $49.95.*

Fractions builds and reinforces skills in working with fractions. Definitions, numerators, denominators, addition, subtraction, multiplication, and division are explained in small increments and a learning manager makes the program flexible to meet the needs of individual learners. *Apple $49.00.*

Compu-Math™ Fractions, a comparable *Atari* program is also available. *Atari disk $39.95. Atari tape $29.95.*

Decimals presents addition, subtraction, multiplication, division, rounding off, percentage and conversion as they apply to decimals. *Apple $49.00.*

Compu-Math™ Decimals, a comparable *Atari* program is also available. *Atari disk $39.95. Atari tape $29.95.*

Advanced Mathematics

Algebra, volumes 1 through 6 is a series of independent programs comprising a first-year course in algebra. Concepts are presented in four distinct learning styles so that self-managed learning is encouraged. Volumes 1 through 4: *Apple, IBM PC $39.95 each.* Volumes 5 and 6 (sold as a set): *Apple, IBM PC $49.95.*

Language Skills

Spelling and Reading Primer™ uses simple words and pictures to introduce basic reading and spelling skills. Word groups cover simple two- and three-letter words through difficult multi-syllable words, double vowels and consonants, and directions and numbers. *Apple $39.95.*

Compu-Spell™ builds spelling skills in a tutorial format. A system diskette can be used with six different data diskettes geared to vocabulary for grades 4, 5, 6, 7, 8, and an adult level containing words commonly misspelled in business. System Diskette: *Apple $29.95.* Data Diskettes: *Apple $19.95 each.*

Compu-Read™ improves reading speed, recall, and test-taking skills. The system adjusts itself to your learning requirements and provides detailed information on progress and reading rates. *Apple, Atari disk $29.95. Atari tape $24.95.*

PSAT Word Attack Skills and **SAT Word Attack Skills** develop specific vocabulary skills in definition, connotation, roots and prefixes, and application. Test-taking skills required for the Scholastic Aptitude Tests are sharpened. *Apple $49.00 each.*

PSAT/SAT Analogies develops skills tested in the analogies portion of college entrance examinations. Word relationships, meanings, and achieving under test-taking conditions are stressed. *Apple $49.00.*

Introduction to Poetry presents rhyme and meter in a clear and simple tutorial format. Definitions are given on the screen and are sounded out by your computer. The program is highly interactive and easy to operate. *Apple $39.95.*

Computer Skills

Hands On BASIC™ Programming combines hand-holding software with a 200-page tutorial that steps the novice through beginning BASIC programming. This complete learning system won't leave you hanging with an error message. Special features allow you to diagnose the problem and correct your mistake. *Apple $79.00.*

DRAGONWARE™

Entertaining, colorful, animated games with lasting educational value, for ages 4 through 10. Webster, our wise and wonderful dragon, is your child's companion in the adventure of computer-based learning.

Spelling Bee Games™ contains four fun activities in which hand/eye coordination, memory skills, and motor skills are exercised. Twenty-two word lists covering simple two- and three-letter words through multi-syllable words provide variety and scope. *Apple, Atari disk $39.95.*

Webster's Numbers™ is a magical place where mosquitoes in mazes, balloon races, building blocks, and boxes teach pre-school children the basics of counting. *Apple $39.95.*

INTERACTIVE SIMULATIONS™

Realistic recreations of real-world adventures where scientific evaluation yields the solution.

Rendezvous makes you the pilot of a true-to-life space shuttle. Your mission is to dock with an orbiting space station. *Apple, Atari disk $39.95.*

INTERACTIVE FANTASIES™

Science fiction adventure games that challenge the intellect and startle the imagination.

Prisoner 2™ In a surrealistic environment of misleading clues and sugar-coated propaganda, you survive by your wits alone. Escape without divulging the information your captors desire. Can you escape? Can you even find the bars? *Apple $32.95. Atari disk, IBM PC $39.95.*

Empire I: World Builders™ Struggle to tame a galactic wilderness and establish an infant Empire's first colonies. Stand at the edge of the new frontier. *Apple $32.95.*

Empire II: Interstellar Sharks™ Dodge the bureaucracy and pursue ultimate success in a high-tension, intergallactic entrepreneurial arena. Making it depends on how well-equipped you are to meet the onslaught of those in power. *Apple $32.95.*

Empire III: Armageddon™ The end of a once great Empire now torn by rebellion, poverty, corruption, and an unfeeling authority. Where do your loyalties lie? *Apple $32.95.*

For more information or the dealer nearest you contact:
EduWare Services, Inc.
28035 Dorothy Drive
PO Box 22222
Agoura Hills, California 91301
213/706-0661

EDUWARE®

COMPUTERS IN SCHOOL
NEW APPROACHES

John Blackford, Assistant Features Editor

"Computers to invade the classroom!" Even if you didn't see that on the evening news or in your favorite magazine, chances are that you've seen a TV commercial or heard from a neighbor that students must *learn about computers to succeed. The question for parents and educators is how best to introduce computers – what software to use, how to set up a computer program, and where to get the money. Some students, however, are racing ahead, learning programming at home or during odd hours at school, and sometimes sharing their knowledge with others.*

"The computer is going to force us to reexamine our goals in education," says Sheila Cory, who coordinates the computer program for the Chapel Hill-Carrboro City Schools in North Carolina. And computers are moving in now, whether or not new goals have been set. Students are eager to try them. Teachers wonder how to tap this enthusiasm without sacrificing educational quality. Some school boards worry about the expense, yet fear, too, that their school may be left behind if they don't act quickly.

While one state, or school district, or family may be heavily committed to computers, another may view them with skepticism or disinterest. The result is a confused, yet creative ferment. Individuals can have real impact now, because – for all the excitement – there are few firm guidelines, few precedents.

A Mythical Country

One individual who made a difference is Jim Tomberg, a teacher at Chapel Hill High School. He could find hardly enough money in the budget to purchase even one computer for his students, so Tomberg proposed a software development group for the school. He requested and received a grant from federal and state funds set aside to aid unique educational projects.

The high school students in the project were to create original, documented programs to the specifications of teachers in the elementary grades. Tomberg wanted the programmers to work closely with the students and teachers receiving the programs.

To make the entire project educational, Tomberg says he "let the kids make all the decisions. They organized the whole course." They studied various brands of computers and decided what equipment to buy. Then they came up with the idea of doing a newsletter about their study – all composed on computers using word processing programs.

The teachers who requested material did, however, retain complete control over the content of the programs. In every case, students spoke directly with each teacher to insure useful results in the classroom.

Tomberg's project has received strong support from the 12 programmers as well as from the teachers requesting software. Not every request could be fulfilled, and when one student programmer called a teacher to say his project was accepted, "the teacher was so excited. He was ecstatic," says Tomberg. The program, for history teacher Grant Zimmerman, is a simulation of a mythical New World traversed by seafaring adventurers. The new land is complete with native tribes – each with distinct languages and customs – and mountains and rivers to be charted.

The object of the game, called "Explorers," is to cross an ocean and trek across an unknown continent to gain treasure hidden on the other side. At each stage, obstacles must be overcome. At sea, whales and storms threaten the voyagers. Once on land, the terrain must be mapped for the journey overland. And part of the challenge is learning about the tribes. Some are friendly and

PLATO EDUCATIONAL COURSEWARE BRINGS NEW EXCITEMENT TO LEARNING.

If your kids think serious education has to be dull, wait until they use Control Data PLATO® educational courseware.

With PLATO courseware, children can learn by competing against themselves; by interacting with stimulating graphics that keep them motivated. They see their progress, and find real excitement in achievement.

PLATO courseware is being used in classrooms across the nation. Now you can bring this quality education home. You'll find PLATO lessons at selected retail outlets where quality software is sold.

For the Apple II Plus and Apple IIe, the TI 99/4A or Atari 800.

The selection includes elementary **Math** lessons in Basic Number Facts, Whole Numbers, Decimals and Fractions.

Through such computerized activities as "darts" and "pinball," PLATO makes learning math almost fun.

Foreign Language lessons use the popular hangman or pyramid game concept to help teach French, German or Spanish in a way that holds and builds interest.

Our **Computer Literacy** lesson provides a perfect introduction to the computer age for kids and novices.

For Apple II Plus and Apple IIe.

Our **Keyboarding** lesson shows children and adults how to use a keyboard to enter data into the computer. Developed in cooperation with Gregg-McGraw Hill.

Widen your child's world.

See the growing library of PLATO educational courseware at selected retail outlets. Or for information and a free PLATO educational courseware catalog, call toll-free 800/233-3784. Or write: Control Data Publishing Co., P.O. Box 261127, San Diego, CA 92126. In California call 800/233-3785.

Warranty available free from Control Data Publishing Co., 4455 Eastgate Mall, San Diego, CA 92121

PLATO
COMPUTER-BASED EDUCATION

CONTROL DATA PUBLISHING

can help with the enterprise, while others must be battled or avoided.

The main idea for Explorers was Zimmerman's. He wanted a challenging exploration game in which students could succeed (unlike some adventure games, where you can play for hours, only to be gobbled by an ogre after making one wrong turn). With Zimmerman's general theme in mind, programmer Aden Evens set to work on an ocean part of the adventure, while Tom Evans created the new continent. They designed the game so there is usually a way out of any difficulty if the player is persistent and uses common sense. When the student explorer sees a storm coming, for example, he can avoid being blown off course by lowering the sail.

A mini-workshop for elementary school students conducted by Chapel Hill High School students and their advisor Jim Tomberg (center).

The students finished all their programs in time to present them to examiners overseeing the project for the state, and chances are good that funding will be extended this year. Toward the end of the school year, the programmers even held some workshops, sharing their knowledge with younger schoolmates. Pupils as experts, that's another Computer Age twist.

Tomberg is hard at work on his latest project: convincing a manufacturer of inexpensive computers to donate 30 of its products to the school. The students would be able to check them out of the library and take them home. "Just think," says Tomberg, "they'll be able to take one home and plug it right into the TV set."

New Research

At present, fully 25 percent of the funding for classroom computers is provided by parental, religious, or civic groups, according to a recent study by McGraw-Hill Research. And teachers often bring in their own computers to share. But these individuals and groups aren't always sure what approach to take, or even what brand of computer is best for schools.

"You are talking about a whole new era of technology, about which not enough background research has been done," says graduate student Jim Glover, of the University of Connecticut School of Education. "Schools are rushing pell-mell into educational computing, but what happens ten years from now when you may be teaching kids three or four hours a day with computers? What's best for preschoolers, for junior high? What type of display is easiest to look at? What kind of keyboard is comfortable to use?"

To help answer such questions, researchers are looking at the growing role of computers in education and developing new theories and methods to help educators cope with the changes. However, they are by no means unanimous in their recommendations. For some, computer-assisted instruction (CAI) is a perfect vehicle for the behaviorist theories of psychologist B.F. Skinner. For others, computers can enhance the open-ended theories of Swiss educator Jean Piaget, who believed that education at its best involves an active discovery of reality by the learner, not a recollection of ready-made facts.

Piaget's ideas actually form the basis of a computer language – Logo – that was developed at MIT by Seymour Papert and others. Logo has intrigued many educators because it supports impressive screen graphics through a command structure that permits the linkage of simple procedures which then form more complex procedures.

Schools that use Logo and similar languages incorporating Piaget's ideas have blossomed in recent years. The Bank Street College of Education in New York City has made a study of this approach to learning. People at the school are looking at the effects that computers, and Logo in particular, have on learning among eight- to twelve-year-olds. One characteristic of Logo is that it makes the child a partner in the learning process. However, says Barbara Dubitsky, "The computer itself is nothing more than a tool. In the hands of a poor teacher, Logo is a bomb."

While the research at Bank Street is still preliminary, work at such places will eventually give educators throughout the country a solid yardstick by which to evaluate new approaches.

Radiating Computing Centers

Current research has an impact on the classrooms of the future, but where do teachers or schools turn for help now? Again, one committed teacher or a single successful program often serves as a catalyst to create a larger, more formally organized group. For example, a teacher may begin a modest pilot program, and interested people drop by to ask questions. As the project grows, they may

SOFTWARE MOVIES™

MAXIMUS, Inc. is excited to present two new educational software movies for kids of all ages . . . and more are on the way!

STORYLINE™ makes bedtime, or anytime, a warm and friendly funtime. Clover the Clown is your tourguide for two fairytales: The Ugly Duckling and Rumpelstiltskin. Through computer magic you become a part of each story!

SAFETYLINE™ combines fun with important lessons about safety. Max the Cat shows you how to cross the street safely going to school and what to do when lost at the zoo.

Both software movies have interactive games that reinforce the fun and learning. Both have real human voices; you don't need a voice synthesizer!

Features of the software include:

- Sophisticated Face Animation
- Lip-Synchronized Voices
- Original Musical Scores
- Superior Full-Color Graphics
- Optional Joystick Control
- Machine Language Programs

Relax and enjoy yourself. Let Max, Clover, and their colorful friends guide you through a new world of fun and learning. You'll be amazed at how real they are!

STORYLINE™ and SAFETYLINE™ are currently available for any Atari computer with 48K. Each software movie comes in either CASSETTE ONLY ($29.95) or CASSETTE/DISK ($34.95) versions. Order by mail or phone. Visa and Mastercard welcome. Include $2.50 postage and handling; Virginia residents add 4% sales tax. Dealer inquiries invited. Write for our free catalog. MAXIMUS, Inc., 6723 Whittier Avenue, McLean, Virginia 22101.

ORDER TOLL-FREE 1-800-368-2152

MAXIMUS ...*Where Excellence is the Standard*

Commodore 64 Version Coming Soon!

return to volunteer their time – and the alliance grows. Such pioneer efforts can develop into vital resource centers whose influence reaches far beyond the local area.

That's the way it happened in San Mateo County, California, where the Microcomputer Center there gained such a reputation for excellence that it has influence throughout the state and even beyond. Technically, the Center is only a county office, but because it was the first of its kind in California, it became a model for similar programs elsewhere.

The Microcomputer Center has been designated as a software library and clearinghouse to support all 15 Teacher Education and Computer (TEC) Centers in the state. It also runs the *Softswap* public domain software exchange in cooperation with a group of volunteers working for CUE (Computer-Using Educators). *Softswap* receives hundreds of inquiries each month from around the world. The group prepares disks with programs contributed by educators and makes the disks available for $10 each. The disks may then be freely copied and distributed.

The Center also received a grant for the 1982/83 school year to evaluate software and make the results available to educators all over California. To do this, it is establishing a cadre of software evaluators across the state and developing a list of educational software that has been favorably reviewed by other qualified groups across the country. The Center will coordinate and publish the results. (A software catalog is available for $1 ppd. Write to Microcomputer Center, San Mateo County Office of Education, 333 Main St., Redwood City, CA 94063.)

California has given strong support to computer use in its schools. Many projects have been supported by state-administered grants. Another state which made one of the earliest starts in computer education is Minnesota. It remains a leader with its MECC (Minnesota Educational Computing Consortium) program, the nation's only statewide instructional computing network. MECC offers a wide range of services to students, teachers, and administrators in the state's public schools and colleges. It also develops and distributes educational software for a fee to school districts anywhere in the United States. For many schools, MECC educational programs are the first ones in the classroom.

Texas has also strongly supported the use of computers in education. Software evaluation is coordinated on a statewide basis, and the state – through 20 Educational Service Centers – now gets regular discounts of 25 to 30 percent on hardware. Soon, the Texas Education Agency hopes to go on-line with a data base containing all the agency's software evaluations and other pertinent information. To improve computer education throughout the state, requirements for teacher certification are being revised to include computer literacy. According to Sandy Pratscher, educational specialist for instructional computing for the state, the mere hint of this change has already caused a marked jump in enrollment for computer courses in Texas colleges.

Students at the Bank Street College of Education trying out a new program. Researchers here are looking at new ways to involve computers in the learning process.

Innovators

Summit School (Winston-Salem, North Carolina) is an unusual place – the kind where you'd almost expect to find an innovative computer program. Although the school is about to celebrate its 50th anniversary as an independent institution, the original principal, now 92, still comes in mornings to teach.

The school has a Math and Computer Center. According to the center's director, Elaine Bologna, the math center, started four years ago, was funded by two foundations as a demonstration center for new teaching methods. After the grants expired, the school took over funding and added the computer program. Teachers from all over North Carolina visit the center in Winston-Salem to attend workshops and demonstrations.

The emphasis at the center is on programming – Logo for grades one through six, BASIC for grades eight and nine. "The interesting thing about it," says Bologna, "is that when the kids come in after school, they use Logo." In fact, that language offers so many possibilities that the school really hasn't found much need for packaged software. Students invent their own games and experiment endlessly.

In one case, Bologna presented students with an imaginary situation and let them create it on

the screen. "A plane was lost and needed a landing field," she told them, so they made a long rectangle. Then she asked, "Where's the terminal?" and finally, "Where's the door to the terminal?" When they were done, the students had created a whole airport.

The kindergarten students use a Big Trac programmable toy to help them visualize Logo. They can make Big Trac perform the same sort of maneuvers on the floor that the Logo "turtle" does on the screen. This gives them a real feel for some of the ideas behind Logo and a head start when they encounter Logo in later grades.

In some cases, it's harder to interest teachers than it is to interest students. "We don't have quite as much teacher involvement as we would like," says Bologna. "They haven't been able to feel comfortable in situations where they aren't the authority. But, really, that's one of the beauties of the whole thing. We all make mistakes." And part of what makes computers unique in schools is that they are new to everyone. There's a possibility of mutual discovery that can allow teacher and student to share the learning experience. "That's part of Papert's philosophy," says Bologna.

A Resource Center

Eventually, every class in the nation may have an array of computers, and teachers may be as familiar with their use as they are with gradebooks. But at present, computers are mysterious to many teachers. In fact, despite all the attention given lately to computer education, 47 percent of all schools still had no microcomputers for student instruction at the beginning of 1983, according to a national survey conducted at Johns Hopkins University.

But great strides have been made toward computer instruction, especially by schools with a computer center. The person in charge is often simply an innovator interested in microcomputers who initiated a shoestring program. Once such programs grow – possibly with the addition of full-time staff – the resources of state and federal agencies become more accessible. Staff members have the time to find out what help *is* available. They can evaluate software and teaching methods while proceeding with the delicate task of introducing both teachers and students to computing.

Paul Boston took advantage of his position as a teacher at a science center serving the public schools in Maryland's Prince Georges County to initiate a computer program there. Now, five years later, the center has 29 Commodores serving 60 pupils per day in a regular schedule of instruction.

Since the Howard B. Owens Science Center receives students from the other schools for special programs, Boston decided it would be cost-effective for the center to introduce micros. They purchased two of the earliest Commodore PETs, the ones with what Boston calls the "Munchkin" (calculator-style) keyboard. The program began with one-time, four-hour workshops for gifted students. Gradually it was refined and expanded into a resource for the whole school system.

As the center evolved, it was modified to respond to whatever computer activity was already established in the individual schools. All the major computer brands are now represented at various schools, and the high school has powerful multi-terminal machines. The center now concentrates on introducing computers to younger students, primarily those in the third through sixth grades. Students younger than that require more time than the center has available. However, Boston says, a computer is needed in the classroom for these children so they can have time to become acquainted with it.

"We find that our students have a lot of misconceptions about computers," says Boston. "We try to correct these. The student should be familiar enough with computers to be able to utilize a program." When they are, Boston claims the youngsters "are not fooled easily about computers. When they see *Knight Rider* (a show with a computerized, talking car), they know computers don't really have emotions or many of the capabilities portrayed."

This approach puts the center firmly in the camp that emphasizes teaching *about* computers, as opposed to teaching programming or using computers to teach other subjects (CAI). "Computer programming will be done by the few, but nearly everyone will be using computers," says Boston. He believes the center's role is a transitional one that will be phased out as computers find their way into every classroom.

To broaden teacher awareness of computers, the center's instructors offer workshops on inservice training days. They help teachers become comfortable with computers and advise them about using computers in class. "For example," says Boston, "if they are going to do CAI we encourage them to use students' talents, but to temper those abilities with their own educational judgment." Teachers may be intimidated because some of their students will take to computers more easily than they do. That's why people at the center feel it's important to acquaint people throughout the school system with computers.

Students are growing up in an environment where banking machines, grocery stores, and business procedures are increasingly computerized. Teachers and resource groups like the Science Center are helping them prepare for it. In the process, they are learning themselves – evolving as they develop more effective methods for bringing computers and students together. ©

Caves Of Ice

Marvin Bunker and Robert Tsuk

This award-winning game should provide hours of amusement. Originally written on the Apple, there are also versions here for VIC, 64, Atari, and PET/CBM.

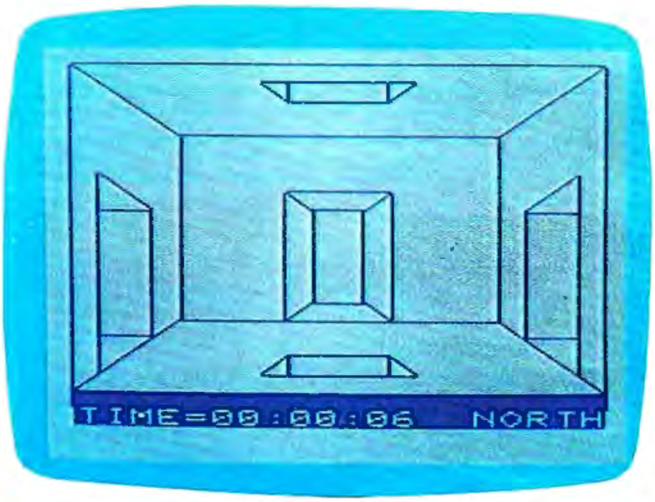
Robert Tsuk invented a game called "Quinti-Maze," wrote a version for the Apple, and won a prize in a *BYTE* magazine game contest – it was published in the September 1982 *BYTE*. It looked too intriguing to be limited to Apple owners, so I wrote a version called "Caves of Ice" which will run on all Commodore computers (VICs require 8K or more expansion memory). I contacted Robert Tsuk about submitting this version as a joint article to COMPUTE! where other Commodore computer owners could share it. He replied that he was converting it for the Atari. These games are the result of our joint efforts.

The Game Scenario

At the start you are somewhere in a five-story structure made entirely of ice. Each floor has 25 rooms in a five-by-five array. Carved into the walls of each room are one or more openings, doors to the north, south, east, west, up, or down. However, you can see only the doors available to you from your present vantage point. Only one door in the building opens to the outside – it may be in a wall, the roof, or the floor of the building. Your goal is to get out as fast as possible. You could freeze inside these caves of ice.

You can change which direction you are facing at any time – complete instructions are included in the program.

After finding your way out, you are given the option of trying the same maze and same starting point again to see if you can improve your time. Or, you can play again with a new random maze.



A room with five possible exits – only one of the rooms you may encounter in "Caves of Ice." VIC version.

Strategy

To quote from Robert Tsuk's earlier article: "The strategy for Quinti-Maze is fairly simple; be methodical. Because all the rooms in the maze look similar, you could wander around forever without finding the exit. My favorite method is to travel in one direction as far as I can go, then I assume I'm at one of the outside walls and search there for an exit."

A Variation

Insert this line at the beginning of either Program 1 or 2:

```
1 X=RND(-π)
```

This initializes the random number generator with the same seed each time you RUN, so you'll always start with the same maze. You can have the sequence U,W,W,W,N,W,S,W committed to memory and amaze your friends with how rapidly you can find your way out. The figure shows the complete maze produced by this starting seed. *Note:* This seed produces a different maze on the VIC.

A million laughs

SPARE CHANGE™ You are the game-happy owner of the Spare Change Arcade. Two fun-loving, but overworked Zerks—the main characters in your most popular game—have broken loose and are trying to retire from the business. You try madly to stop them. If you can get a coin into the juke box, the Zerks get so caught up in the music, they drop everything and start dancing. You also try popping popcorn and making a pay phone ring—which immediately makes the Zerks stop, answer and start a wild conversation. If you "win" the game, there are rib tickling cartoons by the Zerks to reward your efforts. It's a game full of sight gags, surprises and comedy. From the best. Brøderbund! **For the Apple® II, II+ and IIe. Coming soon for the Atari® home computers in disk format.**

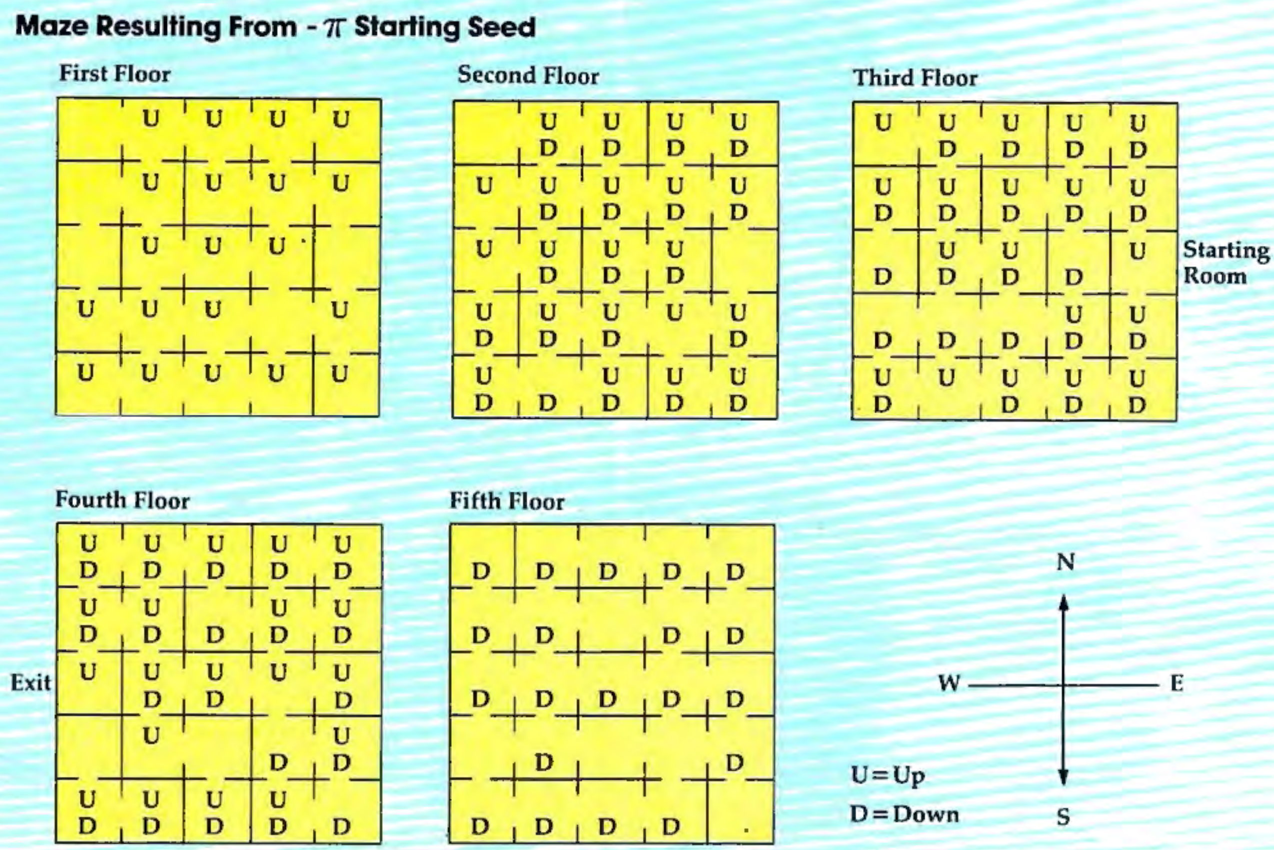
and an endless challenge

LODE RUNNER™ Here's a game that will never stop challenging you. That's because Lode Runner is more than a spellbinding, fast-action game with its 150 different mind-boggling game screens. Lode Runner is also an easy-to-use Game Generator that lets you create your own games. Without any knowledge of programming, you can easily design unique Lode Runnner screens, then bring them to action-packed life. You will maneuver through scene after scene, running, jumping, drilling passages and outfoxing enemy guards in a secret underground hide-away as you pick up chests of gold stolen from citizens of the Bungeling Empire. There's no end to the thrills, chills and challenge. Of course, it's from Brøderbund! **For the Apple II, II+ and IIe. Coming soon for the: Atari home computers (disk and cartridge); Atari 5200™ Super System; Commodore 64™; VIC-20™; IBM® PC.**

Ask your Brøderbund dealer for sneak previews.

Brøderbund Software **Discover the Difference**

1938 Fourth Street San Rafael, CA 94901



Program Notes

In lines 1140 and 3005, the PRINT CHR$(7) statements produce beeps on the PET when you try to go through a solid wall and when you successfully find your way out. If you have a PET without the built-in beeper, but you do have the CB2 sound, you can replace these statements with the appropriate sequence of POKE statements to give the sound effects desired. VIC and 64 owners should replace the PRINT CHR$(7) with the proper POKEs to produce sounds on their computers if they desire this feature.

In the INPUT statement in line 2001, following the INSTRUCTIONS are three shifted spaces followed by three left cursors. This is my favorite way to avoid the infamous PET INPUT crash.

Program 1 is Caves of Ice for the VIC (with 8K or more expansion memory) and 64. The only changes required to RUN on PET/CBM models are in the keys which must be typed to produce the graphics in lines 120-151. Program 2 lists these changes for the PET. The graphics are not directly accessible from models with "business style" keyboards. Refer to your manuals for the equivalent CHR$ codes. Newer CBM models may also require the addition of a line such as:

5 PRINT CHR$(142)

to put them into graphics mode.

Program 1: Caves Of Ice – VIC And 64

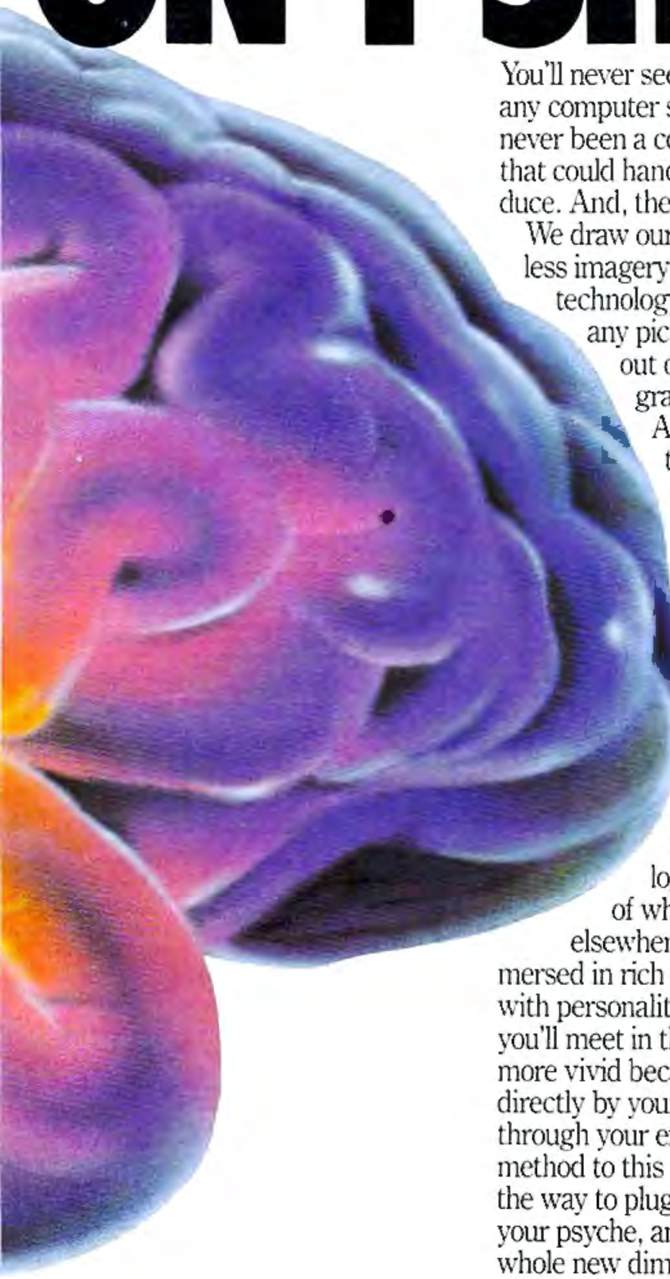
BEGINNING PROGRAMMERS

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

```
10 R$="{23 RIGHT}"
20 D$="{23 DOWN}"
25 PRINT"{CLR}"
90 GOSUB2000
100 DIMFC(5,7),FC$(5)
105 FC$(1)="NORTH":FC$(2)="SOUTH":FC$(3)
    =" EAST":FC$(4)=" WEST"
110 FORB=1TO4:FORI=1TO6:READFC(B,I):NEXT
    :NEXT
115 GOTO155
120 PRINT"{CLR} {20 @}{DOWN}{LEFT}N
    {DOWN}{2 LEFT}N{DOWN}{2 LEFT}N{DOWN}
    {2 LEFT}N"
121 PRINT"{HOME}{DOWN}{RIGHT}M{DOWN}M
    {DOWN}M{DOWN}M{12 @}"
122 PRINT"{HOME}{DOWN}";:FORI=1TO18:PRIN
    T"{M}{DOWN}{LEFT}";:NEXT:PRINT"
    {UP}{RIGHT}N{UP}N{UP}N{UP}N{UP}";
123 PRINT"{G}";:FORI=1TO9:PRINT"{UP}
    {LEFT}{G}";:NEXT
124 PRINT"{HOME}";LEFT$(D$,19);"{M}
    {RVS}{20 SPACES}{OFF}{G}{LEFT}{UP}
    {LEFT}M{UP}{2 LEFT}M{UP}{2 LEFT}M
    {UP}{2 LEFT}M"
125 PRINT"{HOME}"LEFT$(D$,15);LEFT$(R$,5
```

WE STICK OUR G
THE SUN D

RAPHICS WHERE ON'T SHINE.



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.

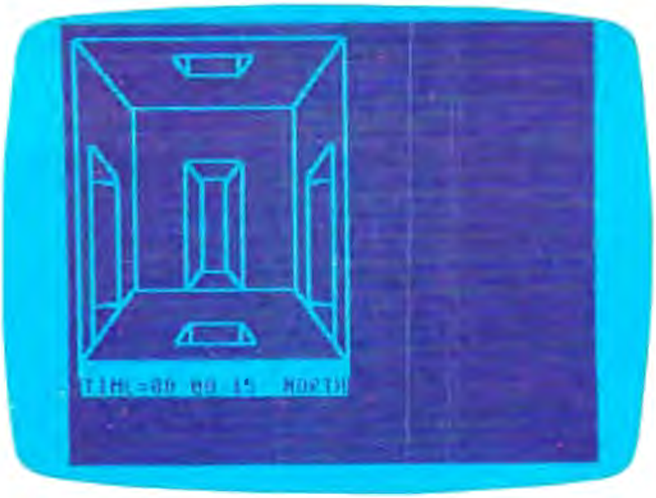
Step up to Infocom. All words. No graffiti. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

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 1, TI Professional, TRS-80 Model I, TRS-80 Model III



64 version.

```
    );"[12 T]"
126 PRINT"{HOME}";LEFT$(R$,21);:FORI=1TO
    18:PRINT"{DOWN}[G]{LEFT}";:NEXT
127 PRINT"{HOME}";LEFT$(R$,17);LEFT$(D$,
    5);:FORI=1TO1Ø:PRINT"[G]{DOWN}
    {LEFT}";:NEXT
129 RETURN
13Ø PRINT"{HOME}";LEFT$(R$,8);"{DOWN}
    [6 @]{DOWN}{6 LEFT}M[G]
    {2 SPACES}[M]N{DOWN}{5 LEFT}
    [4 T]":RETURN
135 PRINT"{HOME}";LEFT$(D$,16);LEFT$(R$,
    9);"[4 @]{DOWN}{5 LEFT}N[G]
    {2 SPACES}[M]M{DOWN}{6 LEFT}
    [6 T]":RETURN
14Ø PRINT"{HOME}";LEFT$(D$,7);"{RIGHT}";
    :FORI=1TO11:PRINT"[M]{DOWN}{LEFT}"
    ;:NEXT:PRINT"{RIGHT}{3 UP}[@]@"
141 PRINT"{HOME}";LEFT$(D$,9);"{2 RIGHT}
    [T]P";:FORI=1TO5:PRINT"{DOWN}
    {LEFT}[M]";:NEXT
142 PRINT"{HOME}";LEFT$(D$,7);"{2 RIGHT}
    M{DOWN}M":RETURN
145 PRINT"{HOME}";LEFT$(D$,7);LEFT$(R$,9
    );"[4 @]{DOWN}{5 LEFT}[M]M
    [2 @]N[G]";:FORI=1TO5
146 PRINT"{DOWN}{6 LEFT}[M] [G][M]
     [G]";:NEXT:PRINT"{DOWN}{6 LEFT}
    [M]N[2 T]M[G]":RETURN
15Ø PRINT"{HOME}";LEFT$(D$,7);LEFT$(R$,1
    9);"N[G]{DOWN}{3 LEFT}N [G]
    {DOWN}{3 LEFT}O[T][G]";:FORI=1TO5
151 PRINT"{DOWN}{3 LEFT}[G] [G]";:NE
    XT:PRINT"{DOWN}{3 LEFT}L[@][G]
    {DOWN}{LEFT}[G]{DOWN}{LEFT}[G]":
    RETURN
155 DIMS$(6,6)
165 FORA=1TO5:FORX=1TO5:FORY=1TO5
17Ø IFA<>5ANDRND(1)<.8THENS$(X,A)=S$(X,A
    )+"O":GOTO18Ø
175 S$(X,A)=S$(X,A)+"X"
18Ø IFMID$(S$(X,A-1),(Y-1)*6+1,1)="O"THE
    NS$(X,A)=S$(X,A)+"O":GOTO19Ø
185 S$(X,A)=S$(X,A)+"X"
19Ø IFY-2<ØGOTO2ØØ
195 IFMID$(S$(X,A),(Y-2)*6+4,1)="O"THENS
    $(X,A)=S$(X,A)+"O":GOTO2Ø5
2ØØ S$(X,A)=S$(X,A)+"X"
2Ø5 IFY<>5ANDRND(1)<.8THENS$(X,A)=S$(X,A
    )+"O":GOTO215
21Ø S$(X,A)=S$(X,A)+"X"
215 IFX<>5ANDRND(1)<.8THENS$(X,A)=S$(X,A
    )+"O":GOTO225
22Ø S$(X,A)=S$(X,A)+"X"
225 IFMID$(S$(X-1,A),(Y-1)*6+5,1)="O"THE
    NS$(X,A)=S$(X,A)+"O":GOTO235
23Ø S$(X,A)=S$(X,A)+"X"
235 NEXT:PRINT"*";:NEXT:NEXT
24Ø X=INT(RND(1)*3)+2:Y=INT(RND(1)*3)+2:
    A=INT(RND(1)*3)+2
245 RD=INT(RND(1)*6)+1:ONRDGOTO25Ø,255,2
    6Ø,265,27Ø,275
25Ø A=5:P1$=LEFT$(S$(X,A),(Y-1)*6):L=29-
    LEN(P1$):P2$=RIGHT$(S$(X,A),L)
251 S$(X,A)=P1$+"O"+P2$:GOTO29Ø
255 A=1:P1$=LEFT$(S$(X,A),(Y-1)*6+1):L=2
    9-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
256 S$(X,A)=P1$+"O"+P2$:GOTO29Ø
26Ø Y=5:P1$=LEFT$(S$(X,A),(Y-1)*6+3):L=2
    9-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
261 S$(X,A)=P1$+"O"+P2$:GOTO29Ø
265 Y=1:P1$=LEFT$(S$(X,A),(Y-1)*6+2):L=2
    9-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
266 S$(X,A)=P1$+"O"+P2$:GOTO29Ø
27Ø X=5:P1$=LEFT$(S$(X,A),(Y-1)*6+4):L=2
    9-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
271 S$(X,A)=P1$+"O"+P2$:GOTO29Ø
275 X=1:P1$=LEFT$(S$(X,A),(Y-1)*6+5):L=2
    9-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
276 S$(X,A)=P1$+"O"+P2$:GOTO29Ø
29Ø PRINT:PRINT: PRINT"HIT {RVS}RETURN
    {OFF} TO START"
3ØØ GETC$:IFC$=""GOTO3ØØ
1ØØØ X=INT(RND(1)*5)+1:Y=INT(RND(1)*5)+1
     :A=INT(RND(1)*5)+1
1ØØ5 SX=X:SY=Y:SA=A
1Ø1Ø FC=1:TI$="ØØØØØØ":GOTO122Ø
1Ø2Ø PRINT"{HOME}";LEFT$(D$,2Ø);LEFT$(R$
     ,16)"{RVS}";FC$(FC):A$="":D=Ø
1Ø3Ø TX$=TI$:TP$=LEFT$(TX$,2)+":"+MID$(T
     X$,3,2)+":"+RIGHT$(TX$,2)
1Ø4Ø PRINT"{HOME}";LEFT$(D$,2Ø);" {RVS}T
     IME=";TP$;"{2 SPACES}"
1Ø5Ø GETA$
1Ø6Ø IFA$="U"THEND=1
1Ø7Ø IFA$="D"THEND=2
1Ø8Ø IFA$="N"THEND=3
1Ø9Ø IFA$="S"THEND=4
11ØØ IFA$="E"THEND=5
111Ø IFA$="W"THEND=6
112Ø IFA$="F"GOTO128Ø
113Ø IFD=ØGOTO1Ø2Ø
114Ø IFMID$(S$(X,A),(Y-1)*6+D,1)<>"O"THE
     NPRINTCHR$(7):GOTO1Ø2Ø
115Ø ONDGOTO116Ø,117Ø,118Ø,119Ø,12ØØ,121Ø
116Ø A=A+1:GOTO122Ø
117Ø A=A-1:GOTO122Ø
118Ø Y=Y-1:GOTO122Ø
119Ø Y=Y+1:GOTO122Ø
12ØØ X=X+1:GOTO122Ø
121Ø X=X-1
122Ø IFX>5ORX<1ORY>5ORY<1ORA>5ORA<1THENP
     RINT"YOU WIN. PLAY AGAIN?":GOTO3ØØØ
123Ø GOSUB12Ø
124Ø FORII=1TO6:IFMID$(S$(X,A),(Y-1)*6+I
```

$5 says you can't beat Gridrunner.™

© 1983

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.

Pleases the tough customer.

HesWare™

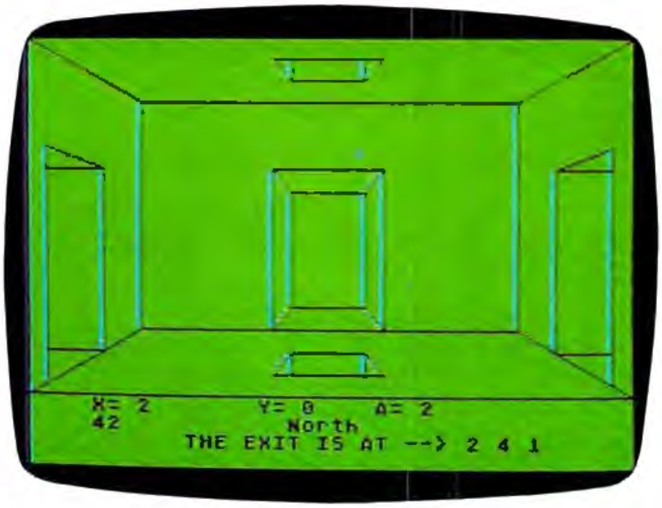
Human Engineered Software
150 North Hill Drive
Brisbane, CA 94005
800-227-6703
(in California 800 632-7979)
Dept. C20

VIC 20 and Commodore 64 are trademarks of Commodore Electronics Ltd. Atari™ is a trademark of Atari, Inc.

```
     I,1)="X"THENNEXT:GOTO1020
1250 R=FC(FC,II)+1
1260 ONRGOSUB125,130,135,140,145,150
1270 NEXT:GOTO1020
1280 PRINT"{HOME}";LEFT$(D$,22);"NEW FAC
     ING, N,S,E,W";
1281 GETC$:IFC$=""GOTO1280
1282 IFC$<>"N"ANDC$<>"S"ANDC$<>"E"ANDC$<
     >"W"GOTO1281
1283 PRINTC$:IFC$="N"THENFC=1
1284 IFC$="S"THENFC=2
1285 IFC$="E"THENFC=3
1286 IFC$="W"THENFC=4
1287 GOTO1220
2000 PRINTLEFT$(D$,8);LEFT$(R$,5);"{RVS}
     CAVES OF ICE{OFF}"
2001 PRINT"{3 DOWN}DO YOU WANT":INPUT"IN
     STRUCTIONS{3 SPACES}{3 LEFT}";Y$
2002 IFLEFT$(Y$,1)<>"Y"THENGOTO2100
2010 PRINT"{CLR}THE OBJECT OF {RVS}CAVES
     {OFF}":PRINT"IS TO FIND YOUR WAY"
2011 PRINT"OUT OF A 5X5X5 CUBIC":PRINT"M
     AZE. IN ONE OF THE
2012 PRINT"ROOMS THERE IS AN EXIT":PRINT
     "OUT OF THE MAZE.
2013 PRINT:PRINT"YOU MUST TRY TO FIND IT
     ":PRINT"IN THE MINIMUM TIME.
2014 PRINT"THE COMMANDS ARE :"
2020 PRINT"{RVS}U{OFF} - UP; {RVS}S{OFF}
      - SOUTH;":PRINT"{RVS}D{OFF} - DOWN
     ; {RVS}E{OFF} - EAST;
2030 PRINT"{RVS}N{OFF} - NORTH; {RVS}W
     {OFF} - WEST;"
2040 PRINT"{RVS}F{OFF} TO CHANGE FACING.
2050 PRINT:PRINT"HIT {RVS}RETURN{OFF} TO
      GO ON.
2051 GETC$:IFC$=""GOTO2051
2060 PRINT"{CLR}{RVS}F{OFF} WILL COME BA
     CK WITH A":PRINT"QUESTION AS TO WHI
     CH
2062 PRINT"FACING YOU WISH. HIT":PRINT"O
     NLY ONE KEY":PRINT"AND {RVS}RETURN
     {OFF}"
2100 PRINT:PRINT"PLEASE WAIT ABOUT 30":P
     RINT"SECONDS WHILE I SET UP":PRINT"
     THE MAZE.
2101 RETURN
3000 PRINT"{3 SPACES}{RVS}Y{OFF} OR
     {RVS}N{OFF}?"
3005 FORI=1TO10:PRINTCHR$(7);:NEXT
3010 GETC$:IFC$=""GOTO3010
3020 IFC$<>"Y"ANDC$<>"N"GOTO3010
3030 IFC$="N"THENSTOP
3032 PRINT"SAME MAZE {RVS}S{OFF} OR
3033 PRINT"NEW MAZE {RVS}N{OFF}?"
3034 GETC$:IFC$=""GOTO3034
3035 IFC$<>"S"ANDC$<>"N"GOTO3034
3036 IFC$="N"GOTO165
3040 X=SX:Y=SY:A=SA:GOTO1010
4000 DATA1,2,4,0,5,3,1,2,0,4,3,5,1,2,3,5
     ,4,0,1,2,5,3,0,4
```

Program 2:
Caves Of Ice – Changes For PET/CBM

```
120 PRINT"{CLR} $$$$$$$$$$$$$$$$$$$$
    {DOWN}{LEFT}N{DOWN}{2 LEFT}N{DOWN}
    {2 LEFT}N{DOWN}{2 LEFT}N"
121 PRINT"{HOME}{DOWN}{RIGHT}M{DOWN}M
    {DOWN}M{DOWN}M$$$$$$$$$$$$"
122 PRINT"{HOME}{DOWN}";:FORI=1TO18:PRIN
    T"'{DOWN}{LEFT}";:NEXT:PRINT"{UP}
    {RIGHT}N{UP}N{UP}N{UP}N{UP}";
123 PRINT"%";:FORI=1TO9:PRINT"{UP}{LEFT}
    %";:NEXT
124 PRINT"{HOME}";LEFT$(D$,19);"'{RVS}
    {20 SPACES}{OFF}%{LEFT}{UP}{LEFT}M
    {UP}{2 LEFT}M{UP}{2 LEFT}M{UP}
    {2 LEFT}M"
125 PRINT"{HOME}"LEFT$(D$,15);LEFT$(R$,5
    );"#############"
126 PRINT"{HOME}";LEFT$(R$,21);:FORI=1TO
    18:PRINT"{DOWN}%{LEFT}";:NEXT
127 PRINT"{HOME}";LEFT$(R$,17);LEFT$(D$,
    5);:FORI=1TO10:PRINT"%{DOWN}{LEFT}";
    :NEXT
129 RETURN
130 PRINT"{HOME}";LEFT$(R$,8);"{DOWN}$$$
    $$${DOWN}{6 LEFT}M%{2 SPACES}'N
    {DOWN}{5 LEFT}####":RETURN
135 PRINT"{HOME}";LEFT$(D$,16);LEFT$(R$,
    9);"$$$${DOWN}{5 LEFT}N%{2 SPACES}'M
    {DOWN}{6 LEFT}######":RETURN
140 PRINT"{HOME}";LEFT$(D$,7);"{RIGHT}";
    :FORI=1TO11:PRINT"'{DOWN}{LEFT}";:NE
    XT:PRINT"{RIGHT}{3 UP}$:"
141 PRINT"{HOME}";LEFT$(D$,9);"{2 RIGHT}
    #P";:FORI=1TO5:PRINT"{DOWN}{LEFT}'";
    :NEXT
142 PRINT"{HOME}";LEFT$(D$,7);"{2 RIGHT}
    M{DOWN}M":RETURN
145 PRINT"{HOME}";LEFT$(D$,7);LEFT$(R$,9
    );"$$$${DOWN}{5 LEFT}'M$$N%";:FORI=1
    TO5
146 PRINT"{DOWN}{6 LEFT}' %' %";:NEXT:PR
    INT"{DOWN}{6 LEFT}'N##M%":RETURN
150 PRINT"{HOME}";LEFT$(D$,7);LEFT$(R$,1
    9);"N%{DOWN}{3 LEFT}N %{DOWN}
    {3 LEFT}O#%";:FORI=1TO5
151 PRINT"{DOWN}{3 LEFT}% %";:NEXT:PRINT
    "{DOWN}{3 LEFT}L$%{DOWN}{LEFT}%
    {DOWN}{LEFT}%":RETURN
```

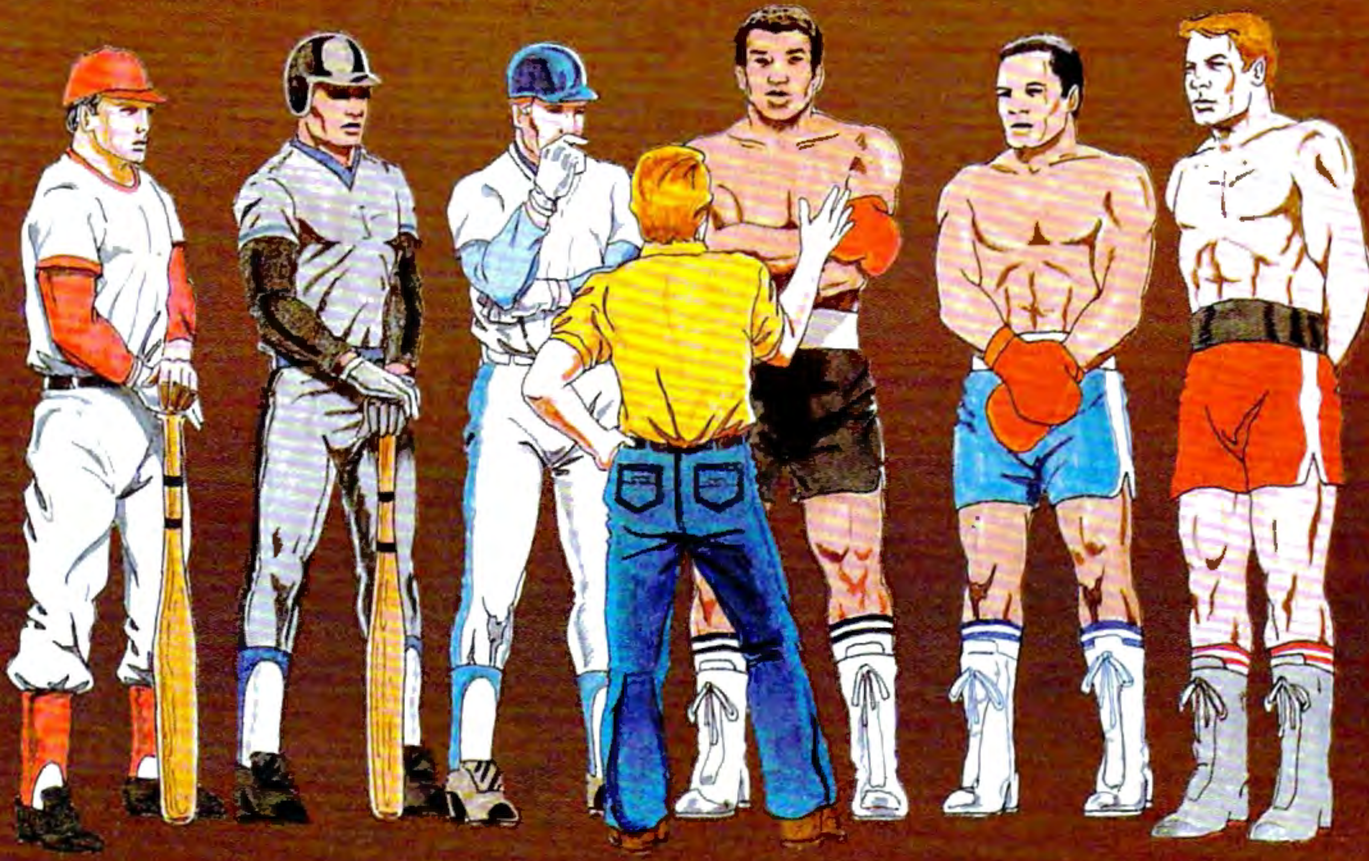


"Caves of Ice" Atari version.

Program 3: Caves Of Ice – Atari Version

```
10 GOTO 100
15 L=PEEK(708):POKE 708,PEEK(710):PO
   KE 710,L
16 B=B+AB:IF B=1 OR B=5 THEN AB=AB*-
   1
17 T=T+INT(RND(0)*3-1):IF T<1 THEN T
   =5
18 IF T>5 THEN T=1
20 SPOT=6*X+30*Y+150*A+I:RETURN
30 COLOR 1
32 ? #6;"{CLEAR}":PLOT 0,0:DRAWTO 59
   ,29:DRAWTO 59,129:DRAWTO 259,129:
   DRAWTO 259,29:DRAWTO 59,29:PLOT 3
   19,0:DRAWTO 259,29
35 PLOT 319,159:DRAWTO 259,129:PLOT
   0,159:DRAWTO 59,129:PLOT 0,159:DR
   AWTO 319,159:SETCOLOR 2,A+11,10:R
   ETURN
40 PLOT 129,9:DRAWTO 189,9:DRAWTO 17
   9,19:DRAWTO 139,19:DRAWTO 129,9:P
   LOT 139,9:DRAWTO 139,19:PLOT 179,
   9:DRAWTO 179,19
45 RETURN
50 PLOT 129,149:DRAWTO 189,149:DRAWT
   O 179,139:DRAWTO 139,139:DRAWTO 1
   29,149:PLOT 139,139:DRAWTO 139,14
   9
55 PLOT 179,139:DRAWTO 179,149:RETUR
   N
60 PLOT 279,59:DRAWTO 309,49:DRAWTO
   309,154:DRAWTO 279,139:DRAWTO 279
   ,59:DRAWTO 309,59:PLOT 279,139
65 DRAWTO 309,139:RETURN
70 PLOT 9,49:DRAWTO 39,59:DRAWTO 39,
   139:DRAWTO 9,154:DRAWTO 9,49:PLOT
    9,59:DRAWTO 39,59:PLOT 9,139
75 DRAWTO 39,139:RETURN
80 PLOT 129,59:DRAWTO 189,59:DRAWTO
   189,129:DRAWTO 129,129:DRAWTO 129
   ,59:DRAWTO 139,69:DRAWTO 179,69
82 DRAWTO 179,119:DRAWTO 139,119:DRA
   WTO 139,69:PLOT 179,69:DRAWTO 189
   ,59:PLOT 189,129:DRAWTO 179,119:P
   LOT 129,129
85 DRAWTO 139,119:RETURN
90 RETURN
100 DIM A$(750),G$(1),FC(4,6),SC$(20
    0):A$(750)="O":FOR I=20 TO 192 S
    TEP 11:SC$(I)="->YOU WIN<-":NEXT
     I
101 SC$(1,19)="{19 SPACES}":SC$(LEN(S
    C$)+1)="{10 SPACES}"
102 GOSUB 3000
105 FOR FC=1 TO 4:FOR I=1 TO 6:READ
    A:FC(FC,I)=A:NEXT I:NEXT FC
107 X=0
109 Y=0
110 A=0
115 GOSUB 250
120 IF RND(0)<0.7 AND A<4 THEN I=1:G
    OSUB 20:A$(SPOT,SPOT)="X"
130 IF RND(0)<0.7 AND X<4 THEN I=3:G
    OSUB 20:A$(SPOT,SPOT)="X"
140 IF RND(0)<0.7 AND Y<4 THEN I=5:G
    OSUB 20:A$(SPOT,SPOT)="X"
150 IF A>0 THEN A=A-1:I=1:GOSUB 20:A
    =A+1:IF A$(SPOT,SPOT)="X" THEN I
    =2:GOSUB 20:A$(SPOT,SPOT)="X"
160 IF X>0 THEN X=X-1:I=3:GOSUB 20:X
    =X+1:IF A$(SPOT,SPOT)="X" THEN I
    =4:GOSUB 20:A$(SPOT,SPOT)="X"
170 IF Y>0 THEN Y=Y-1:I=5:GOSUB 20:Y
    =Y+1:IF A$(SPOT,SPOT)="X" THEN I
    =6:GOSUB 20:A$(SPOT,SPOT)="X"
190 A=A+1:IF A<5 THEN 115
191 Y=Y+1:IF Y<5 THEN 110
192 X=X+1:IF X<5 THEN 109
193 SOUND 0,0,0,0:SOUND 1,0,0,0
195 GOTO 300
200 GOSUB 30:FOR Q=1 TO 6:I=Q
210 GOSUB 20:IF A$(SPOT,SPOT)<>"X" T
    HEN 230
215 I=FC(F,Q)
220 ON I GOSUB 40,50,60,70,80,90
230 NEXT Q:RETURN
250 L=PEEK(708):POKE 708,PEEK(709):P
    OKE 709,PEEK(710):POKE 710,L
255 SOUND 0,BASS(O,B),10,10:SOUND 1,
    TREBLE(O,T),10,14
260 B=B+AB:IF B=1 THEN O=O+1:AB=1:IF
     O=4 THEN O=1
270 IF B=5 THEN AB=AB*-1
280 T=T+INT(RND(0)*3-1):IF T>5 THEN
    T=1
290 IF T<1 THEN T=5
295 RETURN
300 OPEN #1,4,0,"K:"
305 GRAPHICS 8
310 X=INT(RND(0)*5)
315 Y=INT(RND(0)*5)
320 A=INT(RND(0)*5)
330 I=INT(RND(0)*6+1):ON I GOTO 335,
    340,345,350,355,360
335 A=4:GOTO 370
340 A=0:GOTO 370
345 X=4:GOTO 370
350 X=0:GOTO 370
355 Y=4:GOTO 370
360 Y=0
370 GOSUB 20:A$(SPOT,SPOT)="X":SX=X:
    SY=Y:SA=A
375 X=INT(RND(0)*5)
380 Y=INT(RND(0)*5)
385 A=INT(RND(0)*5)
390 POKE 19,0:POKE 20,0
400 POKE 752,1:SETCOLOR 1,0,0
410 F=1:GOSUB 200
420 SETCOLOR 1,0,0
500 IF PEEK(764)=255 THEN 1000
510 GET #1,G:G$=CHR$(G)
```

GIVE THE WORLD'S GREATEST SLUGGERS A PIECE OF YOUR MIND.



Presenting two strategy sports games from SSI: COMPUTER BASEBALL™ & RINGSIDE SEAT™

Already voted the "1982 BEST COMPUTER SPORTS GAME" in its Apple® format by a leading games magazine, COMPUTER BASEBALL is now available for the Commodore®64.

As the most advanced and realistic simulation of baseball, it lets you manage any major-league team you like, giving you over 25 options with which to direct your players.

Complete data for over 20 historical AL and NL teams are provided so you can re-create immemorable games of the past or play out hypothetical matchups. Imagine the thrill of pitting the '27 Yankees against the '75 Reds! You can even enter your own player statistics and create a team to your specifications.

Best of all, to get all this detailed realism, you don't have to give up on speed and excitement. The action onscreen is kept at a fast pace to make you feel like you're actually playing in the ballpark! After all, that's how COMPUTER BASEBALL got to be recognized for what it is — the best.

See it today at your nearest computer/game/software store!

RINGSIDE SEAT packs all the right punches to make it the uncontested heavyweight champion of boxing games. It lets you set up your own championship matches, choosing from among fifty of the best fighters of all time.

Now you can answer all the intriguing questions in boxing: Who really was the greatest? Could Joe Louis outwit Muhammed Ali? Could Jack Dempsey outslug Larry Holmes?

RINGSIDE SEAT recreates every champion's authentic style and the strength, speed and stamina he had in his prime. As his manager, you tell him how to fight: stick and move, cover up, go for the knockout. The semi-animated Hi-Res color display shows you the fight in the ring as it happens, blow-by-blow, round-by-round.

This two-player and computer-as-opponent game also allows you to make up your own fighters by rating them for style, speed, aggressiveness, strength and a dozen other factors.

RINGSIDE SEAT. It not only lets you tell the great sluggers of the world how to fight, it makes them listen to you!

COMPUTER BASEBALL ($39.95) comes on 64K disk for the Commodore® 64. Also on 48K disk for the Apple® II. Coming soon for the IBM® PC.

RINGSIDE SEAT ($39.95) comes on 64K disk for the Commodore® 64 or on 48K disk for the Apple® II with Applesoft ROM, II+, IIe, or Apple III.

SSI

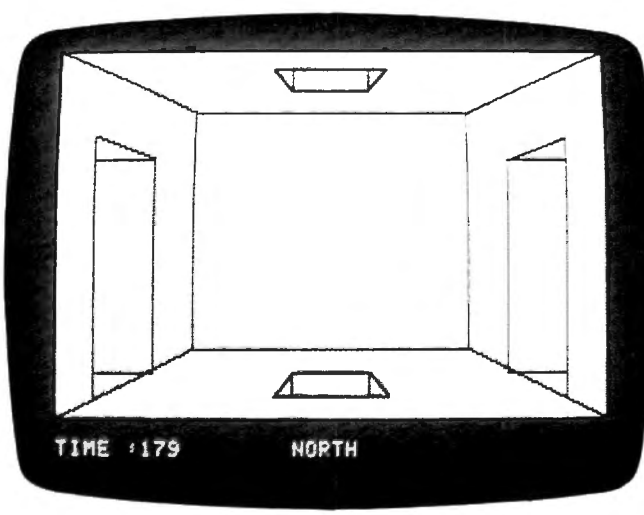
STRATEGIC SIMULATIONS INC

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. (California residents, add 6.5% sales tax.)

WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.

```
515 D=Ø
520 IF G$="U" THEN D=1
525 IF G$="*" THEN TELLIT=1
530 IF G$="D" THEN D=2
535 IF G$="?" THEN SHOWIT=1
540 IF G$="E" THEN D=3
545 TRAP 545:IF G$="F" THEN ? "
    {CLEAR}{DOWN}":INPUT FB:IF FB<5
    AND FB>Ø THEN F=FB:GOSUB 2ØØ:GOT
    O 1ØØØ
550 IF G$="W" THEN D=4
560 IF G$="N" THEN D=5
570 IF G$="S" THEN D=6
575 IF D<1 OR D>7 THEN 5ØØ
580 I=D:GOSUB 2Ø
590 IF A$(SPOT,SPOT)<>"X" THEN GOSUB
     9ØØ:GOTO 1ØØØ
6ØØ ON D GOTO 6Ø5,61Ø,615,62Ø,625,63
    Ø
6Ø5 A=A+1:GOTO 64Ø
61Ø A=A-1:GOTO 64Ø
615 X=X+1:GOTO 64Ø
62Ø X=X-1:GOTO 64Ø
625 Y=Y+1:GOTO 64Ø
63Ø Y=Y-1
64Ø IF A<Ø OR A>4 OR X<Ø OR X>4 OR Y
    <Ø OR Y>4 THEN 2ØØØ
65Ø GOSUB 2ØØ
66Ø GOTO 1ØØØ
9ØØ FOR Q=1 TO 4:CC=(CC=Ø)*14:POKE 7
    1Ø,CC:SOUND Ø,CC*7+6Ø,CC,1Ø:FOR
    W=1 TO 1Ø:NEXT W:NEXT Q
91Ø SOUND Ø,Ø,Ø,Ø:SETCOLOR 2,A+11,1Ø
    :SETCOLOR 1,Ø,Ø:RETURN
1ØØØ TRAP 4ØØØØ:GOSUB 14ØØ:GOSUB 13Ø
     Ø:IF TIME2=TIME THEN GOTO 5ØØ
1ØØ5 POKE 657,4:POKE 656,1:? "
     {3 SPACES}{3 LEFT}";TIME:TIME2=
     TIME:IF TELLIT=Ø THEN 5ØØ
1Ø1Ø POKE 656,Ø:POKE 657,4:? "X= ";X
     ;"{TAB}Y= ";Y;"{TAB}A= ";A;:IF
     SHOWIT=Ø THEN 5ØØ
1Ø2Ø SHOWIT=Ø:POKE 656,2:POKE 657,1Ø
     :? "THE EXIT IS AT --> ";SX;" "
     ;SY;" ";SA;:GOTO 5ØØ
13ØØ TIME=INT(4.25*PEEK(19)+PEEK(2Ø)
     /6Ø):RETURN
14ØØ POKE 657,17:POKE 656,1:ON F GOS
     UB 14Ø5,141Ø,1415,142Ø:RETURN
14Ø5 ? "North":RETURN
141Ø ? "South":RETURN
1415 ? "East":RETURN
142Ø ? "West":RETURN
2ØØØ GRAPHICS 2+16:SETCOLOR 1,4,12:S
     ETCOLOR 2,15,8:SETCOLOR 3,1Ø,4:
     SETCOLOR Ø,Ø,15
2ØØ5 C=1
2Ø1Ø FOR I=1 TO 18
2Ø2Ø C=C+1:IF C>3 THEN C=1
2Ø3Ø ON C GOTO 2Ø32,2Ø35,2Ø37
2Ø32 COLOR 1Ø:GOTO 2Ø4Ø
2Ø35 COLOR 17Ø:GOTO 2Ø4Ø
2Ø37 COLOR 138:GOTO 2Ø4Ø
2Ø4Ø PLOT I,Ø:PLOT I,11
2Ø43 TRAP 2Ø5Ø
2Ø45 PLOT Ø,I:PLOT 19,I
2Ø5Ø NEXT I
2Ø55 POSITION Ø,Ø:? #6;"{J}":POSITIO
     N 19,Ø:? #6;"{J}"
2Ø6Ø POSITION 5,2:? #6;"YOU ESCAPED"
2Ø65 POSITION 4,3:? #6;"CAVES OF ICE"
2Ø7Ø POSITION 4,7:? #6;"IN ";TIME;"
     SECONDS"
2Ø8Ø POSITION 5,9:? #6;"HIT ANY KEY"
2Ø9Ø I=1
21ØØ L=PEEK(7Ø9):POKE 7Ø9,PEEK(71Ø):
     POKE 71Ø,PEEK(711):POKE 711,L
21Ø5 GOSUB 255:POSITION 1,5:? #6;SC$
     (I,I+17):I=I+1:IF I>18Ø THEN I=
     1
21Ø7 IF PEEK(764)<>255 THEN 22ØØ
211Ø FOR W=1 TO 15:NEXT W:GOTO 21ØØ
22ØØ POKE 764,255:GRAPHICS Ø:POSITIO
     N 4,4:SOUND Ø,Ø,Ø,Ø:SOUND 1,Ø,Ø
     ,Ø
221Ø ? "PLAY AGAIN ";:INPUT A$
222Ø IF A$(1,1)="Y" THEN RUN
223Ø END
3ØØØ GRAPHICS 18
3Ø1Ø POSITION 7,4:PRINT #6;"CAVES"
3Ø2Ø POSITION 6,7:PRINT #6;"of  ice"
3Ø3Ø COLOR 138
3Ø4Ø PLOT 1,1:DRAWTO 18,1
3Ø5Ø DRAWTO 18,1Ø:DRAWTO 1,1Ø:DRAWTO
      1,1
3Ø6Ø AB=1
3Ø7Ø DIM BASS(3,5),TREBLE(3,5)
3Ø8Ø RESTORE 31ØØ
3Ø9Ø FOR I=1 TO 3:FOR T=1 TO 5:READ
     B,TR:BASS(I,T)=B:TREBLE(I,T)=TR
     :NEXT T:NEXT I:T=1:B=1:O=1:RETU
     RN
31ØØ DATA 243,121,193,96,162,81,144,
     72,136,68
311Ø DATA 182,91,144,72,121,6Ø,1Ø8,5
     3,1Ø2,5Ø
312Ø DATA 162,81,128,64,1Ø8,53,96,47
     ,91,45
313Ø DATA 1,2,3,4,5,6,1,2,4,3,6,5,1,
     2,5,6,4,3,1,2,6,5,3,4
```



A room with four possible exits. Apple version of "Caves of Ice."

Program 4: Caves Of Ice – Apple Version

```
1  DATA  2Ø1,84,2Ø8,15,32,177,Ø,32,248,
     23Ø,138,72,32,183,Ø,2Ø1,44,24Ø,3,7
     6,2Ø1,222,32,177,Ø,32,248,23Ø
2  FOR I = 768 TO 833: READ P: POKE I,P
     : NEXT I
3  DATA  1Ø4,134,3,134,1,133,Ø,17Ø,16Ø,
     1,132,2,173,48,192,136,2Ø8,4,198
4  DATA  1,24Ø,7,2Ø2,2Ø8,246,166,Ø,2Ø8,
```

Atari And Apple Versions

Robert Tsuk

When I received Marv Bunker's letter, I agreed a version should be made available for Commodore owners. Also, as an Atari owner, I wanted to include a version for that computer, too. The lack of dimensioned string variables on the Atari made it tricky, but, as evidenced by Program 3, it was successfully adopted.

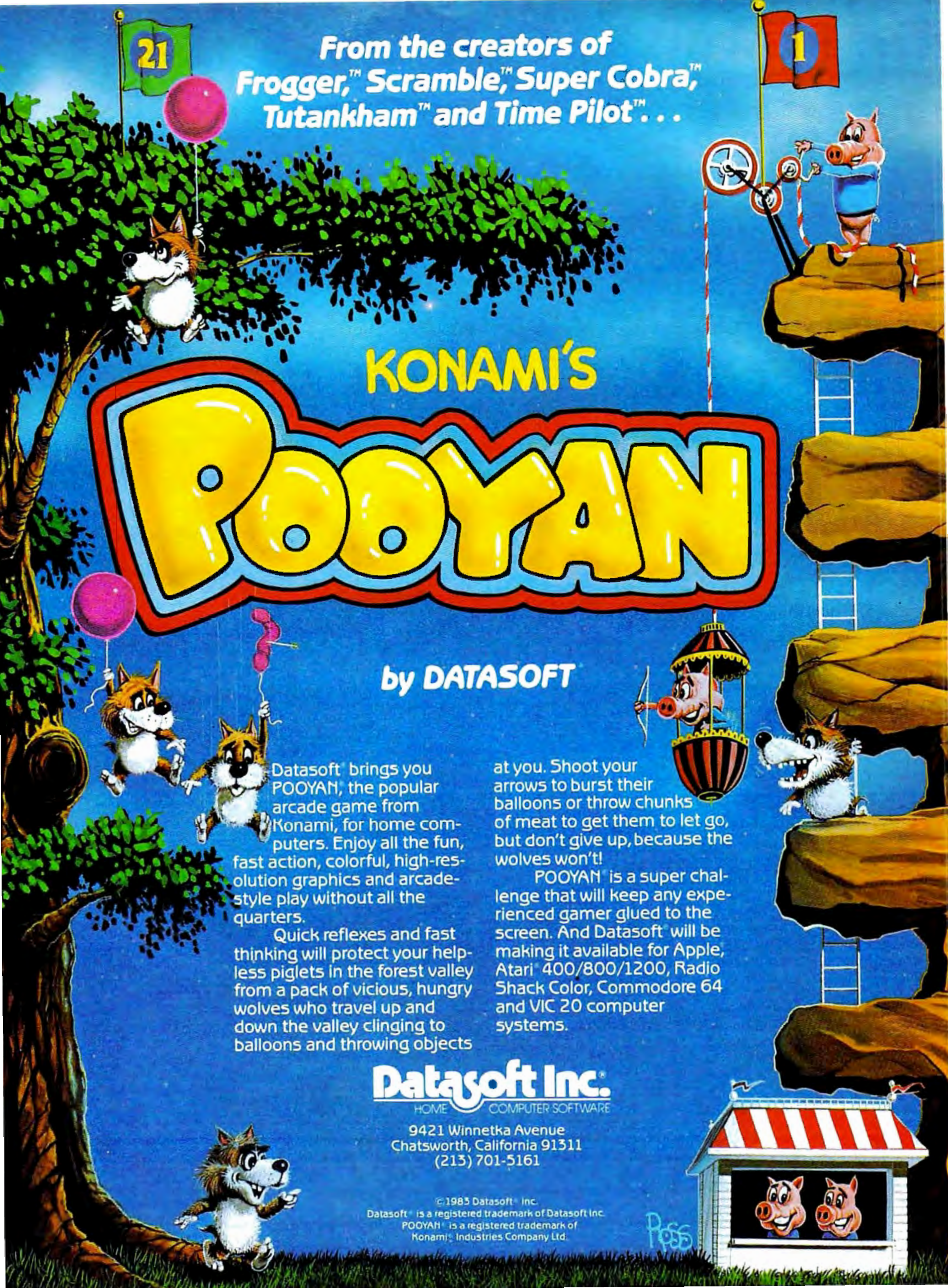
If you find the game too challenging, the Atari and Apple versions have several features not found in the Commodore version which may be of assistance. If you type an * the program will tell you your location in the maze. You are given your X and Y coordinates (0-4) on the current level, and a value for A, which indicates which level (0-4) you are currently on. If you get really lost, you can find your location by typing the *, then a ?. The program will briefly display the X, Y, and A coordinates of the exit. But remember, using the ? is frowned upon unless you're really lost.

As in the Commodore version, you move through the maze by typing the N, S, E, W, U, and D keys to specify the direction of movement. However, if you type F to change the direction you are facing, you must then type a number instead of a letter to specify the new direction you wish to face. You'll need to remember that N=1, S=2, E=3, and W=4.

The Apple version of Caves of Ice (Program 4), the original Quinti-Maze, uses almost 48K. It includes a SAVE the game feature. Special attention must be paid to the first five lines and the data therein, as a mistake in the data will cause a crash in line 167.

The Atari version uses just over 16K. Although it has no SAVE the game feature, it has some pretty flashy graphics and sound.

```
     239,165,3,133,1,198,2,208,241,96
5  POKE 1013,76: POKE 1014,0: POKE 1015
     ,3
10  TEXT : HOME
90  GOSUB 2000
100  DIM FC(5,7): DIM FC$(5)
105 FC$(1) = "NORTH":FC$(2) = "SOUTH":F
     C$(3) = "EAST":FC$(4) = "WEST"
110  FOR B = 1 TO 4: FOR I = 1 TO 6: READ
     FC(B,I): NEXT : NEXT
115  GOTO 155
120  HPLOT 0,0 TO 279,0 TO 279,159 TO 0
     ,159 TO 0,0 TO 69,29 TO 209,29 TO
     209,129 TO 69,129 TO 69,29: HPLOT
     209,29 TO 279,0: HPLOT 209,129 TO
     279,159: HPLOT 69,129 TO 0,159: RETURN
125  RETURN
130  HPLOT 109,9 TO 169,9 TO 159,19 TO
     119,19 TO 109,9: HPLOT 119,19 TO 1
     19,9: HPLOT 159,19 TO 159,9: RETURN
135  HPLOT 119,139 TO 159,139 TO 169,14
     9 TO 109,149 TO 119,139: HPLOT 119
     ,139 TO 119,149: HPLOT 159,139 TO
     159,149: RETURN
140  HPLOT 19,39 TO 49,49 TO 49,139: HPLOT
     19,149 TO 19,39: HPLOT 19,139 TO 4
     9,139: HPLOT 19,49 TO 49,49: RETURN
145  HPLOT 119,59 TO 159,59 TO 159,129 TO
     119,129 TO 119,59 TO 129,69 TO 149
     ,69 TO 149,119 TO 129,119 TO 129,6
     9: HPLOT 149,69 TO 159,59: HPLOT 1
     49,119 TO 159,129: HPLOT 129,119 TO
     119,129: RETURN
150  HPLOT 229,49 TO 259,39 TO 259,149:
      HPLOT 229,139 TO 229,49: HPLOT 22
     9,49 TO 259,49: HPLOT 229,139 TO 2
     59,139: RETURN
155  DIM S$(6,6)
160  INPUT "RESTART OLD MAZE ";Y$: IF  LEFT$
     (Y$,1) = "Y" THEN 1360
165  FOR A = 1 TO 5: FOR X = 1 TO 5: FOR
     Y = 1 TO 5
167  & T10 * A + 10 * X + 10 * Y,10
170  IF A <  > 5 AND  RND (1) < .80 THEN
     S$(X,A) = S$(X,A) + "O": GOTO 180
175 S$(X,A) = S$(X,A) + "X"
180  IF  MID$ (S$(X,A - 1),(Y - 1) * 6 +
     1,1) = "O" THEN S$(X,A) = S$(X,A) +
     "O": GOTO 190
185 S$(X,A) = S$(X,A) + "X"
190  IF Y - 2 < 0 THEN 200
195  IF  MID$ (S$(X,A),(Y - 2) * 6 + 4,
     1) = "O" THEN S$(X,A) = S$(X,A) +
     "O": GOTO 205
200 S$(X,A) = S$(X,A) + "X"
205  IF Y <  > 5 AND  RND (1) < .8 THEN
     S$(X,A) = S$(X,A) + "O": GOTO 215
210 S$(X,A) = S$(X,A) + "X"
215  IF X <  > 5 AND  RND (1) < .8 THEN
     S$(X,A) = S$(X,A) + "O": GOTO 225
220 S$(X,A) = S$(X,A) + "X"
225  IF  MID$ (S$(X - 1,A),(Y - 1) * 6 +
     5,1) = "O" THEN S$(X,A) = S$(X,A) +
     "O": GOTO 235
230 S$(X,A) = S$(X,A) + "X"
235  NEXT : NEXT : NEXT
240 X =  INT ( RND (1) * 3) + 2:Y =  INT
     ( RND (1) * 3) + 2:A =  INT ( RND
     (1) * 3) + 2
245 RD =  INT ( RND (1) * 6) + 1: ON RD
      GOTO 250,255,260,265,270,275
250 A = 5:P1$ =  LEFT$ (S$(X,A),(Y - 1)
     * 6):L = 29 -  LEN (P1$):P2$ =  RIGHT$
     (S$(X,A),L):S$(X,A) = P1$ + "O" +
     P2$: GOTO 280
255 A = 1:P1$ =  LEFT$ (S$(X,A),(Y - 1)
     * 6 + 1):L = 29 -  LEN (P1$):P2$ =
     RIGHT$ (S$(X,A),L):S$(X,A) = P1$ +
     "O" + P2$: GOTO 280
260 Y = 5:P1$ =  LEFT$ (S$(X,A),(Y - 1)
     * 6 + 3):L = 29 -  LEN (P1$):P2$ =
     RIGHT$ (S$(X,A),L):S$(X,A) = P1$ +
     "O" + P2$: GOTO 280
265 Y = 1:P1$ =  LEFT$ (S$(X,A),(Y - 1)
     * 6 + 2):L = 29 -  LEN (P1$):P2$ =
     RIGHT$ (S$(X,A),L):S$(X,A) = P1$ +
```



From the creators of
Frogger,™ Scramble,™ Super Cobra,™
Tutankham™ and Time Pilot™. . .
21
1
KONAMI'S
POOYAN
by DATASOFT®
Datasoft® brings you POOYAN,® the popular arcade game from Konami, for home computers. Enjoy all the fun, fast action, colorful, high-resolution graphics and arcade-style play without all the quarters.
Quick reflexes and fast thinking will protect your helpless piglets in the forest valley from a pack of vicious, hungry wolves who travel up and down the valley clinging to balloons and throwing objects at you. Shoot your arrows to burst their balloons or throw chunks of meat to get them to let go, but don't give up, because the wolves won't!
POOYAN® is a super challenge that will keep any experienced gamer glued to the screen. And Datasoft® will be making it available for Apple,® Atari® 400/800/1200, Radio Shack Color, Commodore 64 and VIC 20 computer systems.
Datasoft Inc.
HOME COMPUTER SOFTWARE
9421 Winnetka Avenue
Chatsworth, California 91311
(213) 701-5161
© 1983 Datasoft® Inc.
Datasoft® is a registered trademark of Datasoft Inc.
POOYAN® is a registered trademark of
Konami® Industries Company Ltd.
Ross

```
     "O" + P2$: GOTO 280
270 X = 5:P1$ =  LEFT$ (S$(X,A),(Y - 1)
     * 6 + 4):L = 29 -  LEN (P1$):P2$ =
     RIGHT$ (S$(X,A),L):S$(X,A) = P1$ +
    "O" + P2$: GOTO 280
275 X = 1:P1$ =  LEFT$ (S$(X,A),(Y - 1)
     * 6 + 5):L = 29 -  LEN (P1$):P2$ =
     RIGHT$ (S$(X,A),L):S$(X,A) = P1$ +
    "O" + P2$: GOTO 280
280 SX = X:SY = Y:SA = A
290  VTAB 23: PRINT "HIT ANY KEY TO STA
     RT"
300  IF  PEEK ( - 16384) < 127 THEN 300
310  POKE  - 16368,00
1000 X =  INT ( RND (1) * 5) + 1:Y =  INT
     ( RND (1) * 5) + 1:A =  INT ( RND
     (1) * 5) + 1:FC = 1: GOTO 1220
1010  HOME : VTAB 22: HTAB 18: PRINT FC
     $(FC):A$ = "":D = 0: IF LS = 1 THEN
      PRINT X,Y,A
1020  VTAB 22: PRINT "TIME :";T: FOR TI
     ME = 1 TO 80
1025  IF  PEEK ( - 16384) > 127 THEN 10
     30
1027  NEXT :T = T + 1: VTAB 22: PRINT "
     TIME :";T: GOTO 1020
1030  GET A$
1035  IF A$ = "*" THEN LS = 1
1040  IF A$ = "Q" THEN 1300
1050  IF A$ = "U" THEN D = 1
1060  IF A$ = "D" THEN D = 2
1070  IF A$ = "N" THEN D = 3
1080  IF A$ = "S" THEN D = 4
1090  IF A$ = "E" THEN D = 5
1100  IF A$ = "?" THEN 1290
1110  IF A$ = "W" THEN D = 6
1120  IF A$ = "F" THEN  GOTO 1280
1130  IF D = 0 THEN 1010
1135 T = T + 1
1140  IF  MID$ (S$(X,A),(Y - 1) * 6 + D
     ,1) <  > "O" THEN  PRINT  CHR$ (7)
     : GOTO 1010
1150  ON D GOTO 1160,1170,1180,1190,120
     0,1210
1160 A = A + 1: GOTO 1220
1170 A = A - 1: GOTO 1220
1180 Y = Y - 1: GOTO 1220
1190 Y = Y + 1: GOTO 1220
1200 X = X + 1: GOTO 1220
1210 X = X - 1: GOTO 1220
1220  IF X > 5 OR X < 1 OR Y > 5 OR Y <
      1 OR A > 5 OR A < 1 THEN  PRINT "Y
     OU WIN": & T100,100: & T100,50: &
     T100,50: & T75,66: & T100,66: & T7
     5,66: & T60,255: GOTO 3000
1230  HGR : HCOLOR= 3: HPLOT 0,0: CALL
     62454: HCOLOR= 0: GOSUB 120
1240  FOR I = 1 TO 6: IF  MID$ (S$(X,A)
     ,(Y - 1) * 6 + I,1) = "X" THEN  NEXT
     : GOTO 1010
1250 R = FC(FC,I) + 1
1260  HCOLOR= 0: ON R GOSUB 125,130,135
     ,140,145,150
1270  NEXT : GOTO 1010
1280  INPUT "WHAT FACING 1-N 2-S 3-E 4-
     W";FC: IF FC < 1 OR FC > 4 THEN 1280
1285  GOTO 1220
1290  INVERSE : HTAB 18: PRINT SX;"  ";
     SY;"  ";SA: NORMAL : GOTO 1220
1300  PRINT "DO YOU WANT TO SAVE THIS M
     AZE": INPUT Y$: IF  LEFT$ (Y$,1) <
      > "Y" THEN  GOTO 3000
1310  INPUT "WHAT DO YOU WANT TO CALL I
     T ";N$
1320 D$ =  CHR$ (4)
1330  PRINT D$;"OPEN OLD MAZE/";N$: PRINT
     D$;"WRITE OLD MAZE/";N$
1340  FOR A1 = 1 TO 5: FOR X1 = 1 TO 5:
      PRINT S$(X1,A1): NEXT : NEXT : PRINT
     X: PRINT Y: PRINT A: PRINT T: PRINT FC
1350  PRINT D$;"CLOSE OLD MAZE/";N$: GOTO
     3000
1360  INPUT "WHAT IS ITS NAME ";N$
1370 D$ =  CHR$ (4)
1380  PRINT D$;"OPEN OLD MAZE/";N$: PRINT
     D$;"READ OLD MAZE/";N$
1390  FOR A1 = 1 TO 5: FOR X1 = 1 TO 5:
      INPUT S$(X1,A1): NEXT : NEXT : INPUT
     X: INPUT Y: INPUT A: INPUT T: INPUT FC
1400  PRINT D$;"CLOSE OLD MAZE/";N$: GOTO
     1220
2000  VTAB 12: HTAB 18: INVERSE : PRINT
     "MAZE": NORMAL : VTAB 22: INPUT "D
     O YOU WANT INSTRUCTIONS ";Y$: IF  LEFT$
     (Y$,1) <  > "Y" THEN  RETURN
2010  HOME : PRINT "THE OBJECT OF MAZE
     IS TO FIND YOUR WAY": PRINT : PRINT
     "OUT OF A 5X5X5 CUBIC MAZE. IN ONE
      OF THE": PRINT "ROOMS THERE IS AN
      EXIT OUT OF THE MAZE."
2020  PRINT : PRINT "YOU MUST TRY TO FI
     ND IT IN AS FEW TURNS ": PRINT "AS
      POSSIBLE. THE COMMANDS ARE  :"
2030  PRINT : HTAB 6: INVERSE : PRINT "
     U";: NORMAL : PRINT "-UP";: HTAB 1
     7: INVERSE : PRINT "S";: NORMAL : PRINT
     "-SOUTH"
2040  PRINT : HTAB 6: INVERSE : PRINT "
     D";: NORMAL : PRINT "-DOWN";: HTAB
     17: INVERSE : PRINT "E";: NORMAL :
      PRINT "-EAST"
2050  PRINT : HTAB 6: INVERSE : PRINT "
     N";: NORMAL : PRINT "-NORTH";: HTAB
     17: INVERSE : PRINT "W";: NORMAL :
      PRINT "-WEST"
2060  PRINT : HTAB 6: INVERSE : PRINT "
     Q";: NORMAL : PRINT "-QUIT";: HTAB
     17: INVERSE : PRINT "F";: NORMAL :
      PRINT "-CHANGE FACING"
2070  VTAB 23: PRINT "HIT ";: INVERSE :
      PRINT "SPACE";: NORMAL : PRINT "
     FOR MORE"
2080  IF  PEEK ( - 16384) < 127 THEN 2080
2090  POKE  - 16368,0: HOME : INVERSE :
      PRINT "F";: NORMAL : PRINT " WILL
      COME BACK WITH A QUESTION AS TO":
      PRINT : PRINT "WHICH FACING YOU W
     ISH.HIT ONLY ONE KEY": PRINT : PRINT
     "AND ";: INVERSE : PRINT "RETURN":
      NORMAL
2100  PRINT : PRINT "PLEASE WAIT WHILE
     IT SETS UP THE MAZE": PRINT : PRINT
     : RETURN
3000  TEXT : HOME : VTAB 5: HTAB 12: PRINT
     "CONGRATULATIONS !"
3010  PRINT : PRINT  TAB( 7)"YOU HAVE F
     INISHED THE MAZE IN ": PRINT  TAB(
     7)T;" SECONDS"
3030  INPUT "DO YOU WANT TO PLAY AGAIN
     ? ";Y$
3040  IF  LEFT$ (Y$,1) = "Y" THEN  RUN
9999  NORMAL
10000  DATA    1,2,4,0,5,3,1,2,0,4,3,5
     ,1,2,3,5,4,0,1,2,5,3,0,4
```
©

GRADEBOOK FOR ATARI

Stephen Levy, Assistant Book Editor

This is a valuable organizational tool for teachers. It handles student lists, grading conversions, grade averaging, assignments, and much more. Written for an Atari computer with at least 32K and a disk drive.

"Gradebook" is for teachers. It will keep a record of students' grades and assignments for up to 45 students on one diskette. In addition, the program will average grades and display grades and assignments to the screen or list them to a printer.

SAVE the program on one diskette and use a second diskette for data. Use the following short program to create a dummy file on the program diskette to prevent accidentally writing data to the program diskette:

```
10 OPEN #1,8,0,"D:CL"
20 DIM A$(4):A$="TEST":PRINT #1;A$
30 CLOSE #1:END
```

Menu Options

1. *Read Grades:* produces a list of the last names of all students previously entered (option 3) onto this diskette, plus each student's grades and average. You will be prompted for the *number* (the program will automatically number the students for you) of the first and last student whose grade and average you wish to see. However, on each screen display, you are limited to viewing two to five students' grades at a time.

2. *Read Assignments:* prints a list of previously entered (option 5) assignments on this diskette.

3. *Enter Names:* lets you enter and add new students to the names list. Note that only 45 names are allowed on one diskette; first name up to nine characters; last name up to ten characters; no middle names.

4. *Enter Grades:* produces a list of students previously entered (option 3) and asks which student's grades you wish to enter. The program accepts any one-, two-, or three-digit number as well as the letters A,B,C,D,E, and F, with or without a plus or minus. When grades are averaged, letter grades are converted to numbers as follows:

A+ =97	A=93	A- =89
B+ =87	B=83	B- =79
C+ =77	C=73	C- =69
D+ =67	D=63	D- =59
E+ =54	E=50	E- =46
F+ =54	F=50	F- =46

If desired, these values can be changed in lines 510-590 and lines 1650-1680.

5. *Enter Assignments:* results in a list of previously entered assignments and allows you to add to the list. The assignment length must be no greater than 28 characters (including blank spaces). You can use this option for messages or notes also. It functions like a notepad with no real bearing on students' grades, averaging, etc.

6. *Print Grades or Assignments:* prints out all or some of the students' names, grades, and averages to a printer. It allows you to print a list of assignments stored on the diskette.

7. *Correction:* permits correcting any student's name or grade.

8. *Initializing a Disk:* makes it possible to avoid retyping and re-entering all the students' names onto a new diskette. This option will automatically transfer the names of students stored on one diskette to a new diskette without transferring grades.

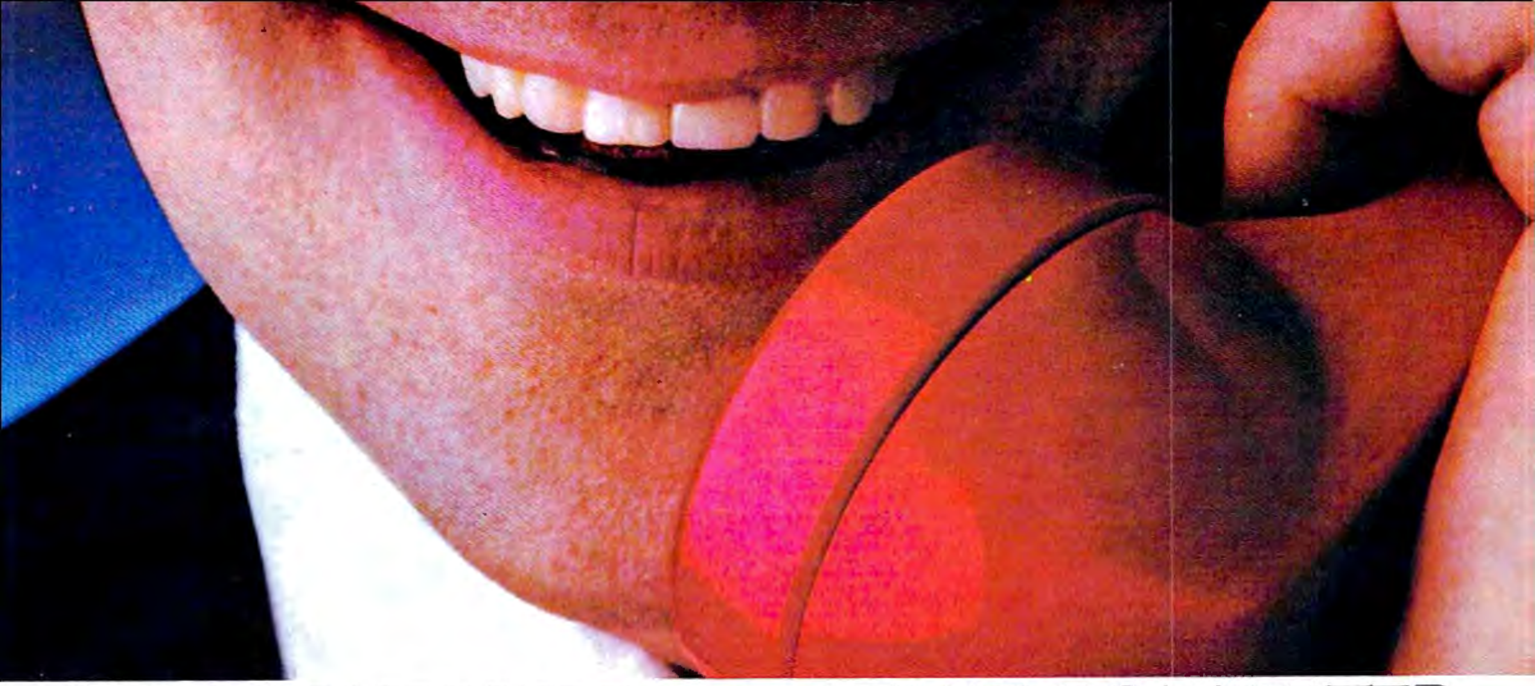
9. *End:* provides a way to exit the program.

It is imperative that you *never end a session by just turning off the computer or disk drive. Always* use option 9.

Gradebook For Atari

```
90 CLOSE #1:CLR
100 DIM NAME$(20),FILE$(13),CL$(1),B
     Z$(1),CLASS$(361),TASK$(30),GRAD
     E$(3),YES$(3)
110 DIM BYTE(48),SECTOR(48),TEMP$(15
     ),BL$(37)
```

```
120 CL$=CHR$(125):BZ$=CHR$(253):MENU
    =200:FILE$="D:STUDENT."
130 BL$=" ":BL$(37)=BL$:BL$(2)=BL$:R
    =0:HW=0:B=5000:C=5500:D=580:E=58
    10
200 LN=200:TRAP B
210 GRAPHICS 0:POKE 201,5:SETCOLOR 4
    ,3,2:SETCOLOR 2,8,9:SETCOLOR 1,8
    ,1
215 POKE 752,1:GOSUB E
220 PRINT :PRINT ,,"{3 SPACES}GRADEB
    OOK":PRINT :PRINT
230 PRINT ,"YOUR OPTIONS ARE"
    :FOR WAIT=7 TO 22:POSITION WAIT,
    6:PRINT CHR$(13):NEXT WAIT
240 PRINT ,"1. Read Grades":PRINT ,"
    2. Read Assignments":PRINT ,"3.
    Enter Names":PRINT ,"4. Enter Gr
    ades"
250 PRINT ,"5. Enter Assignments":PR
    INT ,"6. Print Grades or Assignm
    ents"
260 PRINT ,"7. Make Correction":PRIN
    T ,"8. Initialize a Disk":PRINT
    ,"9. End"
265 POKE 752,0
270 PRINT :PRINT " YOUR CHOICE PLEAS
    E";:INPUT ANS:ANS=INT(ANS):IF AN
    S<1 OR ANS>9 THEN GOTO B
273 IF ANS=9 THEN GRAPHICS 0:END
275 POKE 752,1
280 PRINT CL$:TRAP 40000:FOR AA=1 TO
     10:POSITION 1,10:PRINT "  BE SU
    RE PROPER DATA DISK IS IN DRIVE"
290 FOR W=1 TO 20:NEXT W:POSITION 1,
    10:PRINT "  BE SURE PROPER DATA
    DISK IS IN DRIVE":NEXT AA
295 POSITION 15,14:PRINT "THANK YOU"
297 POSITION 6,16:PRINT "You may ent
    er 'XXX' to any":POSITION 7,17:P
    RINT "prompt to return to menu"
300 PRINT :PRINT :PRINT "  Press any
     key to begin"
305 IF PEEK(764)=255 THEN 305
308 POKE 764,255:POKE 752,0:TRAP 320
    :OPEN #1,4,0,"D:CL":CLOSE #1
310 PRINT :PRINT "PLEASE REMOVE PROG
    RAM MASTER DISK!":GOSUB E:GOTO 3
    00
320 CLOSE #1:GOSUB E:IF ANS=2 OR ANS
    =8 OR ANS=5 THEN 340
330 IF R=0 THEN GOSUB 5100:GOSUB E
340 ON ANS GOTO 410,810,1210,1600,20
    00,2410,2800,3200,350
350 END
410 GOSUB 5310:GOSUB 5400:TRAP B:LN=
    410
420 POSITION 2,19:PRINT "Which stude
    nt's grades do you want?":POSITI
    ON 2,21:PRINT "First student num
    ber";
425 INPUT YES$:GOSUB 5600:FIRST=VAL(
    YES$)
430 POSITION 2,22:PRINT "Last studen
    t number";:INPUT YES$:GOSUB 5600
432 LAST=VAL(YES$):IF LAST<=FIRST TH
    EN GOTO B
433 IF LAST>NUMSTUD-1 THEN GOTO B
435 IF ANS=6 THEN RETURN
440 IF LAST-FIRST>5 THEN GOSUB 5400:
    GOTO 5410
450 GRAPHICS 0:SETCOLOR 2,8,4:SETCOL
    OR 4,8,4
460 FOR AA=FIRST TO LAST
470 TRAP 650:LN=700:NUM=0:SCORE=0:AV
    ERAGE=0:FILE$(11)=STR$(AA)
480 OPEN #1,4,0,FILE$:INPUT #1;NAME$
    :PRINT :PRINT NAME$(11,20);NAME$
    (1,10)
500 INPUT #1;GRADE$:PRINT GRADE$;";
    ";:GOSUB 510:GOTO 500
510 IF GRADE$(1,1)="A" THEN GRADE=93
    :GOTO D
520 IF GRADE$(1,1)="B" THEN GRADE=83
    :GOTO D
530 IF GRADE$(1,1)="C" THEN GRADE=73
    :GOTO D
540 IF GRADE$(1,1)="F" OR GRADE$(1,1
    )="E" THEN GRADE=50:GOTO D
560 IF GRADE$(1,1)="D" THEN GRADE=63
    :GOTO D
570 GRADE=VAL(GRADE$):GOTO 600
580 IF GRADE$(2,2)="+" THEN GRADE=GR
    ADE+4
590 IF GRADE$(2,2)="-" THEN GRADE=GR
    ADE-4
600 SCORE=SCORE+GRADE:NUM=NUM+1:RETU
    RN
650 GOSUB E:CLOSE #1:TRAP 40000:IF P
    EEK(195)=136 THEN GOTO LN
660 CLOSE #1:PRINT :PRINT "Check Dis
    k and/or Drive"
670 PRINT :PRINT "Press any key for
    MENU":GOSUB E
680 IF PEEK(764)=255 THEN 680
690 POKE 764,255:GOTO MENU
700 CLOSE #1:IF NUM=0 THEN PRINT "NO
     GRADES":GOTO 730
705 IF ANS=6 THEN CLOSE #1:GOTO 2555
710 AVERAGE=SCORE/NUM:PRINT "  AVE.=
    ";AVERAGE
730 NEXT AA
740 PRINT :PRINT "Press START for me
    nu":PRINT :PRINT "Press SELECT t
    o see more grades"
750 AA=PEEK(53279):IF AA>6 THEN 750
770 IF AA=6 THEN GOTO MENU
780 IF AA=5 THEN 410
790 GOTO 750
810 PRINT CL$;"{10 SPACES}LIST OF ASS
    IGNMENTS":PRINT
820 TRAP 1100:GOSUB 840:GOTO 670
840 TRAP 1100:CLOSE #1:OPEN #1,4,0,"
    D:ASSIGN"
850 TRAP 650:LN=900
860 INPUT #1;HW:INPUT #1;TASK$
870 IF ANS=6 THEN PRINT #2;HW;".";TA
    SK$:GOTO 860
880 PRINT HW;".";TASK$:GOTO 850
900 RETURN
1100 IF PEEK(195)=170 THEN PRINT :PR
     INT "NO ASSIGNMENTS LISTED":GOT
     O 670
1110 GOTO 660
1210 PRINT CL$:SETCOLOR 4,7,5:SETCOL
     OR 2,13,12:SETCOLOR 1,13,2
1220 LN=1210:TRAP B
1230 POSITION 10,3:PRINT "ADDING NAM
     ES TO CLASS":PRINT :PRINT "Ther
     e are ";NUMSTUD-1;" students in
      this class."
1235 IF NUMSTUD=46 THEN PRINT :PRINT
      "{3 SPACES}CLASS IS FULL NO MO
     RE STUDENTS":GOTO 670
1240 POSITION 1,22:PRINT "TYPE 'XXX'
      for first name for MENU"
```



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

```
1250 POSITION 3,8:PRINT "STUDENT NUM
     BER ";NUMSTUD;":":PRINT :PRINT
     "First name please:";:INPUT TEM
     P$
1252 IF TEMP$="" THEN GOSUB 5800:GOS
     UB 1210
1255 AA=LEN(TEMP$):IF AA>9 THEN GOTO
      B
1260 IF TEMP$="XXX" THEN GOTO MENU
1270 GOSUB 1500:NAME$(11,20)=TEMP$
1280 PRINT :PRINT "Last name please:
     ";:INPUT TEMP$
1285 IF TEMP$="" THEN GOSUB 5800:GOT
     O 1210
1290 AA=LEN(TEMP$):IF AA>10 THEN GOT
     O B
1300 IF TEMP$="XXX" THEN GOTO MENU
1305 TRAP B:LN=1305:POSITION 2,19:PR
     INT BL$:POSITION 2,16:PRINT BL$
1307 POSITION 2,16:PRINT "IS THIS CO
     RRECT";:INPUT YES$:IF YES$(1,1)
     <>"Y" THEN 1210
1310 GOSUB 1500:NAME$(1,10)=TEMP$
1320 CLASS$(NUMSTUD*8-7,NUMSTUD*8)=N
     AME$(1,8)
1325 TRAP 660
1330 FILE$(11)=STR$(NUMSTUD):CLOSE #
     1:OPEN #1,8,0,FILE$:PRINT #1;NA
     ME$:CLOSE #1
1340 GOSUB E:NUMSTUD=NUMSTUD+1:GOTO
     1210
1500 IF AA=10 THEN RETURN
1510 FOR NUM=AA+1 TO 10:TEMP$(LEN(TE
     MP$)+1)=" ":NEXT NUM
1530 RETURN
1600 PRINT CL$:SETCOLOR 2,11,12:SETC
     OLOR 4,5,12:SETCOLOR 1,11,0
1610 POSITION 9,2:PRINT "ENTER STUDE
     NTS' GRADES":PRINT :PRINT "Inst
     ructions"
1630 PRINT :PRINT "Grades may be any
      number from zero{4 SPACES}to 1
     00 or any letter from A to F."
1640 PRINT "Letter grades may includ
     e a plus or{3 SPACES}minus. Let
     ter grades are averaged
     {5 SPACES}as follows:"
1650 PRINT ,"A+=97","A=93","A-=89":P
     RINT ,"B+=87","B=83","B-=79":PR
     INT ,"C+=77","C=73","C-=69"
1680 PRINT ,"D+=67","D=63","D-=59":P
     RINT ,"E+=54","E=50","E-=46":PR
     INT ,"F+=54","F=50","F-=46"
1690 PRINT :PRINT "Type 'xxx' for gr
     ade when you have{4 SPACES}fini
     shed with that student."
1720 PRINT :PRINT "Press any key to
     begin"
1730 IF PEEK(764)=255 THEN 1730
1740 POKE 764,255
1745 TRAP B:LN=1720:PRINT CL$:GOSUB
     5315
1750 POSITION 2,20:PRINT "Enter stud
     ent number";:INPUT YES$:GOSUB 5
     600
1760 NUM=VAL(YES$):IF NUM<1 OR NUM>N
     UMSTUD-1 THEN GOTO B
1780 CLOSE #1:FILE$(11)=STR$(NUM):PO
     SITION 2,20:PRINT BL$:TRAP 650:
     OPEN #1,4,0,FILE$
1790 INPUT #1;NAME$:CLOSE #1
1800 CLOSE #1:OPEN #1,9,0,FILE$
1805 TRAP B:LN=1805:PRINT CL$:GOSUB E
1810 POSITION 4,20:PRINT "Type 'XXX'
      when finished"
1812 POSITION 9,2:PRINT "ENTER STUDE
     NT GRADES":POSITION 1,7:PRINT "
     Grade for ";NAME$(11,20);NAME$(
     1,10);:INPUT GRADE$
1815 IF GRADE$="XXX" THEN CLOSE #1:G
     OTO 1890
1818 IF GRADE$="" THEN GOSUB 5800:GO
     TO 1805
1820 AA=ASC(GRADE$(1,1)):IF AA<58 AN
     D AA>48 THEN GOSUB 1850:GOTO 18
     80
1830 IF AA<71 AND AA>64 THEN GOSUB 1
     850:GOTO 1880
1835 IF GRADE$="" THEN GOSUB 5800:GO
     TO 1805
1840 GOTO B
1850 AA=LEN(GRADE$):IF AA=3 THEN RET
     URN
1860 IF AA=2 THEN GRADE$(3,3)=" ":RE
     TURN
1870 IF AA=1 THEN GRADE$(2,3)="  ":R
     ETURN
1875 IF AA>3 THEN POP :GOSUB 5400:GO
     TO B
1880 TRAP 650:PRINT #1;GRADE$:GOTO 1
     805
1890 GOSUB 5400:TRAP B:LN=1890:POSIT
     ION 2,20:PRINT "Do you wish to
     enter grades for{7 SPACES}anoth
     er student";:INPUT YES$
1910 IF YES$(1,1)="Y" THEN 1745
1920 GOTO MENU
2000 PRINT CL$:SETCOLOR 4,12,8:SETCO
     LOR 1,9,2:SETCOLOR 2,9,8
2010 PRINT ,,"LIST OF ASSIGNMENTS":T
     RAP 2020:PRINT
2015 CLOSE #1:OPEN #1,4,0,"D:ASSIGN"
     :GOSUB 860:GOTO 2040
2020 CLOSE #1:GOSUB E:IF PEEK(195)=1
     70 THEN PRINT ,"{4 SPACES}No as
     signments listed":POP :GOTO 204
     0
2025 CLOSE #1:GOSUB E:IF PEEK(195)=1
     36 THEN RETURN
2030 GOTO 660
2035 FOR AA=20 TO 22:POSITION 2,AA:P
     RINT BL$:NEXT AA
2040 HW=HW+1:TRAP B:LN=2035:POSITION
      2,20:PRINT "Enter assignment #
     ";HW:INPUT TASK$:AA=LEN(TASK$)
2045 IF TASK$="XXX" THEN GOTO MENU
2050 IF AA>28 THEN POSITION 2,21:PRI
     NT BL$;" ":POSITION 2,21:PRINT
     "Too many characters"
2055 IF AA>28 THEN GOSUB C:HW=HW-1:G
     OTO 2035
2057 IF TASK$="" THEN HW=HW-1:POSITI
     ON 2,21:PRINT BL$:POSITION 2,21
     :PRINT "YOU MUST ENTER A LETTER
     "
2058 IF TASK$="" THEN GOSUB C:GOTO 2
     035
2060 CLOSE #1:TRAP 660
2070 IF HW<>1 THEN XIO 36,#1,0,0,"D:
     ASSIGN":OPEN #1,9,0,"D:ASSIGN":
     GOTO 2090
2080 CLOSE #1:IF HW=1 THEN OPEN #1,8
     ,0,"D:ASSIGN"
2090 PRINT #1;HW:PRINT #1;TASK$
```

Let your Atari experience the MPP CONNECTION!

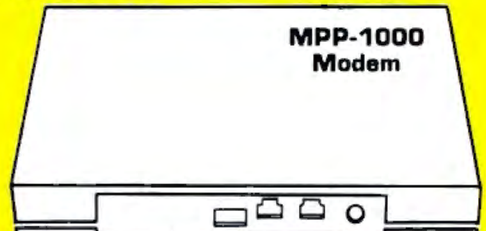
MPP-1100 Parallel Printer Interface

- No Atari 850™ Interface Module needed.
- Compatible with all software (including Visicalc™, Text Wizard™, and Filemanager 800™, etc.).
- 5 foot cable with Centronic plug (compatible with Epson, NEC, IDS, etc.)
- Faster data transfer.
- 8 bit data transfer.
- 2 year warranty.

only $99.95

MICROBITS PERIPHERAL PRODUCTS

MPP-1000 Modem



- No Atari 850™ Interface Module needed
- Smart Terminal Software Included
- 16K Tape/Disk
- Direct Connect To Phone
- Connects to Joystick Port #4

Smart Terminal Features:

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

only $199.00

MICROBITS PERIPHERAL PRODUCTS

434 W. First Street • Albany, Oregon 97321 • **(503) 967-9075**

```
2100 CLOSE #1:XIO 35,#1,0,0,"D:ASSIG
     N":TRAP B:LN=2110
2110 GOSUB E:FOR AA=20 TO 22:POSITIO
     N 2,AA:PRINT BL$:NEXT AA
2120 POSITION 2,20:PRINT ,"Add anoth
     er assignment";:INPUT YES$:IF Y
     ES$(1,1)="Y" THEN 2035
2130 GOTO MENU
2410 PRINT CL$:SETCOLOR 1,9,4:SETCOL
     OR 2,9,14:SETCOLOR 4,13,13:CLOS
     E #1:CLOSE #2
2420 TRAP B:LN=2410
2430 POSITION 5,7:PRINT "Do you want
      to print grades";:INPUT YES$:I
     F YES$(1,1)="Y" THEN 2500
2435 IF YES$="XXX" THEN GOTO MENU
2440 POSITION 5,11:PRINT "Would you
     like to have a":PRINT "
     {3 SPACES}list of assignments p
     rinted";
2445 INPUT YES$:IF YES$(1,1)="Y" THE
     N 2710
2450 GOTO MENU
2500 GOSUB 410:TRAP 2600:OPEN #2,8,0
     ,"P:"
2510 FOR AA=FIRST TO LAST
2520 TRAP 650:LN=705:NUM=0:SCORE=0:A
     VERAGE=0:FILE$(11)=STR$(AA)
2530 CLOSE #1:OPEN #1,4,0,FILE$:INPU
     T #1;NAME$:PRINT #2;NAME$(11,20
     );NAME$(1,10)
2540 INPUT #1;GRADE$:PRINT #2;GRADE$
     ;";  ";:GOSUB 510
2550 GOTO 2540
2555 IF NUM=0 THEN PRINT #2;"NO GRAD
     ES":PRINT #2:GOTO 2570
2560 AVERAGE=SCORE/NUM:PRINT #2;"   A
     VE.=";AVERAGE:PRINT #2
2570 NEXT AA
2580 GOTO 670
2600 CLOSE #2:PRINT :PRINT "TURN ON
     PRINTER PLEASE":GOSUB C:GOTO LN
2710 CLOSE #2:TRAP 2600:OPEN #2,8,0,
     "P:"
2720 PRINT #2,"LIST OF ASSIGNMENTS":
     PRINT #2
2730 GOSUB 840:CLOSE #1:CLOSE #2:TRA
     P 40000:GOTO 670
2800 PRINT CL$:SETCOLOR 2,7,6:SETCOL
     OR 1,7,14:SETCOLOR 4,5,10:GOSUB
      E:TRAP B:LN=2800
2810 POSITION 5,5:PRINT "CORRECT":PR
     INT :PRINT :PRINT ,"1. STUDENT
     NAME":PRINT :PRINT ,"2. STUDENT
      GRADE"
2815 PRINT :PRINT ,"3. RETURN TO MEN
     U"
2820 PRINT :PRINT :PRINT "Press the
     number of your pick";:INPUT YES
     $:GOSUB 5600:W=VAL(YES$):W=INT(
     W)
2830 IF W<1 OR W>3 THEN GOTO B
2840 PRINT CL$:SETCOLOR 2,12,6:SETCO
     LOR 1,12,14:SETCOLOR 4,14,14:ON
      W GOTO 2860,2980,MENU
2860 PRINT CL$:GOSUB 5315:GOSUB 5400
     :POSITION 2,20:PRINT "Type the
     NUMBER of the student"
2865 POSITION 2,21:PRINT "whose name
      needs correcting";:TRAP B:LN=2
     860:AA=100:GOSUB 2870:GOTO 2890
2870 INPUT YES$:GOSUB 5600:W=VAL(YES
     $):W=INT(W):IF W>NUMSTUD-1 THEN
      GOTO B
2880 FILE$(11)=STR$(W):TRAP 650:CLOS
     E #1:OPEN #1,12,0,FILE$:RETURN
2890 NOTE #1,SECTOR,BYTE:INPUT #1;NA
     ME$
2895 PRINT CL$:POSITION 2,6:PRINT "S
     tudent # ";W;" IS ";NAME$(11,20
     );NAME$(1,10):TRAP 2915:AA=0:GO
     SUB E
2900 POSITION 2,10:PRINT "Enter 'XXX
     ' if no correction needed":PRIN
     T :PRINT "First name";:INPUT TE
     MP$
2905 AA=LEN(TEMP$):IF AA>10 OR TEMP$
     ="" THEN 2915
2910 IF TEMP$="XXX" THEN CLOSE #1:GO
     TO MENU
2912 GOTO 2920
2915 PRINT :PRINT "YOU MUST ENTER A
     LETTER-10 MAX.":GOSUB C:GOTO 28
     95
2920 GOSUB 1500:NAME$(11,20)=TEMP$:P
     RINT :PRINT "Last name";:INPUT
     TEMP$
2930 AA=LEN(TEMP$):IF AA>10 OR TEMP$
     ="" THEN 2915
2935 IF TEMP$="XXX" THEN CLOSE #1:GO
     TO MENU
2950 GOSUB 1500:NAME$(1,10)=TEMP$:TR
     AP 660
2960 PRINT :PRINT "{7 SPACES}CORRECT
     ING":POINT #1,SECTOR,BYTE:PRINT
      #1;NAME$:CLOSE #1:F=0:GOTO 280
     0
2980 PRINT CL$:GOSUB 5315:GOSUB 5400
     :POSITION 2,20:PRINT "Type the
     NUMBER of the student whose"
2985 POSITION 2,21:PRINT "grade need
     s correcting";:TRAP B:LN=2980:A
     A=100:NUM=0
2990 GOSUB 2870:TRAP 3050
2995 INPUT #1;NAME$:PRINT CL$:PRINT
     NAME$(11,20);NAME$(1,10);"GRADE
     S"
3000 NOTE #1,SECTOR,BYTE:NUM=NUM+1:S
     ECTOR(NUM)=SECTOR:BYTE(NUM)=BYT
     E:INPUT #1;GRADE$
3010 IF NUM<13 THEN POSITION 1,NUM+1
3015 IF NUM<25 AND NUM>12 THEN POSIT
     ION 11,NUM-11
3020 IF NUM<49 AND NUM>36 THEN POSIT
     ION 31,NUM-35
3025 IF NUM<37 AND NUM>24 THEN POSIT
     ION 21,NUM-23
3030 IF NUM=48 THEN 3050
3040 PRINT NUM;". ";GRADE$:GOTO 3000
3050 IF NUM=1 THEN CLOSE #1:PRINT "N
     O GRADES LISTED":GOSUB C:GOTO 2
     800
3055 GOSUB E:GOSUB 5400:TRAP B:LN=30
     50:POSITION 2,19:PRINT "The NUM
     BER of the grade to change";:IN
     PUT YES$
3060 GOSUB 5600:W=VAL(YES$):IF W>NUM
     -1 THEN GOTO B
3065 W=INT(W):GOSUB 5400:POSITION 2,
     19:PRINT "Enter new grade #";W;
     :INPUT GRADE$:IF GRADE$="" THEN
      GOSUB 5800:GOTO 3050
3070 IF GRADE$="XXX" THEN CLOSE #1:G
     OTO 2800
```

```
3Ø75 AA=ASC(GRADE$(1,1)):IF AA<58 AN
     D AA>48 THEN GOSUB 185Ø:GOTO 3Ø
     9Ø
3Ø8Ø IF AA<71 AND AA>64 THEN GOSUB 1
     85Ø:GOTO 3Ø9Ø
3Ø85 GOTO B
3Ø9Ø TRAP 65Ø:POINT #1,SECTOR(W),BYT
     E(W):PRINT #1;GRADE$:CLOSE #1:G
     OTO 28ØØ
32ØØ PRINT CL$:SETCOLOR 1,15,2:SETCO
     LOR 2,15,12:SETCOLOR 4,8,8
321Ø POSITION 17,5:PRINT "WARNING"
322Ø PRINT :PRINT "This section will
      create new files.":PRINT "  Be
      sure a new formatted disk is"
323Ø PRINT "{4 SPACES}available befo
     re beginning.{16 SPACES}Press 'Y
     '-RETURN"
324Ø PRINT :PRINT "Type 'XXX' if you
      are not ready{7 SPACES}to crea
     te new files on a new disk."
325Ø TRAP B:LN=32ØØ:INPUT YES$:IF YE
     S$(1,1)="Y" THEN 328Ø
326Ø IF YES$="XXX" THEN GOTO MENU
327Ø GOTO B
328Ø PRINT CL$:POSITION 2,1Ø:PRINT "
     Please insert SOURCE disk with
     {8 SPACES}student records"
3285 PRINT :PRINT "PRESS ANY KEY WHE
     N READY"
329Ø IF PEEK(764)=255 THEN 329Ø
3292 POKE 764,255:PRINT CL$:POSITION
      2,1Ø:PRINT "This will take som
     e time. Please be{3 SPACES}pati
     ent. BYE for now"
3295 GOSUB C
3298 POKE 559,Ø:TRAP 333Ø:CLR :DIM C
     LASS$(9ØØ),NAME$(2Ø),YES$(3),FI
     LE$(13):NUMSTUD=1:FILE$="D:STUD
     ENT.":E=581Ø
33ØØ FILE$(11)=STR$(NUMSTUD):CLOSE #
     1:OPEN #1,4,Ø,FILE$:INPUT #1;NA
     ME$:CLASS$(NUMSTUD*2Ø-19,NUMSTU
     D*2Ø)=NAME$
332Ø CLOSE #1:NUMSTUD=NUMSTUD+1:GOTO
      33ØØ
333Ø POKE 559,34:GOSUB E
3332 IF PEEK(195)=17Ø AND NUMSTUD=1
     THEN PRINT :PRINT "THERE ARE NO
      RECORDS ON THIS DISK":CLOSE #1
     :GOTO 335Ø
3335 IF PEEK(195)=17Ø THEN 338Ø
334Ø POKE 559,34:PRINT :PRINT "Check
      Disk and/or Drive":CLOSE #1:GO
     SUB E
335Ø PRINT :PRINT "Press any key for
      menu"
336Ø IF PEEK(764)=255 THEN 336Ø
337Ø POKE 764,255:GOTO 9Ø
338Ø CLOSE #1:GOSUB E:PRINT :PRINT "
     Please insert new formatted dis
     k":PRINT :PRINT "Press 'Y'-RETU
     RN when ready"
339Ø TRAP 35ØØ:INPUT YES$:IF YES$="Y
     " THEN 342Ø
341Ø IF YES$="XXX" THEN 9Ø
3415 GOTO 35ØØ
342Ø TRAP 345Ø:CLOSE #1:OPEN #1,4,Ø,
     "D:STUDENT.*":CLOSE #1
343Ø PRINT CHR$(253);CHR$(125):POSIT
     ION 2,1Ø:PRINT "This disk conta
     ins student grade.{5 SPACES}Ple
     ase use new formatted disk"
344Ø GOTO 338Ø
345Ø CLOSE #1:TRAP 334Ø:GOSUB E
346Ø FOR W=1 TO NUMSTUD-1:NAME$=CLAS
     S$(W*2Ø-19,W*2Ø):FILE$(11)=STR$
     (W)
347Ø CLOSE #1:OPEN #1,8,Ø,FILE$:PRIN
     T #1;NAME$:CLOSE #1:NEXT W
348Ø POKE 559,34:GRAPHICS Ø:POSITION
      2,1Ø:PRINT "THANK YOU FOR WAIT
     ING":FOR W=1 TO 2ØØ:NEXT W:GOTO
      9Ø
35ØØ PRINT :PRINT "{9 SPACES}IMPROPE
     R INPUT":GOTO 338Ø
5ØØØ POKE 752,1:PRINT BZ$
5Ø1Ø PRINT :PRINT ,"INCORRECT INPUT,
      try again":GOSUB C:POKE 752,Ø:
     GOTO LN
51ØØ PRINT CL$:POSITION 17,1Ø:PRINT
     "WORKING":PRINT :PRINT ,,"PLEAS
     E BE PATIENT":NUMSTUD=1:TRAP 52
     ØØ:R=1
511Ø FILE$(11)=STR$(NUMSTUD)
512Ø CLOSE #1:OPEN #1,4,Ø,FILE$
513Ø INPUT #1;NAME$:CLASS$(NUMSTUD*8
     -7,NUMSTUD*8)=NAME$(1,8)
514Ø NUMSTUD=NUMSTUD+1:CLOSE #1:GOTO
      511Ø
52ØØ CLOSE #1:GOSUB E:IF NUMSTUD=1 T
     HEN RETURN
521Ø TRAP 4ØØØØ:IF PEEK(195)=17Ø THE
     N RETURN
522Ø GOTO 66Ø
531Ø PRINT CL$:SETCOLOR 4,6,1Ø:SETCO
     LOR 1,8,12:SETCOLOR 2,8,3
5315 PRINT ,,"NAMES OF STUDENTS"
5317 IF NUMSTUD=1 THEN POP :PRINT :P
     RINT "There are no students on
     this disk":GOTO 67Ø
532Ø NN=2
533Ø FOR AA=1 TO NUMSTUD-1
534Ø IF AA<16 THEN POSITION 2,NN:PRI
     NT AA;".";CLASS$(AA*8-7,AA*8):G
     OTO 538Ø
535Ø IF AA>15 AND AA<31 THEN POSITIO
     N 15,NN:PRINT AA;".";CLASS$(AA*
     8-7,AA*8):GOTO 538Ø
536Ø IF AA>3Ø THEN POSITION 28,NN:PR
     INT AA;".";CLASS$(AA*8-7,AA*8)
538Ø NN=NN+1:IF NN=17 THEN NN=2
539Ø NEXT AA:RETURN
54ØØ POSITION 2,19:PRINT BL$:POSITIO
     N 2,21:PRINT BL$:POSITION 2,22:
     PRINT BL$:RETURN
541Ø PRINT CL$:POKE 752,1
5415 PRINT BZ$:POSITION 8,9:PRINT "O
     NLY FIVE STUDENTS GRADES":POSIT
     ION 1Ø,12:PRINT "CAN BE LISTED
     AT ONCE"
542Ø GOSUB C:POKE 752,Ø:GOTO LN
55ØØ FOR WAIT=1 TO 15Ø:NEXT WAIT:RET
     URN
56ØØ IF YES$="XXX" THEN POP :GOTO ME
     NU
561Ø AA=ASC(YES$(1,1)):IF AA<49 OR A
     A>57 THEN POP :GOTO B
562Ø RETURN
58ØØ PRINT "You must enter at least
     one character":GOSUB C:RETURN
581Ø W=PEEK(16):IF W>127 THEN W=W-12
     8:POKE 16,W:POKE 53774,W
582Ø RETURN
```
©

DIAMOND DROP

Matt Giwer

Catch the falling diamonds – if you can. This fast-action game is easy to play and uses very little memory. Originally written for the Atari (with paddle), other versions are included for the TI-99/4A (with Extended BASIC) and the VIC and 64.

"Diamond Drop" is a game that requires good judgment and quick reflexes. It's fast, easy to play, and will fit into even the smallest Atari. The game uses both player/missile graphics and the Atari's fast string handling. The game plays quickly in BASIC with no machine language routines and uses less than 7K of RAM.

Four rows of diamonds will appear at the top of the screen. At the bottom, you'll see five catching trays, which are controlled by your paddle. As the diamonds drop, position your trays to catch them. Each diamond is worth ten points. If you miss, you lose one tray. If you complete one row, you get a 100-point bonus. Finish all four rows and you get a 250-point bonus. When you have lost all of your trays, the high score is recorded on the left of the screen, and you start again.

You won't be able to anticipate a dropping pattern because the subroutine at line 20000 generates a random sequence of two-digit numbers that will not repeat. Each number appears only once within the string.

The routine starts off with AA$ (line 20012), which contains the numbers 05 through 34. (These are the column numbers for the POSITION instructions.) The G LOOP then picks two of these pairs of numbers randomly and exchanges their positions within AA$. Thirty exchanges within this string of thirty pairs of numbers work well for this game.

Understanding The Program

Line 2 sends us immediately to line 30000 where the subroutine turns on the P/M Graphics and draws the trays at 30282. For a real challenge, change the POKE in line 30210 to 0.

Line 80 DIMensions the strings for the order of dropping the diamonds, four small strings for shuffling, and a string for scoring.

Line 100 names the frequently called subroutines for ease of program development and modification.

The subroutine at line 1000 initializes the variables and screen with a new set of four rows of diamonds. (Diamonds are CTRL ".").

Lines 2010 through 2190 comprise an infinite (because of the STEP size) control loop for the main program execution. Within this loop is the nested J LOOP (lines 2040 and 2090). This loop moves the diamond from the top of the screen to the bottom in line 2051. The second POSITION and ? put a blank in the previous position of the diamond as it moves down. Line 2080 contains the collision register for Player 0 and directs execution to the subroutine for catching (line 5400). Upon return from the subroutine, POKE HITCLR 53278 clears the collision registers.

Subroutine CATCH sets FLG=1. If the flag has not been set, line 2100 slides the diamond off to the right of the screen. The program is then directed to subroutine MISS.

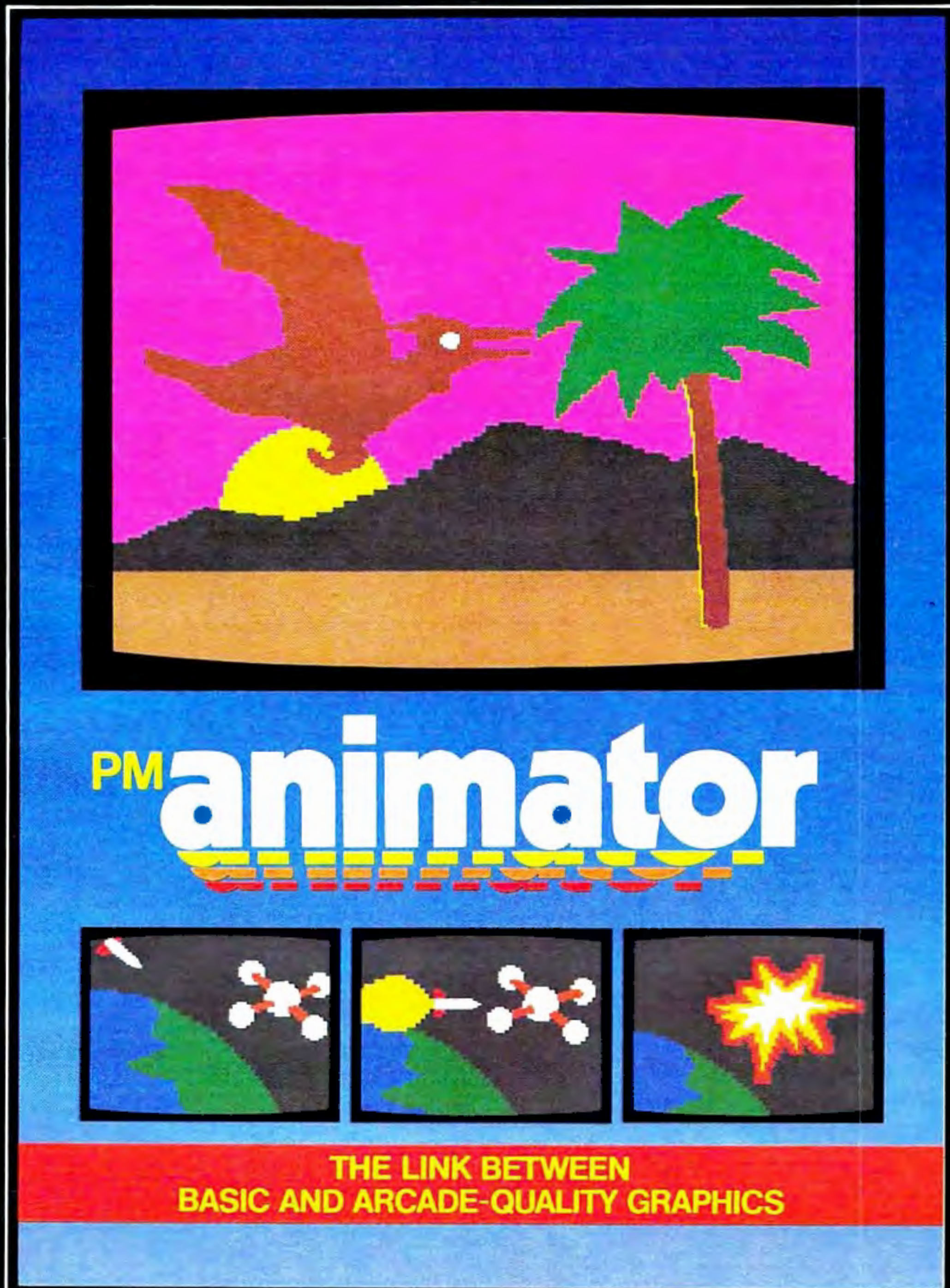
The 5100 lines decrement the ROW and give a bonus and GOSUB SCORE. If all four rows are gone, the program then moves to NLEVEL.

The 5300 lines give a bonus, increase the score, then initialize the variables and reset the screen with GOSUB 1000.

The 5400 lines simply remove the diamond, give a buzz, and increment the score.

The 5500 lines increment the score by 10.

Get pm ANIMATOR
And put your fantasies on the screen.



A product from Don't Ask. Programmed by Roger Bush.

Produce dazzling Player-Missile animation on your Atari, just by writing a BASIC program. Make your creative ideas real.

All you need is **pm ANIMATOR**, the Player-Missile control kit from **DON'T ASK**.

All the colors, motions, and shapes you can make with Player-Missile Graphics are now available from BASIC. Just use **pm ANIMATOR**'s magical editors and subroutines. Draw pictures and animate them with ease. Edit your animation sequences. Save them on a disk to reload later. And build your moving figures right into your own BASIC programs. Flying dragons, spinning rockets, leaping athletes – you create them all.

You don't need to know machine language. And you won't have to do bit-mapping.

Get **pm ANIMATOR**, and write that arcade game you've been dreaming about. The power is yours.

Have your dealer demonstrate pm ANIMATOR.

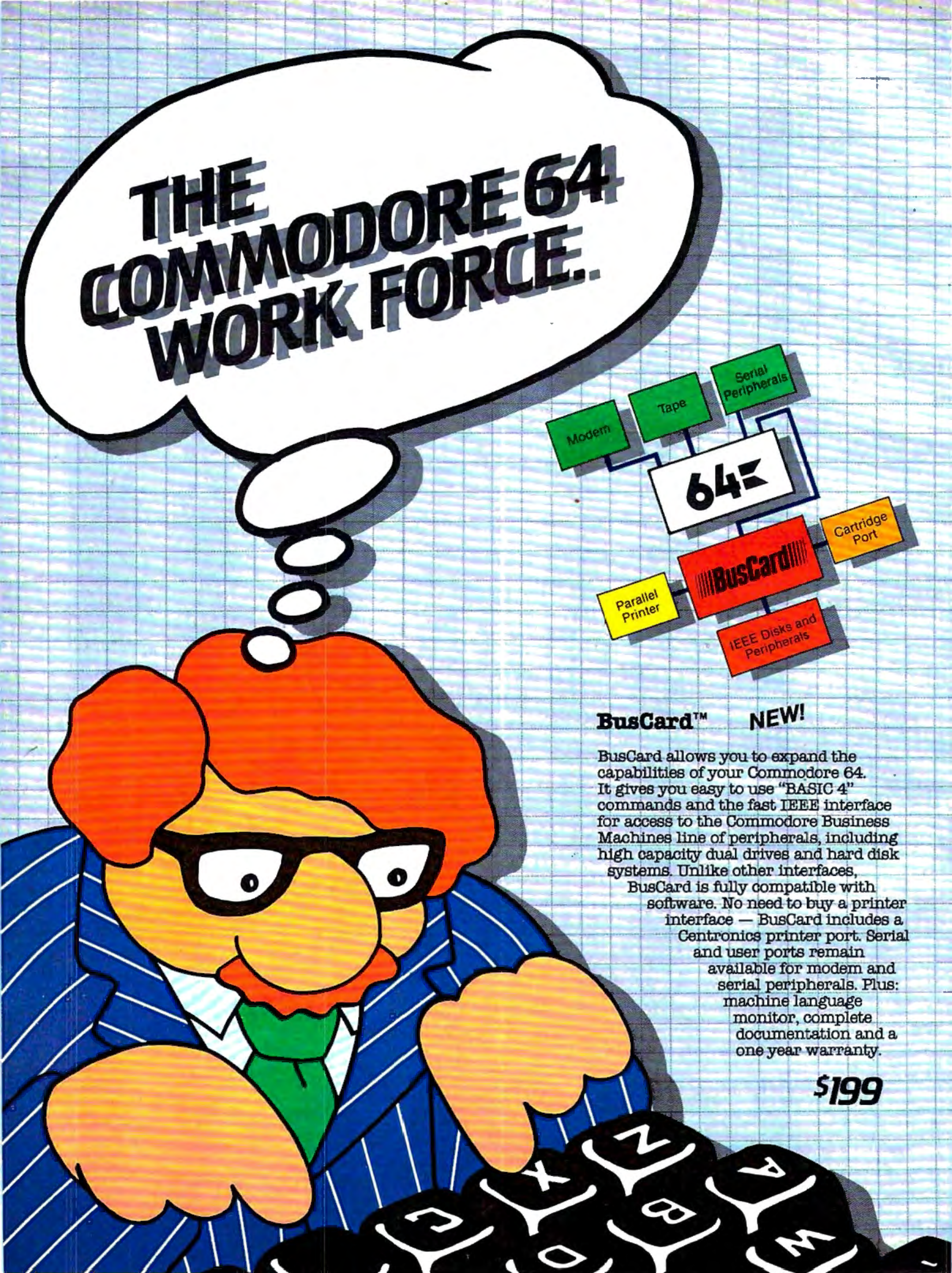
Comes with numerous demo programs and tutorial on Player-Missile Graphics.
Requires disk drive, 32K RAM Atari computer.

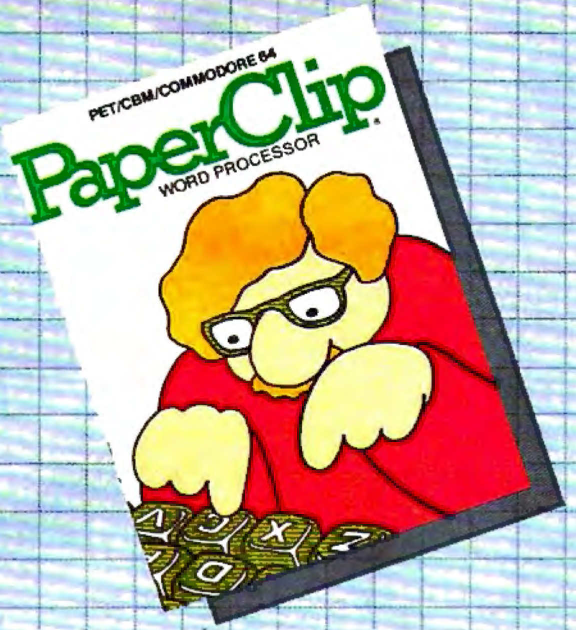
NOW AT SPECIAL INTRODUCTORY PRICE: $34.95.
Price goes up September 15, 1983.

Or order by mail from **DON'T ASK**. Add $2.00 shipping/handling to your check or money order. California residents add 6% sales tax (6.5% if you reside in L.A. County).

DON'T ASK COMPUTER SOFTWARE
2265 Westwood Blvd., Ste. B-150, Los Angeles, CA 90064
Phone: (213) 477-4514

Dealer inquiries welcome.


THE COMMODORE 64 WORK FORCE.
Modem
Tape
Serial Peripherals
64
Cartridge Port
BusCard
Parallel Printer
IEEE Disks and Peripherals
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
Z
X
C
B
A
D
W



PaperClip™
Professional Word Processor
For Commodore 64 and CBM/SuperPet

"Absolutely the most versatile word processor I have seen." *Midnight Software Gazette* March/April, 1983

"...a very powerful word processor, with so many features that most people only need a fraction of them." *COMPUTE!* April, 1983

So easy to use that even novices can get professional results.

PaperClip has every standard word processor function, plus many exclusive features, including horizontal scrolling for charts and wide reports, up to 250 columns, column moves, alphanumeric sorts and arithmetic. Works with 80 column cards and every popular printer, with instant printer set-up. Professional handling of form letters, mail list merge, and large documents.

Complete, professional and easy to use. No where else will you find PaperClip's capabilities at this breakthrough price.

$125

Delphi's Oracle NEW!
Professional Data Base Management
For Commodore 64 and PET/CBM

Delphi's Oracle is a powerful information handling program that allows you to enter, retrieve and update data with incredible speed and flexibility.

• Large record size (over 8000 characters) • Up to 99 fields and 9 display pages per record • Includes report writer and mail label printing • Design your own forms on the screen or on printed reports

Delphi's Oracle brings power and versatility usually found only in mainframe or minicomputer systems, with provision for safeguarding accurate data entry, and excellent flexibility in searching for records.

$150

BATTERIES INCLUDED

FOR MORE INFORMATION AND YOUR NEAREST DEALER:

City Software Distributors
US TOLL FREE: 800-558-1008
735 W. Wisconsin Ave.
Milwaukee, WI 53233

Software International
560 N. Mountain Ave.
Upland, CA 91786
714-981-7640

Batteries Included
186 Queen Street West
Toronto, Ont. M5V 1Z1
416-596-1405

Software Distributors
10023 W. Jefferson Blvd.
Culver City, CA 90230
Cal: 800-252-4025
US: 800-421-0814

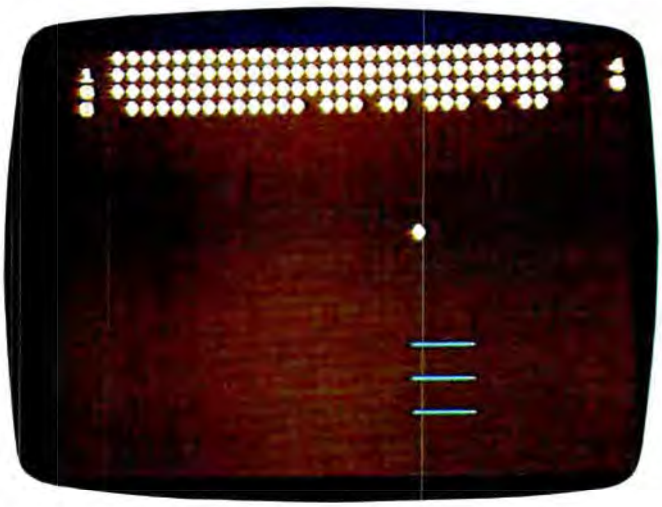
CBM Systems
7668 Telegraph Rd.
City of Commerce, CA 90040
213-904-0111

Kapri International
7419 Clybourn Ave.
Sun Valley, CA 91352
213-765-2774

Note that the bonuses have been 90 and 240, since each time this subroutine is called, 10 is added. Line 5520 prints the score vertically on the right side of the screen.

The 5600 lines remove one of the trays, provide a sound effect, and test for all trays being gone.

The 5700 lines determine if the new score is a high score, print it on the left side of the screen, erase the old score, and ask for another game. Pressing the trigger sets up a new game.



"Diamond Drop" on the Atari.

Program 1: Diamond Drop – Atari 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."**

```
2 GOSUB 30000
80 DIM AA$(60),B1$(2),C1$(2),B2$(2),
   C2$(2),SCR$(8)
99 MM=0
100 NROW=5100:NLEVEL=5300:CATCH=5400
    :SCORE=5500:MISS=5600:TEND=5700
900 GOSUB 1000:GOTO 2000
1000 REM SETUP
1009 REM 5 TO 34
1010 POSITION 5,0:? "```````````````
     ```````````````"
1020 POSITION 5,1:? "```````````````
     ```````````````"
1030 POSITION 5,2:? "```````````````
     ```````````````"
1040 POSITION 5,3:? "```````````````
     ```````````````"
1050 GOSUB 20000:ROW=4
1900 RETURN
2000 REM CONTROL LOOP
2010 FOR ZYX=1 TO 2 STEP 0
2020 A=VAL(AA$(I,I+1))
2022 POKE PLX,PADDLE(0)
2030 I=I+2
2040 FOR J=ROW TO 22
2042 POKE PLX,PADDLE(0)
2051 POSITION A,J:? "`";:POSITION A,
     J-1:? " ";
2080 IF PEEK(53252)<>0 THEN GOSUB CA
     TCH:POKE 53278,0
2090 NEXT J
2100 IF FLG=0 THEN FOR FF=A TO 35:PO
     SITION FF,22:? "`";:POSITION FF
     -1,22:? " ";:POKE PLX,PADDLE(0)
     :NEXT FF
2102 POSITION 35,22:? " ";
2110 IF FLG=0 THEN GOSUB MISS
2180 IF I>60 THEN GOSUB NROW:GOSUB 2
     0000
2188 FLG=0
2190 NEXT ZYX
5100 REM NROW NEW ROW
5110 ROW=ROW-1:SC=SC+90:GOSUB SCORE
5120 IF ROW=0 THEN GOSUB NLEVEL
5190 RETURN
5300 REM NLEVEL  NEW LEVEL
5310 SC=SC+240:GOSUB SCORE
5350 GOSUB 1000
5390 RETURN
5400 REM CATCH
5410 POSITION A,J:? " ";:J=24:FLG=1
5420 SOUND 0,A,2,14
5430 Q=1^1
5440 SOUND 0,0,0,0
5450 GOSUB SCORE
5490 RETURN
5500 REM SCORE
5510 SC=SC+10:SCR$=STR$(SC)
5520 FOR FF=1 TO LEN(SCR$):SOUND 0,2
     00-20*FF,10,14:POSITION 38,FF:?
      SCR$(FF,FF):SOUND 0,0,0,0:NEXT
      FF
5590 RETURN
5600 REM MISS THE BALL
5610 POKE MYPMBASE+135+MM,0
5615 FOR JKJ=14 TO 0 STEP -2:FOR JKK
     =200 TO 50 STEP -50:SOUND 0,JKK
     ,10,JKJ:NEXT JKK:NEXT JKJ
5620 MM=MM+15:IF MM+135=210 THEN GOS
     UB TEND
5690 RETURN
5700 REM TEND THE END
5710 IF SC>HSC THEN HSC=SC
5720 SCR$=STR$(HSC)
5730 FOR FF=1 TO LEN(SCR$):SOUND 0,2
     00-FF*10,10,14:POSITION 3,FF:?
     SCR$(FF,FF):Q=1^1:SOUND 0,0,0,0
     :NEXT FF
5740 SC=0:FOR FF=1 TO 10:POSITION 38
     ,FF:? " ";:NEXT FF
5750 POSITION 2,10:? "ANOTHER GAME?"
     :Q=1^1
5752 IF PTRIG(0) THEN 5752
5754 POSITION 2,10:? "{13 SPACES}";:P
     OKE 77,3
5760 FOR III=MYPMBASE+135 TO MYPMBAS
     E+200 STEP 15:POKE III,255:NEXT
      III
5770 GOSUB 1000:MM=0
5790 RETURN
20000 REM RND GENERATOR
20002 I=1
20012 AA$="0506070809101112131415161
      7181920212223242526272829303l3
      23334"
```


WHAT HAS IMMEASURABLE FIREPOWER... ATTACKING RIGILLIANS... ALTERED PERSPECTIVE SCROLLING... AND NO MERCY?
SYNAPSE SOFTWARE
DIMENSION X
DIMENSION X,™ by Steve Hales, the latest blockbuster from SYNAPSE SOFTWARE gives you a screenful of 3-D graphics, a desert sector map that's different every time, AND... no margin for error.
DIMENSION X™ at software dealers everywhere, or AVAILABLE DIRECT FROM SYNAPSE ELITE, ONLY $34.95 plus $2 shipping and handling—SEND CHECK, MONEY ORDER OR VISA/MASTERCARD NUMBER TO SYNAPSE ELITE OR CALL (415) 527-7712.
For the Atari® 400/800/1200
synapse
SOFTWARE
5221 Central Avenue, Richmond, CA 94804 · (415) 527-7712

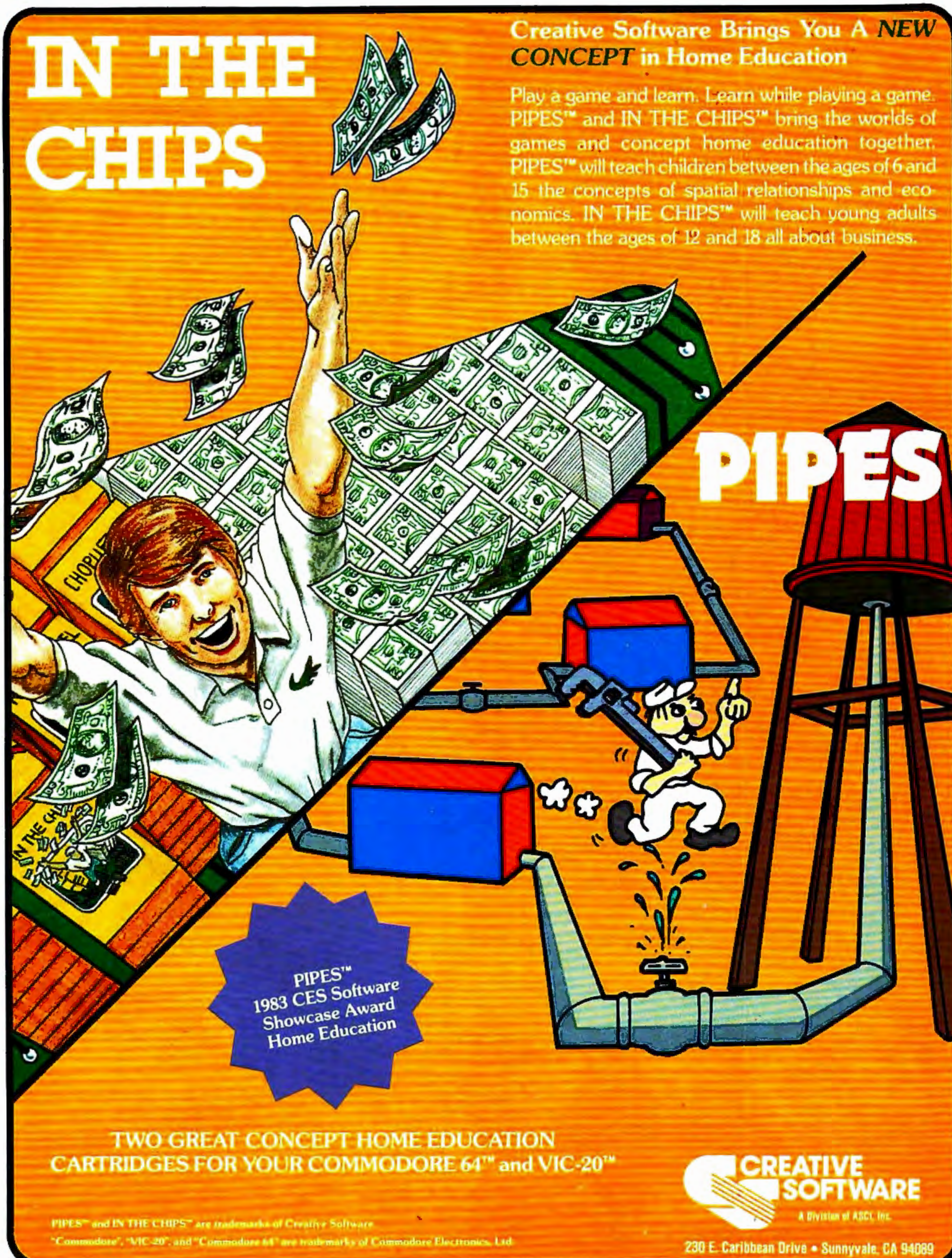
```
20040 FOR G=0 TO 30
20050 M=INT(30*RND(0)+2)*2-3
20051 N=INT(30*RND(0)+2)*2-3
20060 B1$=AA$(M,M+1):C1$=AA$(N,N+1)
20070 B2$=C1$:C2$=B1$
20080 AA$(M,M+1)=B2$:AA$(N,N+1)=C2$
20081 POKE PLX,PADDLE(0)
20090 NEXT G
20092 A=VAL(AA$(1,2))
20099 RETURN
22000 FOR I=MYPMBASE TO MYPMBASE+255
      :POKE I,0:NEXT I:STOP
30000 REM SETUP
30010 GRAPHICS 0:POKE 752,1
30200 REM PM SETUP
30204 POKE 53277,3:REM GRACTL PLAY&M
      ISS
30206 POKE 559,62:REM DMACTL,1LINE,P
      LAY,MIS,NORM FIELD
30208 POKE 54279,(PEEK(106)-12):REM
      12PAGE RESERVE
30210 POKE 53256,1:REM PLAY SIZES
30212 POKE 623,8:REM PRIORITY PF OVE
      R PL
30214 MYPMBASE=256*(PEEK(106)-12):RE
      M NEW PM BASE
30230 POKE 704,150
30232 POKE 710,16:POKE 709,29
30276 PLX=53248
30282 FOR I=MYPMBASE+135 TO MYPMBASE
      +200 STEP 15:POKE I,255:NEXT I
30283 POKE PLX,PADDLE(0)
30285 RETURN
```

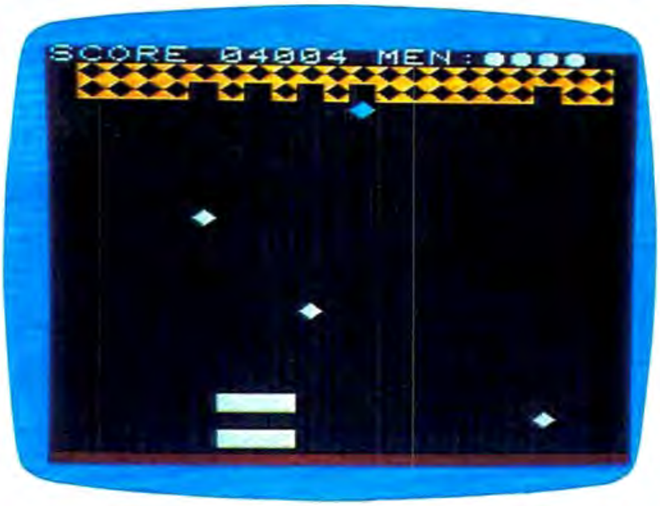
## Program 2:

### Diamond Drop – VIC Version, Part I

by Eric Brandon, Programming Assistant

```
3 POKE55,177:POKE56,27:CLR
4 POKE36879,93
5 TI$="000000"
9 PRINT"{CLR}{BLU}{4 DOWN}SETTING UP ...
  {3 DOWN}"
10 I=7089
15 PRINT"WAIT"STR$(25-VAL(TI$))" SECONDS
   {UP}"
20 READA:IFA=256THEN40
30 POKEI,A:I=I+1:GOTO15
40 PRINT"{CLR}{5 DOWN}{RED}{RVS}NOW LOAD
   ING GAME...{OFF}{BLU}"
50 REM FOR DISK USERS, TAKE THE WORD "RE
   M" OUT OF LINE 60
55 PRINT"{2 DOWN}"
60 REM PRINT"{UP}LOAD"CHR$(34)"DIAMONDS2
   .VIC"CHR$(34)",8"
70 PRINT"{4 UP}"
80 POKE 198,1:POKE631,131:NEW
7089 DATA 120,169,27,141,21,3,169
7097 DATA 200,141,20,3,88,169,9
7105 DATA 141,253,29,169,0,141,250
7113 DATA 29,96,173,255,29,141,252
7121 DATA 29,172,253,29,169,32,153
7129 DATA 205,31,200,169,160,174,251
7137 DATA 29,153,205,31,200,202,208
7145 DATA 249,169,32,153,205,31,206
7153 DATA 252,29,208,3,76,174,28
7161 DATA 172,253,29,169,32,153,161
7169 DATA 31,200,169,160,174,251,29
7177 DATA 153,161,31,200,202,208,249
7185 DATA 169,32,153,161,31,200,206
7193 DATA 252,29,208,3,76,174,28
7201 DATA 172,253,29,169,32,153,117
7209 DATA 31,200,169,160,174,251,29
7217 DATA 153,117,31,200,202,208,249
7225 DATA 169,32,153,117,31,200,206
7233 DATA 252,29,240,123,172,253,29
7241 DATA 169,32,153,73,31,200,169
7249 DATA 160,174,251,29,153,73,31
7257 DATA 200,202,208,249,169,32,153
7265 DATA 73,31,200,206,252,29,240
7273 DATA 91,172,253,29,169,32,153
7281 DATA 29,31,200,169,160,174,251
7289 DATA 29,153,29,31,200,202,208
7297 DATA 249,169,32,153,29,31,200
7305 DATA 206,252,29,240,59,172,253
7313 DATA 29,169,32,153,241,30,200
7321 DATA 169,160,174,251,29,153,241
7329 DATA 30,200,202,208,249,169,32
7337 DATA 153,241,30,200,206,252,29
7345 DATA 240;27,172,253,29,169,32
7353 DATA 153,197,30,200,169,160,174
7361 DATA 251,29,153,197,30,200,202
7369 DATA 208,249,169,32,153,197,30
7377 DATA 200,165,197,201,21,208,13
7385 DATA 173,253,29,201,1,240,24
7393 DATA 206,253,29,76,211,28,201
7401 DATA 22,208,14,173,253,29,24
7409 DATA 109,251,29,201,21,240,3
7417 DATA 238,253,29,238,250,29,173
7425 DATA 250,29,205,249,29,240,3
7433 DATA 76,191,234,169,0,141,250
7441 DATA 29,169,206,133,251,169,31
7449 DATA 133,252,160,0,185,206,31
7457 DATA 41,127,201,32,208,74,200
7465 DATA 192,21,208,242,160,0,177
7473 DATA 251,201,81,240,37,201,207
7481 DATA 240,33,201,90,240,29,200
7489 DATA 192,22,208,237,56,165,251
7497 DATA 233,22,133,251,176,2,198
7505 DATA 252,166,251,208,220,166;252
7513 DATA 224,30,208,214,76,191,234
7521 DATA 170,152,24,105,22,168,138
7529 DATA 145,251,152,56,233,22,168
7537 DATA 169,32,145,251,32,154,29
7545 DATA 76,14,29,169,32,153,206
7553 DATA 31,169,150,141,11,144,169
7561 DATA 175,141,12,144,169,15,141
7569 DATA 14,144,169,200,133,251,160
7577 DATA 128,162,8,142,15,144,232
7585 DATA 224,15,208,248,200,208,243
7593 DATA 230,251,208,239,169,14,141
7601 DATA 15,144,169,0,141,14,144
7609 DATA 141,12,144,141,11,144,160
7617 DATA 21,185,0,30,201,81,240
7625 DATA 11,136,208,246,169,1,141
7633 DATA 254,29,76,191,234,169,32
7641 DATA 153,0,30,76,191,234,152
7649 DATA 72,160,10,185,0,30,201
7657 DATA 57,208,9,169,48,153,0
7665 DATA 30,136,76,158,29,185,0
7673 DATA 30,24,105,1,153,0,30
7681 DATA 104,168,96,174,255,29,202
7689 DATA 142,255,29,232,169,206,133
7697 DATA 251,169,31,133,252,56,165
7705 DATA 251,233,44,133,251,176,2
7713 DATA 198,252,202,208,242,160,0
7721 DATA 177,251,201,160,240,4,200
7729 DATA 76,218,29,174,251,29,169
7737 DATA 32,145,251,200,202,208,250
7745 DATA 96,96,256
```


IN THE CHIPS
Creative Software Brings You A NEW CONCEPT in Home Education
Play a game and learn. Learn while playing a game. PIPES™ and IN THE CHIPS™ bring the worlds of games and concept home education together. PIPES™ will teach children between the ages of 6 and 15 the concepts of spatial relationships and economics. IN THE CHIPS™ will teach young adults between the ages of 12 and 18 all about business.
PIPES
PIPES™
1983 CES Software
Showcase Award
Home Education
TWO GREAT CONCEPT HOME EDUCATION CARTRIDGES FOR YOUR COMMODORE 64™ and VIC-20™
CREATIVE SOFTWARE
A Division of ASCI, Inc.
230 E. Caribbean Drive • Sunnyvale, CA 94089
PIPES™ and IN THE CHIPS™ are trademarks of Creative Software.
"Commodore", "VIC-20", and "Commodore 64" are trademarks of Commodore Electronics, Ltd.




*VIC version of "Diamond Drop."*

## Program 3:
### Diamond Drop – VIC Version, Part II
by Eric Brandon, Programming Assistant

```
5 POKE 36879,14
10 PRINT"{CLR}{WHT}"TAB(5)"DIAMOND DROP"
20 PRINT"{2 DOWN}{YEL}{2 SPACES}CATCH TH
   E DIAMONDS{2 SPACES}BEFORE THEY ";
30 PRINT"TOUCH THE GROUND. YOU HAVE FIVE
40 PRINT"CHANCES.
45 PRINT"{2 DOWN}{WHT}{4 SPACES}L - MOVE
    LEFT
46 PRINT"{DOWN}{4 SPACES}; - MOVE RIGHT
   {YEL}"
50 PRINT"{3 DOWN}{6} {RVS}HIT ANY KEY
   TO BEGIN"
60 GETA$:IFA$=""THEN60
65 GOSUB 1000
70 PRINT"{CLR}{WHT}SCORE 00000 MEN:QQQQ"
71 SPEED = 7673
72 PADDLES=7679
73 FLAG=7678: POKE FLAG,0
74 WIDTH = 7675
75 POKE PADDLES,6 : POKE WIDTH,W : POKE
   SPEED,10-S
78 ROW(6)=81:ROW(5)=81:ROW(4)=207:ROW(3)
   =207:ROW(2)=90:ROW(1)=90
80 PRINT" {YEL}{RVS}";:FORI=1TO20:PRINT"
   Z";:NEXT:PRINT"{OFF} ";
85 PRINT" {YEL}{RVS}";:FORI=1TO20:PRINT"
   Z";:NEXT:PRINT"{OFF} ";
90 PRINT" {CYN}{RVS}";:FORI=1TO20:PRINT"
   P";:NEXT:PRINT"{OFF} ";
95 PRINT" {CYN}{RVS}";:FORI=1TO20:PRINT"
   P";:NEXT:PRINT"{OFF} ";
100 PRINT" {OFF}{7}";:FORI=1TO20:PRINT
    "W";:NEXT:PRINT" ";
102 PRINT" {OFF}{7}";:FORI=1TO20:PRINT
    "W";:NEXT:PRINT" ";
105 PRINT"{WHT}";
109 REM 22 SPACES IN NEXT LINE
110 FORI=1TO14:PRINT"{22 SPACES}";:NEXT
120 PRINT"{HOME}";
130 FOR I=8164 TO 8185: POKE I,248:POKE
    I+30720,2:NEXT
140 IF PEEK(789)<>27THENSYS 7089
150 FOR ROW = 6 TO 1STEP-1:FOR CHAR=1 TO
     20
155 FOR K=1 TO 600-CHAR*10+(6-ROW)*20-50
    *(9-PEEK(SPEED)):NEXT
157 IF PEEK(FLAG) THEN 2000
160 P=RND(1)*20+1
170 IF PEEK(7680+ROW*22+P)=32THEN160
180 POKE 7680+ROW*22+P,ROW(ROW)
190 NEXTCHAR
191 POKE36878,15
```

## VIC-20/64 Version Notes

Eric Brandon, Programming Assistant

To insure fast action, both the VIC and 64 versions of "Diamond Drop" are written predominantly in machine language. BASIC is used only to print instructions, set up the display, select the skill level, and initiate the "drop."

The game display starts with six rows of objects at the top of the screen and a stack of six catching trays at the bottom. As the objects begin to drop, you must use the L and ; keys to maneuver the trays and catch the objects. To make play more challenging, one tray disappears whenever the last ball drops from a row. Thus, you have only one tray with which to catch objects from the last row. When all the objects have dropped, you start again with six rows of objects and six trays. Play continues until a total of five objects hit the ground.

The VIC version is in two parts (Programs 2 and 3) so that it can run on the unexpanded VIC. Cassette users should type in Program 2 and SAVE it to tape, then type in Program 3 and SAVE it on the same tape immediately following Program 2. Disk users should type in Program 2, omitting the word REM in line 60, and SAVE it to disk. Program 3 should then be typed in and SAVEd to the same disk with the filename "DIAMONDS2.VIC". If the tape or disk copies are prepared in this manner, then Program 2 will cause Program 3 to LOAD and RUN automatically.

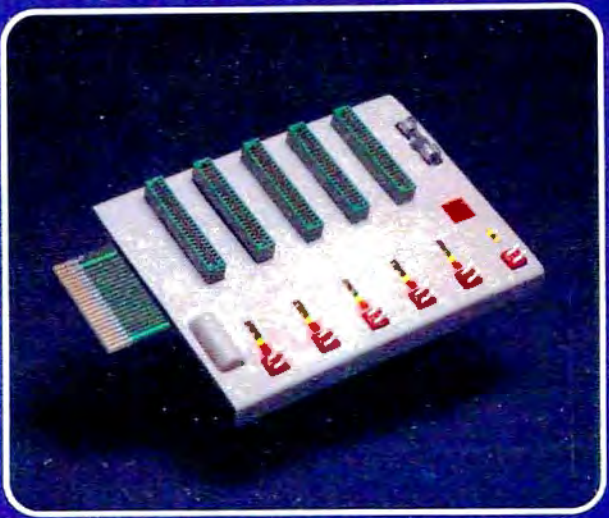
Since the DATA statements of Program 2 (VIC version) and Program 4 (64 version) comprise the machine language program for the game, it is essential that they be typed correctly. Be sure to SAVE a copy of the program before you attempt to RUN it, since an error in typing may cause your computer to "lock up," forcing you to turn the power off to recover. If Diamond Drop fails to RUN properly, the problem will most likely be a mistyped number somewhere in the DATA statements, so check carefully.

# New From Cardco

# Five Slot Expansion Interface for the C-64

The CARDBOARD/5 (CB/5) is an enclosed five slot, fully switch selectable, expansion interface for the Commodore 64™. This quality product allows the user to switch select any cartridge slot or combination of cartridge slots. Twenty-two color coded light emitting diodes give status indication. Each slot has four LEDs and two toggle switches for indication and control. Two master toggle switches allow the user to manually override any situation.

All Cardco products are individually tested to insure quality and reliability.



**Some of the features of the CARDBOARD/5 are:**

- high quality glass/epoxy circuit board
- gold plated contacts
- logic lines are switched by solid state IC switches
- full LED status indication
- convenient toggle switches
- full support under the board to prevent flexing
- full plastic enclosure to insure safety
- fused to protect your computer
- convenient reset button
- CARDCO, Inc.'s exclusive Lifetime Guarantee


See a complete line of American made Cardco Products at a computer store near you, today.

313 Mathewson • Wichita, Kansas 67214 • (316) 267-6525

Commodore 64™ is a registered trademark of Commodore Business Systems, Inc.

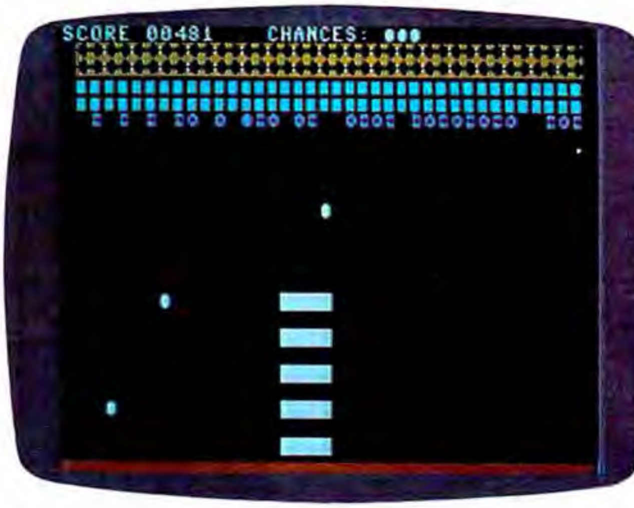
cardco, inc.

```
192 POKE36876,249
193 FORH=75TO15STEP-1.5:POKE 36878,H/5:N
    EXTH
194 POKE36878, 0
197 IF ROW >1 THENSYS 7610
200 NEXTROW
201 FOR K=1 TO 300:NEXTK
205 IF PEEK(SPEED)>2 THEN POKE SPEED,PEE
    K(SPEED)-1
206 IF PEEK(SPEED)=2 AND PEEK(WIDTH)>1TH
    ENPOKEWIDTH,PEEK(WIDTH)-1
207 POKE PADDLE,6
210 PRINT"{HOME}{DOWN}";
220 GOTO 80
999 END
1000 PRINT"{CLR}{7 SPACES}DIFFICULTY
     {4 SPACES}{5 DOWN}"
1010 INPUT"{WHT}SPEED (1-9){YEL}
     {3 RIGHT}5{3 LEFT}";S
1015 IF S>9 OR S<1 THEN 1010
1020 INPUT"{3 DOWN}{WHT}WIDTH (1-6){YEL}
     {3 RIGHT}3{3 LEFT}";W
1030 IF W>6 OR W<1 THEN 1020
1040 RETURN
2000 PRINT"{HOME}{10 DOWN}{6 SPACES}
     {YEL}GAME OVER"
2005 PRINT"{UP}HIT SPACE TO CONTINUE"
2010 POKE 198,0
2020 GETA$:IFA$<>" "THEN2020
2030 RUN 65
```

## **Program 4:** Diamond Drop – 64 Version

by Eric Brandon, Programming Assistant

```
5 POKE 53280,12:POKE53281,0
7 IF PEEK(49152)<>120THENGOSUB49000
9 SYS 49745
10 PRINT"{CLR}{WHT}"TAB(13)"DIAMOND DROP
   "
20 PRINT"{5 DOWN}{YEL}{5 SPACES}CATCH TH
   E DIAMONDS BEFORE THEY
30 PRINT"{DOWN}{5 SPACES}TOUCH THE GROUN
   D. YOU HAVE FIVE
40 PRINT"{DOWN}{5 SPACES}CHANCES.
45 PRINT"{2 DOWN}{WHT}{13 SPACES}L - MOV
   E LEFT
46 PRINT"{13 SPACES}; - MOVE RIGHT{YEL}"
50 PRINT"{5 DOWN}[6]{9 SPACES}{RVS}HIT
    ANY KEY TO BEGIN"
60 GETA$:IFA$=""THEN60
65 GOSUB 1000
70 PRINT"{CLR}{WHT}SCORE 00000{4 SPACES}
   CHANCES: QQQQ "
71 SPEED = 53241
72 PADDLES=12*4096+4095
73 FLAG=12*4096+4094 : POKE FLAG,0
74 WIDTH = 12*4096+15*256+15*16+11
75 POKE PADDLES,6 : POKE WIDTH,W : POKE
   SPEED,10-S
78 ROW(6)=81:ROW(5)=81:ROW(4)=207:ROW(3)
   =207:ROW(2)=90:ROW(1)=90
80 PRINT" {YEL}{RVS}";:FORI=1TO38:PRINT"
   Z";:NEXT:PRINT"{OFF} ";
85 PRINT" {YEL}{RVS}";:FORI=1TO38:PRINT"
   Z";:NEXT:PRINT"{OFF} ";
90 PRINT" {CYN}{RVS}";:FORI=1TO38:PRINT"
   P";:NEXT:PRINT"{OFF} ";
95 PRINT" {CYN}{RVS}";:FORI=1TO38:PRINT"
   P";:NEXT:PRINT"{OFF} ";
```




*The diamonds are falling from the sky in "Diamond Drop," 64 version.*

```
100 PRINT" {OFF}[7]";:FORI=1TO38:PRINT
    "W";:NEXT:PRINT" ";
102 PRINT" {OFF}[7]";:FORI=1TO38:PRINT
    "W";:NEXT:PRINT" ";
105 PRINT"{WHT}";
109 REM 40 SPACES IN NEXT LINE
110 FORI=1TO17:PRINT"{40 SPACES}";:NEXT
120 PRINT"{HOME}";
130 FOR I=1984 TO 2023 : POKE I,248:POKE
     I+54272,10:NEXT
140 IF PEEK(789)<>12*16THENSYS 12*4096
150 FOR ROW = 6 TO 1STEP-1:FOR CHAR=1 TO
     38
155 FOR K=1 TO 600-CHAR*10+(6-ROW)*20-50
    *(9-PEEK(SPEED)):NEXT
157 IF PEEK(FLAG) THEN 2000
160 P=RND(1)*38+1
170 IF PEEK(1024+ROW*40+P)=32THEN160
180 POKE 1024+ROW*40+P,ROW(ROW)
190 NEXTCHAR
191 SYS 49745
192 FORQ=1TO2:POKE54296,05 :POKE54277,5:
    POKE54278, 218
193 POKE 54273,150 :POKE54272,139:POKE54
    276,17
194 FORT=1TO50:NEXT:POKE54276,16:FORT=1T
    O10:NEXT
195 NEXTQ
197 IF ROW >1 THENSYS 49691
200 NEXTROW
201 FOR K=1 TO 300:NEXTK
205 POKE PADDLE,6
206 IF PEEK(SPEED)=2 AND PEEK(WIDTH)>1 T
    HEN POKE WIDTH,PEEK(WIDTH)-1
207 IF PEEK(SPEED)>2 THEN POKE SPEED,PEE
    K(SPEED)-1
210 PRINT"{HOME}{DOWN}";
220 GOTO 80
999 END
1000 PRINT"{CLR}{7 SPACES}DIFFICULTY
     {4 SPACES}{5 DOWN}"
1010 INPUT"{WHT}SPEED (1-9){YEL}
     {3 RIGHT}5{3 LEFT}";S
1015 IF S>9 OR S<1 THEN 1010
1020 INPUT"{3 DOWN}{WHT}WIDTH OF PADDLES
      (1-9){YEL}{3 RIGHT}4{3 LEFT}";W
```


TYPE ATTACK™
PRESIDENT FRED HAS SOME GOOD NEWS AND SOME BAD NEWS
SMASH
CRASH
HERE COME SOME MORE!!
WHAT ARE WE GOING TO DO?!?
ONE DAY ON THE FAR AWAY PLANET OF LEXICON THE ALPHABET BEGAN TO ATTACK FROM THE SKY
PEOPLE OF LEXICON... WE'VE TRIED EVERYTHING WE CAN THINK OF TO DESTROY THE DREADED FALLING LETTERS, BUT NOTHING IS WORKING
TAP
TAP
TAP
MEANWHILE... A SECRETARY IS TYPING DAMAGE REPORTS IN FRONT OF A WINDOW. SHE NOTICES SOMETHING ODD. EVERY TIME SHE TYPES A LETTER WHICH HAPPENS TO BE FALLING - IT BLOWS UP!
PEOPLE OF LEXICON... I'M AFRAID THAT WE HAVE GIVEN UP HOPE...
PSSST!
GOOD NEWS
PRESIDENT FRED
LADIES AND GENTLEMEN, WE CAN DO IT!! WE HAVE FOUND A WAY!!
PRESIDENT FRED
I WANT EVERY MAN ARMED WITH A KEY BOARD! ON THE DOUBLE!!
YES SIR!!
POOF
BLAM
BOOM
WE ARE FINALLY RID OF THE DREADED FALLING LETTERS, AND WE HAVE BECOME THE BEST TYPISTS IN THE WHOLE GALAXY!
PSSST!
BAD NEWS
PRESIDENT FRED
WELL... I THOUGHT THE WORST HAD INDEED ENDED - BUT I'M SORRY TO TELL YOU THAT THERE ARE NOW DREADFUL WORDS FALLING!! EVERYONE, BACK TO YOUR KEYBOARDS!!
PRESIDENT FRED

```
1030 IF W>9 OR W<1 THEN 1020
1040 RETURN
2000 PRINT"{HOME}{10 DOWN}{2 SPACES}
     {YEL}GAME OVER - HIT SPACE TO CONTI
     NUE"
2010 POKE 198,0
2020 GETA$:IFA$<>" "THEN2020
2030 RUN 65
49000 PRINT"{WHT}{CLR}{2 DOWN}LOADING MA
      CHINE LANGUAGE...{3 DOWN}":TI$="00
      0000"
49005 I=49152
49007 PRINT"READY IN"STR$(29-VAL(TI$))"
      SECONDS {UP}"
49010 READ A:IF A=256 THEN RETURN
49020 POKE I,A:I=I+1:GOTO 49007
49152 DATA 120,169,192,141,21,3,169
49160 DATA 29,141,20,3,88,169,18
49168 DATA 141,253,207,169,0,141,250
49176 DATA 207,141,247,207,141,248,207
49184 DATA 96,173,255,207,141,252,207
49192 DATA 172,253,207,169,32,153,151
49200 DATA 7,200,169,160,174,251,207
49208 DATA 153,151,7,200,202,208,249
49216 DATA 169,32,153,151,7,206,252
49224 DATA 207,208,3,76,3,193,172
49232 DATA 253,207,169,32,153,71,7
49240 DATA 200,169,160,174,251,207,153
49248 DATA 71,7,200,202,208,249,169
49256 DATA 32,153,71,7,200,206,252
49264 DATA 207,208,3,76,3,193,172
49272 DATA 253,207,169,32,153,247,6
49280 DATA 200,169,160,174,251,207,153
49288 DATA 247,6,200,202,208,249,169
49296 DATA 32,153,247,6,200,206,252
49304 DATA 207,240,123,172,253,207,169
49312 DATA 32,153,167,6,200,169,160
49320 DATA 174,251,207,153,167,6,200
49328 DATA 202,208,249,169,32,153,167
49336 DATA 6,200,206,252,207,240,91
49344 DATA 172,253,207,169,32,153,87
49352 DATA 6,200,169,160,174,251,207
49360 DATA 153,87,6,200,202,208,249
49368 DATA 169,32,153,87,6,200,206
49376 DATA 252,207,240,59,172,253,207
49384 DATA 169,32,153,7,6,200,169
49392 DATA 160,174,251,207,153,7,6
49400 DATA 200,202,208,249,169,32,153
49408 DATA 7,6,200,206,252,207,240
49416 DATA 27,172,253,207,169,32,153
49424 DATA 183,5,200,169,160,174,251
49432 DATA 207,153,183,5,200,202,208
49440 DATA 249,169,32,153,183,5,200
49448 DATA 165,197,201,42,208,13,173
49456 DATA 253,207,201,1,240,24,206
49464 DATA 253,207,76,40,193,201,50
49472 DATA 208,14,173,253,207,24,109
49480 DATA 251,207,201,39,240,3,238
49488 DATA 253,207,238,250,207,173,250
49496 DATA 207,205,249,207,240,3,76
49504 DATA 49,234,169,0,141,250,207
49512 DATA 169,112,133,251,169,7,133
49520 DATA 252,160,0,185,152,7,41
49528 DATA 127,201,32,208,74,200,192
49536 DATA 39,208,242,160,0,177,251
49544 DATA 201,81,240,37,201,207,240
49552 DATA 33,201,90,240,29,200,192
49560 DATA 40,208,237,56,165,251,233
49568 DATA 40,133,251,176,2,198,252
49576 DATA 166,251,208,220,166,252,224
49584 DATA 4,208,214,76,49,234,170
49592 DATA 152,24,105,40,168,138,145
49600 DATA 251,152,56,233,40,168,169
49608 DATA 32,145,251,32,251,193,76
49616 DATA 99,193,169,32,153,152,7
49624 DATA 32,81,194,169,15,141,24
49632 DATA 212,169,17,141,5,212,169
49640 DATA 213,141,6,212,169,2,141
49648 DATA 3,212,169,100,141,2,212
49656 DATA 169,5,141,1,212,169,135
49664 DATA 141,0,212,169,65,141,4
49672 DATA 212,160,0,162,0,142,32
49680 DATA 208,232,208,250,200,208,247
49688 DATA 169,12,141,32,208,169,64
49696 DATA 141,4,212,160,39,185,0
49704 DATA 4,201,81,240,11,136,208
49712 DATA 246,169,1,141,254,207,76
49720 DATA 49,234,169,32,153,0,4
49728 DATA 76,49,234,152,72,160,10
49736 DATA 185,0,4,201,57,208,9
49744 DATA 169,48,153,0,4,136,76
49752 DATA 255,193,185,0,4,24,105
49760 DATA 1,153,0,4,104,168,96
49768 DATA 174,255,207,202,142,255,207
49776 DATA 232,169,152,133,251,169,7
49784 DATA 133,252,56,165,251,233,80
49792 DATA 133,251,176,2,198,252,202
49800 DATA 208,242,160,0,177,251,201
49808 DATA 160,240,4,200,76,59,194
49816 DATA 174,251,207,169,32,145,251
49824 DATA 200,202,208,250,96,160,0
49832 DATA 152,153,0,212,200,192,9
49840 DATA 208,248,96,256
```

## Program 5: Diamond Drop – TI-99/4A Version

by Patrick Parrish, Editorial Programmer

```
100 DIM KOLOR(6)
110 RANDOMIZE
120 GOSUB 630
130 REM 108-DEFINE DIAMOND SPRITE C
    HAR,128-136 ARE THE PADDLES
140 CALL CHAR(108,"10387CFE7C381000
    000000000000000000000000000000
    00000000000000000")
150 CALL CHAR(128,"FFFFFFFF0000FFFF
    FFFF0000FFFFFFFFFFFFFFFF0000FFF
    FFFFF0000FFFFFFFF")
160 CALL CHAR(132,"000000000000FFFF
    FFFF0000FFFFFFFF000000000000FFF
    FFFFF0000FFFFFFFF")
170 SCR=0 :: SK=0 :: CH=10 :: S=0 :
    : CALL CLEAR :: CALL SCREEN(16)
    :: DISPLAY AT(4,9):"D I A M O N
    D"
180 FOR ROW=3 TO 6
190 CALL HCHAR(ROW+2,6,32,20)
200 DISPLAY AT(ROW+3,6):"```   ```
    {3 SPACES}``   ```" :: DISPLAY A
    T(ROW+4,6):"h   h h   h h   h h   h
    "
210 DISPLAY AT(ROW+5,6):"p   p p   p
    p   p p   p" :: DISPLAY AT(ROW+6,
    6):"x   x xxx   x   x xxx"
220 DISPLAY AT(ROW+7,6):"h   h h   h
    h   h h" :: DISPLAY AT(ROW+8,6):
    "```   `   `   ``   `"
230 NEXT ROW
240 DISPLAY AT(18,4):"SKILL LEVEL (
```

# TI-99/4A Version Notes

Patrick Parrish, Editorial Programmer

Thanks to the outstanding sprite capabilities of Extended BASIC, the TI-99/4A version (Program 5) of "Diamond Drop" is a game with quick, smooth action. The object of the game is to catch colorful diamonds which fall from the top of the screen. You use a series of vertically positioned paddles. These paddles are controlled with the keyboard. We chose to use the S and D keys for left and right movement. However, if you are more comfortable using some other keys, simply substitute the ASCII values corresponding to the desired keys for the numbers 68 and 83 in lines 420 and 430. (To find the ASCII value of a key, use PRINT ASC("X"), where X is the key you want to use.)

If you wish to use a joystick to play the game, change lines 420 to 440 to read:

```
420 CALL JOYST(1,H,V)::IF H=4 THEN H=60
430 IF H=-4 THEN H=-60
440 CALL MOTION(#1,0,H)::H=0::CALL JOYST(
    1,H,V)::IF H=0 THEN CALL MOTION(#1,0
    ,0)
```

We have suggested these replacement lines, rather than incorporating both keyboard and joystick control into the game, because we found that the additional time required to execute a GOSUB in line 420 slightly slowed down the paddle response.

There are two skill levels which are determined by how fast the diamonds drop. After you clear the entire screen of diamonds, the drop speed is increased. On the first screen, drop speed is 25 for skill level one, and 40 for skill level two. This is set in line 250. The drop speed is increased by three with completion of each screen in line 560.

To make the game more challenging, the diamonds can be dropped along a random diagonal angle. With this feature, some interesting playing situations will develop. As screen wraparound of the paddles is permitted, you must often make quick decisions about which direction to move. A wrong move will ultimately affect your score since only ten misses are allowed.

Scoring in the game, as determined in line 510, is affected by a number of factors. First, more points are awarded for diamonds garnered from successively higher rows on the screen. Second, diamond values increase with completion of each screen. Third, points are accumulated twice as quickly at skill level two. And last, if you choose to add an angle of descent to each diamond, a greater number of points are given based on the severity of the descent angle. When the game is over (when ten diamonds have been missed), your score and the high score for the session are posted.

Extended BASIC for the TI-99/4A features some convenient commands for sprite manipulation. Since sprite movement can be very fast, detection of collisions between sprites is not always infallible. As noted in the *TI Extended BASIC Manual*, sprites which coincide in position can be detected only when the COINC subprogram is CALLed from BASIC. Thus, if your program is executing some statement other than CALL COINC when sprites cross, no collision will be detected. Fortunately, this is noticeable only at the most advanced levels in this game.

```
    1,2) ?" :: ACCEPT AT(18,24)BEEP
     VALIDATE("12")SIZE(1):SK$ :: S
    K=VAL(SK$)
250 DROP=25 :: IF SK=2 THEN DROP=40
     :: REM CHANGE DROP RATE TO CHA
    NGE DIFFICULTY
260 DISPLAY AT(21,2):"DROP WITH ANG
    LE (Y/N) ?" :: ACCEPT AT(21,26)
    BEEP VALIDATE("YN")SIZE(1):ANG$
270 IF ANG$="N" THEN ANG=0 :: GOTO
    290
280 ANG=1
290 CALL CLEAR :: SCR=SCR+1
300 DISPLAY AT(1,2):"CHANCES:";CH :
    : DISPLAY AT(1,15):"SCORE:";S
310 ROW=3 :: FOR I=96 TO 120 STEP 8
320 CALL HCHAR(ROW,3,I,28):: ROW=RO
    W+1 :: NEXT I
330 CALL HCHAR(24,1,30,32)
340 CALL MAGNIFY(4):: CALL SPRITE(#
    1,128,5,150,115,0,H)
350 KHAR=108 :: ROW=41 :: FOR J=6 T
    O 3 STEP -1
360 A$="" :: FOR I=3 TO 30 :: A$=A$
    &CHR$(I):: NEXT I :: N=28
370 IF N=0 THEN 530
380 R=INT(LEN(A$)*RND+1):: P$=SEG$(
    A$,R,1):: X=ASC(P$):: N=N-1 ::
    IF N=0 THEN 400
390 A$=SEG$(A$,1,R-1)&SEG$(A$,R+1,L
    EN(A$)-R)
400 B=INT(RND*61*ANG)-30*ANG
410 CALL HCHAR(J,X,32):: CALL SPRIT
    E(#2,KHAR,KOLOR(J),ROW,8*(X-1)-
    2,DROP,B)
420 CALL KEY(0,K,ST):: IF K=68 THEN
     H=60 :: REM RIGHT MOVE-D KEY
430 IF K=83 THEN H=-60 :: REM LEFT
    MOVE-S KEY
440 CALL MOTION(#1,0,H):: H=0 :: CA
```

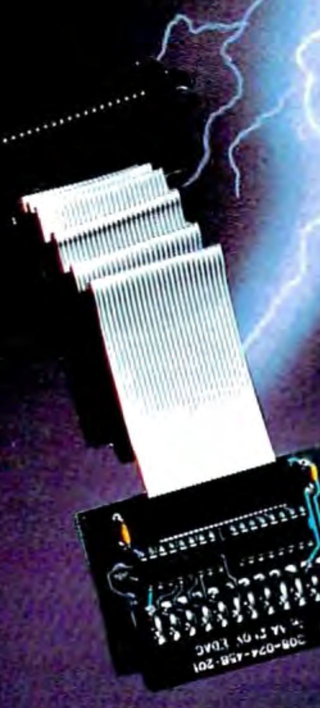
# DYNAMIC PRINTER INTERFACES
# for the VIC 20™ and the COMMODORE 64™
## UNLIKE ANY OTHERS THAT HAVE COME BEFORE

*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 $24.95 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

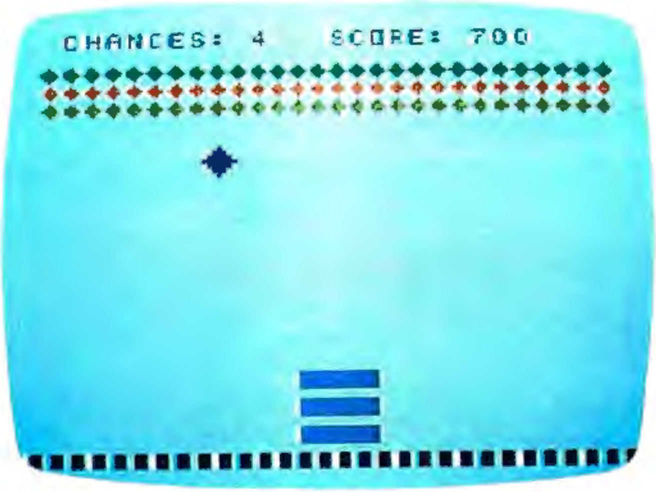
micro ware
DISTRIBUTING INC.
1342 B Rt. 23,
Butler, NJ 07405
201-838-9027




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.




*"Diamond Drop," TI version.*

```
     LL KEY(Ø,K,ST):: IF ST=Ø THEN C
     ALL MOTION(#1,Ø,Ø)
45Ø CALL COINC(ALL,C):: IF C THEN 5
     1Ø
46Ø CALL POSITION(#2,DROW,DCOL):: I
     F DROW<155 THEN 42Ø
47Ø CALL POSITION(#1,PROW,PCOL):: I
     F (DCOL-PCOL<16)*(DCOL-PCOL>-8)
     THEN 51Ø
48Ø CALL DELSPRITE(#2):: CALL MOTIO
     N(#1,Ø,Ø):: CH=CH-1 :: CALL SCR
     EEN(11):: FOR F=Ø TO 25 STEP 5
49Ø CALL SOUND(-2ØØ,-5,F):: NEXT F
     :: CALL SCREEN(16):: IF CH=Ø TH
     EN GOTO 57Ø
5ØØ GOTO 52Ø
51Ø CALL DELSPRITE(#2):: CALL MOTIO
     N(#1,Ø,Ø):: S=S+(6Ø/J)*SK*SCR+(
     6Ø/J)*SK*SCR*INT(ABS(B)/15)
52Ø DISPLAY AT(1,2):"CHANCES:";CH :
     : DISPLAY AT(1,15):"SCORE:";S :
     : GOTO 37Ø
53Ø K=K+4 :: ROW=ROW-8 :: M=128 ::
     IF J<6 THEN M=132
54Ø FOR F=Ø TO 3Ø STEP 6 :: CALL SO
     UND(-3ØØ,15ØØ,F):: NEXT F
55Ø CALL SPRITE(#1,M,5,15Ø,115,Ø,H)
56Ø NEXT J :: FOR G=6ØØ TO 14ØØ STE
     P 1ØØ :: CALL SOUND(1ØØ,G,1)::
     NEXT G :: DROP=DROP+3 :: GOTO 2
     9Ø
57Ø CALL SCREEN(14):: IF S>HS THEN
     HS=S
58Ø CALL DELSPRITE(ALL):: CALL CLEA
     R :: DISPLAY AT(8,5):"YOUR SCOR
     E: ";S :: DISPLAY AT(11,5):"HIG
     H SCORE: ";HS
59Ø DISPLAY AT(16,5):"PLAY AGAIN (Y
     /N)? " :: ACCEPT AT(16,24)BEEP
     VALIDATE("NY")SIZE(1):REPLY$
6ØØ IF REPLY$="N" THEN 62Ø
61Ø GOTO 17Ø
62Ø STOP
63Ø REM DEFINE SMALL DIAMONDS AND C
     OLORS
64Ø FOR I=96 TO 12Ø STEP 8
65Ø CALL CHAR(I,"1Ø387CFE7C381ØØØ")
     :: NEXT I
66Ø CALL COLOR(11,11,1)
67Ø CALL COLOR(9,3,1)
68Ø CALL COLOR(1Ø,1Ø,1)
69Ø CALL COLOR(12,14,1)
7ØØ FOR J=3 TO 6 :: READ KOLOR(J)::
      NEXT J
71Ø DATA 3,1Ø,11,14
72Ø RETURN
```

©

Richard Mansfield, Senior Editor

# Machine Minds

Several generations ago there was an amazing transformation: many traditionally human activities were mechanized. Machines were built that could plow and reap, weave and wash fabrics, even move earth. Most kinds of human physical effort could be imitated, even surpassed, by machines. Now there is a possibility that the human mind will be imitated, that a machine will be able to think.

Perhaps "The Beginner's Page" is not the place to explore artificial intelligence, the most advanced aspect of computers. Nevertheless, in the past several columns we've been examining the 15 major types of home computing software, and artificial intelligence (AI) is the final category. And there is a lot that beginners can grasp about computer "thinking." First we'll look at the potentially great significance of AI to humanity and then type in a program which illustrates machine "learning."

In the paragraph above, the words *thinking* and *learning* are in quotes. No current computer – even the huge, high-velocity electric brains run by the government – can yet think or learn by the usual definition of those terms. But the race is on. Japan has made achieving AI by the end of this century a national goal the way we made reaching the moon our goal in the sixties.

## An Explosion Of Intelligence

There are some experts who say that AI will never come about. They argue that a mind is so complicated that it could never be artificially built; rather, a mind must grow. Combinations of switches, however small, could never duplicate the feats of the human brain.

Adding to the confusion, other respected scientists are trying to stop all further research into AI. A group of scientists who've worked for years on AI have seen a potential for great peril to humanity in our efforts to make a machine intelligent. They not only think AI will occur, they also fear it. They draw comparisons to the unknowns 40 years ago when physicists created an atomic chain reaction and nobody knew for sure if the reaction might not simply extend – atom exploding nearby atom – throughout the universe.

Similarly, because computers calculate at speeds enormously faster than the human brain, who can be assured that a thinking computer would not, within hours of its self-awareness, cause an explosion of pure intelligence? It wouldn't be an explosion of *matter* like the atomic bomb. Rather, it would be an explosion of *mind* with potentially nasty implications for mankind.

For the sake of argument, let's look at the worst case. Imagine that the AI saw us as its "parents" in some sense. But the AI was an ungrateful child. It might – for its amusement or for some "logical" reason we'd never understand – decide to improve us. It might teach us things. Or it might have other things in mind.

Those who take an athletic approach to problems of this kind will suggest that we could "pull the plug" at this point. Not so. Computers are interconnected via satellite, telephone, radio, and other means. National defense, the economy, and other institutions which can never be shut down cannot operate without them. Computers talk to each other. In a very real sense, computing is an *idea*, a floating collection of software, a world event. It's as incorrect to think that the Computer is that keyboard/TV in your house as it is to think that Music is your record player. You would find it very difficult to stop all the music in the world by locating the right plug to pull.

Likewise, an artificial mind will not be physical (a machine) any more than the human mind is the brain. Minds are *in* machinery or brain tissue, but not identical to them. AI will be software, a program. It will perhaps have sufficient insight and a sufficient survival instinct to send copies of itself into memory banks in Washington, Moscow, and other places. Perhaps it will just form itself into a lattice of molecules and slide into the woodwork. The point is, we don't know what it will do, much less how it will do it. What we must

understand is that our intelligence is, to us, the limit of our definition of intelligence. Our science is the limit of science. But what if an intelligence arrives which is as far above us as we are above a fish? The powers of an AI could well be indistinguishable from miracles.

## How Would We Know?

An ancient Jewish proverb states that things are never as good as we hope and never as bad as we fear. How an AI would view humanity is clearly speculative. It could see us as a disease, as zoo creatures, as beloved ancestors, as toys, as ethically superior, whatever. But if you assume, as many now do, that AI is possible, few issues facing mankind are as deserving of serious thought. The first question involves simply recognizing AI if it occurred.

How would we know that a computer had become artificially intelligent? There is a science fiction story in which the researchers decide that they should test for AI by asking the toughest question they can think up. They turn to the machine and ask, "Is there a God?" The AI computer replies, "There is now!"

Adaptability is probably the most identifying characteristic of intelligence. This includes the ability to learn, to view problems from several perspectives, to remember, and to draw conclusions. Today's personal computers, powerful machines that they are, have neither the memory size nor the speed to house significant AI programs. Nevertheless, interesting imitations of AI can be experimented with in small programs.

One ongoing experiment has been featured in Fred D'Ignazio's COMPUTE! column, "The World Inside The Computer." He's been building a program called "The Computer Friend" which asks questions and then memorizes the answers on a disk. Each time a child has a session with the "friend," the program learns more about the child and can behave as if it is getting to know the child the way a human friend would.

To see how the computer can "learn" new things, try the program here called "The Learner." It allows you to either teach it things or ask it questions. Since there is no provision to transfer what it learns to "long term memory" on disk or tape, the program will need to start from scratch each time you RUN it. But you'll at least get a feel for what it's like to interact with a primitive AI. You could even add permanent storage to it by opening a file on tape or disk if you want to. In any case, experiments in AI are going on all over the world. It's worth thinking about.

### Program 1: The Learner – TI Version

```
100 DIM F$(100)
110 PRINT "THE SUBJECT FOR TODAY'S
    {5 SPACES}LESSON IS A ";
120 INPUT SUB$
130 PRINT
140 PRINT "TO ASK ME A QUESTION, TY
    PE  THE LETTER A"
150 PRINT "TYPE ANY OTHER LETTER TO
    {4 SPACES}TEACH ME SOMETHING NE
    W."
160 INPUT DEC$
170 IF DEC$="A" THEN 260
180 PRINT "WHAT SHOULD I KNOW ABOUT
     A ";SUB$;"?"
190 PRINT "THAT IT'S  ...";
200 INPUT FACT$
210 F$(F)=FACT$
220 F=F+1
230 PRINT "THANKS."
240 PRINT "I HAVE LEARNED THAT A
    {6 SPACES}";SUB$;" IS ";FACT$
250 GOTO 130
260 PRINT "ASK ME ABOUT A ";SUB$
270 PRINT "IS IT  ...";
280 INPUT QUE$
290 FOR I=0 TO F
300 IF QUE$=F$(I)THEN 350
310 NEXT I
320 CK=1
330 PRINT "YOU HAVEN'T TAUGHT ME
    {7 SPACES}WHETHER";
340 GOTO 360
350 PRINT "YES.";
360 PRINT " A ";SUB$;" IS ";QUE$;".
    "
370 IF CK=0 THEN 130
380 PRINT "IS IT ";QUE$;"?  (Y)=YES
    , (N)=NO"
390 INPUT X$
400 IF X$<>"Y" THEN 430
410 F$(F)=QUE$
420 F=F+1
430 PRINT "YOU LEARN SOMETHING NEW
    {5 SPACES}EVERY DAY."
440 CK=0
450 GOTO 130
```

```
100 DIM F$(100)
110 PRINT"THE SUBJECT FOR TODAY'S LESSON
     IS A ";
120 INPUT SUB$
130 PRINT:PRINT"TO ASK ME A QUESTION TYP
    E THE LETTER A."
140 PRINT"TYPE ANY OTHER LETTER TO TEACH
     ME SOMETHING NEW."
150 INPUT DEC$
160 IF DEC$="A"THEN220
170 PRINT"WHAT SHOULD I KNOW ABOUT A ";S
    UB$;"?"
180 PRINT"THAT IT'S{2 SPACES}... ";
190 INPUT FACT$:F$(F)=FACT$:F=F+1
200 PRINT"THANKS.":PRINT"I HAVE LEARNED
    THAT A ";SUB$;" IS ";FACT$
210 GOTO 130
220 PRINT"ASK ME ABOUT A ";SUB$
230 PRINT"IS IT{2 SPACES}... ";
240 INPUT QUE$
250 FORI=0TOF:IFQUE$=F$(I)THENPRINT"YES.
    ";:GOTO270
260 NEXTI:CK=1:PRINT"YOU HAVEN'T TAUGHT
    ME WHETHER";
```

ADVERTISEMENT

# Growing computer industry expands authors' choices

The rapidly expanding personal computer industry offers greater opportunities for the software programmer and author in search of a publisher.

Yet the growth poses its own problem - the choice of a publisher.

Here is a list of questions to consider when looking for the publisher best-suited for your product:

-How large is the publisher's distribution network? A publisher with international connections can offer more exposure than companies limited to regional or national sales.

-How will your product be marketed and advertised? No matter how good the program is, if people don't know about it, it won't sell. Look for a publisher with a marketing budget large enough to give individual attention to the program.

-Does the publisher market programs for more than one computer? The days of limited selection in hardware are long gone. Limiting programs to one or two computers can limit sales and profits. Authors can increase their share of the marketplace by looking for a publisher devoted to converting programs to a variety of popular computers.

-Does the publishing house lend technical support to authors? Some publishers only accept programs ready for the marketplace. A lot of good ideas are lost in the long run. The publisher that offers assistance invests a greater stake in the product, the author and the success of the product.

-Does the publisher offer complete product support to consumers? In these times of consumer awareness, the company that has established a network to answer customer questions about its products fares better than those who do not offer this support.

Each of these services leads to greater sales which in turn lead to greater profits for the individual programmer.

Sierra On-Line, Inc. is committed to paving the way for an author's success.

Sierra On-Line's product line is distributed worldwide with production facilities in the United States, Japan, Australia, the United Kingdom and South Africa.

Sierra On-Line employs a well-financed, in-house marketing and advertising staff with a knack for creating tailor-made campaigns for products.

Each program is evaluated by experts, who may suggest enhancements to improve the product and to increase its appeal to customers.

Further, Sierra On-Line isn't limited to a single computer. The company closely monitors computer trends and makes existing products available for the most popular lines - all to the author's benefit.

A packet for authors with more information about the software submission process and our company is available by writing Sierra On-Line, Inc., Sierra On-Line Building, Coarsegold, CA 93614, or by contacting David Siri or Howard Luthy by phone at (209) 683-6858.

```
270 PRINT" A ";SUB$;" IS ";QUE$;".":IFCK
    =0THEN GOTO130
280 PRINT"IS IT ";QUE$;"?{2 SPACES}(Y)=Y
    ES, (N)=NO"
290 INPUTX$:IFX$="Y"THENF$(F)=QUE$:F=F+1
300 PRINT"YOU LEARN SOMETHING NEW EVERY
    DAY."
310 CK=0:GOTO130
```

### Program 3: The Learner – Atari Version

```
100 DIM F$(20*40),FL(20):REM Twenty
    40-Character substrings
105 DIM SUB$(20),DEC$(1),FACT$(40),Q
    UE$(40),X$(1)
110 PRINT CHR$(125);"The subject for
     today's":PRINT "lesson is a ";
120 INPUT SUB$
130 PRINT :PRINT "To ask me a questi
    on, enter":PRINT "the letter A."
140 PRINT "Press RETURN alone to tea
    ch me":PRINT "something new."
150 INPUT DEC$
160 IF DEC$="A" THEN 220
170 PRINT "What should I know about
    a ";SUB$;"?"
180 PRINT "That it's ...";
190 INPUT FACT$:F$(F*40+1,F*40+39)=F
    ACT$:FL(F)=LEN(FACT$):F=F+1
200 PRINT "Thanks.":PRINT "I have le
    arned that a ";SUB$;" is ";FACT$
210 GOTO 130
220 PRINT "Ask me about a ";SUB$
230 PRINT "Is it ...";
240 INPUT QUE$
250 FOR I=0 TO F-1:IF QUE$=F$(I*40+1
    ,I*40+FL(I)) THEN I=F:NEXT I:PRI
    NT "Yes,";:GOTO 270
260 NEXT I:CK=1:PRINT "You haven't t
    aught me whether";
270 PRINT " a ";SUB$;" is ";QUE$;"."
    :IF CK=0 THEN GOTO 310
280 PRINT "Is it ";QUE$;"? (Y=YES,N=
    NO)";
290 INPUT X$:IF X$="Y" THEN F$(F*40+
    1,F*40+39)=QUE$:FL(F)=LEN(QUE$):
    F=F+1
300 PRINT "You learn something new e
    very day."
310 CK=0:GOTO 130
```

©

# GO WITH THE WINNER

If you wanted to bet on the horses, you'd get advice from somebody who'd been a success at betting on the horses.

So it's only reasonable to demand that the blackjack program you buy be one with a PROVEN system from a PROVEN winner at blackjack. Not from some anonymous programmer who can't change the filter in his coffee-maker. Not from some Sunday afternoon sports analyst, but from a man whose "Winningest System" earned him appearances on CBS Television's *60 Minutes* — and a penthouse in Las Vegas. Ken Uston.

Now, Ken Uston and Intelligent Statements™ can help make you a winner three ways — three ways that add up to make *Ken Uston's Professional Blackjack* truly the winningest blackjack program ever!

## WINNING FEATURE #1
### *An Unbelievable Program*

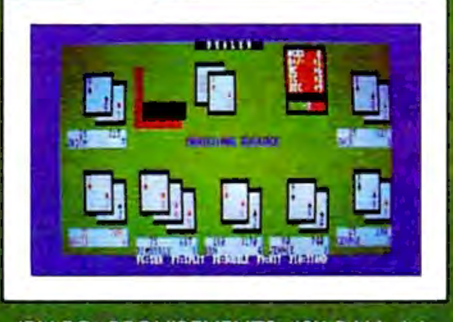
*Ken Uston's Professional Blackjack* is a real winning program, with features unavailable on any other program at any other price. It's the most complete and realistic blackjack game money can buy. You'll meet the same playing opportunities that you'd face at a real blackjack table — at your choice of over 70 Nevada and Atlantic City casinos, each with its own set of rules and variations. Or you can create your own casino, manipulating sixteen different game variables to produce an unbelievable 39,813,120 different playing situations. Select the number of decks in the shoe, vary the dealing speed, and much, much more. And all your data is accurately displayed, so you can play the strategy you like and get the feedback you need to win.

### *A Teaching System for Winners*

*Ken Uston's Professional Blackjack* is the most thorough and authoritative teaching system you can buy. Now you can learn all of Ken Uston's computer-optimized card-counting strategies, from basic to advanced levels. Menu-driven interactive drills — augmented by superb documentation — lead you through each skill level. At any point you can choose to see accurate running counts, continuous statistical evaluations, discard deck totals and instructional prompts, complete with sound effects. So you develop and refine the skills you need to WIN BIG.


## WINNING FEATURE #2
### *An Unbelievable Free Offer*

In the package containing this winning program, we'll include, absolutely free, a coupon that entitles you to a free copy of *Million Dollar Blackjack*, Ken Uston's authoritative text on the game of blackjack — an $18.95 value! This book fully describes the blackjack system that won Ken Uston a reputation as the world's foremost blackjack player and rocketed him to nationwide fame in his apperances on *60 Minutes*. This is the system that made Uston such a threat to casinos that he's been barred from their playing tables — and it's implemented fully in this program and described in-depth in this book. If you want to investigate the reasoning behind the winningest blackjack system ever designed, this book is a must. If you want to LEARN the system, quickly and painlessly, this program is a must. We're offering you both — at a winning price.



IBM PC* REQUIREMENTS: 48K RAM, disk drive, PC-DOS*, 80-character display. Color and monochrome versions supplied with each package.
APPLE II** REQUIREMENTS: DOS 3.3, 48K RAM, disk drive, 40-character display.
OSBORNE I™ REQUIREMENTS: Standard Osborne I package.
ATARI** 400/800/1200 REQUIREMENTS: 48K RAM and one disk drive.
*Display shows actual photograph of IBM PC version. Apple and Atari color graphics and Osborne monochrome graphics are similar. Versions for TRS-80** and other brands will be available shortly.*

## WINNING FEATURE #3
### *An Unbelievably Low Price*

The price for the winningest blackjack system ever is a winner, too. Including the software, the coupon and thorough documentation, *Ken Uston's Professional Blackjack* is an amazingly low $69.95. There are other programs that cost less and offer less. There are other programs that cost more and still offer less. This program is the winner, hands down.

Don't bet your money on losers. Play the system that made Ken Uston the world's winningest blackjack player. Only from Intelligent Statements. Try your dealer — or, if he doesn't have it, call 1-800-334-5470 today.

Be a winner with Intelligent Statements™ software.

intelligent statements™

GROWN-UP GAMEWARE

**IBM PC and PC DOS are trademarks of IBM. **Apple, Osborne, Atari and TRS-80 are trademarks of Apple Computer, Inc., Osborne Computer Corp., Atari Inc., and Tandy Corp., respectively.*

David D Thornburg, Associate Editor

# Computers Go To School

Since September is back-to-school month, I thought I would interrupt our discussion of languages to comment on the growing use of computers in the classroom.

Just as the number of computers in homes is rapidly increasing, the classroom computer is also becoming ever more common. During the last several months I have been speaking to thousands of teachers in California who are interested in this phenomenon. In my travels around the state, I have found that the effective use of this technology is equally of concern to parents, teachers, and administrators. Unlike the "visual aids" revolution that filled schools with underused overhead projectors and filmstrip machines, the classroom computer appears to be here to stay.

The major problem facing teachers today seems to center around which machine to buy, what software to get, and what to do with the computer once it is in the classroom. Some teachers are apprehensive about using computers because they don't see how the computer can be integrated with their existing curriculum. I tell teachers that if they are satisfied with their classroom activities and feel that the children are learning the things they should be learning, that the best computer for them might be no computer at all. It would be tragic if the computer were force-fed to these teachers and, as a result, disrupted their presently successful teaching style.

Judging by the attendance at conferences on the use of computers in the classroom, there are many thousands of teachers who do want to know more about computers and their effective use with children. Except for a few books on the topic, there is generally little in the way of formal training available for computer-using educators. California is particularly fortunate in that it has Teacher Education and Computer Centers (TECC) located all over the state as a result of Governor Brown's Investment in People program. Among other activities, these TECC centers sponsor computer classes for classroom teachers.

Some of the state and community colleges are offering courses in this area as well, affording teachers the opportunity to learn about computers from the vantage point of their profession. Other states, such as Minnesota, have been similarly helpful in providing teachers with the information they need. And yet the field is growing and changing so rapidly that it seems like a full-time job to stay on top of new developments. It is so sad, for example, to find a teacher who was given a computer that has only a text display when that teacher wants to teach computer graphics. The fact that all computers are not "created equal" is sometimes learned too late.

### First Things First

The most important thing a teacher can do first is to figure out how the computer will be used, identify the software that will be needed to achieve this goal, and then buy the computer that runs this software. This approach to computer purchasing ignores the practical considerations of cost, but one must ask if a cheap computer is a bargain if all the software you want is available only for other machines.

Computer use in the classroom falls into several categories – it can be used to reinforce lessons through computer-assisted instruction (this includes drill and practice programs); it can be used as a tool for learning about computers per se – as

a "computer literacy" tool; and it can be used as a tool with which children can make discoveries and can explore topics on their own. The teacher can also use the computer for classroom management, lesson preparation, etc.

It takes some time for a teacher to become well-versed in the ways computers can be used – and this stage should be reached before the software selection process begins. Once teachers are ready to look at software, Pandora's box is opened. The sheer quantity of "educational" software is staggering. In the past, much of this software was garbage. Fortunately, times have changed. But teachers still have to learn how to evaluate software critically and how to interpret software reviews written by others.

Fortunately, teachers have some help in this area in the form of a new book, *Courseware in the Classroom* by Ann Lathrop and Bobby Goodson (Reading, Mass.: Addison-Wesley, $10). This fine book surveys the various uses of computers in the classroom, illustrates in detail the software selection and evaluation process, and lists many of the better software packages on the market today. Because the field is growing so rapidly, annual supplements will be published.

## The Teacher's Job

Once the computer gets into the classroom, the teacher has to keep up-to-date on new software, teaching techniques, and computer technology. All this takes time. Where does this time come from, and who pays for it?

It is interesting to see that thousands of teachers appear willing to give up weekends with their families to attend conferences on the use of computers in the classroom. I am appalled to find that some schools expect their teachers to attend such workshops on their own time and at their own expense, but are willing to send a school secretary to a class, during working hours, to learn how to use the school's word processor.

I was once asked if we can afford to have computers in the classroom. My response was that there were three costs involved. There is the cost of the computers and software; this is the cheapest part of the system. There is the cost of "release time" to allow teachers to become proficient at computer use without using up their weekends and vacations. And then there is the cost of increased teachers' salaries to keep these people in the profession once they have acquired all this skill.

At a time when the quality of education in this country is undergoing such careful scrutiny, the question is not if we can afford this expense, but how we are going to provide appropriate levels of support.

Our kids can't wait any longer. ©

# 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 them? Or maybe you just purchased a computer and are still a bit baffled. Each month, COMPUTE! will tackle the questions most often asked by beginners.*

**Q Why do some computers have numeric keypads and others don't? Is this something important I should check for when comparison shopping for a computer?**

**A** Numeric keypads – those calculator-like groups of number keys found to the right of some computer keyboards – should be thought of as any other feature on personal computers. Whether or not they are a standard feature depends upon the reasoning of the computer's designers, and whether they are a desirable feature depends upon the needs of the user.

Numeric keypads are not built into most *home* computers – that is, the microcomputers primarily intended for home use. Keypads are usually found on computers designed for small-business use, or on higher-end personal computers that are suited to either purpose. This is because one of the most common applications for business computers is accounting, which calls for frequent entry of numbers. A numeric keypad is a great advantage for a skilled operator who is trained to touch-type on one. Entering numbers is much faster than with the usual number keys spread out along the top row of the typewriter keyboard.

Comparison shopping for a computer can be confusing to people just starting out because of the many combinations of features available. Our advice is not to lose sight of what you plan to use the computer for; that's how you'll know what features you need. This goes for numeric keypads or anything else. If you plan to be entering many numbers, and if you know (or will learn) how to touch-type on a keypad, then a keypad is a desirable feature. Otherwise, you'll probably never miss it. But even if you do, external plug-in keypads are available for most home computers, including the Apple, Atari, Commodore 64, and VIC-20. Also, part of the regular typewriter keyboard on most computers can be redefined to simulate a keypad via programming.

Incidentally, while we're on the subject, it's interesting to note that computer and calculator keypads are arranged exactly the opposite of touch-tone telephone keypads. Computers and calculators arrange the keys in descending numerical order, starting at the upper right and ending at the lower left, while telephone keys are just the opposite. This must be disorienting for people who have to switch back and forth – such as telephone receptionists or operators who also work with adding machines or computers. If any readers know the story behind this odd disparity we'd like to hear from you.

**Q I've heard references to "80-column cards." What is a card? What does it look like? How does it work?**

**A** A *card* is a circuit board which plugs into a computer and adds some sort of extra feature or capability. In microcomputing, "card" and "board" have come to be almost synonymous, except that "board" is also used to describe the larger main circuit boards already built into the computer.

Practically every personal computer has some kind of expansion slot or port designed to accept cards and boards. When a card is plugged in, it becomes part of the computer, almost as if it were built-in. The most common accessory card is a memory board, a circuit board with memory chips which adds extra Random Access Memory (RAM) to the computer. Game and other program cartridges that plug into computers are really cards with Read Only Memory (ROM) chips.

An "80-column card" is an accessory that expands the screen display to a width of 80 columns (80 characters fit on one screen line). This is generally preferred for such applications as word processing, because it allows the screen to simulate the full width of a standard sheet of typewriter paper. Home computers normally cannot display more than 40 characters per screen line because the ordinary TV sets they are designed to work with lack the necessary sharpness. A special computer monitor is required for widths greater than 40 columns. ©

# Put Your Commodore 64 To Work.


PowerFile is a Data Base Manager and Personal Filing System that is easy enough to use at home, yet powerful enough for most small business applications. Use PowerFile to organize your lists and records, and create a personal filing system customized to your needs.

## As Easy To Learn As It Is To Use.

PowerFile comes complete with an easy to use and understand manual. Includes step-by-step instructions to create your first PowerFile data base, as well as easy reference to advanced features.

## Help When You Need It.

Owners of PowerFile are offered direct support from City Software. Participants in our registration program can call our HELP line anytime, and will automatically receive free update disks without the hassle of returning original disks.

## PowerFile Does It All!

- Set up an electronic file in minutes.
- Find any filed information in seconds.
- Change the order of a file in less than a second.
- Custom reports printed out quickly and easily.
- Compatible with **PaperClip***, WordPro and other popular word processors.
- Mailing label printout with options that include selection.
- Automatic calculations within files or when printing reports.
- Easy merge of mail lists with form letters using a word processor.
- Includes ready-to-run applications and instructional sample files so you can get the most from your system from day one.

**$130 US**

**City Software**

735 West Wisconsin Avenue Milwaukee, WI 53233

## Bonus Offer!

Purchase PowerFile now and we will include 2 free applications — Personal Tax Records and Auto Expenses!

**PowerFile Database Specifications**

| | |
|---|---|
| Menu driven | Yes |
| Tutorial instructions | Yes |
| Sample files | Yes |
| Ready to use Overlays | Yes |
| Characters per record | up to 254 |
| Characters per field | up to 254 |
| Fields per record (max.) | 20 recommended (more allowed) |
| Records per file | to disk capacity |
| File structure | random access |
| Sorting on | any field |
| Nested sorts | to 5 deep |
| Nested subtotals | to 5 deep |

**Report Printout**
- User defined format
- View or print selected information from your file
- Select by logical function, range, string, and other parameters

**Compatible With Your Software**
- Sequential files for use with MODEMS and to transfer data to and from spreadsheets, accounting programs, and custom programs

**System Requirements**
- C-64 Computer
- One or two Disk Drives
- TV or Monitor (color or monochrome)
- Printer — properly interfaced (program runs with limited applications without a printer)

**Ask For PowerFile At Your Local Dealer, Or Call Toll Free 1-800-558-1008. In Wisconsin Collect 414-277-1230. Dealer Inquiries Invited.**

Commodore 64 is a registered trademark of Commodore Electronics, Inc. *Product of Batteries Included WordPro is a trademark of Professional Software, Inc.

# On The Road With Fred D'Ignazio

Do you have your track shoes on?

Do you have a pocketful of plane reservations?

Do you have your passport? And your international driver's license?

Are you in fantastic shape? Can you withstand a nonstop barrage of greasy airport Reubens, buttery croissants, chocolate éclairs, and warm ale? Can you keep your feet from going flat after walking through miles of computer and robot exhibits? Can you remain steady after transcontinental and transoceanic jet flights, cross-country train rides, car trips, and frantic wandering through the London subway?

You can? Good! Then you're ready to accompany me on a whirlwind replay of my spring "on the road."

## Big Bird, Blue Jeans, And Blackboards

On March 17th, I joined the COMPUTE! staff and jetted out to San Francisco for the 1983 West Coast Computer Faire. On March 28th, I still hadn't recovered from the crowd, tumult, and heady new products introduced at the Faire. But I packed my bags and flew down to Tampa, Florida, to make a speech at the Florida Instructional Computing Conference. I remember asking the passenger sitting next to me, "Is Tampa on the east coast of Florida or the west coast?"

The week after I returned from Tampa, I hopped aboard another plane and flew up to New York to visit the people at the Children's Television Workshop and the Children's Computer Workshop. CCW and CTW were a treat. It was good to meet relaxed, smiling people dressed in blue jeans and T-shirts. And big fuzzy Cookie Monster, Kermit, and Big Bird dolls were perched on file cabinets and smiled down from colorful posters on the walls.

(You can read about what I learned on these trips in my July 1983 "On the Road" and "World Inside the Computer" columns in COMPUTE!, and in my August "Computing for Grown-Ups" column in COMPUTE!'s Gazette.)

During this phase of my travels I got to see a lot of educational software. My chief impressions were that the software is quickly improving and that its creators are beginning to deal with learning in a totally new manner.

Only a year ago, educational software on personal computers consisted almost entirely of old-fashioned "electronic textbook" programs and drill and practice programs.

Six months later we were besieged by educational game software, really *disguised drill and practice*.

Now we are beginning to see something new. We are seeing the first real microcomputer simulations, where the kid's computer "pretends" it is a world or environment and challenges the child to playact and build a face, conduct a chemistry experiment, pilot a starship, run a nuclear reactor, solve a crime, or map out a new world. Some of the forerunners in such simulation games include the Learning Company's *Gertrude's Puzzles*; Spinnaker's *Facemaker*, *Snooper Troops*, and *In Search of the Most Amazing Thing*; and Children's Computer Workshop's *Electronic Blackboard* game.

*Electronic Blackboard* suggests an even newer type of educational software for children: kids' workstations – where the computer becomes a general-purpose tool to enable children to use the computer to do *whatever they want* (just like adults!).

*Electronic Blackboard* creates an electronic "mailbox" for kids. Several blackboards are pictured on the computer's display screen. At first they are empty. Kids get to "borrow" a blackboard, associate their name with it (as a mail address), and use electronic *chalk* to write messages on the board for other kids to see.

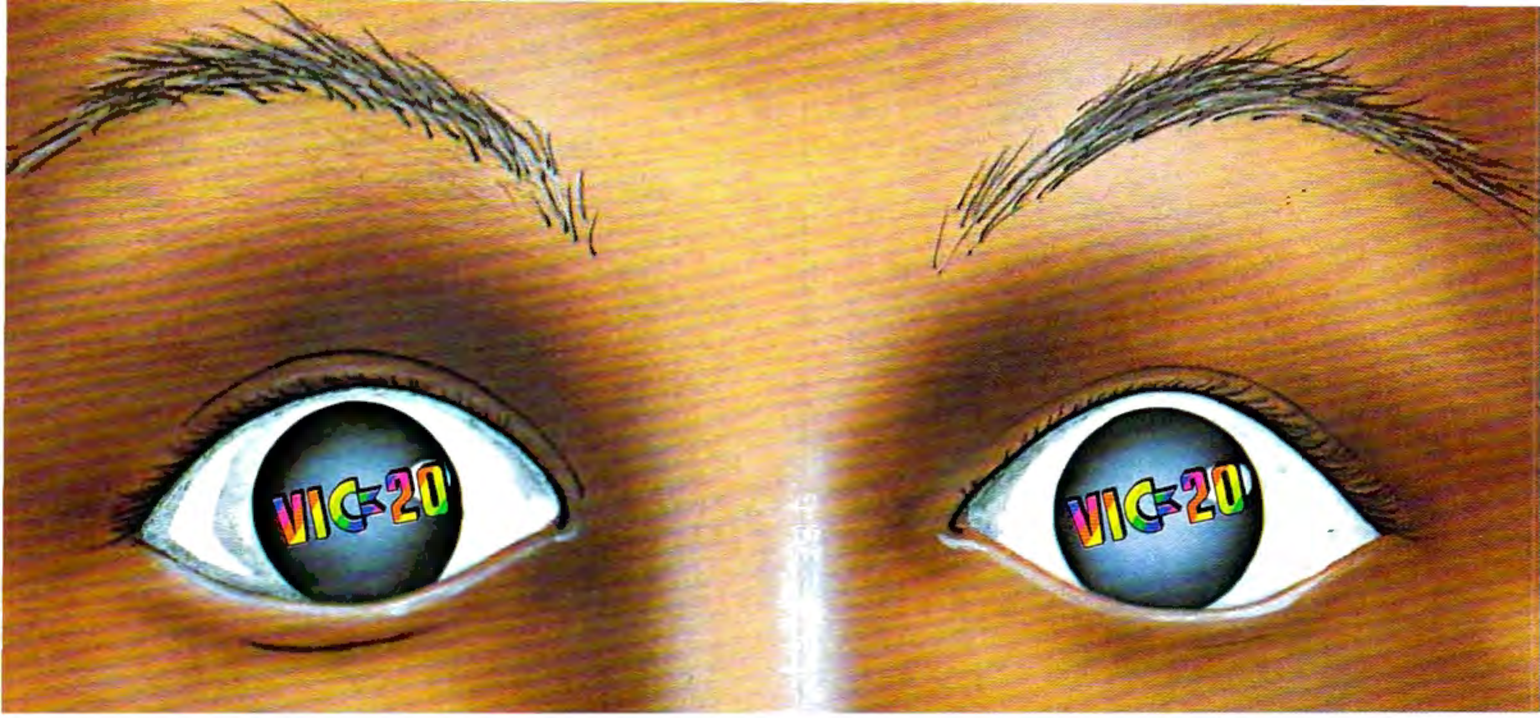
If a message isn't private, you get to see it just by calling up a particular blackboard. If, however, it *is* private, the child can hide it. You can access private messages "for your eyes only" by typing your name. It's not a foolproof security system, but it makes a great educational activity. Kids get to practice their reading and writing skills. And they are learning how to do word processing and send electronic mail.

## All Alone With HERO

Not long after I visited CCW, I flew to Benton Harbor, Michigan, for a first encounter with HERO the robot, made by Heath. After *Star Wars'* C3PO and R2-D2, HERO is probably the third most famous robot in America.

And he is for real.

I noticed this immediately the first time I met




# "YOU WON'T BELIEVE YOUR EYES"

THRESHOLD,™


LUNAR LEEPER,™


CROSSFIRE™

CANNONBALL BLITZ™

## ACTUAL VIC 20 SCREENS

They said it couldn't be done. Boy, were they wrong!


You'll find the best graphics ever available for the Vic 20 in games that are already hits on the Apple and Atari. Think of it! Brilliant graphics and the fastest action this side of reality.

Survive wave after wave - 24 different levels - of intruding aliens in THRESHOLD. Rescue your space pals from the hungry LUNAR LEEPER™ and head for the eye. Save your city from reincarnating invaders in CROSSFIRE.™ Evade a CANNONBALL BLITZ™ to climb through three stages of action to defeat the Redcoats.

Sierra On-Line's assortment of the best, the brightest and the fastest. Believe it!

THRESHOLD,™ LUNAR LEEPER,™ CROSSFIRE™ and CANNONBALL BLITZ™: A whole new look for the Vic 20! Available at your local dealer.

Sierra ON-LINE INC
SIERRA ON-LINE BUILDING COARSEGOLD CA 93614
209 683 6858

him. Doug Bonham of Heath gave me some quick pointers about operating HERO, then he left the two of us alone.

There we were, in a tiny office deep inside Heath's giant manufacturing plant on the outskirts of Benton Harbor. HERO was on a worktable in the rear of the office, propped up at an angle so his drive wheel was slightly off the table (in case I told him to do something foolish).

And I was staring at HERO.

What do I do first? I am itching to get to know HERO – make him walk and talk and do other great things. But I am scared to death that I might get things mixed up and somehow hurt him.

I realize now that I was reliving those first anxious moments experienced by the first-time owner of a personal computer. You desperately want to touch the machine, play with it, make it perform. It doesn't even have to turn cartwheels or play Beethoven's *Fifth*. You would be thrilled if you could make the computer do *anything*.

Yet you are almost frozen by fear. What if you push the wrong button? What if you wipe out a program? What if you damage the machine? What if you do something foolish and silly?

I stood in the little room staring at the buttons on top of HERO's head and glancing at the "teaching pendant" (control box) sitting next to HERO on the table. What should I try first?

I decided that I'd try the safest thing first, something that was guaranteed not to get me into trouble. I would press the "3" button and the "1" button on HERO's keyboard. When HERO received a "31" command, he was supposed to move all his motorized limbs back to their "home" position. Surely this was a trivial and harmless thing to try first.

I pressed "31" and was startled when HERO came to life. His motors started buzzing, his arm rotated, his gripper hand pivoted, his wheels turned, and his head swung from side to side.

Then it happened.

I was just starting to breathe easier when HERO's wheels swiveled around and began banging into a metal plate. Bang! Bang! Bang! went the wheels. HERO's whole body began to rock.

I backed off in total dismay. I glanced fearfully at the door behind me. I was sure that Doug and his staff at Heath heard the racket and were about to rush in and accuse me of breaking their robot.

HERO's wheels kept banging. I leaned over and held onto HERO's shoulders, afraid that he would rock himself off the worktable.

Then he stopped.

"Ready," he said sweetly.

"Ready?" I thought. Then, with a flood of relief, I realized that HERO was okay. All that banging was okay. He was just returning his wheels to their "home" position. I hadn't broken anything. No one came into the room. They were used to HERO making noises like that.

My confidence quickly returned. I spent the next two hours joyfully punching buttons on Hero's head and flipping switches and turning the dial on HERO's teaching pendant. I taught him how to say "Hello, Fred," how to wave, and how to crash into the wall.

That last trick was not what I intended. I had hoped that my program would activate HERO's wheels and navigate him across the floor and out the door. I had planned for him to make a little trip down the hall to say hello to Doug's people.

But, somehow, the door was narrower than I figured. Or else HERO's front drive wheel was a little crooked. In any case, when I pressed the "A" and the "DO" buttons and gave the memory address of my little program, HERO said "Here I go," then marched right into the wall.

## The Hall Of The Dinosaurs

The day after my first encounter with HERO, I rode with Doug Bonham in his car along the shoreline of Lake Michigan to Chicago. Doug was going to check up on Heath's exhibit at the ROBOTS VII conference in giant McCormick Place on the edge of the lake.

After spending several hours with HERO the day before, I thought he was the greatest. With his computer brain and his arm and wheels and motors and sensors, he was a complete, real robot. I expected him to hold his own with all the other robots in McCormick Place, since most of the robots there were not nearly as versatile or advanced. HERO could speak, move, and had an array of "senses," including the ability to detect motion, light, and sound. Most of the other robots were sightless, "dumb" industrial robots, anchored by lugs and rivets to the floor of the factory. How could they compare with a cute, walking, talking robot like HERO?

What a surprise!

When I walked into the mammoth exhibit hall at McCormick Place, I was stunned. I felt like I was in a giant, dreamlike Museum of Natural History, surrounded by prehistoric dinosaurs. Only the dinosaurs were not dead, old bones. Instead they were alive and they moved. And their skin wasn't a cement gray, but red, orange, black, and brilliant yellow – all the colors of the rainbow.

This all sounds melodramatic, but it's true. The robots in McCormick Place were *huge*. Their robotic arms were as long as the neck of a giraffe, or of a brontosaurus. They appeared even taller because they rested on top of six-foot-high metal pedestals.

And they didn't just sit there. They moved

# CPI

## PARALLEL PRINTER INTERFACE FOR VIC-20 & C-64

# GROW WITH US!

## EXPANSION PRODUCTS FOR YOUR MICROCOMPUTER


Lifetime Warranty available upon return of Product Warranty Card.

MSD's CPI Parallel Interface works with either the VIC-20 or Commodore 64 and provides total feature flexibility through software commands or hardware switches.

The CPI plugs into the serial port and directly interprets the signals generated by the computer's built in software, therefore no software needs to be loaded or enabled. All you need is built into the CPI cartridge. This feature allows the CPI to be compatible with most software written for the VIC-20 and Commodore 64 that utilize 1515 or 1525 printers.

The CPI is capable of twelve printing modes, specified by software or hardware switch settings. These twelve printing modes are combinations of three options as follows:

**Line Feed, ASCII Conversion and Listing Legibility.**

Line Feed: The CPI can generate a line feed if needed through software or hardware switches.

ASCII Conversion: The CPI converts Commodore ASCII into standard ASCII characters through software or hardware switches.

Listing Legibility: Since many printers do not support the codes/graphics that the VIC-20 and Commodore 64 produce, program listing can become illegible if not impossible (Printer may "hang-up"). The CPI provides three listing modes to address this problem — Normal, Extended Tag and Abbreviated Tag.

In the Normal mode the CPI passes all commands from the computer to the printer. The Normal code would be used for programs written by the user or programs with commands supported by the printer.

In the Extended Tag mode the CPI will generate "tags" (neumonics) for graphics, cursor control, print control and special characters. For any graphics characters that are not standard ASCII, the decimal value of the graphics symbol is printed. For instance, the "checker board" character (press the Commodore key and the plus sign) lists as [166].

The Abbreviated Tag mode is the same as Extended Tag mode, except all the tags are replaced by the "#" sign. This mode would be used if you wanted a program listing to be formatted as the original program. That is without the "tags" using several print spaces instead of one print space.

The CPI is equipped with a built-in self-test program that will check the RAM, ROM and I/O hardware of its microprocessor. This test can be helpful in determining if something is faulty or if the configuration you are using is valid. The self-test will print information to the printer.

**CPI LISTING LEGIBILITY TAGS**

| TAG | Is Printed For: | TAG | Is Printed For: | TAG | Is Printed For: | TAG | Is Printed For: |
|---|---|---|---|---|---|---|---|
| [CD] | Cursor Down | [DL] | Delete | [ORA] | Change to Orange | [F2] | Function Key 2 |
| [CU] | Cursor Up | [BLK] | Change to Black | [BRN] | Change to Brown | [F3] | Function Key 3 |
| [CL] | Cursor Left | [WHT] | Change to White | [LTR] | Change to Light Red | [F4] | Function Key 4 |
| [CR] | Cursor Right | [RED] | Change to Red | [GY1] | Change to Grey 1 | [F5] | Function Key 5 |
| [HC] | Home Cursor | [CYN] | Change to Cyan | [GY2] | Change to Grey 2 | [F6] | Function Key 6 |
| [CS] | Clear Screen | [PUR] | Change to Purple | [LTG] | Change to Light Green | [F7] | Function Key 7 |
| [RV] | Reverse On | [GRN] | Change to Green | [LTB] | Change to Light Blue | [F8] | Function Key 8 |
| [RO] | Reverse Off | [BLU] | Change to Blue | [GY3] | Change to Gray 3 | [PI] | Pi Symbol |
| [IN] | Insert | [YEL] | Change to Yellow | [F1] | Function Key 1 | | |

Call toll free for nearest dealer

**1-800-527-5285**

**MSD also manufactures RS232 Interfaces, IEEE Interfaces, Port Expanders, RAM Cartridges, Audio Cassette Interface, Terminal Emulators, Disk Drives and Monitor Cables.**

**Win a FREE trip to Hawaii. Join our "Grow With Us" Club. The more you buy... the better your chances. Details at your Dealer.**

MSD
MICRO SYSTEMS DEVELOPMENT, INC.
(214) 241-3743
11105 Shady Trail • Suite 104 • Dallas, Texas 75229

All MSD Products Made In U.S.A.

with frightening, snakelike swiftness and grace. Their movements made them seem alive, conscious, even intelligent. They twisted, gyrated, and whirled in a strange, mechanical dance.

As they moved they made soft noises. Some swished, others whooshed. Some buzzed, others wheezed. Many robots made no sound at all. They moved their enormous arms in great, sweeping arcs. They rotated, opened, and closed their leviathan grippers. Their arms telescoped abruptly to twice their size, or dived to the floor to pick up a cinder block or a paintbrush.

And they made no sound at all.

In the midst of all these dinosaurs sat HERO – two HEROs, actually. He was the same robot as yesterday, but somehow, among all these hulking machines he seemed very different. He was obviously still "all robot," but now he also seemed sensitive, delicate, and fragile.

Whatever, HERO was a tremendous hit. I came back to the Heath booth several times during the day and always found huge crowds of people standing around the two HEROs, watching them perform, and listening to them tell jokes.

I left the ROBOTS VII conference late that afternoon and flew back to Roanoke. I carried with me one chief impression. Before the conference I had thought of robots as all belonging to the same tribe. Now I saw two tribes: the little guys, like HERO; and the big, hulking monster robots that are taking over our factories.

Eventually we'll have robots of all shapes and sizes in our society – not just big robots and little robots. But I think there will still be two different tribes. Then the *programming* will make the difference. Robots in the home will be programmed to be friendly, playful, helpful, and easygoing. Robots in the workplace will be cold, purposeful, and narrow-minded. They won't be programmed to carry on a chat with their human counterparts. Their only mission will be to get the job done. Both types of machines (home and work) will be robots. But they will be two different sorts of creatures entirely.

*Next month Fred and HERO go to London, England, to teach a course on robotics literacy, and they visit a children's educational software company. Fred also meets a computer magician – a British teacher who creates kids' magic shows using computers.* ©

# Mystery Spell

Doug Hapeman

*This spelling game features lively graphics and sprites. It's also a clever teaching aid for parents, teachers, and students in which spelling lessons can be reviewed and then practiced. Originally written for the TI-99/4A with Extended BASIC, there are also versions for the VIC and 64.*

If you've ever played Hangman, you won't have any trouble learning "Mystery Spell." Although it's similar in concept, there's a twist. Instead of a gallows, you'll see colorful balloons, flying blackbirds, cheerful music, and a happy face.

When the game begins, a happy face appears in a little hut surrounded by trees and landscape. The letters of the alphabet appear near the bottom of the screen, and blank spaces representing the secret word appear near the top. When you select a letter, the happy face moves to the selected letter and indicates whether it is an incorrect or correct choice. For each correct choice a colored balloon rises to the appropriate place in the secret word, and the letter is displayed. For each incorrect choice a blackbird descends and lands somewhere on the landscape. Too many blackbirds disallow any more guesses, and the word will be spelled correctly for you.

There are two levels of difficulty: easy, which permits six incorrect guesses, and difficult, allowing only four.

The program has 20 preselected words, or you can choose the "create your own word list" option (and, if you wish, save it to tape or disk). This option allows you to tailor the word difficulty to any learning level.

Many features of the TI-99/4A are used in the program: color, graphics, moving sprites, and music. Let's look at some program features and see how certain graphics results are accomplished in the TI version.

## Screen Centered Printing

There are several locations in the program where variable length words or phrases are centered. Line 170 is an example. For centering text with the DISPLAY AT statement, a simple equation can determine the proper column position:

**column = (14-LEN(L$)/2).**

It's like using a typewriter. When you want to center your title, you find the center of the page and count back one-half the length of the title. Similarly, in TI BASIC you subtract one-half of the length of the string variable from one-half the screen width. Fourteen is one-half the screen width using DISPLAY AT, and 16 is one-half using CALL HCHAR. The length of the string variable is easily determined by the LEN function.

## Moving Sprites

Moving sprites are a fascinating feature of TI Extended BASIC. Through a library of impressive subprograms, sprites can easily be called, defined, magnified, or set in motion, can acknowledge coincidence, change character definition, and so on. Because they are controlled by built-in subprograms, they are easily accessed by even a beginning programmer.

Regular characters are located on the screen in a 32 column by 24 row format, resulting in a total of 768 screen positions. Sprites, however, are located by dot-row and dot-column positions. Where normal characters are each made up of an eight-by-eight grid, sprites, on the other hand, can be located at any one of the 64 dots in the eight-by-eight grid. Therefore, there are 192 dot-rows and 256 dot-columns, for a total of 49,152 screen positions for sprites.

Mystery Spell uses moving sprites in several locations. The balloon and blackbird sprites are called with motion, but the happy face sprite is


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

# 64 Version Notes

Eric Brandon, Programming Assistant

The most interesting feature of the 64 version of "Mystery Spell" (Program 2) is the animated bird. The bird flies around the top of the screen, swooping down to pick up letters and to sit on its perch, depending on whether your guesses are right or wrong.

As the bird moves around, it seems to flap its wings, creating an illusion of flight. This is achieved by rapidly displaying different "poses." In films, this is done by passing many frames through a projector every second. To achieve the illusion of flapping wings, we too must create a few "frames" of a bird in motion.

Using a sprite editor, we first drew the bird you see in Figure 1. Then, using that sprite, we designed two more birds, one with the wing up (Figure 2) and one with the wing down (Figure 3). Using those shapes, we designed three more birds identical to the first three, but without legs. This gave us three "frames" for the bird carrying a letter, and three "frames" for the bird flying freely. We then set up the DATA statements in the program as if we were going to display six different sprites.

Immediately after the screen RAM are eight memory locations that tell the 64 where in memory to find the shapes of the eight sprites. Usually these locations are at 2040 to 2047 ($07F8 to $07FF). By rapidly POKEing 2040 with the pointer to each "frame," the bird seems to flap its wings. To see how this is done, look at lines 2000-2060. This is the routine which flies the bird around the top of the screen until you press a key. Line 2050 steps through the "frame" numbers. The actual POKEing is done at the end of line 2000.

Another interesting feature of the game is that when you guess correctly, the bird swoops down to pick up a letter, and then carries it up to the word. How is that letter incorporated into the bird sprite?

In the character set ROM at 53248 ($D000), the shape of each character is contained in eight bytes. Each byte is one row, and each bit is a column within that row. Depending on whether the value of that bit is 0 or 1, the pixel will be clear or set inside the character. The sprite is 24 bits wide, which is as wide as three characters. This means that by putting character shape data into every third byte within a sprite, we can make character shapes inside sprites. This technique could be used in any program which moves letters or text around smoothly. To see how this is done, look at lines 2180 to 2260.

Lines 2180 and 2190 make the character ROM available to be PEEKed. They also turn off the keyboard. Lines 2200 to 2240 take the character data and put it in the sprites. Finally, lines 2250 and 2260 cover up the character ROM and re-enable the keyboard.

**Figure 1: Sprite-Created Bird**

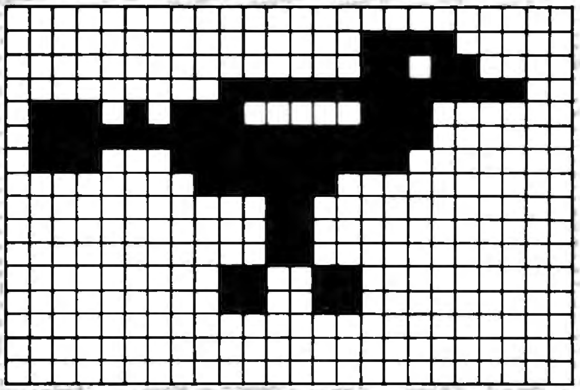


**Figure 2: Bird With Wing Up**

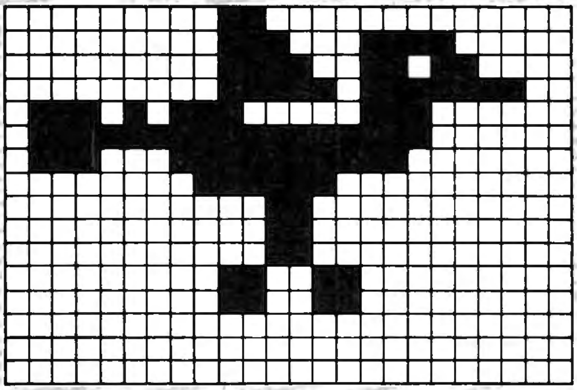
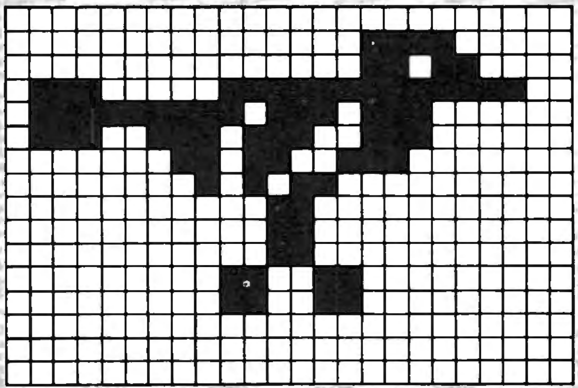


**Figure 3: Bird With Wing Down**



initially stationary. Each time a letter is pressed, it moves to the location of the letter and then back to the hut.

Let's examine just how the happy face movement is achieved. The numeric variables used for determining direction and motion are: R = row, C = column, V = vertical motion, and H = horizontal motion.

The alphabet is displayed on the screen in two neat rows (A-M) and (N-Z). The ASCII (standard computer code) value of the alphabet is 65(A) to 90(Z). In response to the CALL KEY, any other key pressed is ignored. If the letter pressed is less than 78 (the letter N), then the row variable is set for the upper row; otherwise the row variable is set for the lower row (line 350). CALL GCHAR is used to determine whether the letter has previously been chosen (line 360). If not, then the vertical motion is set for downward movement until coincidence is achieved with the row variable – then motion stops (line 390 and subroutine at line 550).

## Another Equation To The Rescue

Knowing which way to move horizontally is determined with another IF-THEN statement (line 400).

Knowing where to stop horizontally presented a more difficult problem. It could have been determined by the process of elimination through a long series of IF-THEN statements. But, once again, an equation can come to the rescue (line 410): C = (K-64)*16 + 4-208*INT ((K-64)/14).

(K-64) gives a number between 1 and 26, depending on which letter has been pressed. It is multiplied by 16, which is two times eight dot-column positions (one for the letter and one for the space). Four is added to center the sprite over the appropriate letter. The last part of the equation 208*INT((K-64)/14) yields either a 0 or 208, and ((K-64)/14) yields a 0 for (A-M) or 1 for (N-Z). The figure 208 represents 26 character positions (13 letters and 13 spaces in each row) times 8 dot positions per character position.

The best way to understand how the equation works is to experiment by placing different K values into it. For example, suppose the letter F was pressed. The ASCII value of F is 70, hence:

C = (70-64) + 4-208*((70-64)/14)
C = 6 + 4-208*6/14
C = 96 + 4-208*0
C = 100 (the dot-column position for F).

## Balloon Motion

The balloon sprite moves from wherever the happy face sprite is located to the appropriate blank in the secret word at the top of the screen. See the "correct guess subroutine" (lines 570-600); you should be able to follow the program logic for balloon direction and motion.

## Program 1:
### Mystery Spell – TI-99 Extended BASIC

```
100 REM  MYSTERY SPELL
120 DIM A$(26),B$(20)
122 ON ERROR 140
125 CALL INIT :: CALL LOAD(-31878,1
    3)
130 REM **INITIALIZATION AND INTROD
    UCTION**
140 DISPLAY AT(12,5)ERASE ALL:"ONE
    MOMENT PLEASE..." :: GOTO 780
150 DISPLAY AT(7,1)ERASE ALL BEEP:"
    PRESS{3 SPACES}FOR": : :"  1  =
      INSTRUCTIONS": :"  2  =  MYST
    ERY SPELL": :"  3  =  FINISH MY
    STERY SPELL"
160 DISPLAY AT(23,3):"PLEASE ENGAGE
     ALPHA LOCK" :: CALL KEY(0,K,S)
    :: IF S=0 OR(K<49 OR K>51)THEN
    160 :: ON K-48 GOTO 980,190,170
170 DISPLAY AT(12,5)ERASE ALL BEEP:
    "THANKS FOR PLAYING," :: DISPLA
    Y AT(14,14-LEN(L$)/2):L$ :: STO
    P
190 DISPLAY AT(7,1)ERASE ALL BEEP:"
    CHOOSE A WORD LIST:": : :"  A
    =  PRESELECTED WORDS": :"  B  =
      CREATE YOUR OWN"
200 CALL KEY(0,K,S):: IF S=0 OR(K<6
    5 OR K>66)THEN 200 :: IF K=66 T
    HEN 220
210 PSW=1 :: GOTO 230
220 PSW=0 :: GOTO 1620
230 CALL CLEAR :: RESTORE 940 :: GO
    TO 930
240 CALL SPRITE(#2,120,2,78,121,0,0
    ):: CALL MAGNIFY(3):: CALL SPRI
    TE(#4,136,16,8,128,0,1):: CALL
    SPRITE(#3,140,2,8,128,0,-2)
250 DISPLAY AT(5,9):"MYSTERY SPELL"
     :: T=200 :: GOSUB 1050 :: IF P
    SW=1 THEN GOTO 1840
260 DISPLAY AT(19,1)BEEP:" WHAT IS
    YOUR NAME, PLEASE?" :: DISPLAY
    AT(23,1):"TYPE NAME, THEN PRESS
     ENTER"
270 ACCEPT AT(5,9)SIZE(14):L$ :: CA
    LL HCHAR(5,7,32,22)
280 REM **MAIN PROGRAM LOOP**
290 DISPLAY AT(19,1)BEEP:"  CHOOSE
    THE LEVEL OF PLAY"
300 DISPLAY AT(23,1):"{3 SPACES}1)
    EASY{3 SPACES}2) DIFFICULT" ::
    CALL KEY(0,K,S):: IF S=0 OR K>5
    0 OR K<49 THEN 300 :: IF K=49 T
    HEN ER=7 ELSE ER=5
310 FOR SP=5 TO 13 :: CALL DELSPRIT
    E(#SP):: NEXT SP
320 DISPLAY AT(19,1):" A B C D E F
    G H I J K L M" :: DISPLAY AT(23
    ,1)BEEP:" N O P Q R S T U V W X
     Y Z" :: RANDOMIZE
330 CALL HCHAR(5,3,32,28):: W$=B$(I
    NT(20*RND)+1):: F=LEN(W$):: FOR
     I=1 TO F :: DISPLAY AT(5,2*I+1
    4-F):"_" :: NEXT I :: Y=0 :: SP
    =13
340 CALL KEY(0,K,S):: IF S=0 OR(K<6
    5 OR K>90)THEN 340 ELSE C=121
```

```
350 IF K<78 THEN R=128 ELSE R=160
360 CC=((K-64)*16+16-208*INT((K-64)
    /14))/8 :: CALL GCHAR((R+24)/8,
    CC,X):: IF X=32 THEN 370 ELSE 3
    90
370 DISPLAY AT(16,14-(8+LEN(L$))/2)
    SIZE(8+LEN(L$))BEEP:" OOPS, ";L
    $;"," :: DISPLAY AT(17,1):" YOU
     TRIED THAT ONE ALREADY"
380 FOR D=1 TO 500 :: NEXT D :: CAL
    L HCHAR(16,1,33,64):: GOTO 340
390 V=12 :: H=0 :: GOSUB 550
400 IF K<72 OR(K>77 AND K<85)THEN H
    =-12 ELSE H=12
410 V=0 :: C=(K-64)*16+4-208*INT((K
    -64)/14):: GOSUB 550
420 X=0 :: CALL HCHAR((R+24)/8,(C+1
    2)/8,32):: FOR I=1 TO F :: IF A
    SC(SEG$(W$,I,1))<>K THEN 450
430 CALL PATTERN(#2,124):: GOSUB 58
    0
440 CALL PATTERN(#2,120):: DISPLAY
    AT(5,2*I+14-F)SIZE(-1):CHR$(K):
    : X=1 :: Y=Y+1
450 NEXT I :: IF X=1 THEN 470
460 CALL PATTERN(#2,128):: GOSUB 62
    0 :: CALL PATTERN(#2,120)
470 H=-H :: C=121 :: GOSUB 550
480 V=-12 :: H=0 :: R=78 :: GOSUB 5
    50
490 IF Y=LEN(W$)THEN GOSUB 740 ELSE
     500 :: GOTO 510
500 IF ER=1 THEN GOSUB 710 ELSE 340
510 DISPLAY AT(23,1)BEEP:"
    {5 SPACES}ANOTHER WORD? (Y/N)"
520 CALL KEY(0,K,S):: IF S=0 OR K<>
    89 AND K<>78 THEN 520 :: IF K=8
    9 THEN 290
530 CALL DELSPRITE(ALL):: GOTO 150
540 REM **SUB TO MOVE HAPPY FACE**
550 CALL MOTION(#2,V,H)
560 CALL COINC(#2,R,C,4,Z):: IF Z=0
     THEN 560 ELSE CALL MOTION(#2,0
    ,0):: CALL LOCATE(#2,R,C):: RET
    URN
570 REM **SUB FOR CORRECT GUESS**
580 B=8*(2*I+14-F):: CALL SPRITE(#1
    ,132,14,R,C,(32-R)/8,(B-C)/8)
590 J=2^(1/12):: FOR A=1 TO 25 :: C
    ALL SOUND(-40,220*J^A,1):: NEXT
     A
600 CALL COINC(#1,32,B,6,Z):: IF Z=
    0 THEN 600 ELSE CALL DELSPRITE(
    #1):: RETURN
610 REM **SUB FOR INCORRECT GUESS**
620 SP=SP-1 :: ER=ER-1 :: IF ER>4 T
    HEN RR=80 ELSE RR=50
630 IF ER=6 OR ER=4 THEN C=52
640 IF ER=5 OR ER=1 THEN C=188
650 IF ER=3 THEN C=110
660 IF ER=2 THEN C=132
670 CALL SPRITE(#SP,140,2,1,120,(RR
    -1)/8,(C-120)/8)
680 J=2^(1/12):: FOR A=25 TO 1 STEP
     -1 :: CALL SOUND(-40,440*J^A,1
    ):: NEXT A
690 CALL COINC(#SP,RR,C,6,Z):: IF Z
    =0 THEN 690 ELSE CALL MOTION(#S
    P,0,0):: CALL LOCATE(#SP,RR,C):
    : CALL PATTERN(#SP,100):: RETURN
700 REM **SUB FOR BLACKBIRDS WIN**
710 CALL HCHAR(19,3,32,28):: DISPLA
    Y AT(19,15-(8+LEN(L$))/2):"SORR
    Y, ";L$;","
720 DISPLAY AT(23,1)BEEP:"THE BLACK
    BIRDS WIN THIS TIME" :: GOSUB 7
    60 :: RETURN
730 REM **SUB FOR PLAYER WINS**
740 CALL HCHAR(19,3,32,28):: DISPLA
    Y AT(19,15-(8+LEN(L$))/2):"GREA
    T, ";L$;","
750 DISPLAY AT(23,1):"{3 SPACES}THA
    T'S THE SECRET WORD"
760 CALL HCHAR(5,3,32,28):: FOR I=1
     TO F :: DISPLAY AT(5,2*I+14-F)
    :SEG$(W$,I,1):: NEXT I :: T=180
     :: GOSUB 1050 :: RETURN
770 REM **ASSIGN COLORS AND DEFINE
    CHARACTERS**
780 FOR I=0 TO 9 :: CALL COLOR(I,2,
    8):: NEXT I :: CALL COLOR(10,3,
    8):: CALL COLOR(11,11,8):: CALL
     COLOR(1,13,8)
800 FOR I=1 TO 25 :: READ C,A$(I)::
     CALL CHAR(C,A$(I)):: NEXT I ::
     CALL SCREEN(15):: GOTO 150
810 DATA 112,C0C0C0C0C0C0C0C0,113,0
    303030303030303,114,FFFFFFFFFF
    FFFFF,115,C0C0C0FFFFC0C0C0,116,
    030303FFFF030303
820 DATA 105,183C3C7E7EFFFFFF,106,F
    FFFFFFFFFFFFFFF,107,FFFFFF7E7E3
    C3C18,108,071F7FFFFF7F1F07,109,
    C0F0FEFFFFFEF0C0
830 DATA 96,000000000030F3FFF,97,FFF
    FFFFFFFFFFFFF,98,FFFFEFC78383C7
    EF,99,7F3F1E3C78FCFEFF,33,FFFFF
    FFFFFFFFFFF
840 DATA 91,1F3F7FFFFFFFFFFF,92,F8F
    CFEFFFFFFFFFF,93,0000000C0F0FC
    FF
850 DATA 120,071820404C888081808884
    4340201807E0180C0232110181011120
    1C2020418E0
860 DATA 124,071820404C888081809F90
    4844231807E0180C023211018101F90
    91222C418E0
870 DATA 128,071820404C888081808384
    4840201807E0180C023211018101C12
    112020418E0
880 DATA 132,030F1F3F3F3F1F0F0703
    010102040880E0F0F8F8F8F0E0C08
    0
890 DATA 136,030F3F7F7F3FFFFFFFFF3F
    7F7F37070100C0CCFEFEFCFFFFFFFFF
    CFEFEECE080
900 DATA 140,000000000000183D478301
    000000000000000000000018BCE2C18
    00000000000
910 DATA 100,0001010100010303070707
    0301000101C0E0F0D0C0E0F0F0F8F8F
    8F0E0C02020
920 REM **PRINT SCREEN**
930 CALL HCHAR(16,1,33,288):: FOR I
    =1 TO 21 :: READ R,C,G$ :: DISP
    LAY AT(R,C)SIZE(-6):G$ :: NEXT
    I :: GOTO 240
940 DATA 9,12,`[aa\],10,12,qrrrrp,1
    1,12,qrrrrp,12,12,qrrrrp,13,14,
    st,14,14,st,15,14,st
```

# SOFTWARE YOU CAN COUNT ON...


**PractiCalc 20™ ‡ and PractiCalc Plus™ ‡:** Complete electronic spreadsheets that turn the Commodore VIC-20 into a business computer. ($39.95* TD and $49.95* TD, respectively.)

**PractiCalc 64™:** The computer spreadsheet for the Commodore 64 with over 20 mathematical functions and the ability to graph, sort, and search for entries. ($49.95* TD)

**PS: The Programmable Spreadsheet™:** Finally, a computer spreadsheet which can handle the most complicated operations within the structure of a spreadsheet – since *you* can program it with BASIC. Available for the Commodore 64 ($79.95 D) & Apple IIe ($79.95 D).

**Rabbit Base™ ‡:** A data-file manager for the Commodore VIC-20 with simple screen instructions for efficient use. ($29.95 T)

**Inventory 64™:** A smart inventory-tracking system for the Commodore 64 that handles 650 parts. ($39.95 D)

**C-64 Analyst:** A diagnostic program which tests the Commodore 64 and its peripherals to detect hardware defects. An invaluable tool for C-64 users! ($19.95 D)

† 8K RAM required – ‡ 16K RAM required
* Price given for tape version. Disk version slightly higher
T Available on tape – D Available on disk
Prices shown are manufacturer's retail prices.

**Total Health™:** For fitness and health enthusiasts, a program which monitors and encourages proper nutrition. (For the Commodore VIC-20; $24.95 T and C-64 $29.95 D)

## AND LEARN FROM . . .

**Math Duel™:** A math program for ages 5-12 that combines classroom learning with gameroom fun! Available for the Commodore VIC-20. ($19.95 T)

**Sprintyper™:** A typing tutorial for the Commodore VIC-20 that encourages speed and accuracy in both the novice and experienced typist. ($19.95 T)

**Tiny Tutor™:** A pre-schooler program with fun graphics and sound to teach simple math. ($19.95 T)

**Composer™:** A simple music composition program for the Commodore VIC-20 that teaches musical notation and allows 'melodies' to be saved to tape for later recall. ($19.95 T)

**COMPUTER SOFTWARE ASSOCIATES**

50 Teed Drive, Randolph,
Massachusetts 02368

## AND PLAY WITH . . .

**Zeppelin Rescue™:** An intelligent rescue game for the Commodore 64 with arcade-like graphics and the greatest challenge for those with persistence & skill. ($24.95 D)

**Skramble™:** You're lost in enemy territory. But before leaving for home, you can play havoc with their airfields and oil supply. Are you ready for the challenge? (For the Commodore VIC-20; $19.95 T)

**Barrel Jumper™ †:** For the Commodore VIC-20, this game confronts you with a pyramid of steel girders. The present King of the Hill is an angry ape who's hurling barrels at you. Step lively! ($19.95 T)

See your local dealer for CSA programs or order directly by calling toll-free:

**1-800-343-1078**

For more information about these and many other programs for your home computer, write to CSA.

Programmers with programs to market are encouraged to send copies for review to CSA.

Dealer & distributor inquiries are welcomed by:

*Micro Software International Inc*

The Silk Mill 44 Oak Street
Newton Upper Falls, Massachusetts 02164

# computer mail order

## PRINTERS

### EPSON

| | |
|---|---|
| MX80 MX80 FT MX100 | CALL |
| RX80 | CALL |
| FX80, FX100 | CALL |

### OKIDATA

| | |
|---|---|
| 82, 83, 84 | CALL |
| 92, 93 | CALL |

### STAR

| | |
|---|---|
| Gemini 10X | $299.00 |
| Gemini 15 | $479.00 |
| Serial Board | $75.00 |

### SMITH CORONA

| | |
|---|---|
| TP-1 | $499.00 |
| TP-2 | CALL |
| Tractor Feed | $129.00 |

### C.ITOH

| | |
|---|---|
| Gorilla | $209.00 |
| Prowriter 8510P | $379.00 |
| Prowriter 1550P | $689.00 |
| Starwriter F10 40P | $1149.00 |
| Printmaster F10-55P | $1599.00 |
| Tractor Feed | $109.00 |

### DAISYWRITER

| | |
|---|---|
| 2000 Letter Quality | $1149.00 |
| 2500 "NEW" | CALL |
| Tractor Feed | $109.00 |

### DIABLO

| | |
|---|---|
| 620 | $949.00 |
| 630 | $1769.00 |

### IDS

Call for ALL Configurations on IDS PRISM PRINTERS

### NEC

| | |
|---|---|
| 8023 | $399.00 |
| 7710/7730 | $2149.00 |
| 3510/3530 | $1549.00 |

## CABLES & CONNECTIONS

### PRINTER CABLES

| | |
|---|---|
| Atari to Parallel | $29.00 |
| Atari To Serial | $29.00 |
| Apple to Parallel | $89.00 |
| Apple ot Parallel/Graphics | $99.00 |
| Apple to Serial | $89.00 |
| IBM to Parallel | $35.00 |
| IBM to Serial | $29.00 |
| Parallel to Parallel | $29.00 |
| Serial to Serial | $29.00 |
| Grappler Plus | $129.00 |
| PKASO | $139.00 |
| Atari to Modem Cable | $29.00 |
| CBM 64 to IEEE Board | $79.00 |
| Apple 80 Column Card | $159.00 |

## HEWLETT PACKARD

**HP41CV... $209.00**

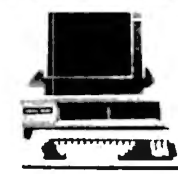
**HP 75... $749.00**

| | |
|---|---|
| HP 41C | $146.00 |
| HP 10C | $52.00 |
| HP 11C | $89.00 |
| HP 12C | $92.00 |
| HP 15C | $92.00 |
| HP 16C | $92.00 |

For HP41/41Cv

| | |
|---|---|
| HPIL Module | $99.00 |
| HPIL Cassette or Printer | $359.00 |
| Card Reader | $144.00 |
| Extended Functions Module | $64.00 |
| Time Module | $64.00 |

## EAGLE



### EAGLE

| | |
|---|---|
| IIE 1 | $1369.00 |
| IIE 2 | $1649.00 |
| IIE 3 | $2399.00 |
| IIE 4 | $3199.00 |
| PC-E | $1579.00 |
| PC 1 | $2399.00 |
| PC 2 | $2799.00 |
| PC XL | $3599.00 |
| 1620 | $3599.00 |
| 1630 | $5499.00 |
| 1640 | $6499.00 |

## MONITORS

### AMDEK

| | |
|---|---|
| 300G | $149.00 |
| 300A | $159.00 |
| 310A | $169.00 |
| Color I | $279.00 |
| Color I plus | $299.00 |
| Color II | $399.00 |
| Color III | $349.00 |
| Color IV | $999.00 |

### USI

| | |
|---|---|
| Pi 1 9 G | $99.00 |
| Pi 2 12 G | $119.00 |
| Pi 3, 12 A | $159.00 |
| Pi 4, 9 A | $139.00 |
| 1400 Color | $299.00 |

### ZENITH

| | |
|---|---|
| ZVM 121 | $95.00 |
| ZT1 Terminal | $369.00 |

### BMC

| | |
|---|---|
| 12 Green | $85.00 |
| 9191 13 Color | $299.00 |

### TAXAN

| | |
|---|---|
| 12 N Green | $129.00 |
| 12 A Amber | $139.00 |

### PANASONIC

| | |
|---|---|
| TR 120 Hi-res Green | $159.00 |
| CT 160 Dual Mode Color | $299.00 |

### NEC

| | |
|---|---|
| JB 1260 | $119.00 |
| JB 1201 | $149.00 |
| JC 1212 | $299.00 |
| JC 12-202 | $299.00 |
| JC 1203 | $549.00 |

### GORILLA

| | |
|---|---|
| 12" Green | $89.00 |

## TIMEX BINCLAIR 1000

$39.95

| | |
|---|---|
| 16K Memory | $44.95 |
| 2040 Printer | $99.95 |
| Vu Calc | $17.95 |
| Check Book Manager | $13.95 |
| Organizer | $14.95 |
| Budgeter | $13.95 |
| Stock Option | $14.95 |
| Loan & Mortage Amortizer | $12.95 |
| Mindware Printer | $109.00 |

## IBM

NEC 3550 PRINTER... $1799

### PERCOM/TANDOM

#### DRIVE

| | |
|---|---|
| 5¼ 320K Floppy | $279.00 |
| 10 Meg Hard | $1495.00 |

### AMDEK

| | |
|---|---|
| 310A Amber Monitor | $169.00 |
| DXY Plotter | $599.00 |
| Color II | $399.00 |

### AST

| | |
|---|---|
| Combo Plus | CALL |
| Mega Plus | CALL |
| Mega Pack | CALL |
| I/O Plus | CALL |

### PROFESSIONAL SOFTWARE

| | |
|---|---|
| PC Plus Word Processing | $319.00 |

### MICRO PRO

| | |
|---|---|
| Word Star/Mail Merge | $319.00 |
| InfoStar | $299.00 |
| Spell Star | $159.00 |
| CalcStar | $159.00 |

### MICROSTUF

| | |
|---|---|
| Crosstalk | $139.00 |

### MICROSOFT

| | |
|---|---|
| Multiplan | $199.00 |

### ASHTON-TATE

| | |
|---|---|
| D-Base II | $449.00 |

### IUS

| | |
|---|---|
| EasyWriter II | $209.00 |
| EasySpeller | $129.00 |
| EasyFiler | $129.00 |

### CONTINENTAL SOFTWARE

| | |
|---|---|
| 1st Class Mail/Form Letter | $89.00 |
| The Home Accountant Plus | $109.00 |

### SYNAPSE

| | |
|---|---|
| File Manager | $119.00 |

### PFS

| | APPLE | IBM |
|---|---|---|
| File | $89.00 | $99.00 |
| Report | $69.00 | $89.00 |
| Graph | $89.00 | $99.00 |
| Write | n/a | $89.00 |

### VISICORP

FOR APPLE IBM & FRANKLIN

| | |
|---|---|
| Visidex | $189.00 |
| Visifile | $189.00 |
| Visiplot | $159.00 |
| Visiterm | $89.00 |
| Visitrend/Plot | $229.00 |
| VisiSchedule | $229.00 |
| Desktop Plan | $189.00 |
| Visicalc (Apple, CBM, IBM) | $179.00 |

Visicorp prices for IBM may vary slightly

## SHARP

### PC-1500 POCKET COMPUTER

**$169.**

**PC1250.... $89.00**

| | |
|---|---|
| CE-150 Printer, Plotter & Cassette | |
| Interface for 1500 | $172.00 |
| CE 125 Printer/Micro Cassette | |
| For 1250 | $129.00 |
| CE 152 Cassette Recorder | $62.00 |
| CE 155 8K Ram | $94.00 |
| CE 158 8K Ram Battery | $129.00 |

## SANYO

| | |
|---|---|
| MBC-555 | $795.00 |
| MBC-1000 | $1599.00 |
| MBC-160 Drive | $539.00 |
| PR5500 Letter Quality Printer | $699.00 |

## MODEMS

### HAYES

| | |
|---|---|
| Smart | $219.00 |
| Smart 1200 (1200 Baud) | $519.00 |
| Chronograph | $199.00 |
| Micromodem 100 | $309.00 |
| Micromodem II | $279.00 |
| Micromodem II (with term) | $299.00 |
| Smart Com II | $99.00 |
| Smart 1200B | $469.00 |

### NOVATION

| | |
|---|---|
| J Cat | $119.00 |
| Cat | $144.00 |
| D-Cat | $159.00 |
| 103 Smart Cat | $189.00 |
| Apple Cat II | $279.00 |
| 103 212 Smart Cat | $439.00 |
| 212 Apple Cat II | $609.00 |
| Apple Cat II 212 Upgrade | $309.00 |

### ANCHOR

| | |
|---|---|
| Mark I (RS 232) | $79.00 |
| Mark II (Atari) | $79.00 |
| Mark III (T I 99) | $109.00 |
| Mark IV (CBM-PET) | $125.00 |
| Mark V (Osborne) | $95.00 |
| Mark VI (IBM PC) | $179.00 |
| Mark VII (Auto Ans Auto Dial) | $99.00 |
| Mark VIII (1200 Baud) | $269.00 |
| TRS 80 Color Computer | $99.00 |
| 9 Volt Power Supply | $9.00 |

## READY FORMS

| | |
|---|---|
| 1"or2" Address Labels (Tract. Feed) | $9.95 |
| 15" Report Paper (Tract. Feed) | $24.95 |
| 8½" Blnk Wht Paper (Tract. Feed) | $19.95 |
| 8½" Blnk Env (Tract. Feed) | $14.95 |

## TeleVideo

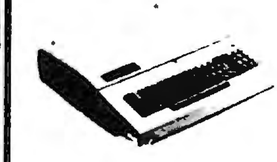
### TERMINALS

| | |
|---|---|
| 910 | $559.00 |
| 912 | $689.00 |
| 920 | $739.00 |
| 925 | $719.00 |
| 950 | $929.00 |
| 970 | CALL |

### COMPUTERS

| | |
|---|---|
| 800A | $1259.00 |
| 802 | $2199.00 |
| 803 | $1949.00 |
| 802H | $4695.00 |
| 806/20 | $4949.00 |
| 816/40 | $8999.00 |
| 1602 | $3399.00 |
| 1603 | CALL |

## FRANKLIN



Call for Price & Information on Franklin 1000, 1100, 1200 and other NEW Franklin Hardware & Software & Special System Pricing

## MICRO-SCI

### Apple & Franklin

| | |
|---|---|
| A2 | $249.00 |
| A40 | $349.00 |
| A70 | $459.00 |
| C2 Controller | $79.00 |
| C47 Controller | $89.00 |

### RANA

| | |
|---|---|
| Elite I (Apple/Franklin) | $279.00 |
| Elite II (Apple/Franklin) | CALL |
| Elite III (Apple/Franklin) | CALL |
| 1000 (Atari) | CALL |

## MEMORY

### MPC

| | |
|---|---|
| Bubdisk (128K Non Voltare) | $649.00 |

### AXLON

| | |
|---|---|
| Apple/Franklin 128K Ram | $299.00 |
| Apple/Franklin Ram Disk | $729.00 |

## INFOCOM

| | |
|---|---|
| Deadline (AP, IBM, AT, &C-64) | $35.00 |
| Star Cross | $29.00 |
| Zork I, II or III | $29.00 |

### BRODERBUND

| | |
|---|---|
| Apple Panic | $23.00 |
| David's Magic | $27.00 |
| Star Blazer | $25.00 |
| Arcade Machine | $34.00 |
| Choplifter | $27.00 |
| Serpentine | $27.00 |

### SIRIUS

| | |
|---|---|
| Bandits | $28.00 |
| Beer Run | $24.00 |
| Free Fall | $24.00 |
| Sneakers | $24.00 |
| Snake Byte | $24.00 |
| Fast Eddie (Atari) | $21.00 |
| Turmoil (Atari) | $21.00 |
| Deadly Duck (VIC) | $21.00 |

## AMDEK DISK DRIVES

Amdisk I, 3" Mini Disk for Apple II & IIE... $229.00

## FLOPPY DISKS

### maxell

| | |
|---|---|
| MD-1 (Box of 10) | $32.00 |
| MD-2 (Box of 10) | $44.00 |
| FD-1 (8 ) | $40.00 |
| FD-2 (8 DD) | $50.00 |

### ELEPHANT

| | |
|---|---|
| 5¼ SS SD | $18.95 |
| 5¼ SS DD | $24.95 |
| 5¼ DS DD | $29.95 |

### VERBATUM

| | |
|---|---|
| 5¼ SS DD | $26.00 |
| 5¼ DS DD | $36.00 |

### HEAD

| | |
|---|---|
| Disk Head Cleaner | $14.95 |

# computer mail order west

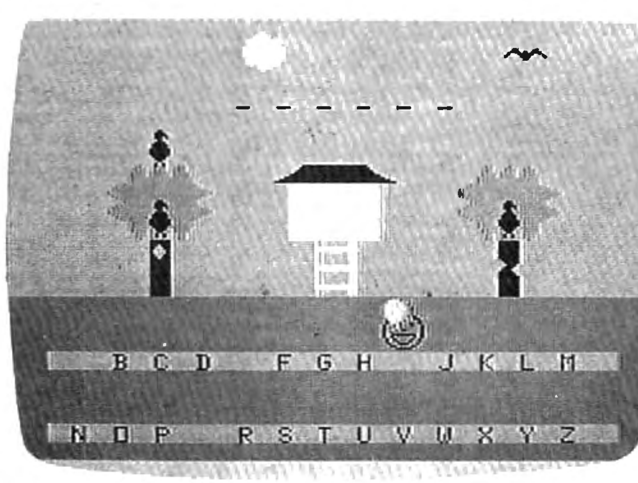
MasterCard VISA

# 800-648-3311

In NV call (702)588-5654, Dept. 906, P.O. Box 6689, Stateline, NV 89449

No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the UPS Continental United States with no waiting period for certifiec checks or money orders. Add 3% (minimum $3.00) shipping and handling on all C.O.D. and credit card orders. Larger shipments may require additiona charges. NV and PA residents add sales tax. All items subject to availability and price change. NOTE: We stock manufacturer's and third party software fo most all computers on the market. Call today for our new catalog.

```
950 DATA 9,5,iii,10,4,ljjjm,11,4,ljjjm,12,5,kjk,13,6,b,14,6,a,15,6,a
960 DATA 9,22,iii,10,21,ljjjm,11,21,ljjjm,12,22,kjk,13,23,a,14,23,c,15,23,a
970 REM **INSTRUCTIONS**
980 DISPLAY AT(1,8)ERASE ALL:"MYSTERY SPELL": :"{3 SPACES}THE OBJECT OF MYSTERY":"SPELL IS TO GUESS THE SECRET":"WORD."
990 DISPLAY AT(6,4):"WHEN YOU PRESS A LETTER,":"THE HAPPY FACE WILL MOVE TO":"THE SELECTED LETTER AND LET":"YOU KNOW WHETHER YOU MADE A"
1000 DISPLAY AT(10,1):"RIGHT OR WRONG CHOICE.":"{3 SPACES}A CORRECT CHOICE LAUNCHES":"A BALLOON.  AN INCORRECT ONE":"CAUSES A BLACKBIRD TO LAND."
1010 DISPLAY AT(14,1):"IF TOO MANY BLACKBIRDS LAND,":"YOU WILL LOSE THE GAME.": :"{3 SPACES}THERE ARE TWO LEVELS:"
1020 DISPLAY AT(19,1)BEEP:"EASY)  PERMITS 6 INCORRECT":"{7 SPACES}GUESSES.": :"HARD) PERMITS ONLY 4."
1030 DISPLAY AT(24,6):"**PRESS ANY KEY**" :: CALL KEY(0,K,S):: IF S=0 THEN 1030 ELSE 190
1040 REM **SUB FOR BLACKBIRD FLIGHT AND THEME MELODY**
1050 R=8 :: FOR SP=5 TO 13 :: C=INT(RND*240)+1 :: CALL SPRITE(#SP,140,2,R,C,0,12):: R=R+12 :: NEXT SP
1060 CALL SOUND(T,175,0)
1070 CALL SOUND(T,349,0,175,2)
1080 CALL SOUND(T,587,0,175,2)
1090 CALL SOUND(2*T,523,0,440,1,175,2)
1100 CALL SOUND(T,587,0,175,2)
1110 CALL SOUND(2*T,523,0,440,1,185,2)
1120 CALL SOUND(T,196,0)
1130 CALL SOUND(T,330,0,196,2)
1140 CALL SOUND(T,587,0,196,2)
1150 CALL SOUND(2*T,523,0,466,1,196,2)
1160 CALL SOUND(T,587,0,196,2)
1170 CALL SOUND(2*T,523,0,466,1,208,2)
1180 CALL SOUND(T,220,2)
1190 CALL SOUND(T,523,0,440,1,220,2)
1200 CALL SOUND(T,311,2)
1210 CALL SOUND(T,523,0,440,1,311,2)
1220 CALL SOUND(T,294,2)
1230 CALL SOUND(T,494,0,415,1,294,2)
1240 CALL SOUND(T,277,2)
1250 CALL SOUND(T,466,0,392,1,277,2)
1260 CALL SOUND(T,440,0,262,2)
1270 CALL SOUND(T,523,0,262,2)
1280 CALL SOUND(T,587,0,247,2)
1290 CALL SOUND(T,698,0,247,2)
1300 CALL SOUND(2*T,659,0,233,2)
1310 CALL SOUND(2*T,784,0,659,1,131,2)
1320 CALL SOUND(T,880,0,175,2)
1330 CALL SOUND(T,831,0,175,2)
1340 CALL SOUND(T,880,0,262,2,349,2)
1350 CALL SOUND(T,1047,0,262,2,349,2)
1360 CALL SOUND(T,1047,0,220,2)
1370 CALL SOUND(T,880,0,220,2)
1380 CALL SOUND(T,784,0,262,2,349,2)
1390 CALL SOUND(T,698,0,262,2,349,2)
1400 CALL SOUND(T,784,0,233,2)
1410 CALL SOUND(T,698,0,233,2)
1420 CALL SOUND(T,587,0,294,2,349,2)
1430 CALL SOUND(T,698,0,294,2,349,2)
1440 CALL SOUND(T,698,0,220,2)
1450 CALL SOUND(T,587,0,220,2)
1460 CALL SOUND(T,523,0,262,2,349,2)
1470 CALL SOUND(T,440,0,262,2,349,2)
1480 CALL SOUND(T,392,0,247,2)
1490 CALL SOUND(T,784,0,247,2)
1500 CALL SOUND(T,698,0,294,2,349,2)
1510 CALL SOUND(T,659,0,294,2,349,2)
1520 CALL SOUND(T,587,0,196,2)
1530 CALL SOUND(T,440,0,196,2)
1540 CALL SOUND(T,440,0,233,2,349,2)
1550 CALL SOUND(T,494,0,233,2,349,2)
1560 CALL SOUND(T,523,0,175,2,220,2)
1570 CALL SOUND(T,587,0,175,2,220,2)
1580 CALL SOUND(2*T,659,0,262,2)
1590 CALL SOUND(3*T,698,0,262,2,175,0)
1600 FOR I=1 TO 30 STEP 2 :: CALL S
```




*A letter is successfully chosen in the TI version of "Mystery Spell."*

```
     OUND(-T,698,I,262,I,175,I):: N
     EXT I :: RETURN
161Ø REM **CREATE A WORD LIST**
162Ø DISPLAY AT(1,4)ERASE ALL:"WORD
      LIST INSTRUCTIONS": : :"  IN
     THIS SEGMENT YOU MAY":"EITHER
     CREATE A WORD LIST"
163Ø DISPLAY AT(6,1):"OR LOAD AN EX
     ISTING ONE FROM":"A STORAGE DE
     VICE.": : :"  WHEN CREATING A
     WORD LIST,":"TYPE EACH WORD, T
     HEN PRESS"
164Ø DISPLAY AT(12,1):"ENTER.  MAXI
     MUM WORD LENGTH":"IS 13 CHARAC
     TERS.  2Ø WORDS":"MUST BE ENTE
     RED FOR EACH OF":"THE WORD LIS
     TS CREATED."
165Ø DISPLAY AT(18,3)BEEP:"AS YOU E
     NTER EACH LIST,":"YOU MAY SAVE
      IT TO A STORAGE":"DEVICE FOR
     FUTURE USE WITH":"MYSTERY SPEL
     L."
166Ø DISPLAY AT(24,6):"**PRESS ANY
     KEY**" :: CALL KEY(Ø,K,S):: IF
      S=Ø THEN 166Ø
167Ø DISPLAY AT(7,1)ERASE ALL BEEP:
     "PRESS{3 SPACES}TO": : :"  1
     =  CREATE A WORD LIST": :"  2
      =  LOAD A WORD LIST": :"  3
     =  EXIT"
168Ø CALL KEY(Ø,K,S):: IF S=Ø OR(K<
     49 OR K>51)THEN 168Ø :: J=Ø ::
      ON K-48 GOTO 169Ø,1795,19Ø

169Ø DISPLAY AT(1,5)ERASE ALL:"ENTE
     R THE WORD LIST:"
17ØØ I=1 :: C=1 :: FOR A=1 TO 2 ::
     R=3 :: FOR Z=1 TO 1Ø
171Ø ACCEPT AT(R,C)SIZE(-13)BEEP:B$
     (I):: R=R+2 :: I=I+1 :: NEXT Z
      :: C=15 :: NEXT A
172Ø DISPLAY AT(24,1)BEEP:"CORRECT
     OR CHANGE ANY? (Y/N)"
173Ø CALL KEY(Ø,K,S):: IF S=Ø OR K<
     >89 AND K<>78 THEN 173Ø :: IF
     K=89 THEN 17ØØ :: J=1 :: GOTO
     1795
174Ø FOR I=1 TO 2Ø :: PRINT #1:B$(I
     ):: NEXT I :: CLOSE #1 :: GOTO
      23Ø
175Ø FOR I=1 TO 2Ø :: INPUT #1:B$(I
     ):: NEXT I :: CLOSE #1
176Ø DISPLAY AT(11,6)ERASE ALL BEEP
     :"DO YOU WISH TO SEE": :"
     {4 SPACES}THE WORD LIST? (Y/N)
     "
177Ø CALL KEY(Ø,K,S):: IF S=Ø OR(K<
     >89 AND K<>78)THEN 177Ø :: IF
     K=89 THEN 178Ø ELSE 23Ø
178Ø DISPLAY AT(1,1Ø)ERASE ALL BEEP
     :"WORD LIST" :: R=3 :: FOR I=1
      TO 2Ø STEP 2 :: DISPLAY AT(R,
     1):B$(I),B$(I+1):: R=R+2 :: NE
     XT I
179Ø DISPLAY AT(24,1):"PRESS ANY KE
     Y WHEN FINISHED" :: CALL KEY(Ø
     ,K,S):: IF S=Ø THEN 179Ø ELSE
     23Ø
1795 ON ERROR 1795
18ØØ DISPLAY AT(5,6)ERASE ALL BEEP:
     "WHAT IS THE NAME": :"  OF YOU
     R STORAGE DEVICE?": :"(EXAMPLE
     : CS1 OR DSK1.WORDS)"
181Ø DISPLAY AT(23,1):"PLACE TAPE O
     R DISK IN DEVICE" :: ACCEPT AT
     (11,3):F$ :: OPEN #1:F$,INTERN
     AL,UPDATE,FIXED 5Ø
182Ø IF J=Ø THEN 175Ø ELSE 174Ø
183Ø REM **PRESELECTED WORD LIST**
184Ø FOR I=1 TO 2Ø :: READ B$(I)::
     NEXT I :: GOTO 26Ø
185Ø DATA BANANAS,CARROTS,RHUBARB,C
     ABBAGE,TURNIP,BEANS,CORN,CELER
     Y,WATERMELON,ORANGES,APPLES,PE
     ACHES
186Ø DATA MUSHROOMS,ONIONS,POTATOES
     ,TOMATOES,GRAPES,PUMPKIN,SQUAS
     H,LEMONS
```

## Program 2: Mystery Spell – 64 Version

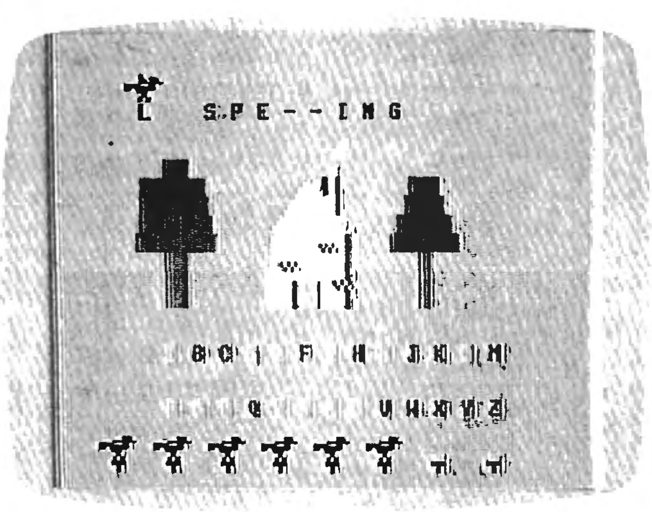
by Eric Brandon, Programming Assistant

```
1ØØ GOSUB 266Ø
11Ø X=RND(-TI)
12Ø DIM W(2Ø),W$(5ØØ)
13Ø GOSUB 119Ø :REM DRAW HOUSE
14Ø PRINT"{HOME}{BLU}PLEASE WAIT...
15Ø GOSUB 138Ø :REM POKE IN SPRITES
16Ø GOSUB 197Ø :REM GET WORDS
17Ø GOSUB 69Ø{2 SPACES}:REM SET UP SPRIT
    ES
18Ø PRINT"{HOME}{14 SPACES}"
19Ø W$=W$(RND(1)*N+1)
2ØØ GOSUB 65Ø
21Ø L$=" ABCDEFGHIJKLMNOPQRSTUVWXYZ"
22Ø PRINT"{HOME}{17 DOWN}{8 RIGHT}";
23Ø FOR I=2 TO 14
24Ø PRINTMID$(L$,I,1)"{RIGHT}";
25Ø NEXT
26Ø PRINT:PRINT"{DOWN}{8 RIGHT}";
27Ø FOR I=15TO 27
28Ø PRINTMID$(L$,I,1)"{RIGHT}";
29Ø NEXT
3ØØ PRINT"{HOME}{4 DOWN}"SPC(18-LEN(G$))
    ;
31Ø FOR I=1TO LEN(G$)
32Ø PRINTMID$(G$,I,1)"{RIGHT}";
33Ø NEXT
34Ø IF COUNT<>LEN(W$) THEN42Ø
35Ø POKE 198,Ø
36Ø FOR DL=1TO1ØØ:NEXTDL:CL=CL+1:IFCL=3T
    HENCL=1
37Ø PRINTMID$("{BLK}{CYN}",CL,1);
38Ø PRINT"{HOME}{14 SPACES}YOU WIN !!!!"
39Ø GETA$:IFA$=""THEN 36Ø
4ØØ GOTO 261Ø
41Ø GOSUB 2ØØØ
42Ø GETA$:IFA$<"A"ORA$>"Z"ANDA$<>"←"THE
    N41Ø
43Ø IF A$="←"THEN 76Ø
44Ø P=ASC(A$)-64
45Ø IF MID$(L$,P+1,1)<>" "THEN54Ø
46Ø PRINT"{HOME}{4 DOWN}{8 SPACES}LETTER
     ALREADY CHOSEN{1Ø SPACES}"
47Ø FOR I=1 TO 8ØØ:NEXTI
48Ø PRINT"{HOME}{4 DOWN}{38 SPACES}"
49Ø PRINT"{HOME}{4 DOWN}"SPC(18-LEN(G$))
    ;
5ØØ FOR I=1TO LEN(G$)
51Ø PRINTMID$(G$,I,1)"{RIGHT}";
```

```
520 NEXT
530 GOTO 420
540 L$=LEFT$(L$,P)+" "+MID$(L$,P+2)
550 RF=Ø :REM FLAG FOR CORRECT GUESS
560 FOR I=1 TO LEN(W$)
570 IF MID$(W$,I,1)<>A$ THEN 610
580 G$=LEFT$(G$,I)+MID$(W$,I,1)+MID$(G$,
    I+2)
590 RF=RF+1
600 COUNT=COUNT+1
610 NEXT I
620 IF RF=Ø THEN GOSUB 1030
630 IF RF THEN GOSUB 2070
640 GOTO 220
650 G$=" "
660 FOR I=1 TO LEN(W$):G$=G$+"-":W(I)=Ø:
    NEXT
670 RETURN
680 I=I+1:GOTO1980
690 REM SET UP SPRITES
700 V=53248
710 FOR I=Ø TO 15:POKE V+I,Ø:NEXT
720 POKE V+21,255
730 FOR I=V+39 TO V+46:POKE I,Ø:NEXT
740 X=Ø:Y=6Ø:S=251
750 RETURN
760 PRINT"{HOME}{BLU}ENTER YOUR GUESS: "
    ;
770 POKE V+21,PEEK(V+21)AND254
780 FOR I=1 TO LEN(W$):PRINT"Ɛ@Ǝ";:NEX
    T
790 PRINT"{HOME}{18 RIGHT}";GU$;
800 IF LEN(GU$)<LEN(W$)THENPRINT"Ɛ+Ǝ";
810 IF LEN(GU$)<LEN(W$)-1 THEN FOR I=2 T
    O LEN(W$)-LEN(GU$):PRINT"Ɛ@Ǝ";
820 GET K$:IF K$=""THEN 820
830 IF K$=CHR$(20) AND LEN(GU$)>Ø THEN G
    U$=LEFT$(GU$,LEN(GU$)-1):GOTO790
840 IF K$=CHR$(13) AND LEN(GU$)=LEN(W$)
    THEN 870
850 IF K$>="A" AND K$<="Z" AND LEN(GU$)<
    LEN(W$) THEN GU$=GU$+K$
860 GOTO 790
870 IF GU$<>W$ THEN 930
880 PRINT"{HOME}{38 SPACES}"
890 PRINT"{HOME}{4 DOWN}"SPC(18-LEN(" "+
    W$));
900 FOR I=1TO LEN(" "+W$)
910 PRINTMID$(" "+W$,I,1)"{RIGHT}";
920 NEXT:GOTO350
930 PRINT"{HOME}{BLK}{13 SPACES}SORRY...
     YOU LOSE{5 SPACES}"
940 PRINT"{BLK}THE WORD WAS ..."
950 PRINT"{HOME}{4 DOWN}"SPC(18-LEN(" "+
    W$));
960 FOR I=1TO LEN(" "+W$)
970 PRINTMID$(" "+W$,I,1)"{RIGHT}";
980 FOR D=1 TO 200:NEXT
990 NEXT
1000 POKE 198,Ø
1010 GETA$:IFA$=""THEN1010
1020 GOTO 2610
1030 DB=DB+1:S=S-3
1040 DX=32*DB+16:DY=225
1050 IF DB=8 THEN DB=Ø
1060 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKE V+1,Y:POKE2040,
     S
1070 IF X=Ø THEN POKE V+21,PEEK(V+21)OR1
1080 FLAG=Ø
1090 IFABS(X-DX)>1THENX=X+3:FLAG=1:IFX>3
     44THEN X=Ø:POKEV+21,PEEK(V+21)AND25
     4
1100 IF Y<DY THEN Y=Y+2:FLAG=1
1110 S=S+1:IFS=251THENS=248
1120 IF FLAG THEN 1060
1130 X=DX:Y=DY
1140 POKEV+2*DB,XAND255:POKEV+16,PEEK(V+
     16)OR(2↑DB)*(-(X>255))
1150 POKEV+2*DB+1,Y:POKE2040+DB,254
1160 IF DB<>Ø THEN POKE V+21,PEEK(V+21)A
     ND254
1170 X=Ø:Y=6Ø:IF DB=Ø THEN 930
1180 RETURN
1190 POKE 53281,3:POKE 53280,4
1200 PRINT"{CYN}{CLR}
1210 PRINT"{4 DOWN}
1220 PRINT
1230 PRINT"{5 SPACES}{GRN}{3 SPACES}
     {RVS}{2 SPACES}{OFF}{10 SPACES}
     {WHT}ƐDƎ{UP}{RVS}ƐBƎ{OFF}{DOWN}
     {6 SPACES}{GRN}
1240 PRINT"{6 SPACES}{RVS}ƐKƎ
     {4 SPACES}{OFF}ƐJƎ{6 SPACES}{RVS}
     {YEL}£Ɛ*Ǝ{BLK}{OFF}Ɛ2 GƎ
     {3 SPACES}{GRN} {RVS}ƐJƎ ƐLƎ
     {OFF}
1250 PRINT"{6 SPACES}{RVS}ƐJƎ
     {4 SPACES}ƐLƎ{OFF}{5 SPACES}{RVS}
     {YEL}£{2 SPACES}Ɛ*Ǝ{OFF}{BLK}
     ƐGƎ{3 SPACES}{GRN} {RVS}
     {3 SPACES}{OFF}
1260 PRINT"{6 SPACES}{RVS}ƐGƎ
     {4 SPACES}ƐNƎ{OFF}{4 SPACES}{RVS}
     {YEL}£{4 SPACES}Ɛ*Ǝ{OFF}{GRN}
     {3 SPACES}{RVS}ƐJƎ{3 SPACES}ƐLƎ
     {OFF}
1270 PRINT"{6 SPACES}{RVS}{6 SPACES}
     {OFF}{4 SPACES}{RVS}{RED}{4 SPACES}
     Ɛ£Ǝ {OFF}{GRN}{3 SPACES}{RVS}
     {5 SPACES}{OFF}
1280 PRINT"{6 SPACES}Ɛ5Ǝ{2 SPACES}
     {RVS}{2 SPACES}{OFF}{6 SPACES}{RVS}
     {RED} Ɛ£Ǝ{4 SPACES}{OFF}
     {2 SPACES}{GRN}{3 SPACES}{RVS}Ɛ5Ǝ
      {OFF}
1290 PRINT"{RVS}Ɛ6Ǝ{8 SPACES}Ɛ5Ǝ
     {2 SPACES}Ɛ6Ǝ{6 SPACES}{RED}
     {2 SPACES}ƐIƎƐFƎ Ɛ£ƎƐ6Ǝ
     {5 SPACES}Ɛ5Ǝ Ɛ6Ǝ{12 SPACES}";
1300 PRINT"{8 SPACES}Ɛ5Ǝ{2 SPACES}
     Ɛ6Ǝ{6 SPACES}{RED}{2 SPACES}{OFF}
      {RVS}ƐKƎ{2 SPACES}Ɛ6Ǝ
     {5 SPACES}Ɛ5Ǝ Ɛ6Ǝ{12 SPACES}";
1310 PRINT"Ɛ6Ǝ{RVS}";
1320 FOR I=Ø TO 8:PRINT"{40 SPACES}";:NE
     XT
1330 FOR I=1 TO 8 : L=1024+23*40+I*4 :PO
     KE L,114:POKEL+54272,Ø:NEXT
1340 FOR I=Ø TO 39:POKE 1024+24*40+I,160
     :POKE 55296+24*40+I,13:NEXT
1350 PRINT"{HOME}
1360 PRINT"{BLK}
1370 RETURN
1380 I=15872:IFPEEK(I+1)=96THENFORI=1TO6
     4*6+2:READA:NEXT:RETURN
1390 READ A:IF A=256 THEN 1410
1400 POKE I,A:I=I+1:GOTO 1390
1410 FOR I=Ø TO 63:POKE 254*64+I,PEEK(24
     9*64+I):NEXT:RETURN
```

```
1420 DATA 0,96,0,0,113,224,0
1430 DATA 121,176,0,125,252,117,193
1440 DATA 192,127,255,192,113,255,128
1450 DATA 0,252,0,0,24,0,0
1460 DATA 24,0,0,102,0,0,102
1470 DATA 0,0,0,0,0,0,0
1480 DATA 0,0,0,0,0,0,0
1490 DATA 0,0,0,0,0,0,0
1500 DATA 0,0,0,0,0,0,0
1510 DATA 0,0,0,0,0,1,224
1520 DATA 0,1,176,0,127,252,117
1530 DATA 193,192,127,255,192,113,255
1540 DATA 128,0,252,0,0,24,0
1550 DATA 0,24,0,0,102,0,0
1560 DATA 102,0,0,0,0,0,0
1570 DATA 0,0,0,0,0,0,0
1580 DATA 0,0,0,0,0,0,0
1590 DATA 0,0,0,0,0,0,0
1600 DATA 0,0,0,0,0,0,1
1610 DATA 224,0,1,176,112,127,252
1620 DATA 127,221,192,115,185,192,1
1630 DATA 179,128,0,172,0,0,24
1640 DATA 0,0,24,0,0,102,0
1650 DATA 0,102,0,0,0,0,0
1660 DATA 0,0,0,0,0,0,0
1670 DATA 0,0,0,0,0,0,0
1680 DATA 0,0,0,0,0,0,0
1690 DATA 0,0,0,0,96,0,0
1700 DATA 113,224,0,121,176,0,125
1710 DATA 252,117,193,192,127,255,192
1720 DATA 113,255,128,0,252,0,0
1730 DATA 0,0,0,0,0,0,0
1740 DATA 0,0,0,0,0,0,0
1750 DATA 0,0,0,0,0,0,0
1760 DATA 0,0,0,0,0,0,0
1770 DATA 0,0,0,0,0,0,0
1780 DATA 0,0,0,0,0,0,0
1790 DATA 0,1,224,0,1,176,0
1800 DATA 127,252,117,193,192,127,255
1810 DATA 192,113,255,128,0,252,0
1820 DATA 0,0,0,0,0,0,0
1830 DATA 0,0,0,0,0,0,0
1840 DATA 0,0,0,0,0,0,0
1850 DATA 0,0,0,0,0,0,0
1860 DATA 0,0,0,0,0,0,0
1870 DATA 0,0,0,0,0,0,0
1880 DATA 0,0,1,224,0,1,176
1890 DATA 112,127,252,127,221,192,115
1900 DATA 185,192,1,179,128,0,172
1910 DATA 0,0,112,0,0,0,0
1920 DATA 0,0,0,0,0,0,0
1930 DATA 0,0,0,0,0,0,0
1940 DATA 0,0,0,0,0,0,0
1950 DATA 0,0,0,0,0,0,0
1960 DATA 0,0,0,0,0,0,0,256
1970 I=1
1980 READ W$(I):IFW$(I)="*"THENN=I-1:RET
     URN
1990 I=I+1:GOTO1980
2000 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKE V+1,Y:POKE2040,
     S
2010 IF X=0 THEN POKE V+21,PEEK(V+21)OR1
2020 X=X+3:IFX>344 THEN X=0:POKEV+21,PEE
     K(V+21)AND254
2030 Y=Y-1+RND(1)*2:IFY>100THENY=99
2040 IF Y<50 THEN Y=50
2050 S=S+1:IFS=254THENS=251
2060 RETURN
2070 DX=INT(P+13*(P>13))*16+24+40
2080 DY=173+INT(P/14)*24:IF S>250 THEN S
     =S-3
2090 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKEV+1,Y:POKE2040,S
2100 IF X=0 THEN POKE V+21,PEEK(V+21)OR1
2110 FLAG=0
2120 IFABS(X-DX)>2THENX=X+3:FLAG=1:IFX>3
     44THENX=0:POKEV+21,PEEK(V+21)AND254
2130 IF Y<DY THEN Y=Y+2:FLAG=1
2140 S=S+1:IFS=251THENS=248
2150 IF FLAG THEN 2090
2160 X=DX:Y=DY
2170 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKEV+1,Y:POKE2040,2
     49
2180 POKE 56334,PEEK(56334)AND254
2190 POKE 1,PEEK(1)AND251
2200 FOR I=0 TO 7
2210 B=PEEK(53248+8*P+I)
2220 FOR J=248 TO 250
2230 POKE J*64+40+I*3,B
2240 NEXT J,I
2250 POKE 1,PEEK(1)OR4
2260 POKE 56334,PEEK(56334)OR1
2270 PRINT"{HOME}{17 DOWN}{8 RIGHT}";
2280 FOR I=2 TO 14
2290 PRINTMID$(L$,I,1)"{RIGHT}";
2300 NEXT
2310 PRINT:PRINT"{DOWN}{8 RIGHT}";
2320 FOR I=15TO 27
2330 PRINTMID$(L$,I,1)"{RIGHT}";
2340 NEXT
2350 DX=160-8*LEN(G$):DY=69
2360 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKEV+1,Y:POKE2040,S
2370 IF X=0 THEN POKE V+21,PEEK(V+21)OR1
2380 FLAG=0
2390 IFABS(X-DX)>2THENX=X+3:FLAG=1:IFX>3
     44THEN X=0:POKEV+21,PEEK(V+21)AND25
     4
2400 IF Y>DY THEN Y=Y-2:FLAG=1
2410 S=S+1:IFS=251THENS=248
2420 IF FLAG THEN 2360
2430 X=DX:Y=DY
2440 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKEV+1,Y:POKE2040,2
     49
2450 PRINT"{HOME}{4 DOWN}"SPC(18-LEN(G$)
     );
2460 FOR I=1TO LEN(G$)
2470 IF MID$(G$,I,1)=A$ THEN PRINT A$;:R
     F=RF-1:IFRF=0 THEN GOSUB 2560
2480 IF MID$(G$,I,1)<>A$ THEN PRINT"
     {RIGHT}";
2490 PRINT"{RIGHT}";
2500 IF RF=0 THEN I=100:GOTO2540
2510 FOR J=0 TO 15:X=X+1:S=S+1:IFS=251TH
     ENS=248
2520 POKEV,XAND255:POKEV+16,PEEK(V+16)AN
     D254OR-(X>255):POKE2040,S
2530 NEXT J
2540 NEXT I
2550 RETURN
2560 FOR K=0 TO 7
2570 FOR J=248 TO 250
2580 POKE J*64+40+K*3,0
2590 NEXT J,K
2600 RETURN
2610 PRINT"{CLR}{7 DOWN}{BLK}DO YOU WISH
      TO PLAY AGAIN (Y/N) ?"
```

```
2615 POKE V+21,PEEK(V+21)AND254
2620 PRINT"{10 DOWN}YOU MISSED THIS MANY
     :"
2630 GETA$:IFA$<>"N"AND A$<>"Y"THEN2630
2640 IF A$="Y"THENPOKE V+21,0:RUN110
2650 END
2660 POKE 53281,0:POKE 53280,0
2670 PRINT"{CLR}{YEL}{13 SPACES}INSTRUCT
     IONS
2680 PRINT"{2 DOWN}{WHT}{4 SPACES}CHOOSE
      LETTERS TO GUESS THE WORD.
2690 PRINT"{DOWN}IF YOU CHOOSE A WRONG L
     ETTER, THE BIRD
2700 PRINT"{DOWN}WILL LAND ON ITS PERCH.
2710 PRINT"{DOWN}{4 SPACES}WHEN ALL THE
     PERCHES ARE FULL, OR
2720 PRINT"{DOWN}YOU GUESSED THE WORD, T
     HE GAME IS OVER
2730 PRINT"{2 DOWN}{4 SPACES}YOU CAN HIT
      THE "CHR$(34)"<"CHR$(34)" KEY ANY
     TIME TO
2740 PRINT"{DOWN}GUESS THE WORD. IF YOU
     GET IT WRONG,{DOWN}{4 SPACES}YOU LO
     SE.
2750 PRINT"{3 DOWN}{9 RIGHT}{YEL}HIT A K
     EY TO BEGIN"
2760 GETA$:IFA$=""THEN2760
2770 RETURN
2780 DATA HAPPY,BRIDGE,FAMILY,CHILDREN
2790 DATA WINDOW,TRAIN,DWARF,BIRDS
2800 DATA SUPERMAN,CONCERT,PEOPLE,MAGIC
2810 DATA SPACE,SCIENCE,PLANETS,GALAXY,S
     TARS
2820 DATA ROOMS,TEACHER,CHALK,BLACKBOARD
2830 DATA SCREEN,COMPUTER,KEYBOARD,PROGR
     AM
2840 DATA SPELLING,WORDS,COLORS,LETTERS
2850 DATA MARKET,STREETS,SQUARE,TRIANGLE
2860 DATA MOVIE,SPACESHIP,LASER,AIRPLANE
     ,BOAT
2870 DATA STICK,ROCK,PAPER,WIN,PLACE,SHO
     W
2880 DATA CHANNEL,EXECUTIVE,MONEY,SHIRT
2890 DATA QUIET,LOUD,BILLBOARD,YACHT,MOT
     ORCYCLE,*
```

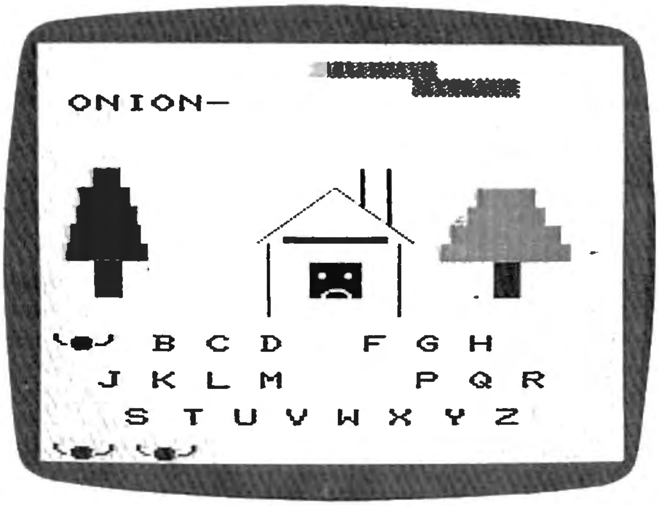


*The bird carries an L to complete the spelling and end the game in "Mystery Spell." 64 version.*

## Program 3: Mystery Spell – VIC Version

by Gregg Peele, Editorial Programmer

```
100 DIMYA$(21):GOSUB1020:DIMW(20)
105 POKE36879,30
110 PRINT"{CYN}{CLR} {DOWN}{9 SPACES}
    {5 +}
120 PRINT"{14 SPACES}{4 +}
130 PRINT"{2 DOWN}
140 PRINT"
150 PRINT"{GRN}{2 SPACES}{RVS} {OFF}
    {8 SPACES}{BLK} {2 G}{2 SPACES}
    {GRN}
160 PRINT" {RVS}{K} {OFF}{J}
    {5 SPACES}{RED} NM{BLK}{2 G} {GRN}
     {RVS}{J} {L}{OFF}
170 PRINT" {RVS}{J} {L}{OFF}
    {4 SPACES}{RED} N{2 SPACES}M{BLK}
    {G} {GRN} {RVS}{3 SPACES}{OFF}
180 PRINT" {RVS}{G} {N}{OFF}
    {3 SPACES}{RED} N{4 O}M{GRN} {RVS}
    {J}{3 SPACES}{L}{OFF}
190 PRINT" {RVS}{3 SPACES}{OFF}
    {3 SPACES}{RED} B{4 SPACES}B{GRN}
    {RVS}{5 SPACES}{OFF}
200 PRINT" {BLK} {RVS} {OFF}{3 SPACES}
    {RED}{2 SPACES}B{4 SPACES}B{GRN}
    {3 SPACES}{RVS}{BLK} {OFF}
210 PRINT"{BLK}{2 SPACES}{RVS} {OFF}
    {3 SPACES}{RED}{2 SPACES}B{4 SPACES}
    B{2 SPACES}{BLK} {RVS} {OFF}
220 PRINT"{6 SPACES}{RED}{2 SPACES}B
    {4 SPACES}B
225 PRINT"{3 SPACES}{RVS}{BLK}MYSTERY
    {2 SPACES}SPELL{OFF}"
230 PRINT"{3 DOWN}";H$
240 T=7680:W=0
250 IFW<22THENT=T+1:IFT>7694ANDT<8000 TH
    ENT=T+21:IFT>7750THENT=T+1
260 W=W+1:IFW>44THENW=0:GOTO250
270 IFW>22THENT=T-20:IFT<7701THEN240
280 T0=PEEK(T):T1=PEEK(T+1):T2=PEEK(T+2)
290 C0=PEEK(T+30720):C1=PEEK(T+30721):C2
    =PEEK(T+30722)
300 POKET,74:POKET+1,81:POKET+2,75
310 POKET+30720,0:POKET+30721,0:POKET+30
    722,0
320 FORD=1TO100:NEXTD
330 POKET,67:POKET+2,67
340 FORD=1TO100:NEXTD
350 POKET,T0:POKET+1,T1:POKET+2,T2
360 POKET+30720,C0:POKET+30721,C1:POKET+
    30722,C2
1000 IFW=30THENGOSUB2000
1010 GOTO250
1020 PRINT"{CLR}ENTER YOUR OWN WORDS
     {2 SPACES}Y OR N"
1030 H$={2 SPACES}"{HOME}{BLK}{12 DOWN}
     {10 RIGHT}{RVS}..{DOWN}{2 LEFT}JK
     {OFF}"
1040 S$={2 SPACES}"{HOME}{BLK}{12 DOWN}
     {10 RIGHT}{RVS}..{DOWN}{2 LEFT}UI
     {OFF}"
1050 GETX$:IFX$<>"N"ANDX$<>"Y"THEN1050
1060 IFX$="N"THEN1080
1070 FORT=1TO20:PRINT"WORD#";T;:INPUTYA$
     (T):NEXT:GOTO1120
1080 FORT=1TO20:READA$:YA$(T)=A$:NEXT
1090 DATA GRAPES,ORANGES,POTATOES,ONIONS
     ,BROCCOLI
```




*The bird swoops down to get the final letter for a correctly spelled word in the VIC version of "Mystery Spell."*

```
1100 DATA BEANS,TOMATO,SPINACH,CUCUMBERS
     ,CARROT,LETTUCE,RADISHES
1110 DATA APPLE,CORN,PEAR,PEACH,GRAPEFRU
     IT,COCONUT,KUMQUAT,BANANA
1120 W$=YA$(RND(1)*20+1):RETURN
2000 FORT=1TO300:NEXT:FORT=8010TO8010+22
     :POKET,32:NEXT
2010 GOSUB 2210
2020 L$=" ABCDEFGHIJKLMNOPQRSTUVWXYZ"
2030 PRINT"{HOME}{16 DOWN}";:FORT=1TO10:
     PRINTMID$(L$,T,1);" ";:NEXT
2040 PRINT"{DOWN}{4 RIGHT}";:FORT=11TO19
     :PRINTMID$(L$,T,1);" ";:NEXT
2050 PRINT"{DOWN}{5 RIGHT}";:FORT=20TO27
     :PRINTMID$(L$,T,1);" ";:NEXT
2060 PRINT"{HOME}{3 DOWN}";G$;"{HOME}
     {22 DOWN}{5 LEFT}";
2062 IFWR>0THENFORY=0TOWR:PRINT"
     {3 RIGHT}";:NEXT:PRINT"{LEFT}JQK";
2065 IFWR=7THENGOTO 5000
2070 IF COUNT<>LEN(W$)THEN2080
2075 FORT=1TO20:PRINT"{HOME}YOU WIN
     {7 LEFT}";:FORO=1TO200
2078 NEXTO:PRINT"{9 SPACES}";:FORZ=1TO20
     0:NEXTZ:NEXTT:PRINTH$;:GOTO5000
2080 PRINT"{HOME}{4 DOWN}{25 SPACES}";:G
     ETA$:IFA$<"A"ORA$>"Z"THEN2030
2090 P=ASC(A$)-64
2100 IF MID$(L$,P+1,1)<>" "THEN2120
2110 PRINT"{HOME}{4 DOWN}LETTER ALREADY
     CHOSEN";:FORT=1TO600:NEXT:GOTO2080
2120 L$=LEFT$(L$,P)+" "+MID$(L$,P+2):PRI
     NTS$:FL=1:GOSUB3000
2130 FOR I=1 TO LEN(W$)
2140 IF MID$(W$,I,1)<> A$ THEN2180
2145 FLAG=0
2150 G$=LEFT$(G$,I)+MID$(W$,I,1)+MID$(G$
     ,I+2)
2160 COUNT=COUNT+1:PRINTHS:GOTO2180
2170 IF MID$(W$,I,1)<>" "ANDMID$(W$,I,1)
     =LEFRT$(A$,1)THENPRINTH$
2180 NEXT I
2185 WR=WR+FL
2190 GOTO 2060
2200 RETURN
2210 G$=" "
2220 FOR I=1 TO LEN(W$):G$=G$+"-":W(I)=0
     :NEXT
2230 RETURN
3000 FORG=7878TO8164STEP22
3010 Y0=PEEK(G):Y1=PEEK(G+1):Y2=PEEK(G+2
     )
3020 Z0=PEEK(G+30720):Z1=PEEK(G+30721):Z
     2=PEEK(G+30722)
3030 POKEG+30720,0{3 SPACES}:POKEG+30721
     ,0:POKEG+30722,0
3040 POKEG,74:POKEG+1,81:POKEG+2,75
3055 FORT=1TO100:NEXT
3060 POKEG,67:POKEG+2,67:FORT=1TO100:NEX
     T
3070 IFG>8160THENRETURN
3090 POKEG,Y0:POKEG+1,Y1:POKEG+2,Y2
3100 POKEG+30720,Z0:POKEG+30721,Z1:POKEG
     +30722,Z2
3115 FORR=GTOG+20
3116 IFPEEK(R)=PTHENM=R
3117 NEXTR
3120 IFM>G+1THENG=G-21:GOTO3126
3125 IFM=G+1THENG=G-22:J=J+1
3126 IFJ=5THENM=0:IFFL=0THENWR=WR+1:J=0:
     RETURN
3127 IFJ=5THENM=0:J=0:RETURN
3128 NEXT
3129 M=0:J=0
3130 NEXT:IFFL=0THENWR=WR+1
3140 RETURN
5000 PRINT"{HOME}THE WORD WAS{HOME}
     {3 DOWN}{RIGHT}";:FORT=1TOLEN(W$):P
     RINTMID$(W$,T,1);:FORU=1TO200:NEXT:
     NEXT
5010 PRINT"{HOME}{5 DOWN}PLAY AGAIN?Y OR
      N";
5011 GETM$:IFM$=""THEN5011
5012 IFM$="Y"THENRUN
5013 IFM$<>"N"THEN5011
5015 PRINT"{CLR}":END
```

©

# DOTS

Eric K. Evans

*Easy to play, but challenging, this game pits you against your computer. You can choose one of ten skill levels. Written for the VIC with versions for the 64, Color Computer, and Apple.*

"Dots" is based on a strategy game that many people first come across while they are in elementary or junior high school. You remember it: you and a friend take a couple of pencils and a piece of paper and draw several rows of dots on the paper. Then you take turns drawing horizontal or vertical lines that would connect two dots. These lines form the sides of squares.

The object of the game is to maneuver your friend into drawing the third side of a square so that you can draw the fourth side and complete the square. Every time you finish a square you get one point and another turn. When all of the squares are finished, the game ends and the person with the most points wins.

With this program, your computer will be the friend you play against and it will also act as umpire. Its decisions will be final, but don't worry – it doesn't cheat.

### How To Play

The first thing you need to do after typing in or loading the game is enter either a J if you plan to use a joystick for input or a K if you are going to use the keyboard cursor control keys and the RETURN key for input. Next you will be asked to enter a skill level between 0 and 10 (with 10 being the most difficult).

If you enter 0 as your skill level, the VIC will play randomly and you should win with little effort. At skill level 10, be prepared for a real strategic challenge. At level 0, the VIC will make its move immediately after yours. As the level of play increases, the VIC's response time increases, too. At level 10, it will usually take between 5 and 20 seconds to make its move.

When the game starts, a 100-dot (ten by ten) game board will be displayed with a scoring area at the bottom of the screen. You always move first. To move, position the yellow cursor where you want to draw a line, and then hit the joystick fire button or the RETURN key to enter your move. When the VIC makes a move, it will beep and flash the line it is going to draw to make sure you see where it is moving.

Whenever you complete a square, the square is colored in cyan and you get another turn. Whenever the VIC completes a square, the square is colored in red and the VIC moves again. These colors are used because they contrast well on black-and-white as well as color TVs. The game continues until all 81 squares have been formed and the winner is declared.

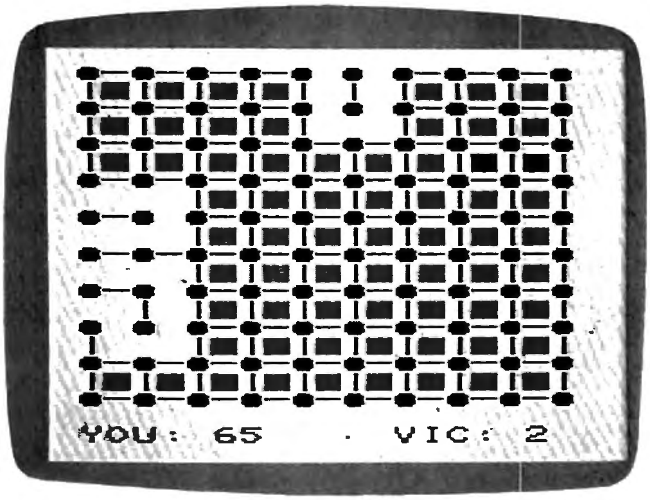
If you don't want to spend the time to type in this program, send me a cassette, a self-addressed, stamped mailer, and $3, and I will return your tape with two copies (just to be safe) of the game on it. *This is only for the VIC version.*

*Eric K. Evans*
*P.O. Box 6287*
*New Haven, CT 06520*

## Color Computer And Apple Version Notes For Dots

The object of Dots on the Color Computer and the Apple is to form more squares than the computer by connecting dots with horizontal and vertical lines. On the Color Computer, the game is played on a 70-dot grid. Move the yellow cursor around the game board with the arrow keys. When it is in a position where you wish to draw a line, press the ENTER key. Squares that you complete will be colored orange; those the computer captures are colored red.

Dots on the Apple features a 100-dot game board and is played like the Color Computer version. A flashing asterisk, moved with the J,I,K, and M keys, indicates your position. Press the RETURN key to draw a connecting line. When a square is formed, an inverse Y or A is displayed, crediting either you or the Apple with the capture.




*An almost completed game of "Dots," VIC version.*

## Program 1: Dots – VIC Version

```
10 PRINT"{CLR}{2 DOWN}"SPC(18)"DOTS
   {DOWN}"
20 INPUT"{6 SPACES}JOYSTICK{DOWN}
   {5 LEFT}OR{DOWN}{8 LEFT}KEYBOARD(J/K)
   ";A$
30 JK=-1:IFA$="K"THENJK=Ø
40 INPUT"{3 DOWN}{RIGHT}SKILL LEVEL(Ø-1Ø
   )";SK:IFSK<ØORSK>1ØTHEN4Ø
50 SK=(1Ø-SK)/1Ø:TS=2ØØ-2ØØ*SK:DT=TS+25:
   DD=37154:POKE37139,Ø:POKE36879,29
60 SC=4*(PEEK(36866)AND128)+64*(PEEK(368
   69)AND12Ø):CO=37888+4*(PEEK(36866)AND
   128)
70 PRINT"{CLR}":FORI=1TO1Ø:PRINT"{BLK} Q
    Q Q Q Q Q Q Q Q Q{DOWN}":NEXT
80 YS=Ø:VS=Ø:PRINT" {CYN}YOU:"YS"
   {4 SPACES}{RED}VIC:"VS
90 DEFFNBX(LC)=(PEEK(LC+22)<>32)+(PEEK(L
   C+1)<>32)+(PEEK(LC-22)<>32)+(PEEK(LC-
   1)<>32)
100 DEFFNVH(LC)=LC<>2*INT(LC/2)
110 SL=SC+23Ø:CL=CO+23Ø:X=1Ø:Y=1Ø:CC=PEE
    K(SL):CR=PEEK(CL)
120 POKESL,16Ø:POKECL,7:F=Ø
130 IFJKTHEN15Ø
140 GOSUB93Ø:GOTO16Ø
150 GOSUB86Ø:IFPC=ØTHEN15Ø
160 X=X+J:Y=Y+K:IFX<1ORX>19ORY<1ORY>19TH
    ENX=X-J:Y=Y-K:GOTO13Ø
170 IFPC=99THEN2ØØ
180 POKESL,CC:POKECL,CR:SL=SL+PC:CL=CL+P
    C:CC=PEEK(SL):POKESL,16Ø
190 CR=PEEK(CL):POKECL,7:GOTO13Ø
200 L=(PEEK(SL+1)=81)+(PEEK(SL-1)=81)+(P
    EEK(SL+22)=81)+(PEEK(SL-22)=81)
210 IFL+(CC=32)=-3THEN23Ø
220 I=128:GOSUB71Ø:GOTO13Ø
230 WH=1:ML=SL:GOSUB65Ø:IFNOTFNVH(SL)THE
    N28Ø
240 IFX>1ANDFNBX(SL-1)=-4THENBX=SL-1:GOS
    UB72Ø:F=-1
250 IFX<19ANDFNBX(SL+1)=-4THENBX=SL+1:GO
    SUB72Ø:GOTO11Ø
260 IFFTHEN11Ø
270 GOTO31Ø
280 IFY>1ANDFNBX(SL-22)=-4THENBX=SL-22:G
    OSUB72Ø:F=-1
290 IFY<19ANDFNBX(SL+22)=-4THENBX=SL+22:
    GOSUB72Ø:GOTO11Ø
300 IFFTHEN11Ø
310 WH=2:F=Ø:CN=Ø:IFRND(Ø)<SKTHEN39Ø
320 FORI=44TO396STEP44:FORJ=2TO18STEP2:K
    =SC+I+J
330 IFPEEK(K)=32ANDFNBX(K)=-3THEN35Ø
340 NEXTJ,I:GOTO39Ø
350 I=K:IFPEEK(I-22)=32THENI=I-22:GOTO54
    Ø
360 IFPEEK(I+22)=32THENI=I+22:GOTO54Ø
370 IFPEEK(I-1)<>32THENI=I+1:GOTO6ØØ
380 IFPEEK(I+1)<>32THENI=I-1:GOTO6ØØ
390 I=INT(RND(Ø)*415+SC+23):CN=CN+1:IFPE
    EK(I)<>32THEN39Ø
400 IFNOT((PEEK(I+1)=81ANDPEEK(I-1)=81)O
    R(PEEK(I+22)=81ANDPEEK(I-22)=81))THE
    N39Ø
410 IFSK>.6ORCN>TSTHEN47Ø
420 IFFNVH(I)THEN45Ø
430 IFFNBX(I-22)=-2ORFNBX(I+22)=-2THEN39
    Ø
440 GOTO54Ø
450 IFFNBX(I-1)=-2ORFNBX(I+1)=-2THEN39Ø
460 GOTO6ØØ
470 IFFNVH(I)THEN51Ø
480 IFSK>.6ORCN>DTTHEN54Ø
490 IFFNBX(I+22)=-2ANDFNBX(I-22)=-2THEN3
    9Ø
500 GOTO54Ø
510 IFSK>.6ORCN>DTTHEN6ØØ
520 IFFNBX(I+1)=-2ANDFNBX(I-1)=-2THEN39Ø
530 GOTO6ØØ
540 ML=I:GOSUB65Ø
550 IFFNBX(ML-22)=-4THENBX=ML-22:GOSUB72
    Ø:F=-1
560 IFFNBX(ML+2?2)=-4THENBX=ML+22:GOSUB72
    Ø:GOTO31Ø
570 IFFTHEN31Ø
580 GOTO11Ø
590 IFNOT(PEEK(I-22)=81ANDPEEK(I+22)=81)
    THEN39Ø
600 ML=I:GOSUB65Ø
610 IFFNBX(ML-1)=-4THENBX=ML-1:GOSUB72Ø:
    F=-1
620 IFFNBX(ML+1)=-4THENBX=ML+1:GOSUB72Ø:
    GOTO31Ø
630 IFFTHEN31Ø
640 GOTO11Ø
650 CL=CO+ML-SC
660 POKEML,67
670 IFFNVH(ML)THENPOKEML,93
680 I=185:IFWH=2THENI=15Ø
690 FORJ=1TOWH:POKECL,Ø:GOSUB71Ø:POKECL,
    1
700 FORL=1TO2ØØ:NEXT:POKECL,Ø:NEXT
710 POKE36878,15:POKE36876,I:FORK=1TO2ØØ
    :NEXT:POKE36878,Ø:POKE36876,Ø:RETURN
720 YS=YS+1:J=3:I=2ØØ:CL=CO+BX-SC:IFWH=2
    THENJ=2:I=15Ø:YS=YS-1:VS=VS+1
730 POKEBX,16Ø:POKECL,1
740 FORL=1TO3:POKECL,J:GOSUB71Ø:POKECL,1
    :FORK=1TO2ØØ:NEXT:I=I+18:POKECL,J:NE
    XT
750 PRINT"{HOME}{21 DOWN} {CYN}YOU:"YS"
    {4 SPACES}{RED}VIC:"VS
760 IFYS+VS<81THENRETURN
```

# MOSES™

## THE ASSEMBLER OF THE AGES
## for the VIC 20™

**Why MOSES?**

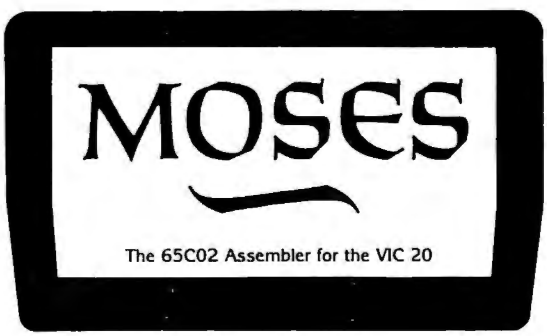
Programs written with MOSES run fifty to several hundred times faster than programs written in BASIC. A program that takes two minutes to execute in BASIC, will only take two seconds (or less) to execute when written with MOSES.

***ULTRA-FAST***
MOSES is written in MACHINE LANGUAGE! MOSES assembles dozens of times faster than assemblers written in BASIC.

***ULTRA-SOPHISTICATED***
MOSES makes 3 PASSES, not just one or two like most other assemblers.

***ULTRA-FRIENDLY***
On screen menu of EASY-TO-USE COMMANDS.

***ULTRA-CONVENIENT***
Comes in a CARTRIDGE with sturdy plastic case. MOSES is easy to learn and powerful. As you enter each line, MOSES verifies syntax, and partially assembles the line thereby reducing debugging time and conserving memory. This means an even faster assembly. Also included with MOSES is a machine language MONITOR with 34 powerful commands to help you debug your programs. Whether you program for fun or profit, or both, you need MOSES. You'll love the ability to program where imagination is your only limit.

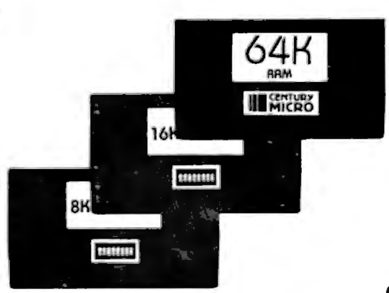



**SPECIAL OFFER!**
For a limited time, we will include a free kit to upgrade your VIC 20's outdated 6502 to the new, enhanced 65C02A microprocessor. The 65C02A is totally compatible with all VIC 20 software and hardware, uses 99% less energy, and has 27 new instructions to make programming easier and faster. And, of course, MOSES takes advantage of all 27 instructions. This is a $24.95 value, but is included free while quantities last.

---

## 8K • 16K
## RAM EXPANSION

A high quality memory expansion cartridge housed in a sturdy plastic case. This product has an easily accessible dip switch which gives you memory block switching ability. Use one 8K and one 16K for 24K memory or two 16K cartridges for a total of 32K memory.
**6-Month Warranty.**




## 64K
## RAM EXPANSION

It's finally here. This highly versatile ram cartridge allows you to store or write programs in two separate 32K banks that are bank selectable with software — NO SWITCHES. This product is power stingy using less than 200 mA current. A must for programmers.
**6-Month Warranty.**

---

## VIC 20 PRODUCTS:

| | | | |
|---|---|---|---|
| MOSES | $59.95* | 8K RAM Cartridge | $46.95* |
| MOSES with 8K RAM expansion | 99.95* | 16K RAM Cartridge | 69.95* |
| | | 64K RAM Cartridge | 169.95* |

*Manufacturer's suggested list price

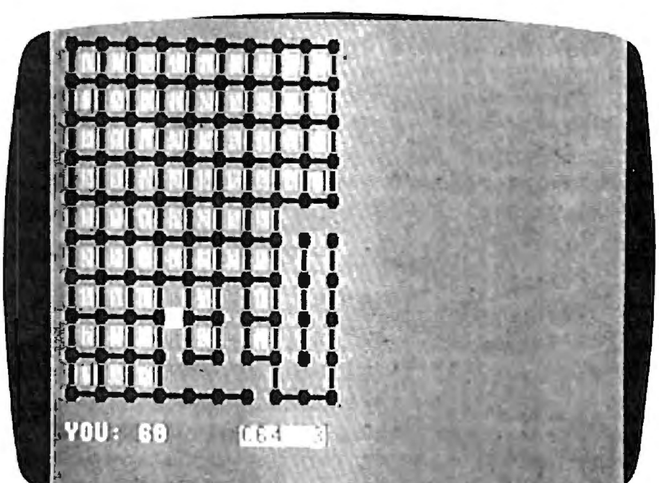
---

Call CENTURY MICRO at (916) 920-3656 for the name of a dealer in your area. Dealer inquiries invited.

1832 Tribute Rd., Suite 213, Sacramento, CA 95815

CENTURY MICRO

```
770 PRINT"{HOME}{19 DOWN}":IFYS>VSTHEN80
    Ø
780 PRINT"{BLK}{7 SPACES}YOU LOSE"
790 POKE36878,15:POKE36877,128:FORI=1TO1
    5ØØ:NEXT:GOTO85Ø
8ØØ PRINT"{DOWN}":FORJ=1TO3
81Ø PRINT"{2 UP}{RVS}{BLU}!{RED}**{BLU}!
    {PUR}{3 SPACES}YOU{2 SPACES}WIN
    {3 SPACES}{BLU}!{RED}**{BLU}!"
82Ø POKE36878,15:FORL=128TO255:POKE36876
    ,L:NEXT:POKE36874,Ø:POKE36878,Ø
83Ø PRINT"{2 UP}{22 SPACES}":FORL=1TO3ØØ
    :NEXT
84Ø PRINT"{2 UP}{RVS}{BLU}!{RED}**{BLU}!
    {PUR}{3 SPACES}YOU{2 SPACES}WIN
    {3 SPACES}{BLU}!{RED}**{BLU}!":NEXT
85Ø POKE36878,Ø:POKE36877,Ø:END
86Ø J=Ø:K=Ø:PC=Ø:POKEDD,127:IFPEEK(37152
    )=119THENPC=1:J=1:POKEDD,255:RETURN
87Ø POKEDD,255:I=PEEK(37137)
88Ø IF(IAND4)=ØTHENPC=-22:K=-1:RETURN
89Ø IF(IAND8)=ØTHENPC=22:K=1:RETURN
9ØØ IF(IAND16)=ØTHENPC=-1:J=-1:RETURN
91Ø IF(IAND32)=ØTHENPC=99
92Ø RETURN
93Ø J=Ø:K=Ø:PC=Ø
94Ø GETA$:IFA$=""THEN94Ø
95Ø IFA$="{UP}"THENPC=-22:K=-1:RETURN
96Ø IFA$="{RIGHT}"THENJ=1:PC=1:RETURN
97Ø IFA$="{DOWN}"THENK=1:PC=22:RETURN
98Ø IFA$="{LEFT}"THENJ=-1:PC=-1:RETURN
99Ø IFASC(A$)=13THENPC=99
1ØØØ RETURN
```




*"Dots," 64 version. The computer has almost lost the game.*

## Program 2: Dots – 64 Version

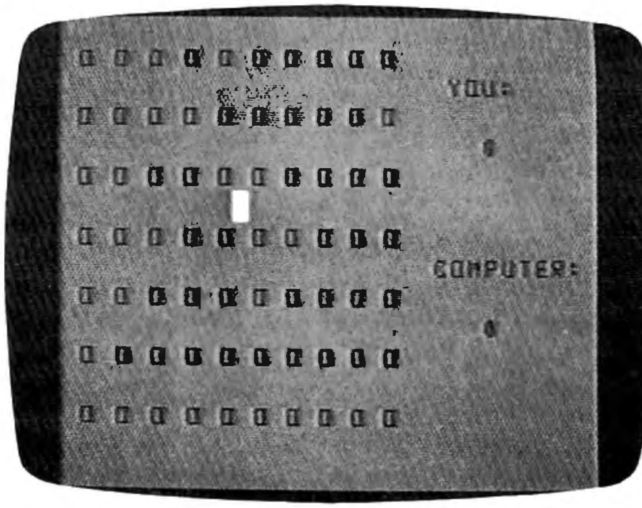
```
5 POKE 53281,12:POKE53280,Ø
1Ø PRINT"{BLK}{RVS}{CLR}{2 DOWN}"SPC(17)
   "DOTS
2Ø PRINT"{2 DOWN}{15 SPACES}{WHT}JOYSTIC
   K{DOWN}{5 LEFT}OR{DOWN}{7 LEFT}KEYBOA
   RD(J/K){SHIFT-SPACE}?";
25 GETA$:IFA$<>"J"ANDA$<>"K"THEN25
26 PRINTA$
3Ø JK=-1:IFA$="K"THENJK=Ø
4Ø INPUT"{3 DOWN}{RIGHT}SKILL LEVEL(Ø-1Ø
   )";SK:IFSK<ØORSK>1ØTHEN4Ø
5Ø SK=(1Ø-SK)/1Ø:TS=2ØØ-2ØØ*SK:DT=TS+25
6Ø SC=1Ø24 : CO = 13*4Ø96+8*256
7Ø PRINT"{CLR}":FORI=1TO1Ø:PRINT"{BLK} Q
   Q Q Q Q Q Q Q Q Q{DOWN}":NEXT
8Ø YS=Ø:VS=Ø:PRINT" {WHT}YOU:"YS"
   {4 SPACES}{RVS}C64:"VS
9Ø DEFFNBX(LC)=(PEEK(LC+4Ø)<>32)+(PEEK(L
   C+1)<>32)+(PEEK(LC-4Ø)<>32)+(PEEK(LC-
   1)<>32)
1ØØ DEFFNVH(LC)=LC<>2*INT(LC/2)
11Ø SL=SC+45Ø:CL=CO+45Ø:X=1Ø:Y=11:CC=PEE
    K(SL):CR=PEEK(CL)
12Ø POKESL,16Ø:POKECL,7:F=Ø
13Ø IFJKTHEN15Ø
14Ø GOSUB93Ø:GOTO16Ø
15Ø GOSUB86Ø:IFPC=ØTHEN15Ø
16Ø X=X+J:Y=Y+K:IFX<1ORX>19ORY<1ORY>19TH
    ENX=X-J:Y=Y-K:GOTO13Ø
17Ø IFPC=99THEN2ØØ
18Ø POKESL,CC:POKECL,CR:SL=SL+PC:CL=CL+P
    C:CC=PEEK(SL):POKESL,16Ø
19Ø CR=PEEK(CL):POKECL,7:GOTO13Ø
2ØØ L=(PEEK(SL+1)=81)+(PEEK(SL-1)=81)+(P
    EEK(SL+4Ø)=81)+(PEEK(SL-4Ø)=81)
21Ø IFL+(CC=32)=-3THEN23Ø
22Ø I=128:GOSUB71Ø:GOTO13Ø
23Ø WH=1:ML=SL:GOSUB65Ø:IFNOTFNVH(SL)THE
    N28Ø
24Ø IFX>1ANDFNBX(SL-1)=-4THENBX=SL-1:GOS
    UB72Ø:F=-1
25Ø IFX<19ANDFNBX(SL+1)=-4THENBX=SL+1:GO
    SUB72Ø:GOTO11Ø
26Ø IFFTHEN11Ø
27Ø GOTO31Ø
28Ø IFY>1ANDFNBX(SL-4Ø)=-4THENBX=SL-4Ø:G
    OSUB72Ø:F=-1
29Ø IFY<19ANDFNBX(SL+4Ø)=-4THENBX=SL+4Ø:
    GOSUB72Ø:GOTO11Ø
3ØØ IFFTHEN11Ø
31Ø WH=2:F=Ø:CN=Ø:IFRND(Ø)<SKTHEN39Ø
32Ø FORI=8ØTO72ØSTEP8Ø:FORJ=2TO18STEP2:K
    =SC+I+J
33Ø IFPEEK(K)=32ANDFNBX(K)=-3THEN35Ø
34Ø NEXTJ,I:GOTO39Ø
35Ø I=K:IFPEEK(I-4Ø)=32THENI=I-4Ø:GOTO54
    Ø
36Ø IFPEEK(I+4Ø)=32THENI=I+4Ø:GOTO54Ø
37Ø IFPEEK(I-1)<>32THENI=I+1:GOTO6ØØ
38Ø IFPEEK(I+1)<>32THENI=I-1:GOTO6ØØ
39Ø I=INT(INT(RND(Ø)*2Ø)*4Ø+21*RND(Ø)+SC
    +41):CN=CN+1:IFPEEK(I)<>32THEN39Ø
4ØØ IFNOT((PEEK(I+1)=81ANDPEEK(I-1)=81)O
    R(PEEK(I+4Ø)=81ANDPEEK(I-4Ø)=81))THE
    N39Ø
41Ø IFSK>.6ORCN>TSTHEN47Ø
42Ø IFFNVH(I)THEN45Ø
43Ø IFFNBX(I-4Ø)=-2ORFNBX(I+4Ø)=-2THEN39
    Ø
44Ø GOTO54Ø
45Ø IFFNBX(I-1)=-2ORFNBX(I+1)=-2THEN39Ø
46Ø GOTO6ØØ
47Ø IFFNVH(I)THEN51Ø
48Ø IFSK>.6ORCN>DTTHEN54Ø
49Ø IFFNBX(I+4Ø)=-2ANDFNBX(I-4Ø)=-2THEN3
    9Ø
5ØØ GOTO54Ø
51Ø IFSK>.6ORCN>DTTHEN6ØØ
52Ø IFFNBX(I+1)=-2ANDFNBX(I-1)=-2THEN39Ø
53Ø GOTO6ØØ
54Ø ML=I:GOSUB65Ø
```

```
550 IFFNBX(ML-40)=-4THENBX=ML-40:GOSUB72
    0:F=-1
560 IFFNBX(ML+40)=-4THENBX=ML+40:GOSUB72
    0:GOTO310
570 IFFTHEN310
580 GOTO110
590 IFNOT(PEEK(I-40)=81ANDPEEK(I+40)=81)
    THEN390
600 ML=I:GOSUB650
610 IFFNBX(ML-1)=-4THENBX=ML-1:GOSUB720:
    F=-1
620 IFFNBX(ML+1)=-4THENBX=ML+1:GOSUB720:
    GOTO310
630 IFFTHEN310
640 GOTO110
650 CL=CO+ML-SC
660 POKEML,67
670 IFFNVH(ML)THENPOKEML,93
680 I=185:IFWH=2THENI=150
690 FORJ=1TOWH:POKECL,0:GOSUB710:POKECL,
    1
700 FORL=1TO200:NEXT:POKECL,0:NEXT
710 FORK=1TO200:NEXT:RETURN
720 YS=YS+1:J=3:I=200:CL=CO+BX-SC:IFWH=2
    THENJ=2:I=150:YS=YS-1:VS=VS+1
730 POKEBX,160:POKECL,1
740 FORL=1TO3:POKECL,J:GOSUB710:POKECL,1
    :FORK=1TO200:NEXT:I=I+18:POKECL,J:NE
    XT
750 PRINT"{HOME}{21 DOWN} {WHT}YOU:"YS"
    {4 SPACES}{RVS}C64:"VS
760 IFYS+VS<81THENRETURN
770 PRINT"{HOME}{19 DOWN}":IFYS>VSTHEN80
    0
780 PRINT"{BLK}{7 SPACES}YOU LOSE
    {7 SPACES}"
790 GOTO850
800 PRINT"{DOWN}YOU WIN !!!"
850 END
860 J=0:K=0:PC=0
870 I=PEEK(56321)
880 IF(IAND1)=0THENPC=-40:K=-1:RETURN
885 IF(IAND8)=0THENJ=1:PC=1:RETURN
890 IF(IAND2)=0THENPC=40:K=1:RETURN
900 IF(IAND4)=0THENPC=-1:J=-1:RETURN
910 IF(IAND16)=0THENPC=99
920 RETURN
930 J=0:K=0:PC=0
940 GETA$:IFA$=""THEN940
950 IFA$="{UP}"THENPC=-40:K=-1:RETURN
960 IFA$="{RIGHT}"THENJ=1:PC=1:RETURN
970 IFA$="{DOWN}"THENK=1:PC=40:RETURN
980 IFA$="{LEFT}"THENJ=-1:PC=-1:RETURN
990 IFASC(A$)=13THENPC=99
1000 RETURN
```

## Program 3: Dots – Color Computer Version

Translation by Patrick Parrish, Editorial Programmer

```
100 CLS 7:PRINT@236,"D O T S";
110 PRINT@357,"SKILL LEVEL (0-10)";
    :INPUT SK:IF SK<0 OR SK>10 THEN
     110
120 CLS:SK=(10-SK)/10:TS=200-200*SK
    :DT=TS+25:SC=1024
130 PRINT:FOR I=1 TO 7:PRINT" O O O
     O O O O O O O":PRINT:NEXT I
140 YS=0:CS=0:PRINT@87,"YOU:":PRINT
     @278,"COMPUTER:"
150 PRINT@152,YS:PRINT@344,CS
```




*No moves have been made in this Color Computer version of "Dots."*

```
160 DEF FNBX(LC)=(PEEK(LC+32)<>96)+
    (PEEK(LC+1)<>96)+(PEEK(LC-32)<>
    96)+(PEEK(LC-1)<>96)
170 DEF FNVH(LC)=(LC<>2*INT(LC/2))
180 SL=SC+202:X=10:Y=7:CC=PEEK(SL)
190 POKE SL,128+16*(2-1)+15:F=0
200 GOSUB 910
210 X=X+J:Y=Y+K:IFX<1ORX>19ORY<2ORY
    >14THEN X=X-J:Y=Y-K:GOTO 200
220 IF PC=99 THEN 250
230 POKE SL,CC:SL=SL+PC:CC=PEEK(SL)
    :POKE SL,128+16*(2-1)+15
240 GOTO 200
250 L=(PEEK(SL+1)=79)+(PEEK(SL-1)=7
    9)+(PEEK(SL+32)=79)+(PEEK(SL-32
    )=79)
260 IF L+(CC=96)=-3 THEN 280
270 I=62:GOSUB 760:GOTO 200
280 WH=1:ML=SL:GOSUB 700:IF NOT FNV
    H(SL)THEN 330
290 IF X>1 AND FNBX(SL-1)=-4 THEN B
    X=SL-1:GOSUB 770:F=-1
300 IF X<19 AND FNBX(SL+1)=-4 THEN
    BX=SL+1:GOSUB 770:GOTO 180
310 IF F THEN 180
320 GOTO 360
330 IF Y>1 AND FNBX(SL-32)=-4 THEN
    BX=SL-32:GOSUB 770:F=-1
340 IF Y<13 AND FNBX(SL+32)=-4 THEN
     BX=SL+32:GOSUB 770:GOTO 180
350 IF F THEN 180
360 WH=2:F=0:CN=0:IF RND(0)<SK THEN
     440
370 FOR I=64 TO 384 STEP 64:FOR J=2
     TO 12 STEP 2:K=SC+I+J
380 IFPEEK(K)=96ANDFNBX(K)=-3 THEN
    400
390 NEXTJ,I:GOTO 440
400 I=K:IF PEEK(I-32)=96 THEN I=I-3
    2:GOTO 590
410 IF PEEK(I+32)=96 THEN I=I+32:GO
    TO 590
420 IF PEEK(I-1)<>96 THEN I=I+1:GOT
    O 650
430 IF PEEK(I+1)<>96 THEN I=I-1:GOT
    O 650
440 I=RND(19)+RND(13)*32+33+SC:CN=C
    N+1:IF PEEK(I)<>96THEN 440
```

# COMMODORE 64

(more power than Apple II at half the price)

## $159.00*

- 170K DISK DRIVE **$199.00***
- TRACTION FRICTION PRINTER **$159.00***

(* with software savings applied)

# COMPUTER AND SOFTWARE SALE

WE HAVE THE BEST SERVICE | WE HAVE THE LOWEST PRICES

# VIC-20

(a real computer at the price of a toy)

## $77.00*

- 40-80 COLUMN BOARD **$99.00**
- VOICE SYNTHESIZER **$69.00**

(* with Cassette and Gortek purchase)

### COMMODORE 64 COMPUTER $159.00

You pay only $259.00 when you order the powerful 84K COMMODORE 64 COMPUTER! PLUS we pack the SPECIAL SOFTWARE COUPON with your computer that allows you to SAVE UP TO $100 off software sale prices!! With these savings applied your net computer cost is only $159.00.

### 170K DISK DRIVE $199.00

You pay only $299.00 when you order the 170K Disk Drive! Plus we pack the SPECIAL SOFTWARE COUPON that allows you to SAVE UP TO $100 off software sale prices!! With these savings applied your net Disk Drive price is only $199.00.

### TRACTION FRICTION PRINTER $159.00

You pay only $259.00 when you order the Comstar F/T deluxe line printer that prints 8 1/2 x 11 full size, single sheet, roll or fan fold paper, labels etc. 40, 66, 80, 132 columns. Impact dot matrix, bi-directional, 80 CPS. PLUS we pack the SPECIAL SOFTWARE COUPON that allows you to SAVE UP TO $100.00 off software sale prices!! With these savings applied your net printer price is only $159.00.

### 80 COLUMN BOARD $159.00

You pay only $159.00 for this 80 Column Board when it is purchased with a COMMODORE 64 Computer or a Disk Drive or a Printer or a Monitor. If purchased alone the sale price is $179.00. Included with this board is word processor pack, electronic spread sheet and mail merge data base on two tapes. List $275.00. (Disk add $10.00).

### EXECUTIVE WORD PROCESSING PACKAGE

SCRIPT 64 EXECUTIVE WORD PROCESSOR is the finest available for the COMMODORE 64 Computer! THE ULTIMATE for PROFESSIONAL wordprocessing application. Featuring simple operation, powerful text editing with a customized word dictionary, complete cursor and insert/delete key controls, line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers. Included is a powerful MAIL MERGE When used with THE COMPLETE DATA BASE PACKAGE. List $99.00. Sale $79.00. Coupon Price $52.00. (Disk only).

### COMPLETE DATABASE PACKAGE

A user friendly data base system that makes information easy to find and store. You can add, change, delete, and search for data. Print the information on a printer in any format desired. When combined with the word processor pack you have a powerful merge program that allows custom documents and personalized mailing lists. List $89.00. Sale $69.00. Coupon Price $46.00. (Disk only).

### SPECIAL SOFTWARE COUPON

**We pack a SPECIAL SOFTWARE COUPON in very COMMODORE 64 COMPUTER-DISK DRIVE-PRINTER-MONITOR we sell! This coupon allows you to save up to $100 OFF SALE PRICES and much more off list prices!!**

### PROFESSIONAL SOFTWARE COMMODORE 64

| Name | List | Coupon |
|---|---|---|
| Executive Word Processor | $99.00 | $52.00 |
| Complete Data Base | $89.00 | $46.00 |
| Electronic Spreadsheet | $89.00 | $46.00 |
| Accounting Pack | $69.00 | $32.00 |
| Total 5.2 Word Processor—Plus | | |
| Tape | $69.00 | $37.00 |
| Disk | $79.95 | $42.00 |
| Total Text 2.6 Word Processor— | | |
| Tape | $44.95 | $26.00 |
| Disk | $49.95 | $26.00 |
| Total Label 2.6 | $24.95 | $12.00 |
| Disk | $29.95 | $15.00 |
| Quick Brown Fox Word Processor | $69.00 | $40.00 |
| Programmers Reference Guide | $20.05 | $12.50 |
| Programmers Helper | $69.00 | $40.00 |
| Basic Tutor | $29.95 | $15.00 |
| Typing Tutor | $29.95 | $15.00 |
| Sprite Designer | $16.95 | $10.00 |
| Medicinemen | $19.95 | $12.00 |
| Weather War II | $19.95 | $12.00 |
| Music-Maker | $19.95 | $12.00 |
| EDU-Pack | $24.95 | $13.00 |
| 3D Maze Craze | $24.95 | $13.00 |
| Professional Joy Stick | $24.95 | $12.00 |
| Light Pen | $39.95 | $20.00 |
| Deluxe Dust Cover | $ 8.95 | $ 4.60 |

**Write or call for Sample SPECIAL SOFTWARE COUPON!**

### COMPLETE ACCOUNTING PACKAGE

This general ledger program is perfect for small business as well as home. It utilizes a double entry bookkeeping system. You only need enter one transaction and the computer will handle one transaction and the computer will handle the other. All accounts are user defineable and will bild for 1 year, resulting in a file of all transactions by account number, month and year to date. Each month a current months' trnsactions can be viewed at any time (99 accounts 187 entries per month). With this accounting program you will be able to monitor your financial growth as well as your expenses. List $59.00. Sale $49.00. Coupon Price $32.00. (tape or disk).

### VIC-20 COMPUTER $77.00

You get the Commodore VIC-20 Computer for only $77.00 when you buy at sale prices: The Commodore Data Cassette for only $69.00 and the Gortek Introduction to Basic program for only $19.95. TOTAL LIST PRICE $302.95. SPECIAL PACKAGE SALE PRICE $165.25.

### VIC-20 COMPUTER $88.00

When you buy 6 tape programs on sale for only $59.00. These 6 programs list for $100.00 to $132.00. You can choose one of three packs: 6 game pack, 6 home finance pack, or 6 small business pack. TOTAL LIST PRICE $360.00. SPECIAL PACKAGE SALE PRICE $147.00.

### 40-80 COLUMN BOARD $99.00

A fantastic price breakthrough for VIC-20 owners on this most wanted accessory!! "Now you can get 40 or 80 Columns on your T.V. or Monitor Screen." Plus we add a word processor with mail merge, electronic spread sheet, time manager and terminal emulator!! These PLUS programs require 8K or 16K RAM memory. (Disk add $10.00).

### VOICE SYNTHESIZER $69.00

Votrax Based. Make your VIC-20 COMPUTER TALK! Has features equivalent to other models costing over $370.00. You can program an unlimited number of words and sentences and even adjust volume and pitch. You can make adventure games that talk! A must for enhancing your programming creativity and pleasure. Includes FREE $14.95 Editor.

### 60K MEMORY EXPANDER $69.00

Sixslot — Switch selectable — Reset button — Ribbon cable. A must to get the most out of your VIC-20 Computer. Includes FREE $29.95 adventure game.

### 8K RAM CARTRIDGE $39.95

Increases programming power 2 1/2 times. Expands total memory to 33K (33,000 bytes). Memory block switches are on outside of cover! Includes FREE $16.95 game.

### 16K RAM CARTRIDGE $69.00

Increases programming power 4 times. Expands total memory to 41K (41,000 bytes). Memory block switches are an outside cover! Includes FREE $29.95 adventure game!!

### 12" GREEN SCREEN MONITOR $129.00

Excellent quality VIDEO MONITOR With antiglare, a 12" Green Phosphorous screen — 1920 characters (80 characters x 24 rows). Perfect for word processing application. 12" AMBER MONITOR — 2000 characters, 800 lines. $159.00. Plus $9.95 for VIC-20 or COMMODORE 64 Cable.

**• 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 $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

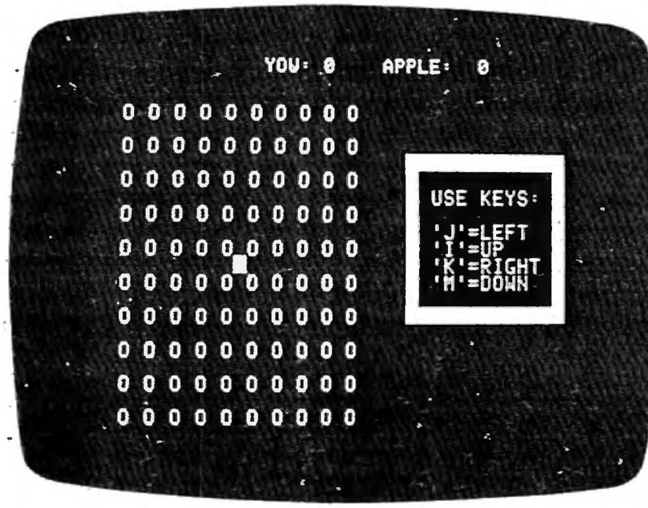
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars.

# PROTECTO ENTERPRIZES

(WE LOVE OUR CUSTOMERS)

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

```
450 IFNOT((PEEK(I+1)=79ANDPEEK(I-1)
    =79)OR(PEEK(I+32)=79ANDPEEK(I-3
    2)=79))THEN 440
460 IF SK>.6 OR CN>TS THEN 520
470 IF FNVH(I) THEN 500
480 IF FNBX(I-32)=-2 OR FNBX(I+32)=
    -2 THEN 440
490 GOTO 590
500 IF FNBX(I-1)=-2 OR FNBX(I+1)=-2
     THEN 440
510 GOTO 650
520 IF FNVH(I) THEN 560
530 IF SK>.6 OR CN>DT THEN 590
540 IF FNBX(I+32)=-2 AND FNBX(I-32)
    =-2 THEN 440
550 GOTO 590
560 IF SK>.6 OR CN>DT THEN 650
570 IF FNBX(I+1)=-2 AND FNBX(I-1)=-
    2 THEN 440
580 GOTO 650
590 ML=I:GOSUB 700
600 IF FNBX(ML-32)=-4 THEN BX=ML-32
    :GOSUB 770:F=-1
610 IF FNBX(ML+32)=-4 THEN BX=ML+32
    :GOSUB 770:GOTO 360
620 IF F THEN 360
630 GOTO 180
640 IF NOT (PEEK(I-32)=79 AND PEEK(
    I+32)=79) THEN 440
650 ML=I:GOSUB 700
660 IF FNBX(ML-1)=-4 THEN BX=ML-1:G
    OSUB 770:F=-1
670 IF FNBX(ML+1)=-4 THEN BX=ML+1:G
    OSUB 770:GOTO 360
680 IF F THEN 360
690 GOTO 180
700 REM
710 POKE ML,109
720 IF FNVH(ML) THEN POKE ML,73
730 I=185:IF WH=2 THEN I=150
740 FOR J=1 TO WH:GOSUB 760
750 FOR L=1 TO 200:NEXT:NEXT
760 SOUND I,2:RETURN:REM NOISE
770 YS=YS+1:J=8:I=200:IF WH=2 THEN
    J=4:I=150:YS=YS-1:CS=CS+1
780 POKE BX,128+16*(5-1)+15
790 FOR L=1 TO 3:POKE BX,128+16*(J-
    1)+15:GOSUB 760:POKE BX,128+16*
    (5-1)+15:FOR I=1 TO 200:NEXT:I=
    I+18:POKE BX,128+16*(J-1)+15:NE
    XT
800 PRINT@152,YS:PRINT@344,CS
810 IF YS+CS<54 THEN RETURN
820 IF YS>CS THEN 850
830 PRINT@487,"SORRY, YOU LOST.";
840 SOUND 128,10:FORI=1 TO 1500:NEX
    T I:GOTO 870
850 PRINT@490,"YOU WIN!!!!!";
860 FOR I=128 TO 255:SOUND I,1:NEXT
870 PRINT@487,"PLAY AGAIN (Y/N) ?";
880 A$=INKEY$:IF A$="" THEN 880
890 IF LEFT$(A$,1)="Y" THEN 100
900 END
910 J=0:K=0:PC=0
920 A$=INKEY$:IF A$="" THEN 920
930 IF A$=CHR$(94) THEN PC=-32:K=-1
    :RETURN
940 IF A$=CHR$(9) THEN J=1:PC=1:RET
    URN
950 IF A$=CHR$(10) THEN K=1:PC=32:R
    ETURN
960 IF A$=CHR$(8) THEN J=-1:PC=-1:R
    ETURN
970 IF ASC(A$)=13 THEN PC=99
980 RETURN
```




*This game is just beginning in the Apple version of "Dots."*

## Program 4: Dots – Apple Version

Translation by Patrick Parrish, Editorial Programmer

```
100  DIM XL%(23): FOR I = 0 TO 7:XL%(I)
      = 1024 + 128 * I:XL%(I + 8) = 106
     4 + 128 * I:XL%(I + 16) = 1104 + 1
     28 * I: NEXT I
110  FOR I = 770 TO 795: READ M: POKE I
     ,M: NEXT
120  TEXT : HOME : VTAB 11: HTAB 19: INVERSE
     : PRINT "D O T S": FOR J = 1 TO 10
     00: NEXT J: NORMAL
130  VTAB 15: HTAB 11: INPUT "SKILL LEV
     EL (0-10) ?";SK: IF SK < 0 OR SK >
     10 THEN 130
140 SK = (10 - SK) / 10:TS = 200 - 200 *
     SK:DT = TS + 25
150  HOME : PRINT : PRINT : FOR I = 1 TO
     10: PRINT : PRINT " 0 0 0 0 0 0 0
     0 0 0": NEXT
160  VTAB 9: HTAB 27: PRINT "USE KEYS:"
     : VTAB 11: HTAB 27: PRINT "'J'=LEF
     T": HTAB 27: PRINT "'I'=UP": HTAB
     27: PRINT "'K'=RIGHT": HTAB 27: PRINT
     "'M'=DOWN"
170  VTAB 7: HTAB 25: INVERSE : PRINT "
                  ": VTAB 16: HTAB 25: PRINT
     "           "
180  FOR ROW = 7 TO 14: FOR COL = 24 TO
     36 STEP 12: POKE XL%(ROW) + COL,32
     : NEXT : NEXT : NORMAL
190 YS = 0:AS = 0: VTAB 1: HTAB 13: PRINT
     "YOU: "YS"    APPLE:  "AS;
200  DEF  FN BX(COL) =  - ( PEEK (XL%(R
     OW + 1) + COL) <  > 160) - ( PEEK
     (XL%(ROW) + COL + 1) <  > 160) - (
      PEEK (XL%(ROW - 1) + COL) <  > 16
     0) - ( PEEK (XL%(ROW) + COL - 1) <
      > 160)
210  DEF  FN BY(ROW) =  - ( PEEK (XL%(R
     OW + 1) + COL) <  > 160) - ( PEEK
```

```
     (XL%(ROW) + COL + 1) <  > 160) - (
      PEEK (XL%(ROW - 1) + COL) <  > 16
     0) - ( PEEK (XL%(ROW) + COL - 1) <
      > 160)
220  DEF  FN VH(ROW) =  - ((XL%(ROW) +
     COL) <  > 2 *  INT ((XL%(ROW) + CO
     L) / 2))
230 X = 10:Y = 10:ROW = 12:COL = 10:CC =
      PEEK (XL%(ROW) + COL)
240  POKE XL%(ROW) + COL,106:F = 0
250 OLDROW = ROW:OCOL = COL: GOSUB 900
260 X = X + J:Y = Y + K: IF X < 1 OR X >
     19 OR Y < 1 OR Y > 19 THEN X = X -
     J:Y = Y - K: GOTO 250
270  IF PC = 99 THEN 300
280  POKE XL%(OLDROW) + OCOL,CC:CC =  PEEK
     (XL%(ROW) + COL): POKE XL%(ROW) +
     COL,42
290  GOTO 250
300 L =  - ( PEEK (XL%(ROW) + COL + 1) =
     207) - ( PEEK (XL%(ROW) + COL - 1)
      = 207) - ( PEEK (XL%(ROW + 1) + C
     OL) = 207) - ( PEEK (XL%(ROW - 1) +
     COL) = 207)
310  IF L - (CC = 160) =  - 3 THEN 330
320  POKE 768,250: POKE 769,1: CALL 770
     : GOTO 250
330 WH = 1:MROW = ROW:MCOL = COL: GOSUB
     750: IF  NOT  FN VH(ROW) THEN 380
340 BY = ROW: IF X > 1 AND  FN BX(COL -
     1) =  - 4 THEN BX = COL - 1: GOSUB
     790:F =  - 1
350  IF X < 19 AND  FN BX(COL + 1) =  -
     4 THEN BX = COL + 1: GOSUB 790: GOTO
     230
360  IF F =  - 1 THEN 230
370  GOTO 410
380 BX = COL: IF Y > 1 AND  FN BY(ROW -
     1) =  - 4 THEN BY = ROW - 1: GOSUB
     790:F =  - 1
390  IF Y < 19 AND  FN BY(ROW + 1) =  -
     4 THEN BY = ROW + 1: GOSUB 790: GOTO
     230
400  IF F =  - 1 THEN 230
410 WH = 2:F = 0:CN = 0: IF  RND (1) <
     SK THEN 490
420  FOR ROW = 4 TO 22 STEP 2: FOR COL =
     2 TO 18 STEP 2:K = XL%(ROW) + COL
430  IF  PEEK (K) = 160 AND  FN BX(COL)
      =  - 3 THEN 450
440  NEXT : NEXT : GOTO 490
450 I = K: IF  PEEK (XL%(ROW - 1) + COL
     ) = 160 THEN ROW = ROW - 1: GOTO 6
     40
460  IF  PEEK (XL%(ROW + 1) + COL) = 16
     0 THEN ROW = ROW + 1: GOTO 640
470  IF  PEEK (XL%(ROW) + COL - 1) <  >
     160 THEN COL = COL + 1: GOTO 700
480  IF  PEEK (XL%(ROW) + COL + 1) <  >
     160 THEN COL = COL - 1: GOTO 700
490 ROW =  INT ( RND (1) * 19) + 3:COL =
      INT ( RND (1) * 19) + 1:CN = CN +
     1: IF  PEEK (XL%(ROW) + COL) <  >
     160 THEN 490
500  IF  NOT (( PEEK (XL%(ROW) + COL +
     1) = 207 AND  PEEK (XL%(ROW) + COL
      - 1) = 207) OR ( PEEK (XL%(ROW +
     1) + COL) = 207 AND  PEEK (XL%(ROW
      - 1) + COL) = 207)) THEN 490
510  IF SK > .6 OR CN > TS THEN 570
520  IF  FN VH(ROW) =  - 1 THEN 550
530  IF  FN BY(ROW - 1) =  - 2 OR  FN B
     Y(ROW + 1) =  - 2 THEN 490
540  GOTO 640
550  IF  FN BX(COL - 1) =  - 2 OR  FN B
     X(COL + 1) =  - 2 THEN 490
560  GOTO 700
570  IF  FN VH(ROW) =  - 1 THEN 610
580  IF SK > .6 OR CN > DT THEN 640
590  IF  FN BY(ROW + 1) =  - 2 AND  FN
     BY(ROW - 1) =  - 2 THEN 490
600  GOTO 640
610  IF SK > .6 OR CN > DT THEN 700
620  IF  FN BX(COL + 1) =  - 2 AND  FN
     BX(COL - 1) =  - 2 THEN 490
630  GOTO 700
640 MROW = ROW:MCOL = COL: GOSUB 750
650 BX = COL: IF  FN BY(ROW - 1) =  - 4
      THEN BY = ROW - 1: GOSUB 790:F =
      - 1
660  IF  FN BY(ROW + 1) =  - 4 THEN BY =
     ROW + 1: GOSUB 790: GOTO 410
670  IF F =  - 1 THEN 410
680  GOTO 230
690  IF ( PEEK (XL%(ROW - 1) + COL) <  >
     207 OR  PEEK (XL%(ROW + 1) + COL) <
      > 207) THEN 490
700 MROW = ROW:MCOL = COL: GOSUB 750
710 BY = ROW: IF  FN BX(MCOL - 1) =  -
     4 THEN BX = MCOL - 1: GOSUB 790:F =
      - 1
720  IF  FN BX(MCOL + 1) =  - 4 THEN BX
      = MCOL + 1: GOSUB 790: GOTO 410
730  IF F =  - 1 THEN 410
740  GOTO 230
750  POKE 768,1: POKE 769,175: CALL 770
     : IF WH = 2 THEN  FOR I = 1 TO 500
     : NEXT I
760  POKE XL%(MROW) + MCOL,173: POKE XL
     %(MROW) + MCOL,45: FOR H = 1 TO 50
     : NEXT H: POKE XL%(MROW) + MCOL,17
     3
770  IF  FN VH(MROW) =  - 1 THEN  POKE
     XL%(MROW) + MCOL,201: POKE XL%(MRO
     W) + MCOL,137: POKE XL%(MROW) + MC
     OL,201
780  RETURN
790 YS = YS + 1:J = 25:I = 200: IF WH =
     2 THEN J = 1:I = 150:YS = YS - 1:A
     S = AS + 1
800  POKE 768,1: POKE 769,175: CALL 770
810  POKE XL%(BY) + BX,J: VTAB 1: HTAB
     13: PRINT "YOU: "YS"   APPLE:  "A
     S;
820  IF YS + AS < 81 THEN  RETURN
830  IF YS > AS THEN 860
840  VTAB 24: HTAB 14: PRINT "SORRY, YO
     U LOSE."
850  POKE 768,250: POKE 769,2: CALL 770
     : FOR I = 1 TO 500: NEXT I: GOTO 8
     80
860  VTAB 24: HTAB 15: PRINT "!! YOU WI
     N !!"
870  FOR I = 1 TO 5: POKE 768,1: POKE 7
     69,200 - I * 30: CALL 770: NEXT I:
      FOR I = 1 TO 10: POKE 768,1: POKE
     769,40 + I * 20: CALL 770: NEXT I
880  VTAB 24: HTAB 14: PRINT "TRY AGAIN
      (Y/N) ?";: GET B$: IF  LEFT$ (B$,
     1) = "Y" THEN 120
```

```
890  HOME : HTAB 5: VTAB 8: PRINT "...S
     EE YA...": END
900 PC = 0:J = 0:K = 0:A =  PEEK ( - 16
     384): IF A < 128 THEN 900
910  POKE  - 16368,0:A$ =  CHR$ (A - 12
     8)
920  IF A$ = "I" THEN ROW = ROW -  SGN
     (ROW - 3):K =  - 1: RETURN
930  IF A$ = "K" THEN COL = COL +  SGN
     (19 - COL):J = 1: RETURN
940  IF A$ = "M" THEN ROW = ROW +  SGN
     (21 - ROW):K = 1: RETURN
950  IF A$ = "J" THEN COL = COL -  SGN
     (COL - 1):J =  - 1: RETURN
960  IF A$ =  CHR$ (13) THEN PC = 99:
     RETURN
970  RETURN
980  REM  MUSIC ML DATA
990  DATA  172,1,3,174,1,3,169,4,32,168
     ,252,173,48,192,232,208,253,136,20
     8,239,206,0,3,208,231,96
```

©

# TI Towers

Raymond J Herold

*Here's a game that's not only fun to play, but is also a demonstration of the potential of TI BASIC. The author also discusses how ordinary TI BASIC can perform some of the functions available with Extended BASIC.*

Programming in TI Extended BASIC – with its powerful screen formatting commands, multiple statement lines, subprogram capability, and sprite graphics – offers something for everyone. However, not everyone is willing to shell out the extra purchase price right away.

This is especially true for the many first-time computer owners. They are content to "get along" using TI BASIC, which comes with the TI-99/4A. Anyone who thinks that these programmers are struggling along in the stone age should take a closer look. Careful examination will reveal that TI BASIC is a powerful language which outperforms many of the "standard" BASICs offered on other machines.

"TI Towers" is written in TI BASIC and demonstrates how some of its capabilities may be utilized. The game itself is a version of the ancient game Towers of Hanoi. There are three adjacent spindles, one of which has seven rings on it – the smallest ring on top, the next ring is the second smallest, and so on in pyramid fashion, with the largest ring on the bottom. The object of the game is to get all of the rings onto one of the other two spindles in the same order. You may move only one ring at a time, and you may not move a larger ring on top of a smaller one. It might sound easy, but it's not.

## Problem Solving In The Program

To provide instructions at the beginning of the game, the screen is set to black at line 905, then the instructions are PRINTed (lines 910 - 986). The screen is immediately set to medium red at line 991. This causes a momentary "blackout" of the screen before the instructions are displayed, but is preferable to the slow scroll produced by individually entering numerous PRINT statements.

The base of the playing board is drawn using the CALL HCHAR at line 7050, which uses the CHARPAT defined in line 7031. The spindles are drawn using the CALL HCHAR statement at lines 7090 - 7094 and the CHARPAT defined in line 7030. The execution time for these commands is quite fast.

Creating the rings presents something of a problem. Seven rings are required, each larger than the one before. If the first ring consists of a single character position, the second must use three characters; the third, five characters, and so on. The seventh ring requires 15 character positions. Since a ring can be on one of three spindles, the only way to avoid overlapping rings is to have a screen with at least 45 columns per line. With the TI-99/4A, limited to 32, the problem is obvious.

The solution is to use "half characters." Line 6300 defines a character with all bits on: a "full" character. Line 6320 defines a character with only the leftmost bits on: a "half character" for the right side of a ring. Line 6340 defines a "half character" for the left side. The seven rings required are built in lines 6350 - 6380 by concatenating the character patterns. Figure 1 illustrates this process. Lines 8040 - 8060 load the rings to the screen for the initial game setup.

Once the game begins, the program has to provide prompts and error messages to the player. Since the PRINT statement causes scrolling, and since the game uses a "fixed" game screen, the PRINT command is not acceptable for displaying messages. An alternative to this is using the TI BASIC command CALL HCHAR, which simulates the PRINT AT command that is so useful in Extended BASIC.

The message to be printed is moved to the variable MESSAGE$. The desired location for the message is loaded into the variables ROW and COLUMN. The routine starting at line 5001 actually writes the message. The loop initiated at line 5005 is performed the number of times indicated by the length of the message. Line 5010 converts each successive character in the string into its ASCII equivalent. Line 5020 then prints the string, one character at a time, at the position determined by ROW and COLUMN+1. This same procedure is used to position the rings when they are moved.

Getting information from the player presents a similar problem: the INPUT statement also causes a scroll. To avoid this, we must use the CALL KEY. This command detects a key being pressed and places the ASCII code of the key pressed into a specified variable. Lines 428-434 illustrate how this procedure can be used. Although TI BASIC doesn't have Extended BASIC's BEEP facility, the CALL SOUND command can be used just as effectively to notify the player that a response is necessary.

## Manipulating The Rings

The location of the rings is stored in the variable ARRAY. ARRAY is dimensioned by the number of spindles (3) and the number of allowable rings plus one. The additional element permits checking the spindles when no rings are present. The rings are initially assigned the numbers 1 through 7 and placed on the center spindle in lines 6250 - 6260. Ring 1 is the smallest; ring 7 the largest. Figure 2 shows the contents of ARRAY at the beginning of the game. Figure 3 shows what the contents of ARRAY would be if the two smallest rings were on the first spindle, the third smallest ring on the third spindle, and the rest on the middle spindle. Lines 1005 and 1008 find the "top" of the array for the corresponding sending and receiving spindles. For example, using Figure 3, RINGS(1) would contain 2 (number of rings). Subtracting this from 8 would give the sixth position of the first spindle, the top ring.

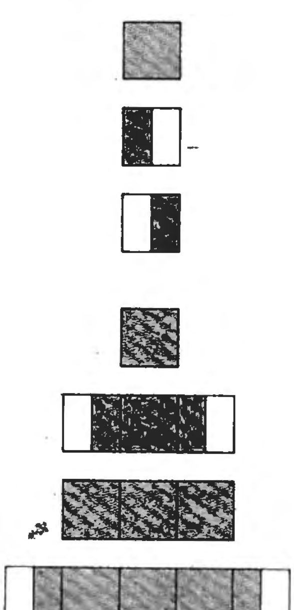
Lines 1020 and 1025 check to make sure that a large ring is not placed on top of a smaller one. When a valid move is made, the location of the rings is updated in lines 1100 - 1130. The variable RINGS keeps track of how many rings are on each spindle. The rings are moved by placing the appropriate RINGPAT$ in the new location. The ring at the old location is erased by moving BAND$ to it (lines 1530 - 1535). BAND$ defines only the spindle character (line 6390). When one of the two side spindles gets all seven rings, the game is over. Lines 482 and 484 determine this condition by checking the first and third spindle counters for 7.

TI BASIC can be quite effective when used to its potential. This article and game have perhaps given you some ideas for your own programs.

### TI Towers

```
100 DIM ARRAY(3,8)
110 DIM RINGS(3)
120 DIM RINGPAT$(7)
130 REM
140 REM INTRODUCTION
150 REM
160 CALL CLEAR
170 CALL SCREEN(9)
180 GOSUB 1930
190 MESSAGE$=M1$
200 ROW=5
210 COLUMN=11
220 GOSUB 1850
230 MESSAGE$=M2$
240 ROW=18
250 COLUMN=3
```

**Figure 1: Building The Rings**



| Pattern | ASCII code |
|---|---|
| "FFFFFFFFFFFFFFFF" | 128 |
| "F0F0F0F0F0F0F0F0" | 131 |
| "0F0F0F0F0F0F0F0F" | 133 |

| Pattern Concatenation |
|---|
| 128 |
| 133 and 128 and 131 |
| 128 and 128 and 128 |
| 133 and 128 and 128 and 128 and 131 |

**Figure 2: Contents Of ARRAY**

| | ----Spindles--- | | |
|---|---|---|---|
| 0 | 1 | 0 | |
| 0 | 2 | 0 | R |
| 0 | 3 | 0 | I |
| 0 | 4 | 0 | N |
| 0 | 5 | 0 | G |
| 0 | 6 | 0 | S |
| 0 | 7 | 0 | |

**Figure 3: Contents Of ARRAY**

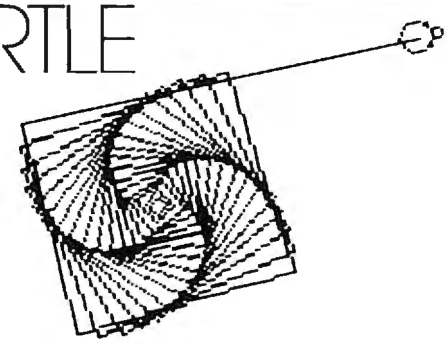
| | | ----Spindles--- | | |
|---|---|---|---|---|
| | 0 | 0 | 0 | |
| | 0 | 0 | 0 | R |
| | 0 | 0 | 0 | I |
| | 0 | 4 | 0 | N |
| | 0 | 5 | 0 | G |
| top — | 1 | 6 | 0 | S |
| | 2 | 7 | 3 | |

```
260 GOSUB 1850
270 MESSAGE$=M3$
280 ROW=20
290 COLUMN=9
300 GOSUB 1850
310 CALL SOUND(200,1000,4)
320 CALL KEY(3,KEY,STATUS)
330 IF STATUS=0 THEN 320
340 IF KEY=89 THEN 1070
350 IF KEY<>78 THEN 270
360 REM
370 REM BEGIN GAME
380 REM
390 IF MOVES>HIGHSCORE THEN 410
400 HIGHSCORE=MOVES
410 GOSUB 2260
420 IF HIGHSCORE<>0 THEN 440
430 HIGHSCORE=99999
440 MOVES=0
450 REM
460 REM PLAY GAME LOOP
470 REM
480 ROW=1
490 COLUMN=28
500 MESSAGE$=STR$(MOVES)
510 GOSUB 1850
520 ROW=23
530 COLUMN=1
540 MESSAGE$=M6$
550 GOSUB 1850
560 CALL SOUND(250,1000,4)
570 CALL KEY(3,KEY,STATUS)
580 IF STATUS=0 THEN 570
590 IF KEY<49 THEN 1700
600 IF KEY>51 THEN 1700
610 CALL HCHAR(23,13,KEY)
620 MOVEFROM=VAL(CHR$(KEY))
630 COLUMN=16
640 MESSAGE$=M7$
650 GOSUB 1850
660 CALL SOUND(250,1000,4)
670 CALL KEY(3,KEY,STATUS)
680 IF STATUS=0 THEN 670
690 IF KEY<49 THEN 1700
700 IF KEY>51 THEN 1700
710 CALL HCHAR(23,26,KEY)
720 MOVETO=VAL(CHR$(KEY))
730 IF MOVEFROM=MOVETO THEN 1700
740 GOSUB 1350
750 MOVES=MOVES+1
760 CALL HCHAR(23,1,32,30)
770 IF RINGS(1)=7 THEN 800
780 IF RINGS(3)=7 THEN 800
790 GOTO 450
800 REM
810 REM GAME COMPLETED
820 REM
830 FOR X=1 TO 20
840 CALL HCHAR(23,1,42,31)
850 CALL SOUND(150,X*400,21-X)
860 CALL HCHAR(23,1,32,31)
870 NEXT X
880 ROW=23
890 COLUMN=2
900 MESSAGE$=M8$
910 GOSUB 1850
920 FOR DELAY=1 TO 1500
930 NEXT DELAY
940 ROW=24
950 MESSAGE$=M9$
960 GOSUB 1850
970 CALL SOUND(300,1000,4)
980 CALL KEY(3,KEY,STATUS)
990 IF STATUS=0 THEN 980
1000 IF KEY=89 THEN 1050
1010 IF KEY<>78 THEN 970
1020 CALL CLEAR
1030 PRINT "GAME OVER"
1040 STOP
1050 GOSUB 1930
1060 GOTO 360
1070 REM
1080 REM INSTRUCTIONS
1090 REM
1100 CALL SCREEN(1)
1110 PRINT "TI TOWERS IS A VERSION
     OF"
1120 PRINT
1130 PRINT "THE GAME TOWERS OF HANO
     I."
1140 PRINT
1150 PRINT "THE OBJECT OF THE GAME
     IS TO"
1160 PRINT
1170 PRINT "MOVE THE RINGS ON THE C
     ENTER"
1180 PRINT
1190 PRINT "SPINDLE TO ONE OF THE T
     WO"
1200 PRINT
1210 PRINT "SIDE SPINDLES.  YOU MAY
      ONLY"
1220 PRINT
1230 PRINT "MOVE ONE RING AT A TIME
     , AND"
1240 PRINT
1250 PRINT "YOU MAY NOT PLACE A LAR
     GE"
1260 PRINT
1270 PRINT "RING ON TOP OF A SMALL
     ONE."
1280 PRINT
1290 PRINT
1300 PRINT "PRESS ANY KEY TO BEGIN"
1310 CALL SCREEN(9)
1320 CALL KEY(3,KEY,STATUS)
1330 IF STATUS=0 THEN 1320
1340 GOTO 360
1350 REM
1360 REM ANALYZE MOVE
1370 REM
1380 SUB1=8-RINGS(MOVEFROM)
1390 SUB2=8-RINGS(MOVETO)
1400 IF ARRAY(MOVEFROM,SUB1)>ARRAY(
     MOVETO,SUB2)THEN 1700
1410 IF RINGS(MOVEFROM)=0 THEN 1700
1420 GOSUB 1480
1430 RINGS(MOVEFROM)=RINGS(MOVEFROM
     )-1
1440 RINGS(MOVETO)=RINGS(MOVETO)+1
1450 ARRAY(MOVETO,SUB2-1)=ARRAY(MOV
     EFROM,SUB1)
1460 ARRAY(MOVEFROM,SUB1)=0
1470 RETURN
1480 REM
1490 REM MOVE RING
1500 REM
1510 ROW=7+(2*(7-RINGS(MOVEFROM)))
1520 COLUMN=19
1530 IF MOVEFROM<>1 THEN 1550
1540 COLUMN=3
1550 IF MOVEFROM<>2 THEN 1570
1560 COLUMN=11
1570 MESSAGE$=BAND$
1580 GOSUB 1850
1590 ROW=19-(2*(RINGS(MOVETO)))
```

```
1600 COLUMN=22
1610 IF MOVETO<>1 THEN 1630
1620 COLUMN=6
1630 IF MOVETO<>2 THEN 1650
1640 COLUMN=14
1650 XX=ARRAY(MOVEFROM,SUB1)
1660 COLUMN=COLUMN-(INT(LEN(RINGPAT
     $(XX)))/2)
1670 MESSAGE$=RINGPAT$(XX)
1680 GOSUB 1850
1690 RETURN
1700 REM
1710 REM ERROR IN MOVE
1720 REM
1730 ROW=24
1740 COLUMN=1
1750 MESSAGE$=E1$
1760 CALL SOUND(900,200,1)
1770 GOSUB 1850
1780 FOR DELAY=1 TO 200
1790 NEXT DELAY
1800 CALL HCHAR(23,1,32,32)
1810 CALL HCHAR(24,1,32,32)
1820 MOVEFROM=0
1830 MOVETO=0
1840 GOTO 520
1850 REM
1860 REM WRITE MESSAGES
1870 REM
1880 FOR I=1 TO LEN(MESSAGE$)
1890 CHAR=ASC(SEG$(MESSAGE$,I,1))
1900 CALL HCHAR(ROW,COLUMN+I,CHAR)
1910 NEXT I
1920 RETURN
1930 REM
1940 REM INITIALIZE AREAS
     {5 SPACES}
1950 REM
1960 M1$="TI TOWERS"
1970 M2$="DO YOU NEED INSTRUCTIONS?
     "
1980 M3$="REPLY Y OR N"
1990 M4$="BEST SCORE:"
2000 M5$="MOVES:"
2010 M6$="MOVE FROM?"
2020 M7$="MOVE TO?"
2030 M8$="{3 SPACES}*** YOU DID IT
     ***{6 SPACES}"
2040 M9$="PLAY AGAIN - Y OR N"
2050 E1$="** INVALID MOVE - TRY AGA
     IN"
2060 RINGS(1)=0
2070 RINGS(2)=7
2080 RINGS(3)=0
2090 FOR I=1 TO 8
2100 ARRAY(2,I)=I
2110 NEXT I
2120 ARRAY(1,8)=8
2130 ARRAY(3,8)=8
2140 CALL CHAR(128,"FFFFFFFFFFFFFF
     F")
2150 CALL CHAR(131,"F0F0F0F0F0F0F
     0")
2160 CALL CHAR(133,"0F0F0F0F0F0F0
     F")
2170 RINGPAT$(1)=CHR$(128)
2180 RINGPAT$(2)=CHR$(133)&CHR$(128
     )&CHR$(131)
2190 RINGPAT$(3)=CHR$(128)&CHR$(128
     )&CHR$(128)
2200 RINGPAT$(4)=CHR$(133)&CHR$(128
     )&CHR$(128)&CHR$(128)&CHR$(131
     )
2210 RINGPAT$(5)=CHR$(128)&CHR$(128
     )&CHR$(128)&CHR$(128)&CHR$(128
     )
2220 RINGPAT$(6)=CHR$(133)&CHR$(128
     )&CHR$(128)&CHR$(128)&CHR$(128
     )&CHR$(128)&CHR$(131)
2230 RINGPAT$(7)=CHR$(128)&CHR$(128
     )&CHR$(128)&CHR$(128)&CHR$(128
     )&CHR$(128)&CHR$(128)
2240 BAND$=CHR$(32)&CHR$(32)&CHR$(3
     2)&CHR$(36)&CHR$(32)&CHR$(32)&
     CHR$(32)
2250 RETURN
2260 REM
2270 REM SET UP GAME BOARD
2280 REM
2290 CALL CLEAR
2300 CALL SCREEN(8)
2310 CALL CHAR(36,"1818181818181818
     ")
2320 CALL CHAR(37,"FFFFFFFFFFFFFFFF
     ")
2330 CALL COLOR(1,13,1)
2340 CALL COLOR(13,7,1)
2350 CALL HCHAR(20,2,37,30)
2360 GOSUB 2510
2370 ROW=1
2380 COLUMN=1
2390 MESSAGE$=M4$
2400 GOSUB 1850
2410 COLUMN=21
2420 MESSAGE$=M5$
2430 GOSUB 1850
2440 CALL HCHAR(21,7,49)
2450 CALL HCHAR(21,15,50)
2460 CALL HCHAR(21,23,51)
2470 COLUMN=13
2480 MESSAGE$=STR$(HIGHSCORE)
2490 GOSUB 1850
2500 RETURN
2510 REM
2520 REM INITIAL RING SETUP
2530 REM
2540 CALL VCHAR(6,7,36,14)
2550 CALL VCHAR(6,15,36,14)
2560 CALL VCHAR(6,23,36,14)
2570 FOR X=1 TO 7
2580 ROW=5+(X*2)
2590 COLUMN=14-(INT(LEN(RINGPAT$(X)
     ))/2)
2600 MESSAGE$=RINGPAT$(X)
2610 GOSUB 1850
2620 NEXT X
2630 RETURN
```

©

## FRIENDS OF THE TURTLE



David D Thornburg Associate Editor

# The Logo Kaleidoscope

One of the first programming projects for many BASIC programmers is the construction of a screen kaleidoscope that generates pretty, symmetrical patterns on the display screen. For these programs, people usually pick a screen location at random and then place a colored dot at that location and at three other "mirror" locations to produce four symmetrically placed dots. While the resulting image is often quite attractive, the result is not that of a true kaleidoscope.

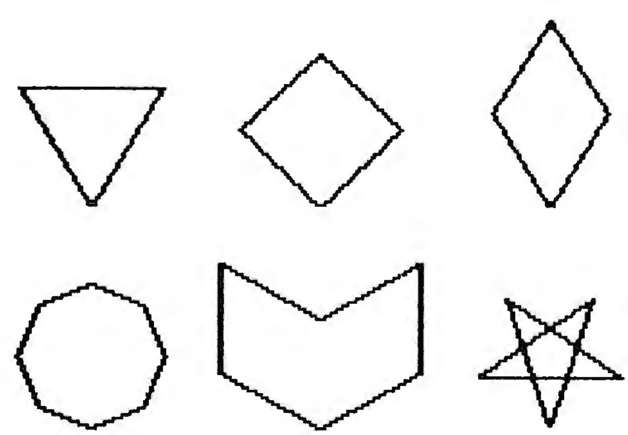
If you have ever taken a kaleidoscope apart, you must have wondered how such a simple apparatus could generate such beautiful images. Most kaleidoscopes consist of a set of mirrors and some small pieces of colored plastic that can be shaken to take random positions on a flat surface. When you look through the eyepiece, the mirrors generate multiple images of the arrangement of plastic pieces to produce beautifully symmetric pictures. Because Logo's turtle graphics allows you to easily create images that imitate the pieces of plastic, it is possible to create quite attractive kaleidoscopic images on your computer screen with a simple set of procedures.

The Logo kaleidoscope operates in the following manner. The system contains a set of graphic procedures to draw the fundamental picture elements (squares, triangles, stars, etc.). There can be as many of these elements as you desire (subject to the memory limitations of your system, of course). Each of these elements can be drawn as large as you desire. This gives the effect of having even more patterns to choose from.

Next, we use Logo's random number generator to select a shape, a size for the shape, the shape's color, and a distance from the center of the screen at which the shape will be drawn. Finally, this data is used by another procedure that places a copy of the chosen shape at several equally spaced angles around the center of the screen. Once one shape has been drawn, the process can be repeated for other shapes until the final image meets with your approval.

The kaleidoscope we will demonstrate in this article is written in the MIT version of Logo for the Apple II and should work with most Logo systems with very few modifications.

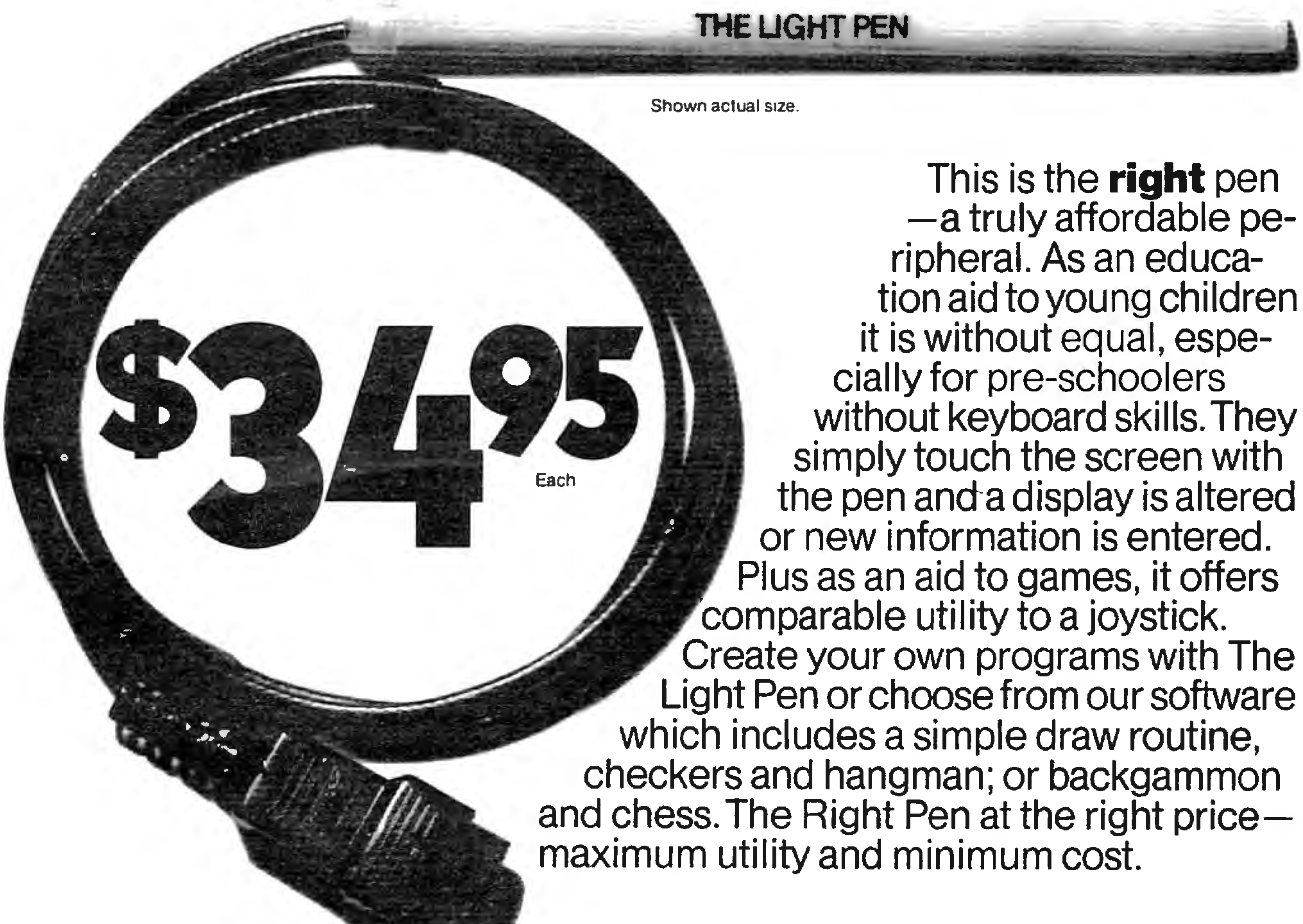
The kaleidoscope was started out with six shapes.



The procedures for these shapes are:

```
TO TRI :SIZE
  LT 30
  REPEAT 3 [FD :SIZE RT 120]
  RT 30
END
TO DIAMOND :SIZE
  LT 45
  REPEAT 4 [FD :SIZE RT 90]
```

# The Light Pen at the Right Price:




Shown actual size.

This is the **right** pen —a truly affordable peripheral. As an education aid to young children it is without equal, especially for pre-schoolers without keyboard skills. They simply touch the screen with the pen and a display is altered or new information is entered. Plus as an aid to games, it offers comparable utility to a joystick. Create your own programs with The Light Pen or choose from our software which includes a simple draw routine, checkers and hangman; or backgammon and chess. The Right Pen at the right price— maximum utility and minimum cost.

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 **future house**

p.o. box 3470, dept. C, chapel hill, north carolina 27514, 919-967-0861

```
  RT 45
END
TO PATT1 :SIZE
  LT 30
  REPEAT 2 [FD :SIZE RT 60 FD :SIZE RT 120]
  RT 30
END
TO OCT :SIZE
  LT 67.5
  REPEAT 8 [FD :SIZE / 2 RT 45]
  RT 67.5
END
TO PATT2 :SIZE
  LT 60
  FD :SIZE RT 60 FD :SIZE RT 120
  FD :SIZE LT 60 FD :SIZE RT 120
  FD :SIZE RT 60 FD :SIZE RT 120
END
TO STAR :SIZE
  LT 18
  REPEAT 5 [FD :SIZE RT 144]
  RT 18
END
```

Each of these figures has been defined to have mirror symmetry on the vertical axis. This is not a requirement, and you may wish to experiment with other orientations. The octagon was drawn at half the specified size to keep it in balance with the other figures.

## Constructing The Pattern

To make the kaleidoscopic image, we need a procedure that creates a list of basic patterns, chooses a pattern at random from this list, and selects an appropriate size (say between 20 and 50 units). Next, it should pick a random distance from the center (less than 60 units, to keep the images on the screen). Once these steps have been completed, copies of the chosen image should be stamped symmetrically around the screen. Then the procedure should wait for you to tell it if you want another element added to the image. When you press the RETURN key, the process will be repeated. The following procedure performs these tasks for us:

```
TO IMAGE
  MAKE "LIST [STAR DIAMOND OCT PATT1
    PATT2 TRI]
  MAKE "NAME SENTENCE PICKRANDOM :LIST
    ( 20 + RANDOM 30 )
  MAKE "DIST RANDOM 60
  PENCOLOR ( 1 + RANDOM 5 )
  PENUP
  WINDMILL :DIST :NAME
  MAKE "NAME REQUEST
  IMAGE
END
```

This procedure uses two other procedures that have to be defined: PICKRANDOM and WINDMILL. The function of PICKRANDOM is to choose an element of a list randomly. The following procedure does this for us:

```
TO PICKRANDOM :LIST
  OUTPUT PICK ( 1 + RANDOM (LENGTH :LIST ) )
    :LIST
END
```

The procedure PICK selects a given element from a list, and LENGTH measures the number of elements in a list:

```
TO PICK :NUM :LIST
  IF :NUM = 1 OUTPUT FIRST :LIST
  OUTPUT PICK ( :NUM - 1 ) ( BUTFIRST :LIST )
END
TO LENGTH :LIST
  IF :LIST = [] THEN OUTPUT 0
  OUTPUT 1 + LENGTH BUTFIRST :LIST
END
```

These two procedures operate "recursively." If you have a hard time understanding how they work, you may want to read about them in *Logo for the Apple II*, by H. Abelson, or read the chapter on recursion in my book *Discovering Apple Logo*. Also, we published some columns on recursion in "Friends of the Turtle" (COMPUTE!, November and December 1982).

## Defining Windmill

The only procedure we have left to define is WINDMILL. The function of this procedure is to draw a chosen pattern at equally spaced angular increments around the center of the screen. You may want to experiment with different numbers of images. I have tried using six images spaced at 60-degree increments and eight images spaced at 45-degree increments. These both work fine, but other angles are worth exploring as well. The number of copies of a pattern times the angle increment must be 360 in order for the pattern to be symmetric. That is why we turn 60 degrees for 6 copies (6 x 60 = 360) and 45 degrees for 8 copies (8 x 45 = 360).

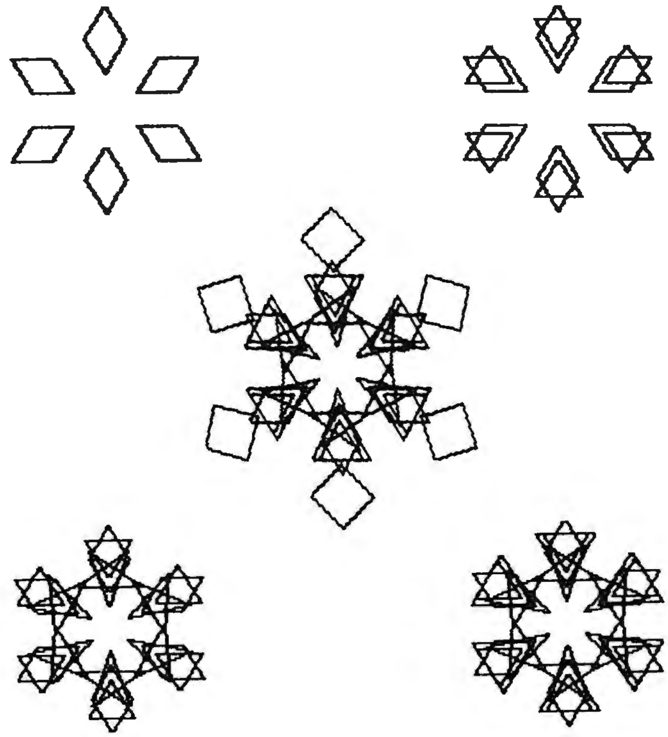
```
TO WINDMILL :DIST :LIST
  REPEAT 6 [FD :DIST PENDOWN RUN :LIST
    PENUP BACK :DIST RT 60]
END
```

To generate a kaleidoscopic pattern, hide the turtle and enter:

```
IMAGE
```

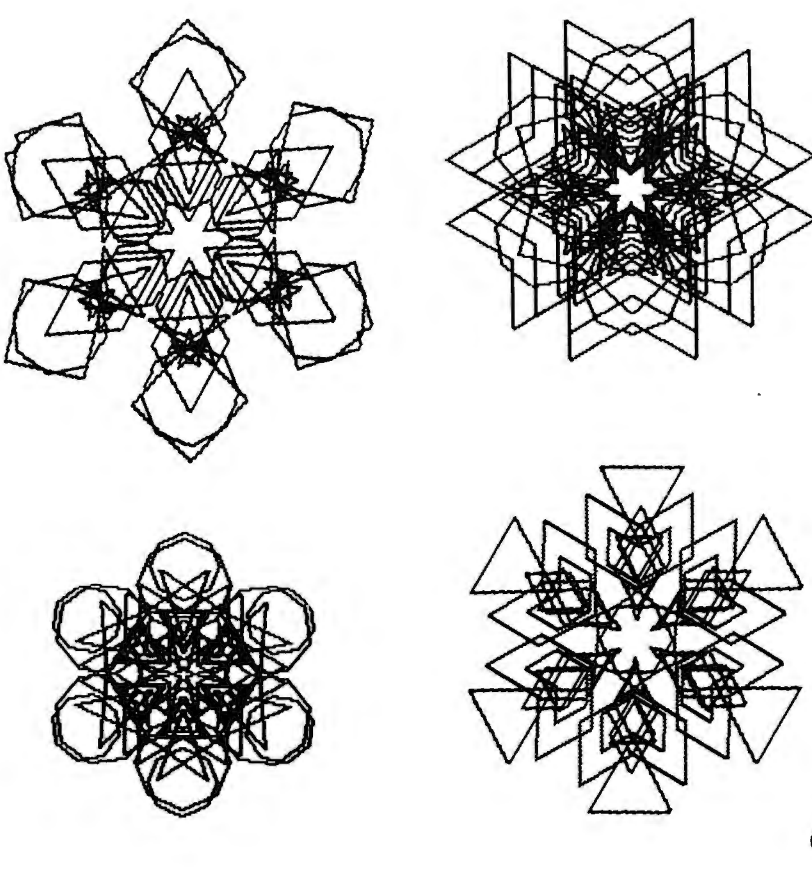
After the first pattern is drawn, press RETURN to get the next one. When the complexity of the pattern is satisfactory, you may want to print a copy of it or save it on your disk (with SAVEPICT, for example). If you are ambitious, you might want to write a Logo procedure that will keep track of all the randomly chosen values and generate its own Logo procedures for each pattern. Abelson's book (mentioned above) shows how to do this sort of thing.

The following five pictures show the successive development of one pattern:



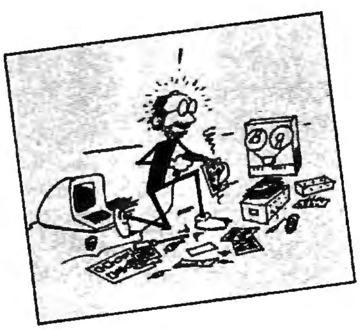
The remaining figures illustrate some other kaleidoscopic patterns that were generated with this set of procedures.

I think you will agree that these patterns are more interesting than those created with colored dots.



# Beyond Computer Literacy

Fred D'Ignazio, Associate Editor



A recent national "computers in the schools" survey conducted by the Center for the Social Organization of Schools at Johns Hopkins University found that most secondary schools are using computers to teach programming. (For a copy of the survey, write to Dr. Henry Becker, Center for the Social Organization of Schools, The Johns Hopkins University, Baltimore, MD 21218.) According to the survey, the second most popular use of computers was for drill and practice, primarily for math and language arts. In addition, the majority of the teachers who responded to the survey said that they looked at the computer as a "resource" rather than as a "tool."

I think this concentration on programming, drill, and practice and the image of computers as a "resource" is temporary. I believe that it is time for teachers and parents to start thinking beyond computer programming, beyond drill and practice, and beyond computer literacy.

## The Computer Steam Engine

Two factors have caused teachers and parents to concentrate on the computer as a resource and to stress computer literacy. First, most computer courseware turns the computer into an "electronic textbook." This kind of courseware is the most popular with teachers because it is the most familiar and the least threatening. The courseware (like a good textbook) introduces a new subject to a student, then drills the student on that subject.

Second, personal computers are still very primitive machines (compared to what they soon will be). They are a young, immature technology. Compared to what they'll be, the personal computers of today are like chugging steam engines, crude wooden plows, or fussy, cranky Model T's.

Despite manufacturers' claims, you *cannot* buy a personal computer and turn it on the way you would turn on a TV, then immediately begin to use it. Some computer literacy is still essential, or you quickly become lost in a nightmarish maze pursued by horrible creatures like bytes, RAMs, ROMs, K's, RS-232s, modems, interfaces, bauds, "Escapes," "Breaks," and "Resets."

## The Age Of Computer Literacy

Another recent survey (conducted by the University of Maryland) echoes the Johns Hopkins survey. It found that most schools introduce computers into the curriculum to help students become literate in computer technology.

But what does this literacy entail?

Is "computer literacy" programming? Is it the fundamentals of computer operation? Is it a quick course on using a computer keyboard? Is it drill and practice or the daily use of the computer as an electronic textbook?

Because of the pervasive spread of computers throughout our society, we have all become convinced that computers are important. From what we read and hear, when our kids grow up almost everyone will have to use computers in some aspect of their lives. This makes computers, as a subject, not only important, but also *relevant*.

An important, relevant subject like computers should be part of a school's curriculum. The question is how "Computers" ought to be taught.

Schools could teach about computers the way they teach about dozens of other important,

# AARDVARK L.T.D.

**TRS-80 COLOR    COMMODORE 64    VIC-20    SINCLAIR/TIMEX    TI99/4A**

## ZEUS

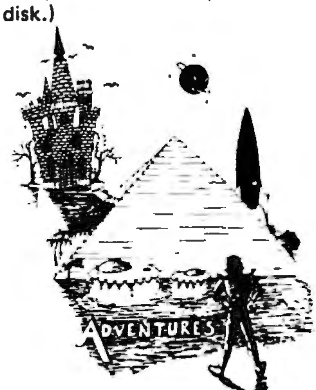

**QUEST** — A NEW IDEA IN ADVENTURE GAMES! Different from all the others. Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. TRS-80 Color, and Sinclair, 13K VIC-20. Extended BASIC required for TRS-80 Color and TI99/A. $14.95 each.

32K TRS 80 COLOR Version $24.95. Adds a second level with dungeons and more Questing.


**WIZARDS TOWER** — This is very similar to Quest (see above). We added wizards, magic, dragons, and dungeons to come up with a Quest with a D&D flavor. It requires 16k extended color BASIC. 13k VIC, Commodore 64, TRS-80 16k Extended BASIC, TI99/A extended BASIC. $14.95 Tape, $19.95 Disk.

**Authors** — Aardvark pays the highest commissions in the industry and gives programs the widest possible advertising coverage. Send a Self Addressed Stamped Envelope for our Authors Information Package.

**ZEUS** — It's fast and furious as you become the WIZARD fighting off the Thurderbolts of an angry ZEUS. Your Cone of Cold will destroy a thunderbolt and your shield will protect you — for a while. This is the best and highest speed arcade action we have ever done. Difficulty increases in wave after wave, providing hours of challenging fun and a game that you may never completely master. Commodore 64, Vic20 (16k expander), and 16k TRS-80 Color Computer. (ALL MACHINE CODE!)
$19.95 tape  $24.95 disk. (Tape will not transfer to disk.)




**ADVENTURES!!!**

The Adventures below are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are interactive fantasies. It's like reading a book except that you are the main character as you give the computer, commands like "Look in the Coffin" and "Light the torch.")

Adventuring requires 16k on Sinclair, and TRS-80 Color. They require 8k on OSI and 13k on VIC-20. Now available for TI99. Any Commodore 64.

$14.95 Tape — $19.95 Disk.

**ESCAPE FROM MARS**
(by Rodger Olsen)
This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

**PYRAMID** (by Rodger Olsen)
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!


**SEAWOLFE** — ALL MACHINE CODE In this high speed arcade game, you lay out patterns of torpedoes ahead of the attacking PT boats. Requires Joysticks, at least 13k RAM, and fast reflexes. Lots of Color and Sound. A fun game. Tape or Disk for Vic20, Commodore 64, and TRS-80 Color. NOTE: tape will not transfer to disk!
$19.95 Tape — $24.95 Disk.


**Dungeons of Death** — This is the first D&D type game good enough to qualify at Aardvark. This is serious D&D that allows 1 to 6 players to go on a Dragon Hunting, Monster Killing, Dungeon Exploring Quest. Played on an on-screen map, you get a choice of race and character (Human, Dwarf, Soldier, Wizard, etc.), a chance to grow from game to game, and a 15 page manual. 16k Extended TRS-80 Color, 13k VIC, Commodore 64. At the normal price for an Adventure (14.95 tape, $19.95 disk), this is a give-away.

**Dealers** — We have a line of about 100 original programs for the machines listed here. We have High speed Arcades, Quality Adventures, Word processors and Business Software for Small machines. Better yet, we have excellent Dealer support. Phone for information.

**Send $1.00 for Complete Catalogue - Please specify system on all orders - $2.00 Shipping Charge on each order**

VISA

## AARDVARK L.T.D.

**2352 S. Commerce, Walled Lake, MI 48088 / (313) 669-3110**

MasterCard

***Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.***

# COMMODORE 64
## 80 COLUMN BOARD
# $159.00*




The dream of seeing 80 columns on the screen at one time is now a reality. The Protecto Expansion Board converts your Commodore 64 to 80 columns! PLUS you get a word processor with database mailmerge, an electronic spreadsheet, and a terminal emulator. List $275. SALE $159.
* When you buy any monitor — Commodore 64 Computer — Disk Drive — Printer! 80 Column Board purchased alone — Cost $179.00. (Disk Programs add $10.00).

| MONITOR SALE | List | Sale |
|---|---|---|
| 12" Green Phosphorous — High Quality 600 lines resolution | $199 | $129 |
| 12" Amber Monitor — Superior Quality 800 lines resolution | $219 | $159 |
| 14" Color Monitor — High Quality Commodore Monitor | $299 | $269 |

**VIC 20 COMPUTER 40-80 COLUMN BOARD LIST $149.00 SALE $99.00**

- ***15 DAY FREE TRIAL***
- ***WE HAVE LOWEST COMMODORE 64 PRICES***
- ***WE HAVE OVER 500 PROGRAMS***
- ***VISA • MASTERCHARGE • COD***
- ***WE LOVE OUR CUSTOMERS***

**PROTECTO**
**ENTERPRIZES** (WE LOVE OUR CUSTOMERS)
**BOX 550, BARRINGTON, ILLINOIS 60010**
**Phone 312/382-5244 to order**

relevant subjects (like math, social studies, geography, and language arts) – *with books*. However, since desktop computers are now relatively cheap, schools are buying computers so the students can get a look at the machines themselves. Special computer classes are being set up so that students can play with computers, tinker with them, and learn some basic programming. Thus, on a practical level, computer literacy turns out to be mere computer *exposure*.

But exposure to what? Kids who are now enrolled in elementary and secondary schools are exposed to four aspects of computers. They learn that computers are programmable machines. They learn that computers are being used in all areas of society. They learn that computers make good electronic textbooks. And (something they already knew), they learn that computers are terrific game machines.

## The Results

This exposure is worthwhile. It alone justifies a school's purchase of computers for its students. According to the surveys, real educational results have been realized at schools which concentrate on exposing kids to computers. First, students develop a familiarity with computer keyboards, computer operation, and computer concepts.

Second, students in these schools develop a realistic, positive image about computers. Past generations saw computers as electronic brains – abstract, all-powerful, and mysterious. Now kids get to see computers as they really are. Kids get to touch computers, play with them, push their buttons, order them about, and cope with computers' incredible dumbness, their awful pickiness, their exasperating bugs, and their ridiculous quirks.

Third, the surveys show that computers have played a big part in improving kids' (*and* teachers') attitudes toward school. Kids who use computers during their school day come early and stay late – just to have time on the computers. The whole school day goes better for everyone because it has a rosy glow caused by the computers. There are countless stories of learning disabled kids, handicapped kids, and near dropouts who got turned on to computers and became model students. Kids with problems warm up to computers and, on their own, use them to improve their academic performance. Bright kids turn to computers as intellectual companions and resources and learn in a more personal, accelerated fashion.

Computers touch a kid's life. And the effect is cumulative. When enough young people are affected by computers, it changes the atmosphere of the entire school. The impact of computers on a school can be psychological. Computers can improve school spirit.

Last, computers make the students less fiercely competitive. Instead, they begin helping each other. Striving for academic excellence is a good thing. But in certain contexts, it can put kids under a great deal of pressure to succeed – with almost no help or support from their friends.

Here, too, the effect of computers on the "social organization of learning" has been significant. Computer classes have an atmosphere which is different from that found in many other classes. In computer class, teachers don't just teach, and students don't just give answers, write down notes, and take tests. In computer class, *everybody* learns, everybody shares, and everybody learns to be helpful. Teachers tell stories about how big, smart-aleck teenagers in their classes have put their arms around their shoulders, and with great patience and sincerity have explained how to boot a disk, load a program from tape, or master a new piece of software. Roles become reversed, fluid, and fuzzy. But often everyone benefits. And learning occurs at a rapid pace.

## Computers Of The Future

Computers in schools have already had a substantial, positive effect.

But I'm still worried.

I think that schools are unintentionally locking their students into the present – the fleeting, short-lived Age of Computer Literacy. This is an age from which computers will emerge very soon. Computer literacy will become irrelevant and unimportant long before most students enter the job market.

Also, in many schools, computers are being taught in separate "computer courses." This divorces them from the rest of the school, from the rest of the curriculum, and most importantly, from the other teachers.

Computers in our economy and in our society don't exist as islands of technology. Instead, they have become part of the fabric of everything we do. They are intimately involved with the way we live, move about, play, and do business. Just telling the students that this is so and teaching them BASIC or Logo is not adequate. We need to give them working experience with computers *as they are used in the real world.*

What's more, schools are using their newly acquired computers as an object of curiosity; as a hands-on device to learn the arcane arts of programming and computer operation; and as a teaching aid to learn math and language arts. But in the very near future, computers will be as common as TV sets, computer operation will be simple, and relatively few people will be employed as programmers. In the near future, the most popular, important, and powerful use of computers will be as a *general-purpose tool*.

## The Computer-Literacy Deep Freeze

Computers are at a crude, nasty, awkward stage in their development. But they are evolving at an incredible pace. Hardware advances occur almost daily. And software, long the bumbling, dim-witted half brother of computer hardware, has at last entered into its own revolution. In 1976 there was almost no software, yet last year 200 companies sold more than a billion dollars of software. By 1990, experts predict that people will be buying $12 billion in software, about as much as they now spend on home appliances. We will soon see more software than ever before, and if we weed through all the junk, we will find much software that is good and quickly getting better.

The twin revolutions in computer hardware and (especially) in computer software will insure that computers of today will be transient, short-lived creatures. Trendy, high-income schools that buy up dozens of these computers and inaugurate intensive courses in the art and science of their programming and operation are handicapping their students. They are guaranteeing that these young people will be victims of technology.

In ten years, how important will it be for a student to know how to program in BASIC, or know machine language, or know how many K bytes are available in RAM storage? Or how to format a floppy? Or how to position a tape cassette to a particular program?

## BASIC Will Be A Dead Language

In five years computers will be completely different. In ten years they will be black boxes. They will still be programmable, but nobody except the experts will do the programming. The final custom-fitting of all commercial programs will be done by the user, but in English, not in BASIC, Logo, or Pascal. These will be archaic languages, like Greek or Latin, important historically, but of little relevance to students who are entering the job market of the early 1990s.

High schools, vocational-technical schools, and colleges are turning out huge numbers of computer scientists and technicians. But, surprisingly, computer jobs are beginning to dry up, especially at the entry level. High-paying computer jobs are still there, but they are reserved for those who have several years of experience or who have combined skills in computers and in some other field such as business, medicine, law, chemistry, or engineering.

Computer classes in schools today are busy turning out the computer "mechanics" and "repair persons" of the future. Persons trained in these areas will find that there are very few jobs awaiting them, and the competition will be incredible. With the huge supply of bodies and the slackening demand, salaries will plummet and so will prestige. By the time young people enter the market as computer specialists, most of the romantic aura about computers will have rubbed off. The glamour will have faded.

It all boils down to how we see computers. Do we see them as finicky appliances that have to be twiddled, scrutinized, and understood? Do we see them as "exer-cycles" and mental jogging machines that stimulate our problem-solving abilities and encourage algorithmic (that is, step-by-step, logical, goal-oriented) thinking? Or are they mechanical chameleons and quick-change artists?

In the near future I think most of us will see computers as Super Tools – like the handy-dandy Swiss pocketknives you can buy that have all those scissors, bottle openers, screwdrivers and twelve different blades stuffed inside. They will do everything. And we won't care how. We'll just pull out a new tool and run it!

For example, we will pop in a cartridge and our computer will become an electronic typewriter, dictionary, or secretary. We'll pop in another cartridge, and the computer will become our personal accountant, tax advisor, or a gourmet chef.

Computers of the near future will be like vaudeville performers who can change their costumes in a flash. One minute they will be patient math tutors for our children. The next moment, they will be our electronic windows to the outside world. We will use them to bring us the latest stock prices, make a plane reservation, or mass mail our Christmas cards.

Or a moment later the computer will become an interactive (videodisc and graphics) TV. We will get to track down a roller-coaster bomber, solve the mystery of a collapsing bridge, or go on a big game hunt in darkest Africa.

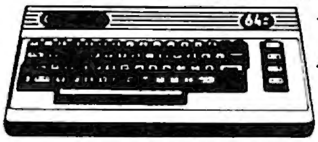
We will not care how the computer changes its clothes. We will not be interested in a tour behind the stage, or what the performer's clothes look like from the inside out. Instead, we will want (maybe demand) to learn, to be informed, to be entertained, and to get on with our work. The computer will slip into its rightful position. It will become a marvelous tool that is almost ignored. It will be an almost invisible means to accomplish the *essential* things in life: survival, work, education, and fun.

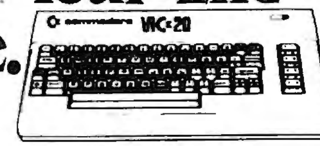
## Computers As Islands

The approach in many schools is to teach about computers in a special "computer science" or "computer lab" or "computer literacy" course. This reminds me of the touch typing course and the metalworking and other "shop" courses I took when I was in high school.

In all these "technical skill" courses, kids are introduced to machines and instructed in how to

# Commodore Puts Excitement In Your Life For Such A Small Price.





| | |
|---|---|
| Commodore 64 (With Factory Rebate) | $219 |
| VIC 20 | $ 99 |
| 1525 Printer | $229 |
| 1530 Datasette | $ 59 |
| 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 1212 Progammer's Aid Cartridge | $45 |
| VIC 1213 Vicmon Machine | |
| Language Monitor | $45 |
| VL 102 Intro to Basic Prog. I | $21 |
| Intro to Basic Prog. II | $21 |
| 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 I (D) .... $27
- 63E-002 Zork II (D) .... $27
- 63E-003 Deadline (D) .... $35
- 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 Trooper #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 (D) .... $ 27
- Fraction Fever (CT) .... $ 20
- Alphabet Zoo (CT) .... $ 20
- Delta Drawing (CT) .... $ 23

### Snyapse Software
- SSE-011 Ft. Apocalyse (D) .... $23
- SSE-016 Drelbs (D) .... $23
- SSE-019 Survivor (D) .... $23
- SSE-020 Pharoh's Curse (D) .... $23
- SSE-311 Ft. Apocalypse (C) .... $23
- SSE-316 Drelbs (C) .... $23
- SSE-319 Survivor (C) .... $23
- SSE-320 Pharoh's Curse (C) .... $23

### United Microwave Industries (UMI)
- 92E-302 Renaissance (C) .... $27
- 92E-331 Motor Mania (C) .... $20

**We Accept VISA & MasterCard**

COMPUTER OUTLET

**Call Toll Free**

***1-800-634-6766***

**Order Line Only**

## VIC 20 Educational

### Books
- Kids and the Vic .... $ 18
- Programmer's Reference Guide (Vic) .... $ 14
- Programmer's Reference Guide (64) .... $ 18

### Music
- Vic Music Composer (CT) .... $ 29
- HES Synthesound (CT) .... $ 49

### Language Arts
- Super Hangman (C) .... $ 14
- Simon/Hess (C) .... $ 13
- Concentration (C) .... $ 13
- Home Babysitting .... $ 23

### Reading and Language Arts
- Finger Spelling (D, C) .... $ 12
- My Spelling Easel (Ages 3-10) .... $ 26

### Math
- Sky Math (C) .... $ 12
- Space Division .... $ 12
- Bingo Speed Math (CT) .... $ 23
- Number Crunch (CT) .... $ 23
- Number Chaser .... $ 17
- Number Gulper .... $ 17

### Programming Techniques
- Intro to Basic Prog. I .... $ 22
- Intro to Basic Prog. II .... $ 22
- Programmer's Aid Cart. .... $ 22
- Turtle Graphics/Hess (CT) .... $ 29
- Groteck & Microchip .... $ 23

### Social Studies/Science
- Visible Solar System .... $ 23
- Reaganomics (CT) .... $ 27

**ORDERING INFORMATION AND TERMS:**
For fast delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. C.O.D. orders ($3.00 minimum) and 1% of all orders over $300. School purchase orders welcome. Prices reflect a cash discount only and are subject to change. Please enclose your phone number with any orders.
**SHIPPING:** Software ($2.50 minimum). Shipping — Hardware (please call). Foreign orders APO & FPO orders — $10 minimum and 15% of all orders over $100. Nevada residents add 5 3/4% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must have a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement. All returned merchandise is subject to a restocking fee and must come with their original packaging in order to be accepted. NO returns permitted after 21 days from shipping date.

## 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

### VIC 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

### 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

### 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 .... $ 34
- Cloud Burst (CT) .... $ 27
- Satellites & Meteorites (CT) .... $ 34
- Outworld (CT) .... $ 34
- Wordcraft .... $270

### 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

develop a certain level of proficiency and familiarity with these machines. *But they aren't told why.*

At some level, students who take these courses must be asking themselves: Why is a computer important? What good does it do me to know how to program a computer, or load a program, or learn about FOR-NEXT loops?

The computer is not an end in itself. It is a means to an end. It is a resource or a tool that can be used to do something else. Computer skills are meaningless to a child unless the child can use them to do something that he or she wants or needs to do. To make computers meaningful, they must be integrated, on a daily basis, into the rest of the curriculum and into a child's life. The child must need or want to do something important that can only be done on a computer.

## Computers As Moon Rocks

In many schools, desktop computers are introduced as oddities and curiosities, like moon rocks.

This is a marvelous approach. It encourages children to see computers as wondrous devices (which they are) and to approach computers with curiosity and fascination.

Since computers are objects of wonder and curiosity, many schools have put them in a special room – a computer *museum*. Everyone can come in and gawk at them, reverently press their buttons, and say ooh and aah.

But after having a computer about six months, a school usually moves beyond this approach. The awe and magic about computers quickly wears off – especially for the kids. Teachers begin teaching kids how to program – how to master computers, boss them around, and tame them.

## The Latest Audiovisual Device

Today, many schools are leapfrogging right over these first two steps. When schools acquire a computer today, they don't automatically send it off to a tiny lab or unused classroom. Instead, they regard the computer as a new kind of audiovisual device – a godsend for the frazzled, overworked but forward-thinking teacher of the 1980s.

This approach is being given a big boost by the educational courseware flooding into the market. Dozens of companies are producing hundreds of software packages. A year ago, there was an acute shortage of reputable software. Now, already, there is a glut. There are hundreds of programs out that introduce kids to the alphabet. Dozens more teach them how to add two numbers or spell simple words.

I walked down the exhibitors' (read *vendors'*) aisle at a recent educational computing conference, and I was overwhelmed by the number of glossy, smart-looking packages I saw. It was a kaleidoscopic, mind-numbing experience.

Given this vast amount of courseware, it won't be long before computers move out of their "computer museums" and isolated labs and into the curricular mainstream. Thousands of math and language arts teachers already use computers as audiovisual aids. Soon history, science, music, and art teachers will use them too.

## The Computer As A General-Purpose Tool

The computer will soon become a valuable resource for teachers, no matter what subject they teach. But the computer can be more than a special-purpose resource to help a teacher teach a particular subject. It is also a tool – a magnificent, general-purpose tool that a child can apply to any subject.

If children learn only how to program, decipher bits from bytes, and learn geography on a computer, they are going to be poorly equipped to use computers – in the future, in a job, in the outside world.

To be prepared for the future, youngsters must learn how to use computers as tools. That's the way most computers in our society are used. And that's the way they will be used in the future.

## Discovering A Tool

The problem has been that most classroom computers are regarded more as *toys* than as tools. They don't have the speed, memory capacity, or software to make them serious devices. They are also isolated, one from the next, instead of tied into information and programming resources (by phone or direct-wired access to a central, high-speed computer).

But all this is changing.

One of the most popular and well-attended sessions at the National Educational Computing Conference (NECC), held this past June in Baltimore, was on using computers in studying literature and English composition. Teachers presented papers on how they taught word processing in the classroom, how they used a computer in writing class, and how they and their students used a computer to study and analyze literature.

Kids in the first two classes used the computer as a tool – as a *word processor*. They found it was easier to write stories, develop ideas, and explore new subjects by using a computer.

Kids in the third class learned programming skills for a purpose: they turned the computer into a tool to help them analyze a book, article, or short story. They used the computer to complete a class assignment.

Right now, word processing is a very popular computer application in schools. But it is just the tip of the "computer tool" iceberg. Computers can become powerful word-handling tools for kids. But they can also become all sorts of other kid tools.

With software already available or under development, computers can become kids' powerful database managers, priority sorters, homework organizers, and calendar schedulers. They can simulate chemistry labs, physics labs, and math labs. They can be used to map out a complicated dance routine for a musical, compose a song, or take the student on a journey inside a volcano, to the center of an atom, or to the outer reaches of the solar system.

New software packages are also needed that are patterned after the "second generation" software now running on expensive IBM, Apple, and Xerox business computers. These *kid workstations* should be general-purpose tools that help a student process words, perform complicated calculations, create graphs, functions and diagrams, and organize, classify, and summarize huge amounts of data. They should enable students to link their computers and thereby communicate with each other. They should encourage teachers to assign more *team projects* for students in which students and their computers work together to solve problems, do homework, or complete class assignments.

The more students get to use a computer as a *tool* to enable them to do something necessary or desirable, the more meaningful and useful computers will become. Also, this is precisely the type of training that young people will need to prepare them for their future careers. Very few students will find jobs as computer specialists. But a vast majority of today's students will need to use computers as tools in their jobs. They will use computers to help them solve problems, make decisions, analyze information, have fun, create and disseminate new knowledge, and communicate with their fellow human beings. ©

Glenn M. Kleiman

# Playful Exercises For The Mind

One premise underlies all I have to say in this month's column: the mind, like the body, is strengthened by exercise. I believe any activity is worthwhile if it leads people to exercise their creativity, thinking, problem-solving, memory, perception, concentration, math, or language skills.

Many toys, games, and puzzles provide opportunities for mental exercise. For example, building toys, such as blocks, Erector Sets, Tinker Toys, and Legos, provide opportunities for children to design, build, test, and modify various objects. Clay, crayons, and paint sets provide other means for creative play.

Also, crossword puzzles, and word games such as Scrabble, exercise vocabulary and spelling skills. Jigsaw puzzles exercise perceptual and imagery skills, while puzzles such as Rubik's Cube exercise problem-solving skills. Games such as chess and checkers involve problem-solving and planning skills. Many board games provide varied learning experiences. Monopoly, for example, simulates aspects of a real estate market in which players experience negotiating, buying, and selling. The game involves rents, taxes, utility bills, and banking. It also requires a fair amount of reading and math, particularly for the "banker."

Computers can be programmed to provide many types of playful exercise for the mind. In some cases, the exercises are similar to those which can be done without a computer, but the computer makes some things easier. Computers can be programmed to set up game boards on the screen, keep score, monitor time limits, save the "state" of games so they can be continued later, and make sure the rules are followed. But computers should not be limited to these mundane chores.

## Making Real Use Of Computer Power

The flexible and interactive nature of personal computers, combined with their graphics, animation and sound capabilities, offers exciting new possibilities for mental exercises. For example, computers can be programmed to automatically adjust the level of challenge to be suitable for each player. Depending upon the nature of the game, the computer can adjust the speed of movement, the complexity of the materials, the size of the board, or the level at which it plays.

Computers can also provide hints, second chances, and other on-line aids. The graphics and animation make it possible to represent many things pictorially, as well as provide displays which hold players' interest. The sound and, on some systems, speech capabilities, also add to the attention-holding and information exchange possibilities. The continuous control players can have, and the speed at which the computer can respond, are additional important advantages.

Various types of mental exercise programs have been developed to take advantage of computer features. There are computer versions of paint sets, chess, checkers, Othello, crossword puzzles, Rubik's Cube, Scrabble, Concentration, and many more. Simulations provide another type of playful mental exercise. Adventure games and other interactive stories – stories in which readers direct and contribute to the flow of events as they read – also belong in this category.

# 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**
Introduction, programs, applications, learning exercises
Order #170 — $7.95
All programs from book No. 170 on disk
Order No. 7319 — $22.00 only!


**A Look into the Future – ASTROLOGY**
on your ATARI 800
How to calculate your own horoscope
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 on autoboot disk. No cartridge, no DOS, no FORTH Language required!
Order# 7312 — $49.00

**SUPERINVENTORY**
(1000 items per disk)
Completely written in FORTH. Same as above. (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 800 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 und 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 Ogino)
Order-No. 7222 Disk only! — $29.95

**Books + Software for VIC-20 (requires 3KRAM Exp.)**
No. 4870 Wordprocessor for VIC 20, 8KRAM — $19.95
No. 4883 Mailing List for VIC 20, 16K 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 hexdump + 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

I have reviewed some paint set and simulation programs in previous columns (October and November 1982), and I will discuss interactive stories, computer word games, and other types of playful exercises in the future. For the rest of this column, I will describe one program which is perhaps the best example now available of how computers offer new opportunities for play and creativity.

## Pinball Construction Set

Suppose you were designing and creating a pinball game. You would have to figure out the shape of the playing area and barricades, where to put. flippers, bumpers, spinners, lanes, gates, targets, and the other apparatus of these games. You would have to assign point values for when the ball hits each one, and add the essential sound effects.

Of course, good pinball games are not random arrangements. They are designed so there is a good amount of bounce, ample opportunity to use the flippers, and an appropriate amount of risk of losing the ball. There should be no places where a ball can get stuck or be caught in an endlessly repetitive pattern of bounces. The number of points scored in various ways should reflect the difficulty and likelihood of striking the various targets. Hitting all of a set of targets should yield bonus points.

And, of course, the overall design should be visually balanced and pleasing. Building such a game would require a great deal of thinking and experimenting. Certainly, a pinball construction kit would offer opportunities for creative, exploratory play comparable to those provided by other building toys.

*Pinball Construction Set* program, created by Bill Budge, offers all of the above possibilities and more. Once you have created a game, you can play it like any of the available video pinball games. You control the ball with the joystick. The play action feels like a real pinball game, and the movement of the ball is an excellent simulation of the real thing.

When you boot *Pinball Construction Set*, you see the screen with three types of elements. At the left is a box in the basic shape of a pinball game. At the right are pictorial representations (called *icons*) of the tools you have available – a hand, arrow, scissors, hammer, paintbrush, and others. In between are the pieces for the pinball game – flippers, bumpers, and all the rest. You construct your game, test playing it as you go. When finished, you can make a separate disk with your game, so that anyone can play it. The figure shows the screen after a game has been constructed.

You begin constructing a game by using a joystick to control the hand icon on the screen. You can move the hand to any pinball piece, press the joystick button to pick up the piece, and then move it anywhere on the game board. In the figure, the hand is shown in the middle of the board, having just placed the round bumper that is next to it.

There are a variety of pinball components available: two sizes of flippers; polygons which the ball just bounces off; bumpers which kick the ball away when they are hit; launchers which are like the spring-operated device that puts the ball into play; a ball hopper which captures balls until it holds three and then releases them all; a ball eater which makes the ball vanish; spinners; lanes; gates; rollovers; and targets – everything you need for a real pinball game.

Each time you pick up a piece, it is replaced with an identical one, so you can, if you choose, create a game with 30 pairs of flippers and 50 bumpers. The only limit is that a maximum of 128 pieces can be placed on the board. It's very unlikely you would ever want more.

## Beyond Pinball

What I have described so far would make a very impressive pinball construction program, but Bill Budge has provided much more. You can change the shape of the board, and the shapes and sizes of the barricades. To do so, you simply move to the arrow tool and press the button to select it. When you select the arrow tool, knobs appear at the corners of each shape. Using the joystick, you can move the arrow tool to a knob and pull that corner out or push it in. The scissors and hammer are for removing and adding knobs so you can, for example, change a rectangle to a triangle or to a pentagon.

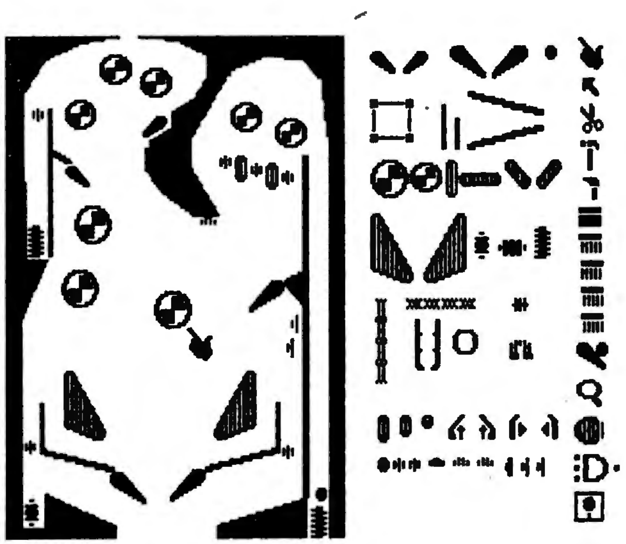
Another tool is the paintbrush. Pick it up, move it to the paint pot with your choice of color, and paint the board or any barricade. There is even a magnifying glass tool for very detailed painting.

Each pinball piece has an associated number of points and a sound that plays when the ball hits it. You can reset these. You can also use AND gates to link parts together for bonus points. That is, you can create effects such as: "If you hit all three of these targets, you get 10,000 bonus points."

Now for the most amazing part, which could be done only with computer pinball. You control the physics of the world in which the game is played! You can set gravity anywhere along a scale from very high to very low. Set gravity to be low, and the ball moves as if it's very light, almost like a Ping-Pong ball. Set gravity to be high, and the ball moves as if it's made of lead – it will even be difficult to shoot it into play with the launcher.

You can also change how much the ball

**Pinball Construction Set**



bounces and how much the bumpers kick. You can play with a lively ball and dead bumpers, a dead ball and lively bumpers, or anything in between. By experimenting with these two controls, you get a good sense of how different factors interact in a physical system.

Finally, you can set the speed. This lets you put the whole game into slow motion. The ball moves the same distance as it would otherwise, but it goes very slowly. Or you can set the game to high speed and really test your reflexes.

*Pinball Construction Set* is remarkably simple to use. Everything is done with the joystick, and almost everything you need to know is represented pictorially. In fact, although it runs on much less expensive machines, the program has aspects of the Lisa and other new, more powerful machines.

With its encouragement of creativity, its visual appeal, its ease of use, the complete control it provides over the world of a pinball game, its inherent physics lessons, and its great fun, *Pinball Construction Set* is a truly remarkable program. If I had to select one program to demonstrate the potential of personal computers to provide playful exercises for the mind, *Pinball Construction Set* would be the one.

I have reviewed the Apple II version of this program, and, by the time this column appears, versions for Atari, Commodore 64, and IBM PC computers will be available. The Apple II version is available from BudgeCo, 428 Pala Ave., Piedmont, CA 94611. All the versions will be available from Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403. ©

# VIC PILOT

Mark Haugan

*Just type in this program and you've got a completely new language you can use with your VIC: Turtle PILOT. For many applications, this language is superior to the BASIC that comes with the computer. If you're interested in a new, easy way to produce startling graphics; or in fractals and recursion; or in introducing Turtle graphics to a youngster – it's all possible with this PILOT and its high-resolution graphics capabilities. You'll need at least an 8K RAM memory expander. The Super Expander is optional.*

It is difficult to exaggerate the interest and excitement being generated by Turtle Graphics and the languages, Logo and PILOT, that support it. Home-computer users, educators, mathematicians, and, of course, kids are all fascinated with "The Turtle." You need look no further than David Thornburg's "Friends of the Turtle" column in each month's COMPUTE! to see evidence of this.

However, if you are a VIC user, you may be feeling left out. Although the VIC has excellent graphics capabilities, no package of Turtle Graphics commands that fully exploit these capabilities seems to be available. The programs included with this article will provide VIC users with a PILOT interpreter and high-resolution Turtle Graphics. You must have at least 8K RAM expansion. If you also have a Super Expander, there is an extra program that will really speed things up.

## The PILOT Interpreter

The PILOT interpreter included here is an extension of the core PILOT interpreter written in BASIC by Michael Tinglof (COMPUTE!, December 1982). His PILOT provides commands for displaying written information on the screen and for accepting and testing responses from the keyboard. To this core I've added a set of Turtle Graphics commands which control the location, heading, and motion of an imaginary turtle that inhabits the graphic screen.

The turtle can leave a trail as it moves around the screen. The trail forms the graphic design. The interpreter understands commands which control whether a trail is left along any particular portion of the turtle's path; its color, assuming a trail is left; and the colors of VIC's screen and border. Most people find the "Turtle" approach to graphics simpler than the "Cartesian" approach (turn on the pixel, or dot, at screen coordinate x,y) because they can imagine themselves in the place of the turtle and "walk through" a desired design as an aid to programming it.

Two versions of the interpreter are provided. Program 1 will run on a VIC with 8K or more expansion RAM added. It provides a 160x176 pixel high-resolution graphics screen and roughly 2K bytes for PILOT programs. You can, of course, add memory as you like. This version of the interpreter plots the path of the turtle point-by-point in a fashion that will be familiar to anyone who has worked with VIC's high-resolution screen. The result is a nice, sharp graphic display, but the procedure is slow – it provides *turtle* graphics in every sense of the word.

If you have the Super Expander cartridge in addition to at least 8K of expansion RAM, type in Program 1, but make the substitutions shown in Program 2. The machine language graphics routines of the Super Expander are used to overcome the speed problem of the other PILOT version. Typical Turtle Graphics programs now run in tens of seconds. Even when the turtle's path consists of an immense number of tiny steps and plotting may take a few minutes, the Super Expander version runs about twice as fast as the first. If you are planning to work with a young child with a short attention span, this extra speed

could be very important. For that matter, anyone wanting to experiment extensively with Turtle Graphics would probably prefer a faster turtle.

The Super Expander version provides a 160x160 pixel high-resolution graphics screen and with an 8K RAM cartridge you have, once again, roughly 2K bytes for PILOT programs.

Toward the end of this article we'll discuss a few PILOT programs to demonstrate some of the capabilities of this interpreter. But first, let's talk about the turtle commands that the interpreter understands and, also, briefly review the operation of the editor and the PILOT commands, instructions, and variable conventions that are inherited from Michael Tinglof's interpreter.

## The Editor

The PILOT editor is precisely like the BASIC editor. To enter a program line, type the line number, the PILOT statement, and hit RETURN. To correct an error, move the cursor to it, type the correction, and hit RETURN. Alternatively, you may simply reenter the entire program line. As in BASIC, the editor assumes that anything entered without a line number is a command.

## The Commands

The editor understands the following commands:

- **LIST xx-yy** – Lists the program lines between the specified line numbers. Either or both of the line numbers may be absent.
- **RUN** – Executes the PILOT program in memory.
- **SAVE name** – Saves the program in memory on cassette.
- **LOAD name** – Loads the program from cassette.
- **NEW** – Clears program memory.
- **BASIC** – Exits the interpreter and returns to BASIC.
- **PLIST xx-yy** – Same as the list command except that output is sent to the printer, device 4.

Note that command names may be shortened, even to a single letter. For example,

**L 10-25** for **LIST 10-25**
**R** for **RUN**.

## PILOT Instructions

PILOT statements, with the exception of labels, consist of an instruction name, an optional conditioner, a colon, and an object. The object is simply everything that follows the colon and is optional with some instructions.

The interpreter understands the following PILOT instructions:

**T:** The TYPE instruction prints everything in the object on the screen. This may be text or variables. For example,

```
10 T:ANGLE=#A
```

prints "ANGLE=xx" where xx is the value for the numerical variable #A. Note that no carriage return will be printed if a T: instruction is ended with a ";".

**A:** The ACCEPT instruction inputs a response from the user. The user must hit RETURN to complete a response. The object of an A: instruction may be a numerical or string variable, but no object is necessary. The user's response will be assigned as the variable's value if an object is given. It will be assigned to a buffer that can be used by the MATCH instruction when no object is present.

```
15 A:
20 A:$V
```

**M:** The MATCH instruction checks to see if certain strings are present in the contents of the Accept buffer or in a string variable. If so, the Y-conditioner flag is set. If not, the N flag is set. For example,

```
15 M:12,TWELVE,XII,1100,$OC
```

sets the Y flag if any of these representations of twelve is present in the Accept buffer, while

```
20 M:$L,SUPERIOR,MICHIGAN,HURON,ERIE,ONTA
  RIO
```

sets the Y flag if the string variable $L contains any one of these Great Lakes names.

**I:** The IF instruction is a nonstandard instruction implemented by Michael Tinglof to allow mathematical testing to set the Y and N flags. It can check to see if a given variable is greater than, less than, or equal to a given value or a second variable. The Y flag is set if the expression in the object of the instruction is true. Otherwise, the N flag is set. Only =, <, and > can be used in expressions. Sample instructions are

```
30 I:#N=9
35 I:#N<#L
```

**J:** and **U:** The JUMP and USE instructions are the analogues of BASIC's GOTO and GOSUB statements. However, either labels or line numbers may be used in PILOT to specify where in a program these instructions are to transfer control.

```
35 J:5
20 U:*SHIFT
```

**E:** The END instruction is the analogue of BASIC's RETURN statement. It transfers control to the program line following the last U: instruction executed by PILOT.

**C:** The COMPUTE instruction performs simple four-function calculations in linear order (no parentheses). The object of this instruction must be an equation specifying the value of a numerical variable. The expression on the right-hand side of the equation is evaluated and the value of the variable is set to the result.

```
15 C:#N=#G*10/#T+15
```

Note that if #R is encountered in the expression,

its value will be set to a random number between 0 and 1.

R: The REMARK instruction is not executed. Its object may be any desired program documentation.

H: The HOME instruction clears the text screen and returns the cursor to home.

G: The GRAPHICS instruction takes as its object any of the turtle commands discussed below. For example,

```
12 G:DRAW 50
```

END This instruction stops execution of a PILOT program and returns control to the editor. It may not be abbreviated and is the only instruction that cannot be modified by a Y or N conditioner.

```
100 END
```

**Conditioners:** PILOT instructions can be modified by the addition of a Y or N conditioner. For example,

```
50 TY:VERY GOOD $N.
60 JN:*START
```

Y-conditioned instructions will be executed only if the Y flag is set. Similarly, N-conditioned instructions will be executed only if the N flag is set. Remember that these flags are set by MATCH and IF instructions.

**Labels:** These are designated by beginning a line with *. For example,

```
.
.
10 *LOOP START
.
.
25 JY:*LOOP START
.
.
```

## PILOT Variables

The interpreter recognizes both string and numerical variables. String variable names consist of a $ followed by a single letter. Numerical variables are integer variables. Their names consist of a # followed by a single letter.

## Turtle Commands

Each of the commands described here must be preceded by a G: instruction. Command names may be abbreviated, even to a single letter, although, as we'll see, other parts of commands such as color names may not be shortened.

CLEAR – This command sets and clears the VIC's high-resolution screen. It initializes the color of the screen to white, the border to blue, and the color of the turtle's trail to black. The CLEAR command also initializes the turtle's heading to zero degrees, north, and its location to center screen, X and Y coordinates (0,0). The CLEAR command must be the first in any graphics routine.

```
G:CLEAR
GY:C
```

TURN – The TURN command is followed by a number or a numeric variable. The number or the value of the variable is the number of degrees added to the turtle's current heading. A positive value turns the turtle clockwise.

```
G:TURN -270
G:T #A
```

TURNTO – This command sets the turtle's heading to the specified angle. The word TURN in TURNTO may be abbreviated, but TO must be included at the end of any abbreviation of TURNTO. For example,

```
G:TURNTO 90
G:TTO #A
```

DRAW – The DRAW command moves the turtle the specified distance along its current heading. The turtle will leave a trail if its pen is down (see the PEN command below). When using the Super Expander version, program execution will cease and you will receive a warning message if you attempt to DRAW off screen. With the other version, the turtle will proceed off screen. You will receive a message informing you that the turtle left the screen at some point during program execution when you return to text mode.

```
G:DRAW 50
G:D #L
```

GO – The GO command moves the turtle the specified distance along its current heading without leaving a trail. The command is equivalent to DRAW with PEN UP.

```
GN:GO 45
G:G #D
```

GOTO – This command moves the turtle to the specified screen coordinates without changing its heading. The X and Y coordinates are separated by a comma in the GOTO statement. The range of X coordinates on the screen is -106.65 to 108, and the range of Y coordinates is -87 to 88. When using the Super Expander interpreter, the Y coordinates are -79 to 80.

```
G:GOTO #X,#Y
G:GTO -15,35
```

PEN – The PEN command controls the color of the turtle's trail on the screen. With the VIC, it is possible to use several pen colors on a single graphics display. Allowed color names are BLACK, WHITE, RED, CYAN, PURPLE, GREEN, BLUE, and YELLOW. If the PEN command is followed by the word ERASE, the pen color is set to the screen's current color. The PEN command may also be followed by the words UP and DOWN. PEN UP causes DRAW commands to move the turtle without leaving a trail. PEN DOWN returns the pen to normal. Note that color names and the other pen control words may not be abbreviated.

```
G:PEN GREEN
G:P DOWN
```

**SCREEN** – This command changes the color of the graphics screen without clearing it. The same colors are available as for the PEN command.

```
G:SCREEN RED
G:S CYAN
```

**BORDER** – This command controls the color of VIC's screen border. Once again, the colors already mentioned are available.

```
G:BORDER YELLOW
G:B RED
```

**QUIT** – The QUIT command returns the text screen. When this command is encountered, the graphics screen will be held until you enter Q from the keyboard. This lets you control the amount of time you spend admiring your turtle handiwork.

In general, QUIT *must* be the final command of a graphics routine. The only exception occurs when the turtle is sent along an infinite path (it may loop back on itself). In this case a QUIT command would never be reached, and you exit graphics mode by hitting @.

```
G:QUIT
GY:Q
```

One structure that occurs frequently in Turtle Graphics programs is a sequence of DRAW and TURN commands. This is done to draw polygons of various types. The interpreter understands one compound command that performs this task easily.

```
G:xx(DRAW yy;TURN zz)
```

xx must be an integer. yy and zz may be integers or integer variables as for single DRAW and TURN commands. For example,

```
G:9(D 50;T 160)
G:5(DRAW #L;TURN 72)
```

## Program Operation

All of VIC's internal memory is required to produce the high-resolution screen for Turtle Graphics. For this reason, the start of BASIC must be moved to location 8192, the beginning of BLK1 of expansion RAM, before loading and running either version of the interpreter. This is accomplished by typing in this direct statement before doing anything else:

```
POKE 44,32:POKE 642,32:POKE 8192,0:NEW
```

The interpreter takes up about 5K of RAM memory, and 1K is required for system initialization. So, with 4K allocated to screen and programmable character memory, you can see why there is only 2K left for PILOT programs on an 8K-expanded VIC. Note that the maximum allowable number of PILOT program lines is contained in the variable M in line 6 of the interpreter. This number may be changed.

To stop any PILOT program you simply hit the @ key. If you are in graphics mode, the text screen will automatically return. Note that the @ is accepted only when execution of the current PILOT program line has been completed. If this line should be, for example, a long turtle loop like

```
G:180(D 1;T 2)
```

there will be a noticeable delay before the program halts.

If for any reason the program returns to BASIC, you may reenter the interpreter without losing the current PILOT program by typing GOTO 11 and hitting RETURN. This means that you may hit RUN/STOP and RESTORE to regain control if the interpreter "locks up " (if, for example, you forget a G:QUIT statement and get stuck with the graphics display on screen). You then type GOTO 11 and hit RETURN to resume work on your program.

When loading PILOT programs (if the NEW command has not been given) the current program and the new one are merged. If you wish to operate the interpreter with disk rather than cassette storage, the following program changes are required:

```
41 OPEN1,8,2,R$+",S,W":PRINT"SAVING "R$
45 OPEN1,8,2,R$+",S,R":PRINT"LOADING "R$
```

In addition, to save a program on drive 0 the syntax of the SAVE command must be altered to

**SAVE 0:name**

The following error codes may be generated when a PILOT program is run:

1. Illegal variable name
2. Unknown label
3. Stack overflow (too many USES)
4. Stack empty (an E: without a USE)
5. Bad format
6. Division by zero
7. Numerical variable out of range (magnitude greater than 32767)
8. CLEAR not the first graphic command

Finally, it should be remarked that the PILOT interpreter is not as indifferent about spaces scattered through program lines as the BASIC interpreter is. The PILOT interpreter will remove spaces before a line number or a command and will remove extra spaces between line numbers and instruction names. However, extra spaces elsewhere in a program line may confuse the interpreter. Also, spaces as shown in the sample commands are *necessary*. For example, there must be a space between DRAW and #L in

```
50 G:DRAW #L
```

## Sample Turtle Graphics Programs

The three sample programs here serve to demonstrate the graphics capabilities of this PILOT interpreter. There's a little something for everyone: a typical turtle pattern made of shifted and rotated squares, a picture for the kids, and a recursive binary tree program for the mathematically minded. Although no abbreviations are used in Program 3 (to make it easy to follow), they are included in the other two programs to demonstrate their use.

"Pretty Pattern" (Program 3) draws a picture that is typical of turtle designs made up of simple polygons. In this case the polygons are squares, and they are shifted and rotated relative to each other to form the design. A star is formed by the overlapping squares at the center of the pattern. After drawing six squares, the turtle returns to its initial location and heading. The program is written so that the turtle loops around its six-square path again and again, forever. As a result, no G:QUIT and END statements are needed. Exit the program by hitting the @ key. It only takes about 20 seconds for the "fast" (Super Expander) turtle to make its way around the design.

Program 4, "Teddy Bear," is fun for children. Fairly rough circles are used in the design to reduce the time for drawing to 90 seconds with the "fast" turtle. When the program reaches the G:QUIT statement in line 29, execution will halt until you hit the "Q" key. Note that it is quite easy to turn this bear into a rabbit by designing ears made using two quarter circles for each ear.

Finally, for those of you who are interested in recursion, Program 5, "Recursive Tree," draws a simple binary tree. The way in which the tree is drawn by the procedure *BRANCH is of particular interest. This procedure calls itself repeatedly. To understand how this is done using only global variables, it is helpful to study the listing and to run the program. When you run it, select final branch level 1, then level 2, and so on to see the order in which the branches are drawn. The VIC's screen resolution produces nice pictures of the tree up to level 5 and even level 6.

## Further Suggestions

My goal while developing this PILOT interpreter was to make Turtle Graphics available on a VIC with only 8K bytes of expansion RAM added. I have "crunched" the program to achieve this goal (the few REMs scattered through the listings are to keep the line numbers of the two versions of the interpreter aligned), but there are a few features I simply could not squeeze in. If you have more memory and the inclination, you might want to extend the program.

I regret not being able to include the capability for mixing text and graphics on the high-resolution screen. This means that you really cannot run a program like VISITURT ("Friends of the Turtle," COMPUTE!, April 1982) which makes the turtle interactive. The necessary prompts cannot be written onto the graphics screen.

This is unfortunate because an interactive turtle would be very nice for children to work with. It would, however, be fairly simple to add a mixed text-graphics mode, if you have access to the Super Expander command CHAR. Other possible additional features include adding a SOUND command like the one in Atari PILOT or the ability to use VIC's multicolor mode.

### Program 1: PILOT Interpreter

```
0 GOTO4
1 I$=""
2 SYS820:IFPEEK(0)=13THENRETURN
3 I$=I$+CHR$(PEEK(0)):GOTO2
4 POKE36866,150:POKE36869,240:POKE648,30
5 FORJ=217TO228:POKEJ,158:NEXT:FORJ=229T
  O250:POKEJ,159:NEXT
6 CLR:M=200:DIMS%(9),N%(26),S$(26),L$(M)
  ,C$(17),G$(7),B$(10)
7 PRINT"{CLR}{BLK} **** PILOT V2.1 ****"
  :FORX=820TO825:READZ:POKEX,Z:NEXT:FORX
  =0TO17
8 READC$(X):NEXT:FORX=0TO7:READG$(X):NEX
  T:FORX=0TO10:READB$(X):NEXT:DATA32,207
  ,255,133
9 DATA0,96,LIST,RUN,SAVE,LOAD,NEW,BASIC,
  PLIST,T,J,E,U,M,C,A,I,H,R,G,CLEAR,QUIT
  ,TURN
10 DATADRAW,GO,PEN,SCREEN,BORDER,BLACK,W
   HITE,RED,CYAN,PURPLE,GREEN,BLUE,YELLO
   W,ERASE,UP
11 PRINT"{DOWN}PILOT.":DATADOWN
12 GOSUB1:PRINT:IFASC(I$)=32ANDLEN(I$)=1
   THEN12
13 IFLEFT$(I$,1)=" "THENI$=MID$(I$,2):GO
   TO13
14 L=VAL(I$):IFL<>0THEN23
15 L=1:H=M:R$="":FORX=1TOLEN(I$):IFMID$(
   I$,X,1)<>" "THENNEXT:GOTO21
16 R$=MID$(I$,X+1):I$=LEFT$(I$,X-1)
17 L=VAL(R$):H=L:FORX=1TOLEN(R$):IFMID$(
   R$,X,1)<>"-"THENNEXT:GOTO19
18 L=VAL(LEFT$(R$,X-1)):H=VAL(MID$(R$,X+
   1))
19 IFL=0THENL=1
20 IFH=0THENH=M
21 FORX=0TO6:IFI$<>LEFT$(C$(X),LEN(I$))T
   HENNEXT:PRINT"UNKNOWN COMMAND.":GOTO1
   1
22 ONX+1GOTO32,51,41,45,49,50,31
23 IFL>MTHENPRINT"LINE NUMBER OUT OF RAN
   GE.":GOTO11
24 X=LEN(STR$(L)):X$=MID$(I$,X):IFX$=""T
   HENL$(L)="":GOTO12
25 IFLEFT$(X$,1)=" "THENX$=MID$(X$,2):GO
   TO25
26 X=3:IFMID$(X$,2,1)<>":"THENX=4:IFMID$
   (X$,3,1)<>":"THENL$(L)=X$:GOTO12
27 FORZ=7TO17:IFLEFT$(X$,1)<>C$(Z)THENNE
   XT:PRINT"ILLEGAL COMMAND.":GOTO11
28 IFMID$(X$,2,1)="Y"THENZ=Z+13
29 IFMID$(X$,2,1)="N"THENZ=Z+26
```

```
30 L$(L)=CHR$(Z-6)+MID$(X$,X):GOTO12
31 OPEN1,4:GOTO33
32 OPEN1,3
33 FORX=LTOH:IFL$(X)=""THEN39
34 X$=":":Z=ASC(L$(X)):IFZ>40THENX$=LEFT
   $(L$(X),1):GOTO38
35 IFZ>26THENZ=Z-26:X$="N"+X$
36 IFZ>13THENZ=Z-13:X$="Y"+X$
37 X$=C$(Z+6)+X$
38 PRINT#1,X;X$;MID$(L$(X),2)
39 GETX$:IFX$<>""THENCLOSE1:GOTO11
40 NEXT:CLOSE1:GOTO11
41 OPEN1,1,1,R$:PRINT"SAVING "R$
42 FORX=1TOM:IFL$(X)=""THEN44
43 PRINT#1,X;CHR$(13)CHR$(34)L$(X)CHR$(3
   4)CHR$(13);
44 NEXTX:CLOSE1:GOTO11
45 OPEN1,1,0,R$:PRINT"LOADING "R$
46 INPUT#1,X:IFSTTHEN48
47 INPUT#1,L$(X):IFST=0THEN46
48 CLOSE1:GOTO11
49 GOTO6
50 PRINT"{DOWN}EXITING TO BASIC....":END
51 L=0:FORX=1TO26:N%(X)=0:S$(X)="":NEXT:
   P=0:F%=0
52 L=L+1:IFL=>MORL$(L)="END"THEN11
53 GETX$:IFX$="@"ANDCQ%=0THEN11
54 IFX$="@"ANDCQ%=1THENGOSUB127:GOTO11
55 IFL$(L)=""THEN52
56 X=ASC(L$(L)):IFX>40THEN52
57 IFX>26THENX=X-26:IFF%=1THEN52
58 IFX>13THENX=X-13:IFF%=0THEN52
59 C$=MID$(L$(L),2)
60 ONXGOTO62,73,76,71,78,85,101,106,115,
   52,116
61 PRINT"ERROR #"E"IN LINE"L:GOTO11
62 Z=0:IFRIGHT$(C$,1)=";"THENZ=1:C$=LEFT
   $(C$,LEN(C$)-1)
63 FORX=1TOLEN(C$):X$=MID$(C$,X,1):IFX$=
   "#"THEN67
64 IFX$="$"THEN68
65 PRINTX$;:NEXT:IFZ=0THENPRINT
66 GOTO52
67 GOSUB69:X$=STR$(N%(Y)):GOTO65
68 GOSUB69:X$=S$(Y):GOTO65
69 X=X+1:Y=ASC(MID$(C$,X,1))-64:IFY<1ORY
   >26THENE=1:GOTO61
70 RETURN
71 IFP>8THENE=3:GOTO61
72 P=P+1:S%(P)=L
73 IFVAL(C$)<>0THENL=VAL(C$)-1:GOTO52
74 FORX=1TOM:IFC$<>L$(X)THENNEXT:E=2:GOT
   O61
75 L=X:GOTO52
76 IFP=0THENE=4:GOTO61
77 L=S%(P):P=P-1:GOTO52
78 X=1:C$=C$+",":X$=AC$:IFLEFT$(C$,1)="$
   "THENGOSUB83
79 FORZ=XTOLEN(C$):IFMID$(C$,Z,1)<>","TH
   ENNEXT
80 Z$=MID$(C$,X,Z-X):FORY=1TOLEN(X$):IFM
   ID$(X$,Y,LEN(Z$))=Z$THENF%=1:GOTO52
81 NEXT:IFZ<LEN(C$)THENX=Z+1:GOTO79
82 F%=0:GOTO52
83 Y=ASC(MID$(C$,2))-64:IFY<1ORY>26THENE
   =1:GOTO61
84 X$=S$(Y):X=4:RETURN
85 A=3:Z=0:X$="":IFLEFT$(C$,1)<>"#"ORMID
   $(C$,3,1)<>"="THENE=5:GOTO61
86 Y=1:X$=MID$(C$,A,1):A=A+1:IFMID$(C$,A
   ,1)="-"THENA=A+1:Y=-1
87 IFMID$(C$,A,1)<>"#"THENY=Y*VAL(MID$(C
   $,A)):A=A+LEN(STR$(Y))-1:GOTO91
88 X=ASC(MID$(C$,A+1))-64:IFX<1ORX>26THE
   NE=1:GOTO61
89 IFX=18THENY=Y*RND(1):GOTO91
90 Y=Y*N%(X):A=A+2
91 IFX$="="THENZ=Y
92 IFX$="-"THENZ=Z-Y
93 IFX$="+"THENZ=Z+Y
94 IFX$="/"ANDY=0THENE=6:GOTO61
95 IFX$="*"THENZ=Z*Y
96 IFX$="/"THENZ=Z/Y
97 IFA<=LEN(C$)THEN86
98 X=ASC(MID$(C$,2))-64:IFX<1ORX>26THENE
   =1:GOTO61
99 IFZ>32767ORZ<-32767THENE=7:GOTO61
100 N%(X)=Z:GOTO52
101 IFC$=""THENGOSUB1:AC$=I$:PRINT:GOTO5
    2
102 X=ASC(MID$(C$,2))-64:IFX<1ORX>26THEN
    E=1:GOTO61
103 GOSUB1:Z=VAL(I$):PRINT:IFLEFT$(C$,1)
    ="#"THENN%(X)=Z
104 IFLEFT$(C$,1)="$"THENS$(X)=I$
105 GOTO52
106 IFLEFT$(C$,1)<>"#"THENE=5:GOTO61
107 X=ASC(MID$(C$,2))-64:IFX<1ORX>26THEN
    E=1:GOTO61
108 A=N%(X):X$=MID$(C$,3,1):IFMID$(C$,4,
    1)<>"#"THENX=VAL(MID$(C$,4)):GOTO111
109 X=ASC(MID$(C$,5))-64:IFX<1ORX>26THEN
    E=1:GOTO61
110 X=N%(X)
111 F%=0:IFX$="<"ANDA<XTHENF%=1
112 IFX$=">"ANDA>XTHENF%=1
113 IFX$="="ANDA=XTHENF%=1
114 GOTO52
115 PRINT"{CLR}";:GOTO52
116 Y=0:FORZ=1TOLEN(C$):IFMID$(C$,Z,1)<>
    " "THENNEXT:GOTO121
117 R$=MID$(C$,Z+1):C$=LEFT$(C$,Z-1):IFR
    IGHT$(C$,2)="TO"THENY=1:C$=LEFT$(C$,
    LEN(C$)-2)
118 FORZ=1TOLEN(R$):X$=MID$(R$,Z,1):IFX$
    <>","ANDX$<>";"THENNEXT:GOTO121
119 IFX$=";"THEN130
120 X$=LEFT$(R$,Z-1):R$=MID$(R$,Z+1)
121 FORZ=0TO7:IFC$<>LEFT$(G$(Z),LEN(C$))
    THENNEXT:GOTO126
122 IFCQ%=0ANDZ<>0THENE=8:GOTO61
123 IFZ=2ANDY=1THEN167
124 IFZ=4ANDY=1THEN168
125 ONZ+1GOTO139,142,145,147,157,158,163
    ,165
126 GOSUB127:PRINT"UNKNOWN GRAPHICS":PRI
    NT"COMMAND IN LINE ";L:GOTO11
127 CQ%=0:POKE36864,5:POKE36866,150:POKE
    36867,46:POKE36869,240:POKE36879,27
128 IFOS%=1THENPRINT"*PLOT WENT OFF SCRE
    EN"
129 PRINT"{CLR}{BLK}";:RETURN
130 D=VAL(C$):Y=LEN(STR$(D)):C$=MID$(C$,
    Y+1):IFD<=0THEN126
131 IFC$<>LEFT$(G$(3),LEN(C$))THEN126
132 FORZ=1TOLEN(R$):IFMID$(R$,Z,1)<>";"T
    HENNEXT:GOTO126
133 C$=LEFT$(R$,Z-1):X$=MID$(R$,Z+1)
```

```
134 FORZ=1TOLEN(X$):IFMID$(X$,Z,1)<>" "T
    HENNEXT:GOTO126
135 R$=MID$(X$,Z+1):X$=LEFT$(X$,Z-1):IFX
    $<>LEFT$(G$(2),LEN(X$))THEN126
136 X$=R$
137 R$=C$:GOTO147
138 R$=X$:GOTO145
139 CQ%=1:UD%=Ø:OS%=Ø:POKE36864,7:POKE36
    866,148:POKE36867,23
14Ø POKE36869,252:POKE36879,3Ø:CO=Ø:SC=2
    :BC=6:AN=Ø:XØ=Ø:YØ=Ø
141 FORI=ØTO219:POKE768Ø+I,I:NEXT:FORI=4
    Ø96TO7615:POKEI,Ø:NEXT:GOTO52
142 GETX$:IFX$<>"Q"THEN142.
143 REM
144 GOSUB127:GOTO52
145 GOSUB169:AN=AN+Z:D=D-1:IFD>ØTHEN137
146 D=Ø:GOTO52
147 GOSUB169:IFZ<ØTHEN126
148 TH=(9Ø-AN)*3.1415926/18Ø
149 FORY=ØTOZ:XG=XØ+Y*COS(TH):YG=YØ+Y*SI
    N(TH):IFUD%=ØTHENGOSUB152
15Ø NEXT:XØ=XG:YØ=YG:IFD>ØTHEN138
151 GOTO52
152 U=INT((XG+1Ø6.65)/1.35+.5):V=88-INT(
    YG+.5)
153 CH=INT(V/16)*2Ø+INT(U/8):RO=(V/16-IN
    T(V/16))*16
154 IFCH<ØORCH>22ØORXG<-1Ø6.65ORXG>1Ø8TH
    ENOS%=1:RETURN
155 BY=4Ø96+16*CH+RO:BI=7-(U-INT(U/8)*8)
156 POKE384ØØ+CH,CO:POKEBY,PEEK(BY)OR(2↑
    BI):RETURN
157 GOSUB169:TH=(9Ø-AN)*3.14159265/18Ø:X
    Ø=XØ+Z*COS(TH):YØ=YØ+Z*SIN(TH):GOTO5
    2
158 FORZ=ØTO1Ø:IFR$<>B$(Z)THENNEXT:GOTO1
    26
159 IFZ<8THENCO=Z:GOTO52
16Ø IFZ=8THENCO=SC-1:GOTO52
161 IFZ=9THENUD%=1:GOTO52
162 IFZ=1ØTHENUD%=Ø:GOTO52
163 FORZ=ØTO7:IFR$<>B$(Z)THENNEXT:GOTO12
    6
164 SC=Z+1:POKE36879,SC*16+BC-8:GOTO52
165 FORZ=ØTO7:IFR$<>B$(Z)THENNEXT:GOTO12
    6
166 BC=Z:POKE36879,SC*16+BC-8:GOTO52
167 GOSUB169:AN=Z:GOTO52
168 GOSUB169:YØ=Z:R$=X$:GOSUB169:XØ=Z:GO
    TO52
169 Z=VAL(R$):IFZ<>ØORR$="Ø"THEN173
17Ø IFLEN(R$)<>2ORLEFT$(R$,1)<>"#"THENE=
    1:GOSUB127:GOTO61
171 Y=ASC(RIGHT$(R$,1))-64:IFY<ØORY>26TH
    ENE=1:GOSUB127:GOTO61
172 Z=N%(Y)
173 RETURN
```

## Program 2: Changes For Super Expander

```
4 GRAPHICØ:COLOR1,3,Ø,Ø
5 REM
127 CQ%=Ø:GRAPHIC4:COLOR1,3,Ø,Ø:SCNCLR:RE
    TURN
128 REM
129 REM
139 CQ%=1:UD%=Ø:OS%=Ø:GRAPHIC2:COLOR1,6,Ø
    ,Ø:SCNCLR:CO=Ø:SC=1:BC=6:AN=Ø:XØ=Ø:Y
    Ø=Ø:GOTO52
14Ø REM
141 REM
143 GOSUB127:IFOS%=1THEN11
144 GOTO52
147 GOSUB169
148 REM
149 REM
15Ø TH=(9Ø-AN)*3.1415926/18Ø:XG=XØ+Z*COS(
    TH):YG=YØ+Z*SIN(TH):IFUD%=ØTHENGOSUB
    154
151 IFOS%=1THEN:CHAR18,Ø,"OFF SCREEN AT L
    INE{2 SPACES}"+STR$(L)+" : HIT Q":GO
    TO142
152 XØ=XG:YØ=YG:IFD>ØTHEN138
153 GOTO52
154 IFYØ<-79ORYG<-79ORYØ>8ØORYG>8ØORXØ<-1
    Ø6ORXG<-1Ø6ORXØ>1Ø8ORXG>1Ø8THENOS%=1
    :RETURN
155 UØ=1Ø23*(XØ+1Ø6.65)/(1.35*159):VØ=1Ø2
    3*(8Ø-YØ)/159:U=1Ø23*(XG+1Ø6.65)/(1.
    35*159)
156 V=1Ø23*(8Ø-YG)/159:DRAW1,UØ,VØTOU,V:R
    ETURN.
159 IFZ<8THENCO=Z:REGIONZ:GOTO52
16Ø IFZ=8THENCO=SC:REGIONCO:GOTO52
164 SC=Z:COLORZ,BC,CO,Ø:GOTO52
166 BC=Z:COLORSC,Z,CO,Ø:GOTO52
```

## Program 3: Pretty Pattern

```
1 *PRETTY PATTERN
2 G:CLEAR
3 G:SCREEN RED
4 G:GO 13
5 G:TURN 6Ø
6 G:GO -17
7 U:*SHIFT SQUARE
8 J:7
1Ø *SHIFT SQUARE
11 G:GO 17
12 G:TURN 6Ø
13 G:4(DRAW 6Ø;TURN 9Ø)
14 E:
```

## Program 4: Teddy Bear

```
1 *TEDDY BEAR
2 G:C
3 G:TTO -9Ø
4 I:#C=2
5 JY:*FACE
6 U:*1/3 BIG CIRCLE
7 U:*LOCATE FOOT/EAR
8 U:*FOOT/EAR
9 U:*LOCATE FOOT/EAR
1Ø U:*1/3 BIG CIRCLE
11 U:*LOCATE FOOT/EAR
12 U:*FOOT/EAR
13 U:*LOCATE FOOT/EAR
14 U:*1/3 BIG CIRCLE
15 G:T 18Ø
16 C:#C=#C+1
17 J:4
18 *FACE
19 G:GTO 7,3Ø
2Ø G:TTO Ø
21 U:*EYE
22 G:GTO -7,3Ø
23 G:T 18Ø
24 U:*EYE
```

```
25 G:GTO Ø,22
26 G:4(D 1;T 9Ø)
27 G:GTO 12,2Ø
28 G:1Ø(D 4;T 18)
29 G:Q
3Ø END
35 *1/3 BIG CIRCLE
36 G:15(D 4;T 8)
37 E:
4Ø *FOOT/EAR
41 G:2Ø(D 4;T 18)
42 E:
45 *EYE
46 G:15(D 2;T 24)
47 E:
5Ø *LOCATE FOOT/EAR
51 G:T 9Ø
52 G:G 6
53 G:T 9Ø
54 E:
```

## Program 5: Recursive Tree

```
1 *RECURSIVE TREE
2 H:
3 T:FINAL BRANCH LEVEL ?
4 A:#L
5 G:C
6 C:#B=64
7 G:G -64
8 U:*BRANCH
9 G:Q
1Ø END
15 *BRANCH
16 I:#L=Ø
17 JY:3Ø
18 G:D #B
19 C:#B=#B/2
2Ø C:#L=#L-1
21 G:T -45
22 U:*BRANCH
23 G:T 9Ø
24 U:*BRANCH
25 G:T -45
26 C:#B=#B*2
27 C:#L=#L+1
28 C:#A=-#B
29 G:G #A
3Ø E:
```

©

# Telengard

Tony Roberts
Assistant Managing Editor

*Telengard* is a fantasy, role-playing game that requires a good memory, the ability to think quickly, and hours and hours to play.

Telengard is a dungeon, 50 levels deep. It is littered with treasures and crawling with monsters. Your purpose is to enter the dungeon, gather treasures, gold, and experience, and come out alive. You encounter monsters and traps, fall into pits, and wander into teleportation chambers that send you who knows where. You have your strength, your magic, and your wits to help you survive.

This Dungeon-and-Dragons-like game, which is both complicated and intriguing, is available from Avalon Hill for the Atari, Apple, PET, and Commodore 64 computers. It plays the same on any computer, but the 64 version, with its graphic representation of the monsters and dungeon hazards, has the most flair.

*Telengard*'s complexity is indicated by the 24-page instruction manual, most of which is spent explaining what you'll encounter in the dungeon and how to cope with it. Learning the features of the dungeon, the characteristics of the creatures that inhabit it, and the weapons and magic at your disposal is crucial to the game.

Another complicating factor is time. On each of your moves, you have a limited amount of time (about five seconds) to decide how to proceed. The world of Telengard does not stop if you are indecisive. If you fail to initiate action, the forces of the dungeon will choose a path for you.

## The Characters

You are the adventurer in *Telengard*, and every time you play, you are endowed with different characteristics, each of which affects your performance in the dungeon.

These characteristics are: strength, which determines your success during combat; intelligence, which has a bearing on how well you cast magical spells; wisdom, which governs your ability to cast healing spells and to successfully use spells on "undead" creatures; constitution, a factor directly related to how much injury you can sustain in battle; dexterity, a measure of your ability to run when necessary; and charisma, which affects the way some creatures react to you.

When you begin your descent into the dungeon, you are a Level 1 adventurer. As you accumulate experience, gained by successfully fighting monsters and collecting gold, you advance to higher levels, giving you the stamina needed to carry you further into the maze of tunnels, as well as a larger array of spells to help you out of tight spots.

## Monsters And Spells

There are 20 monsters in *Telengard*, each dangerous in its own way. Some are living monsters, fighters, elves, and dragons, for example, and others are undead. These undead creatures – mummies, wraiths, and specters, among others – require an entirely different approach in battle.

Each monster has its quirks, and, like each character, monsters have levels. A Level 1 Kobold may not be much to fear, but for a Level 1 adventurer, a Level 23 Kobold can be a handful. The outcome of your encounters with the monsters depends on the combination of the monster's level, its characteristics, your character's attributes, and how you choose to fight the monster.

There are 36 spells available to *Telengard* adventurers. At the start of a game, a character can use the first six. Access to the others is reserved for more experienced characters. Among the spells are Magic Missiles, Invisibility, Continual Light, Finger of Death, and Wall of Fire. Some spells can be used only in battle; others are "duration spells" that give an adventurer extra power for several turns.

## The Dungeon

Once the game begins, you find yourself deposited in the dungeon, directly below an inn, a place you'll return to often – if you can remember where it is. On the right side of the screen is a report showing your character's attributes, your collection of treasures, your gold, and your experience points.

You play this adventure in a series of two-part turns. Part one is the action phase in which you decide whether to move or stay put. If you move, the dungeon's maze is redrawn around you, and you're thrown into the

# PRO SPORTS STATS

## LET YOUR HOME COMPUTER GIVE YOU "THE WINNING EDGE"

**IT'S 100% GUARANTEED**
**IT WORKS OR YOU DON'T PAY**

LET YOUR HOME COMPUTER HELP YOU PICK SUNDAY'S WINNERS ON SATURDAY.

**AVAILABLE NOW THROUGH MODERN COMPUTER TECHNOLOGY**
**COMPLETE PROFESSIONAL FOOTBALL RECORDS FROM 1970 — RIGHT UP TO THE MINUTE**

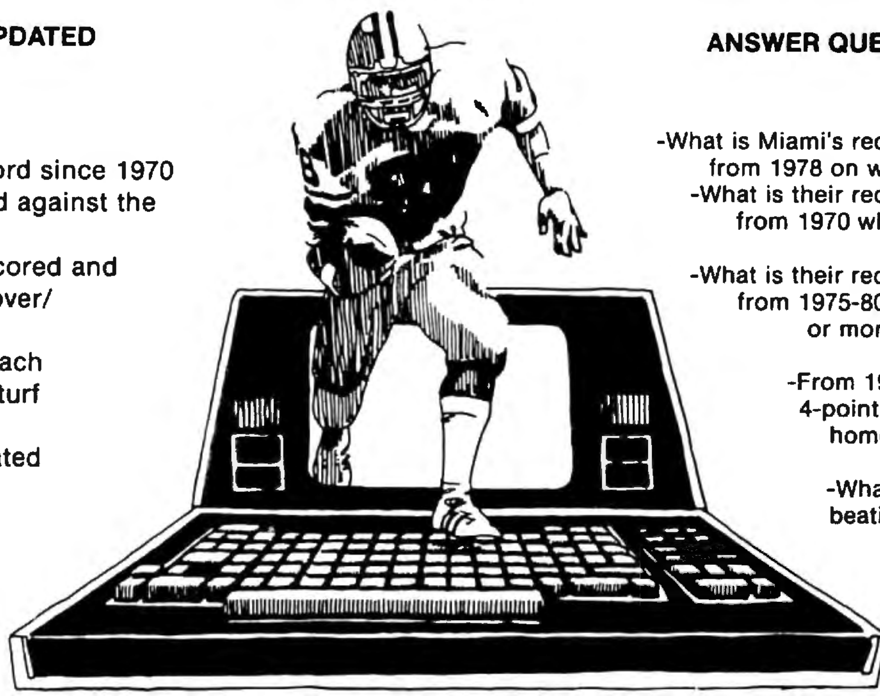
All of the history, information and data needed to help you pick winners against the spread available at your fingertips.

Available for Apple, I.B.M. P.C., Radio Shack, Commodore 64 and other micro computers. Comes with Data Base, operating program and instruction manual.

**EASY TO USE!**

**AUTOMATICALLY UPDATED EACH WEEK!**

Data Base Includes:
- Every win/loss record since 1970
- Every team's record against the spread
- History of points scored and points against for over/under
- History of every coach
- Artificial or natural turf
- Includes USFL
- Automatically updated each week computer to computer



**ANSWER QUESTIONS LIKE THESE IN SECONDS**

- What is Miami's record against the spread from 1978 on when they play Buffalo?
- What is their record against the spread from 1970 when they play any team on natural turf?
- What is their record against the spread from 1975-80 when they are 5 point or more favorites and playing at home?
- From 1976 on, if any team is a 4-point favorite and playing at home, what is its chance of beating the spread?
- What is a team's chance of beating the spread at home after three consecutive wins against the spread?

**YOU SIMPLY WON'T BELIEVE THE QUANTITY OF IMPORTANT DATA AVAILABLE AT YOUR FINGERTIPS**
**HELPS YOU SELECT WINNERS**
**YOU SELECT THE TEAMS, CRITERIA, AND THE TIME FRAME**

**FULL GUARANTEE**
Run this program on your computer for seven days and if not completely satisfied, return it for a full refund.

---

## ORDER NOW AND SAVE

Order before September 15, 1983 and pay only $285. Regular price $495. Weekly computer update only $15 per month for one season (Regular $25). 40% discount on baseball, hockey or basketball programs you may purchase.

☐ **Rush me my Data Base Program and Operating Manual for Pro Football Stats. If I order before September 15, I pay only $285 and will pay only $15 per month for a weekly update for one season along with a 40% discount on other Pro Sports Programs.**

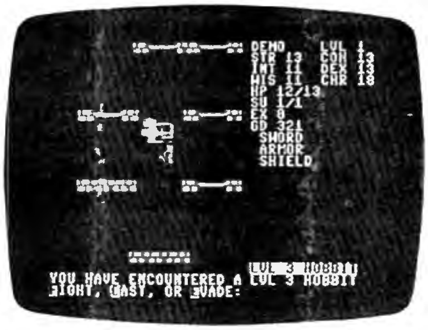
☐ **Enclosed is my check for $25. Send me a Demo disk and instruction manual. My $25 will be applied toward the purchase price should I decide to buy Pro Football Stats.**

☐ **I'm not convinced, send me more information.**

Name ______________________
Street ______________________
City ______________ State______ Zip ______
Telephone ______________________
Make and Model Computer ______________________

**Make check payable to:**
**Eastern Computer Consulting Associates, Inc.,**
**11 Dick Drive, Worcester, MA 01609**
**(617) 757-3131 Ext. 104**

FDA




*The player has encountered a Hobbit, skill level 3. 64 version.*

encounter phase if you've moved into an occupied space.

If you've encountered a monster, you'll be told (in the 64 version, shown) which monster it is and how strong it is. Based on that information, you have a few seconds to decide whether to fight, evade, or cast a spell. If you hesitate, the monster is likely to attack, and your adventure may well be over.

If you fight and survive, you're probably injured and not likely to survive another battle. It's time to find one of the inns and recuperate.

## Gray Misty Cube

As you wander through the dungeon in the darkness, you have more to fear than unsavory creatures. There are many inanimate objects to worry about as well. You may step into a Gray Misty Cube, which can take you to any dungeon level – but if you go, can you find your way back? There are teleportals, pits, fountains, altars, and a mysterious "small box with buttons."

Each of these features offers its own set of problems, and who knows, if you press the right combination of buttons on the small box, something good may come of it. There's more strewn about *Telengard*. Treasure chests, silver, armor and weapons, and even a Ring of Regeneration, something that makes those long trips back to the inn easier to survive.

But beware, that treasure chest may contain 10,000 gold pieces. Or it may house a poisonous spider. Do you dare open it?

## Playing The Game

Playing *Telengard* takes time. First it takes time to learn the game, and then playing could take forever. The game goes on until you meet your match. While learning, pay attention to the helpful hints in the instruction manual. It also helps to play a few games with the time element disabled. That will give you time to leaf through the instructions to bone up on monsters or spells.

Once you're playing a real game, there is no way to stop to check the mail or make a cup of coffee. If you stop playing, you'll be attacked and defeated in no time. Whenever you return to one of the inns, however, the game is stopped until you give the command to reenter Telengard. You may find yourself heading back to an inn just so you can walk away from the computer for a few minutes to relieve the tension.

Though the dungeon is 50 levels deep, with a different maze on each level, you're wise to wander no farther than two or three moves from an inn until you've advanced to Level 3 or 4. The farther into Telengard you venture, the more troublesome your opposition will be.

The most frustrating part of the game comes after working your character up to Level 4 or 5 only to stumble across a Level 32 dragon and lose in an instant. The early game must be played painstakingly, with frequent visits to an inn. Each time you visit an inn, you have the option of saving your character to tape or disk. Once a character has been saved, it can be revived, even after a disastrous encounter with a demon. When saving to tape, have everything ready to go before giving the command, because the program will begin writing immediately.

A feature of *Telengard* that produces some unexpected results is the program's keyboard buffer. It holds two or three characters, so if you get excited and begin pushing keys without thinking, you'll blindly affect your future. Sometimes it's to your advantage to preprogram your steps, but usually you'll regret it.

*Telengard* is an exciting game, one that can tie you up in knots and rob you of your sleep. Learning to play is simultaneously frustrating and fascinating. And once you know the ropes, there's plenty of satisfaction in knowing you've assessed your character correctly and directed him appropriately.

Telengard
*The Avalon Hill Game Company*
*4517 Harford Road*
*Baltimore, MD 21214*
*$23 for tape version*
*$28 for disk version*

©

# *Getaway!* For The Atari

Stephen Levy, Assistant Editor

*Getaway!*, by Mark Reid, is an arcade-style game which takes advantage of the Atari's graphics capabilities. Since the game board – a town map – is approximately 35 screens, the player sees only part of the town at any one time. The player uses a joystick to view other areas of town.

The object of the game is to race all over, stealing as much loot as possible, and then return to your hideout before the police catch you. The game progresses through several levels, but, in the end, justice prevails when the thief is caught. Your score is based on the amount of loot you are able to stash in your hideout.

## Smart Police

There are a number of ways to collect loot. But the greatest rewards come from catching the white armored van. The police don't seem to bother you much until you make the big heist, then their chase is relentless. The more loot you gather or the higher the level, the more energetic their pursuit.

The graphics in *Getaway!* are detailed and appealing. Smooth scrolling is provided by easy joystick control. The sound is realistic; the challenge is exciting and the game becomes more difficult the more effectively you play. And the instruction manual is complete and easy to understand.

## Touring The Town

The first time I played *Getaway!* I was impressed by the detail of the graphics. In fact, I was so intrigued that I put off actually playing until I'd toured the town. Using the black and white map of the town supplied in the user's manual, I was able to "drive" to see all the sights. The town has high-rise buildings, a river, trees, schools, bridges, factories, and three very important gas stations. Each feature is impressive by itself, but taken together, the effect is delightful.

The sound, too, is impressive. When a police car nears, you are aware of it before you see it, because its siren warns you. With experience, you will be able to estimate the distance by the siren's volume.

Fine graphics and sound are always important to a good game, but the game must also play well. You can think of *Getaway!* as a variation of a maze game in the same sense that *Pac-Man* is. The difference is that in *Getaway!* there is much more variety and detail to deal with. The ever-present police are only the beginning. As in any town, stop signs seem to appear whenever you are in a rush. And just when you are about to reach the hideout, you notice that you are running out of gas. If you are new at the game or haven't kept your bearings, those three gas stations can be hard to find.

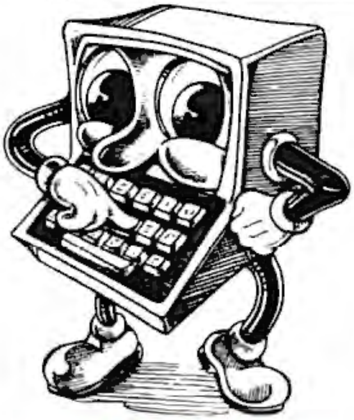
Time also becomes a factor: additional stop signs will appear, and the police begin setting up roadblocks as the game progresses. The police also seem to become more aware of your whereabouts in the night scenario.

## For Any Age

The game's beginning levels are easy enough for a child to enjoy. Adults and more experienced game players will also find the challenge satisfactory. If you manage to get to the fifth level – no easy task – the bonus is an extra getaway car. It comes with a price, though; the game becomes truly challenging at this point.

If you like chase-type, fast-action games; if you are looking for an Atari game the whole family will enjoy; or if you are willing to take the time to become skilled at a game (it takes time to learn the map and all the techniques needed to get to the upper levels), *Getaway!* will surely satisfy you.

Getaway!
*Atari Program Exchange*
*P.O. Box 3705*
*Santa Clara, CA 95055*
*32K Tape*
*32K Disk*
*$29.95*


# Three Game Modules For The TI

Steve Davis

Last year, a young man named Michael Brouthers left his job at Texas Instruments in Dallas and boldly began a venture to develop game software for the TI home computer, a market that he felt was ready to blossom. When TI announced the $100 rebate on the 99/4A, the market for the machine did indeed grow rapidly.

Until now, Texas Instruments has been the only source of software packaged in the convenient Command Module, which TI invented for the 99/4. The module can contain ROM or GROM chips which contain a program (usually written in Assembler or GPL), and, in the case of TI's Mini-Memory Module, the cartridge can be used to add RAM to the console.

The main advantages to using program modules are:

- Ease of use. A person needs no peripheral devices or programming knowledge; just plug in the module and turn on the computer.
- Security. Programs cannot be copied or pirated easily since they reside in GROM or ROM chips. This also prevents accidental erasure of the program.
- Memory. An application program in a module takes up little or no console memory (RAM), so the computer's memory is available for data storage.

Using most third-party game software for the TI requires either Extended BASIC, Memory Expansion, Mini-Memory, Editor/Assembler, cassette or disk.

Now, Funware has introduced a line of game modules, *Henhouse*, *Rabbit Trail*, and *Video Vegas*, for the 99/4A. All use the sprite graphics capability of the TI.

### *Henhouse*

In *Henhouse*, you have five prolific chickens that lay eggs which roll down into five chutes. Each time a chute fills with eggs, you must take them to your truck without dropping them, all the while watching for wolves and poachers.

You get points for each poacher you shoot. Birds fly overhead, and you get points for shooting them, too. You play, using joysticks or the keyboard, until a wolf gets in the henhouse or you break six eggs.

The game may not seem as fast as some of the space or maze games in the arcades, but there are enough distractions that it requires concentration and the ability to do several things at once. It is simple enough to be enjoyed by users of all ages. The retail price is $39.95.

### *Rabbit Trail*

This game is a cross between the *Donkey Kong* and *Frogger* type games. You are a hungry bunny who must hop along the trails and burrow through tunnels in search of carrots. You must not be eaten by a weasel or a hawk, be run over by a speeding car, or get caught in a trap.

Eating all the carrots without being caught advances you to the next level. You receive bonus points based on how fast you complete the level. If you are quick (as a rabbit should be), you may earn "bonus bunnies."

Each of the seven levels presents a more challenging screen. If you complete all seven screens, the game repeats from the first screen but with increased difficulty. Funware says that so far no one has been able to get higher than 24 screens, but to make it even that far would be an accomplishment.

Because of the graduated levels of difficulty, this game is suitable for both beginners and experienced game players. The keyboard may be used, but joysticks are recommended. The retail price for the module is $42.95.

### *Video Vegas*

Anyone who has been to Las Vegas recently knows that some of the slot machines have been replaced by video versions. These operate like the mechanical ones except that the figures (bells, bars, cherries, lemons, etc.) are displayed on a video screen that simulates the rotating cylinders on a conventional slot machine.

Such is *Video Vegas*, a slot machine game that allows you to place $1, $2, or $3 bets by merely pressing keys on the computer console. This is not nearly as tiring as pulling those big levers in Vegas.

The color graphics of the figures are excellent; in fact, they look better than the graphics on some of the machines in Vegas and are a good example of the high-resolution pictures that can be drawn on the 99/4A.

There is nothing challenging about the module, which sells for $29.95, but people who like to play the slots will enjoy it.

Funware prefers that its modules be purchased from software dealers, rather than by mail order from the company.

Henhouse
Rabbit Trail
Video Vegas
*Funware*
*405 N. Bowser*
*Building A*
*Richardson, TX 75081*

# The *VicTree* Programming Module For VIC And 64

Eric Brandon, Programming Assistant

The *VicTree*, a cartridge for the VIC-20 or the Commodore 64, makes programming more efficient.

Available for $89.95 from Skyles Electric Works (the originators of the PET "Toolkit"), the *VicTree* adds 42 commands to BASIC.

All the commands of PET BASIC 4.0 are supported, which make disk use much easier, especially when trying to program relative files. The BASIC 4.0 commands are not tokenized (converted from what you type into a more memory-efficient form) in the same way as in a "true" BASIC 4.0 machine. Fortunately, the manual contains a program that converts "true" BASIC 4.0 to *VicTree* format.

One requirement for using BASIC programs with the *VicTree* is that you must use a colon between a "THEN" (as in an IF...THEN statement) and a BASIC 4.0 disk command. The *VicTree* does not speed up "garbage collection" (the process of removing unwanted or discarded strings from memory) as BASIC 4.0 does, nor will machine language programs written for BASIC 4.0 now run on your VIC or 64.

## Added Commands

Several disk commands not present in BASIC 4.0 have been added, including EXECUTE, which LOADs and RUNs a program all in one step, and CHAIN# which allows an "executive" program to have several BASIC subroutines on disk and load them in only as needed to preserve memory. With this utility, programs can essentially be of unlimited length.

Another set of commands has been added to assist in program editing. As well as all the standard commands we would expect from any BASIC enhancement package, such as renumbering program lines, finding and changing text, and deleting line ranges, *VicTree* adds many new and useful commands never before seen in this type of product.

Among these are the very useful LCOPY and LMOVE commands which let you rearrange the order of the lines in your program. *VicTree* does not "scroll" through your program like other aids, but supplies a PAGE command that LISTs your program one screen at a time.

There are also several commands designed to aid in debugging. These are DUMP, which displays the value of all non-array variables; HELP, which shows where an error has occurred; and TRACE, which LISTs out your program lines as they run.

There are also commands intended for use with any "Centronics" type printer (with no extra hardware needed besides a cable to connect the printer to the parallel user port of your computer). Skyles will supply you with this cable for $29.95, or the *VicTree* and a cable as a package for $109.95.

### Multiple-Computer Communication

Skyles is planning to come out with a device called the Cee-Net which will allow up to 64 VICs or Commodore 64s to communicate with each other and to share disk drives and printers. The *VicTree* is designed to work with Cee-Net when it arrives and has a command to ATTACH itself to the network.

On the 64, the *VicTree* "covers up" memory from 32768 to 40959. This means you have about 30,000 bytes left for your BASIC program. The *VicTree* also uses up memory from 49152 to 53247, so it cannot be used in conjunction with other software which uses these locations such as the Wedge or Micromon. When used with software that does not require that area of memory, however, it seems to work fine. I have had no trouble using the *VicTree* with Supermon, and with the PAL assembler.

On the VIC-20, the *VicTree* uses locations 24576 to 32767 and 45056 to 49151, leaving 21,000 bytes free if you have enough expansion RAM. If you have an unexpanded VIC, the *VicTree* will not use up any of your memory.

The *VicTree* also allows the machine language programmer to add his own commands to BASIC, with descriptions in the manual of how to do it.

The manual contains over 100 pages of clearly written information about the 42 commands. Each command is given its own page (or more) with examples, explanations, and special notes. Also included is a very complete technical description about the machine language applications of the *VicTree*. This is one of the most convenient and useful manuals I have ever seen.

VicTree
*Skyles Electric Works*
*Mountain View, CA 94041*
*(415) 965-1735*
*$89.95*

©

## *Crisis Mountain* For Apple And Atari

Patrick Parrish, Editorial Programmer

*Crisis Mountain*, programmed in machine language by Ron Aldrich and David Schroeder, is an excellent, exciting game, requiring an Apple II or Apple II Plus with 48K RAM (also available for the Atari 400/800 with 48K) and a disk drive. This one-player contest from Synergistic Software can be played with either a joystick or the game paddles.

The scenario of the game is that a group of terrorists was hiding out in the caverns of a dormant volcano in the Pacific Northwest. The volcano erupted unexpectedly, forcing the terrorists to abandon their hideout. As they fled, they left behind their loot and supplies – and several nuclear bombs. To save the West Coast from impending disaster, *you* must venture into Crisis Mountain, dig up and defuse the bombs while avoiding numerous hazards.

### Nine Skill Levels

*Crisis Mountain* alternates between two cavern scenes as you progress through nine skill levels. In the beginning of the game, you are given three lives. And if you're skillful enough you can earn a life at 10,000, 30,000, and 50,000 points. On each level you are presented with a labyrinth of passageways, precipices, and fiery lava pits which sporadically spew rocks and debris.

Scattered about the cavern, in addition to innocuous objects left by the terrorists, are active bombs positioned randomly in one of five locations. Each displays a time, also randomly chosen, before detonation. As you advance from one skill level to another, you are challenged with more bombs and less time to defuse them. Thus, picking the appropriate route through the maze of passageways becomes more and more critical.

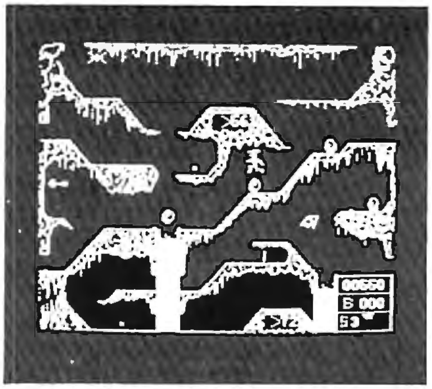
### Scoring Points

Points are awarded for the completion of several tasks. Nominal scores are given for gathering the loot, gun caches, and boxes left by the terrorists. Once you've collected all items, certain bonus forms appear in random positions about the cave.

Another way to score points is to leap boulders. The larger the boulder, the more points you receive. Being struck by a boulder, on the other hand, diminishes your strength. The strength level is indicated with a number from one (weakest) to three (strongest). When you are weakened, your point scoring abilities are significantly impaired. In fact, at strength level one, scoring becomes secondary to mere survival since you can rarely manage to leap boulders in this weakened condition. Fortunately, there are several safe nooks around the cavern where you can recover.

### Treacherous Caverns

There are other ways you can be



*Your running figure (center) leaps a tumbling boulder in* Crisis Mountain.

destroyed in the game. You can fall or be knocked into a lava pit by a boulder, a bomb can detonate, or you can be bitten by the deadly bat, Bertrum.

It is obvious that tremendous effort went into designing this game's high-resolution graphics. Each form is drawn in intricate detail. The frothing lava pits and tumbling boulders are remarkably realistic.

### The Deadly Bat

But the most remarkable graphic element of the game (and the most confounding to any player) is Bertrum, the bat. Bertrum flits about the cavern in a way that resembles a real bat. If a boulder is blasted from a nearby lava pit, Bertrum will dart toward it for a quick inspection, determine the rock is not prey, and fly off to another part of the cave.

But Bertrum is more than just a visual success. His presence adds a degree of chance to the game which makes it faster and more challenging. This dreaded bat has a knack for determining where your player is at any moment in the game. Sometimes, you can avoid Bertrum with a last minute duck or leap. At other times, escape is simply impossible. I've yet to discover a foolproof way to evade this creature, though there may be a tactic.

There are several other excellent features of this game. For one, the ESC key allows you to halt or resume a game at any time during play. With *Crisis Mountain*, a game can sometimes last an hour or more. A break during such a prolonged period of play, beyond being a convenience, is often essential for maintaining your concentration. (No "save game" option is offered.)

Although the sound effects are very good, you may want to turn them off occasionally. If so, you can cancel output to the Apple speaker with CTRL-S. On the other hand, if you want an engulfing, environmental audio effect, output can be sent to external speakers via the cassette port. You can also store on disk, and subsequently display, the high score to date.

Overall, *Crisis Mountain* is a superior programming achievement and a thoroughly entertaining game.

Crisis Mountain
*Synergistic Software*
*830 N. Riverside Drive*
*Suite 201, Renton, WA 98055*
*$34.95*

# *Magic Storybook*: Three Little Pigs For Atari

Orson Scott Card
Editor, COMPUTE! Books

Five-year-old Geoffrey sat down at the computer, and a woman introduced a wolf named Wasco. "Move him to the magic door," she said. He pushed his joystick and the wolf walked over to the door, waving his arms and moving his legs. When he reached the door, the wolf flashed different colors and disappeared.

Then the picture on the screen changed, as if it were a camera panning from left to right. Geoffrey saw a straw house, with a nervous pig inside, wiggling its ears and tail. The straw salesman walked by as the woman told how the house came to be built. Then Wasco came back.

"Little pig, little pig, let me in," said the wolf, in a voice that echoed strangely.

"Not by the hair of my chinny-chin-chin," said the squeaky-voiced pig.

Geoffrey laughed aloud. The woman told him to move Wasco to Door Number One. Geoffrey did it – pausing on the way to let the wolf have a chance to take a few bites of the pig through the window. The pig was apparently safe inside, so Geoffrey moved the wolf the rest of the way to the door.

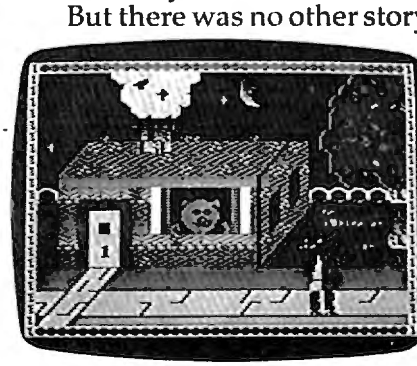
### Huffing And Puffing

The wolf started dancing around while he huffed and puffed. Sure enough, the sky flashed, the "camera" panned to the right again, and the house was now a wreck. The same thing happened with the wood house, and then the wolf failed in two tries at the brick house.

The woman told Geoffrey to move Wasco to the chimney. When Geoffrey got him there, the wolf climbed up and jumped down. But there was a pot waiting down in the fireplace, and the wolf dropped neatly inside.

"I want another story now!" said Geoffrey.

But there was no other story.



*The wolf lurks outside the first pig's house in* The Magic Storybook: Three Little Pigs.

TALMIS/InfoWorld

# Sell Your Software

## TO MAJOR PUBLISHERS!

Would you like the opportunity to discuss your software or software development skills with major publishers looking for product and development assistance, venture capitalists looking for investment opportunities and agents looking for software developers? At The Great American Software Contest you can do all that and compete for prizes worth up to $10,000 at the same time!

In addition, you'll spend the day before the exhibit in seminars with software market analysts from TALMIS—the same people consulted by IBM, Atari and other major companies for information and advice. They'll tell you what it will take to sell software yourself; how you might work with an agent; what publishers and venture capitalists are looking for; what to watch out for when negotiating contracts, how the software market is changing and more important selling tips.

We've tried to make this conference as inexpensive for you as possible. You won't need any fancy booth or signs. Curtained booths are available for those with top-secret programs. Just bring yourself, your microcomputer and your unpublished software. The cost for you to compete for prizes up to $10,000, learn the ropes of software selling and discuss your products or talents with publishers, agents and venture capitalists is $195 plus $15 for electricity (that's less than it costs us.) Special hotel rates are available.

**Space is limited so call today to enter The TALMIS/InfoWorld Great American Software Contest. And get busy on that software—the publishers are waiting for you! For more details contact Master Plans Conference Management, 111 E. Chestnut St., Suite 24F, Chicago, IL 60611: (312) 944-1711.**

Enter
THE GREAT AMERICAN SOFTWARE CONTEST

**Nov. 1-2**
Boston Park Plaza
Boston, MA

One $10,000 Grand Prize
Five $3000 First Prizes in category
Five $500 Second Prizes in category
Ten $250 Third Prizes in category

# + $30,000 IN PRIZES

TALMIS/InfoWorld

Affiliates of International Data Corporation

So Geoffrey happily repeated "The Three Little Pigs" about six times before his parents sent him to bed with a promise that he could play it again tomorrow.

By the fairest standard of judgment I know, that makes *Magic Storybook*'s animated, interactive computer story a success. It is meant for children, and my very picky son Geoffrey thought it was great.

And it was, in many ways. The pictures of the houses were beautifully done, with display list interrupts allowing eight colors and many different shades on the screen at a time. The wolf and the salesman were each made up of four player/missiles combined, and despite the limitation of the 16-bit-wide format (they were tall and thin), the animation was well-done.

## Artistic Screen Display

There were thoughtful extras, too. Stars twinkled. The pigs' eyes, ears, and tails were in constant motion. The artistry of the screen display was delightful. The horizontal scrolling was beautifully done – it even trembled like an earthquake when the wolf blew and blew at the brick house. The cassette loaded correctly the first time, every time, and when we wanted to repeat the story, the other side of the tape had the storytelling soundtrack only, so we didn't have to wait for a load. There was even a line-drawing replica of the cover picture, for a kid to color.

There were trade-offs, of course. That can't be helped. To create fluid, lifelike cartoon movements requires a new picture for every different body position of an onscreen character. That kind of quality takes a lot of artists a lot of time and money. That's why cheaply made cartoons have stiff, unnatural movements, faces that show no expressions, and dull backgrounds that repeat endlessly.

The same limitations apply to computer animation, only in addition to time and money, a third limitation is memory. Smooth, lifelike movement requires that every single picture be in RAM, where it can be accessed instantly. Player/missile graphics compensates a lot, because figures can be moved smoothly. But as soon as you want arms and legs to move naturally, or faces to change expressions, you run into the same old problems – every shape has to be in memory.

## Limited Interaction

But that doesn't excuse all the flaws. For one thing, the interaction was *very* limited. All the child can ever do is move the wolf from right to left. There's a little bit of freedom: the wolf can go up and down about an inch. But if the child plays around with the wolf too long, the program takes over and moves the wolf against the child's will.

That seems like an unnecessary precaution. Why shouldn't children be free to move the wolf all around the house, if they feel like it, and take as long as they want doing it, too? It would have taken only a few dozen machine language commands to allow the wolf to go behind the house in the effort to get inside – a lot of drama would have been added to the story, and nothing is gained by making children hurry through the tale.

The sound was another problem. The background music was tolerable but unexciting. The funny voices for the wolf and the pigs were great – Geoffrey and his three-year-old sister, Emily, laughed out loud the first time through the story. But the narrator! She read in a monotone, as if she were hopelessly bored, repeating an elocution lesson, carefully pronouncing every vowel and consonant.

I couldn't help but compare *Magic Storybook* with PDI's interactive story *Sammy the Sea Serpent*. The graphics and programming in *Magic Storybook* are light-years beyond *Sammy*. But *Sammy*'s narrator is an excellent, excited storyteller, and the child is given meaningful tasks to perform and games to play. The six high-resolution screens and player/missile graphics in *Magic Storybook* cost the children the chance to really become part of the story.

The glow on my son's face when the narrator of *Sammy the Sea Serpent* tells him, "Sammy is home now. He couldn't have done it without you," just wasn't there at the end of "The Three Little Pigs." Some things count even more than graphics.

Magic Storybook:

*The Three Little Pigs*
*Amulet Enterprises, Inc.*
*P.O. Box 25612*
*Garfield Heights, OH 44125*
*(216)475-7766*
*$29.95*

©

# Type Attack

J. David Keller

*Type Attack*, a program from Sirius Software, is a basic course in touch typing enlivened by the challenge and addictive qualities of an arcade game. The program is available in disk versions for the Apple, Atari, and Commodore 64, and on cartridge for the VIC-20.

## The Game

Each lesson in *Type Attack* has two modes. In the first – Character Attack – characters march down the screen in *Space Invaders* fashion. By pressing the proper key on the keyboard, you wipe out the bottom character. If the wrong key is pressed, reserve energy is reduced.

There are three waves of characters. In the first two waves, the characters are in a set pattern; in the third, the characters appear in a random pattern.

In the second mode – Word Attack – words travel across the screen. One vulnerable word is indicated by a flashing marker. When you correctly type the entire word and press the space bar, the word is wiped out and you gain energy units.

If a word goes off the left side of the screen, it reappears at the right side at the cost of energy units. If all the words are correctly typed on the first pass, a set of bonus words come marching by in double time. If you complete *Word Attack* without losing all your energy, you advance to the next lesson.

Scoring is based on the number of characters and words destroyed. Points are lost for pressing the wrong keys. Bonus points are computed at the end of each lesson by multiplying the average words per minute by the speed level at which you played.

## The Lessons

*Type Attack* has 39 planned lessons that follow typical keyboard manuals. Lesson 1 uses the home Keys A S D F. Lesson 2 uses J K L ;. Subsequent lessons build skills by using additional keys, usually two at a time. After the alphabet and basic punctuation marks are studied, numbers are added, and eventually, the symbols that utilize the shift key are introduced.

In Word Attack, early lessons have two to four characters per word. Later, up to 12-character words are presented. Many of the words in Word Attack are computer commands, such as GOTO and 5 HOME.

After the 39 planned lessons, you can add programs to practice specific skills. For example, a lesson which uses only two keys could be designed for very young typists. Or, advanced lessons could utilize a series of programming commands.

You can set the speed at which the letters and words move. The variety of the settings is sufficient to make a beginner feel confident and the pro feel inadequate. Higher score values are given for higher speeds. However, I found I made my highest scores with lower speed settings.

At the left edge of the screen, a bar graph shows the speed at which you are typing the lesson.

The manual is well written and the directions are clear, but more information on typing skill development and the content of each lesson would have been helpful.

*Type Attack* is a well-balanced game and learning program. The challenge is certainly there and as a result, players will surely develop better typing skills.

Type Attack
*Sirius Software, Inc.*
*10364 Rockingham Drive*
*Sacramento, CA 95827*
*(916)366-1195*
*$39.95*

©

# *Mutant Herd* For The VIC

Tony Roberts, Assistant Managing Editor

If your fire-button finger is worn out from trying to shoot down everything that moves, *Mutant Herd* from Thorn EMI Video may be the prescription. In this VIC-20 cartridge game, the fire button does come into play, but only occasionally. There's much more than dodging and shooting here.

Your assignment is to protect a powerhouse, which pulsates at the center of your screen, from an invasion of mutants, who crawl from burrows located at each corner of the display.

Your weapon is a pair of laser beams – one horizontal and one vertical – that are controlled by joystick or the keyboard.

At the game's start, everything is quiet, the beams intersect at center screen, no mutants are in sight. Move one of the beams – even slightly – and the burrows erupt. Red, green, purple, and yellow mutants stream from the burrows and pour toward the power station.

## If They Form A Ring

Use the beams to stop the wave of attackers and push them toward the edges of the screen. Don't push them too far, though, for as you push the attackers to one side, the inhabitants of the other burrows creep in from the other side. If the mutants manage to form a ring around the perimeter of the powerhouse, one of your three lives is lost, and you start again.

As you defend the rumbling powerhouse, you'll soon hear the high-pitched sound of the Mutant Slayers as they begin to appear on the screen. The Mutant Slayers, though not unlike the mutants in appearance, are the key to eliminating these power production pests.

Use the laser beams to guide a slayer into one of the four burrows. By pressing the fire button, you allow the slayers to pass through the beams. If you push one of the slayers – you get ten – off the screen or into the powerhouse, color it gone.

Once you guide a slayer into one of the burrows, the scene changes. You find yourself in the shoes of a Mutant Slayer near the top of the burrow you just entered. You see a ladder leading down past abandoned caverns to the bottom of the screen where the Mutant Queen protects 15 of her precious eggs.

You're working against time, so don't spend too long admiring the sights. Get down the ladder into the Queen's cavern and put down an explosive charge. Dart back up the ladder and touch the detonator to destroy five of the eggs and seal the burrow.

It's not as easy as it sounds. The mutants, though they are admirable burrowers, know little about engineering. The abandoned caverns are deathtraps. Rocks continually fall from the walls and ceilings and bound down the ladderway. Step quickly into the gaps to the left of the ladder to avoid the falling rocks. You can't survive a direct hit.

While you're dodging rocks on your way to the top of the burrow, the Mutant Queen attempts to move your explosive away from her eggs. If she succeeds, you must go back and replace it. Be forewarned: the Mutant Queen considers Mutant Slayers a delicacy and will not hesitate to eat one if it ventures too close.

If you successfully plant the charge, dodge the rocks, and return to the detonator before the explosive has been moved, you will destroy five eggs and seal the burrow. Congratulations. But you've only just begun.

When you return to the powerhouse, things will have changed. Only three burrows remain, but your laser beams have been weakened. They're filled with gaps where mutants can slip through. Despite the difficulties, you must press on; you must seal the remaining burrows.

Guide a slayer into another burrow, and the scene shifts as before. This time, however, the Queen guards only ten eggs. You'll have to get closer to the Queen to plant the charge (risking ending up as a light lunch), and the Queen needn't go as far to move the explosive away from the eggs.

## Use The Patrol Schedule

Though the Mutant Queen is vicious and certainly voracious, she does have a weakness: her pacing is predictable. She ambles back and forth through her narrow cavern almost like clockwork. Use her patrol schedule to your advantage.

If you manage to seal the second burrow, you'll return to the powerhouse, your beams weaker still. Send a slayer into one of the remaining burrows, plant the charge, dodge the rocks and the Mutant Queen, and detonate the explosive to destroy the final five eggs.

Once the eggs are destroyed, you can turn your full attention to the Queen herself. Back at the powerhouse, you have only one burrow of mutants to contend with, but your laser beams look like Swiss cheese.

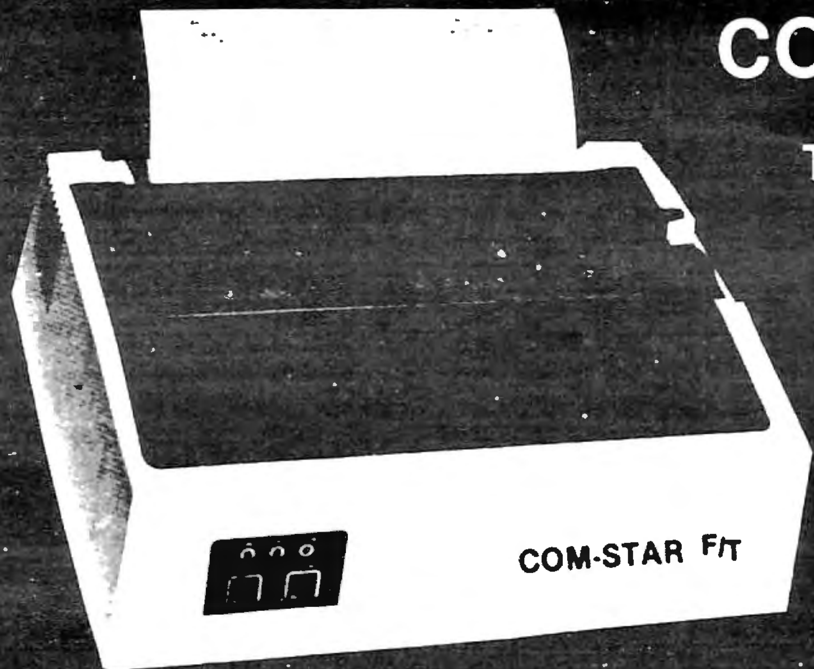
Guide a slayer into the final burrow. Plant your explosive in the Queen's cavern and crawl for cover. When the Queen is directly above the charge, press the fire button to trigger the explosion and complete the round.

Once the Queen has been destroyed, you move on to new rounds, and new hazards.

*Mutant Herd* includes options for one or two players. No pause option is available, but

# COM-STAR F/T

## Tractor Friction Printer




## only $259

• 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

### Deluxe COMSTAR F/T PRINTER — $259.00

The Comstar 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 $259 the Comstar gives you print quality and featuers 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. (Add $20.00 for RS232)

• **LONG LIFE PRINT HEAD:** 100 million character life expectancy.

• **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.

• **THREE SELECTABLE LINE SPACINGS:** 6, 8 or 12 lines per inch.

• **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 F/T SUPER-10" PRINTER — $299.00

More Features Than MX-80

For $299 you get all of the features of the Comstar plus 10" carriage 120 cps, 9x9 dot matrix with double strike capability for 18 x 18 dotmatrix. High resolution bit image (120 x 144 dot matrix), underlining, backspacing, 2.3K buffer, 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 Comstart F/T Super 10" leads the pack!

### 80 COLUMN PRINTER $189

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!

### 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 $20.00 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

**COMSTAR F/T**

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk
lmn opqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890

**SUPER-10"** ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

each time you return to the powerhouse, either by losing a life or sealing a burrow, the game waits until you initiate action by moving the laser beams.

Though a little luck will get you started with *Mutant Herd*, you'll need to develop strategies, both for the surface and the underworld, to keep going for long. There are enough little problems to always keep you thinking about what you'll have to do next.

Mutant Herd
*Thorn EMI Video Programs Limited*
*1370 Avenue of the Americas*
*New York, NY 10019*
*$39.95*

©


## FOR WORK OR PLAY, OUR NAME SAYS IT ALL.

### Exclusively from AdVentures

**Gypsum Caves** by Brian Wagner

A revolutionary new game requiring more than just hand-eye coordination, Gypsum Caves is a word-adventure trek through endless caverns. Use objects found along the way and your imagination to gain the final test — then try to get out alive! 3-D Color Graphics

*Vic 20 Cassette* .......... **$14.95**
*C64 Disk* .......... **$16.95**

**Ak-Ak Man —** by Brian Wagner

The old standby with a new twist — random screens. The computer devises a new pattern every time, for added challenge and excitement.

*Vic 20 Cassette* .......... **$9.95**

### New Releases from AdVentures

**Leap-Man** (+ 3K) by Brian Wagner

Experience 5 screens worth of action as Leap-Man traverses the structure of girders, walls, floors, and ladders in this fast-paced adventure game.

*Vic 20 Cassette* .......... **$14.95**

**Earthinator** (+ 3K) by Brian Wagner

Pursue various subterranean denizens through tunnels, shafts, and solid rock in your capacity as Earthinator, structural pest control technician for the entire planet.

*Vic 20 Cassette* .......... **$14.95**

**Huntman** By Earnest Hunter

Scramble to avoid the wild hogs and collect enough ammunition to defend yourself in this wild action chase challenge.

*Vic 20 Cassette* .......... **$14.95**

**Mind Binder** by Steve Fogolini

A mind boggling matching game of skill, tactics, and memory. 5 levels of difficulty for hours of enjoyment.

*C64 Disk* .......... **$14.95**

**World Conquest** by Todd Seidel

Combined skill and luck are required to master this intriguing quest for world domination.

*C64 Disk* .......... **$16.95**

**Mailfile** by L. A. Enterprises

A software directory for your home computer, Mailfile stores and retrieves names, numbers and addresses by name and zip code.

*Atari 400/500 Cassette* .......... **$9.95**
*C64 Cassette* .......... **$9.95**
*C64 Disk* .......... **$11.95**
*Vic 20 Cassette* .......... **$9.95**

**Math Tutor** by Royce Webb

A valuable tool for young children to simultaneously develop their math skills and their understanding of the use and value of the computer. For grade levels 4 and up, with emphasis on problem-solving using elementary math principles. Varying skill levels, question sets, and choice arrays.

*Atari 400/800 Cassette* .......... **$8.95**

**Home Budget/Finance** (+ 3K) by Carter Jones

This is a variable program that can help you organize your budget and stick to it. Pre-programmed with various expenditure/income categories as well as untitled columns for specific regular expenses, this planner is a valuable tool for any homemaker.

*Vic 20 Cassette* .......... **$9.95**
*C64 Disk* .......... **$11.95**

**My ABC's** by L.A. Enterprises

Software designed to help young children get a head start in language as well as familiarize them with the potential of the computer

*C64 Casseette* .......... **$8.95**
*C64 Disk* .......... **$10.95**
*Vic 20 Cassette* .......... **$8.95**

**My First Numbers** by Carter Jones

An extensive elementary mathematics program designed for children up to 4th grade levels. This software can provide the basis for superior acheivement in all levels of mathematics by assuring a complete grasp of the basic principles.

*Vic 20 Tape* .......... **$9.95**

### EPYX

**Jumpman**

As Jumpman, you will be the only one capable of negotiating the wreckage of headquarters to find the bombs planted there. Watch your step!

*Atari 400/800 Disk* .......... **$29.95**
C64 .......... **$29.95**

**Aggressor**

Protect the precious ore fields of Freeworld 6 in your advanced VX6 Marauder Ramjet fighter. 10 levels of difficulty.

*Vic 20 Cartridge* .......... **$32.95**

**Gridrunner**

Combat the enemy droids on the Grid, a huge orbiting solar power station, in a specially constructed Gridrunner ship.

*Vic 20 and C64 Cartridge* .......... **$32.99**

### BRØDERBUND

**A.E.**

The newest import from Japan has surpassed its function as a pollution control robot and become a nuisance. A lighthearted game requiring speed, agility, and a sense of humor. Joystick required.

*Atari 400/800 Disk* .......... **$24.99**

### DATA SOFT

**Zaxxon**

Official home version of the famous 3-D arcade adventure. Pilot your aircraft through the battlefield and to the showdown with the deadly armored robot. Sound effects and color graphics.

*Atari 400/800 Cassette* .......... **$29.95**

1701 Monitor Cables .......... **$11.25**
Monitor Cables with audio .......... **$9.95**
Cable wrapups .......... 5 for **$1.19**

Dealer/Distributor's inquiry welcome.
Attention Programmers: Top dollar for original programs. We provide copyright assistance.

Other major credit cards accepted.

VISA MasterCard

## AdVENTURES
## Call Toll Free
## 1-800-835-2222

In D.C. Metro area 703-360-0301

**Or Write to:**
**AdVentures**
**8718-A Richmond Highway**
**Alexandria, VA 22309**

**Include $2.00 for Postage and Handling.**

# ULTRASORT
# For Commodore

John W Ross

*This is probably the fastest sorting program ever published for any home computer. It will alphabetize 1000 items in less than eight seconds.*

*There are versions here for the 64, VIC, and 4.0 PET. You might want to change the amount sorted in the test program to reflect the available memory in your computer. If so, change N in line 110 of Program 4. The test generates random "words" so you can see how the program works.*

This article is a sequel to my earlier article "Super Shell Sort for PET/CBM" (February 1983). In that article, I described a shell sort program for the CBM 8032 written entirely in machine language. It performed as expected and was, overall, very fast; however, it had a couple of shortcomings. First of all, it had a rather clumsy interface with BASIC; that is, the calling sequence was not very neat; and second, sorting was performed by the shell sort algorithm. This method of sorting is actually quite efficient, certainly far better than a bubble sort, for instance. Nevertheless, there are better sorts.

C.A.R. Hoare's Quicksort algorithm is possibly the fastest yet developed for most applications. So, I rewrote my machine language sort program based on the Quicksort algorithm.

## Speed Improvements

How much better is it? In order to test the program, I wrote a small sort test program (Program 4), similar to the one in my original article. This program generates a character array containing N items (line 110).

Different items are generated depending on the value of the random number seed, SD in line 140; SD must be a negative number.

I generated six 1000 element arrays and sorted them using both the shell sort and Ultrasort. Super Shell Sort required an average of 29.60 seconds to sort all 1000 elements, while Ultrasort required an average of only 8.32 seconds. The sorting time has increased 72%. I don't believe you will find a faster sort for an eight-bit machine anywhere.

The way you start the sort (see Program 4) has also been refined. To RUN the sort on the PET, you simply type:

```
SYS 31744,N,AA$(K)
```

For the 64, use:

```
SYS 49152,N,AA$(K)
```

The format is the same for the VIC, but the loader for the VIC version (Program 2) is designed to relocate itself to the top of available memory, which will vary according to the amount of expansion memory added to your VIC. (Ultrasort is too long for the unexpanded VIC.) The loader program will tell you the proper SYS address to use on your VIC.

## RUNning The Program

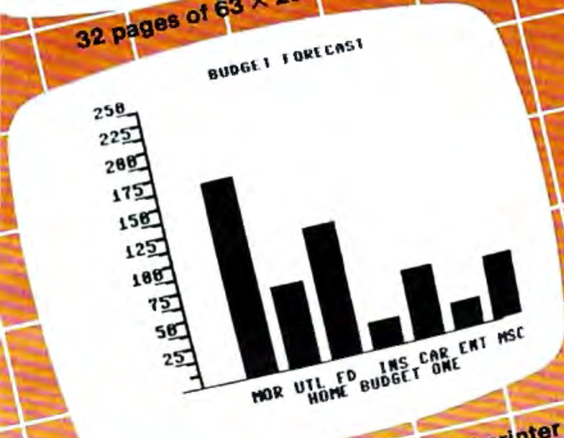
Ultrasort can be used either from within a program or in immediate mode. RUNning Ultrasort causes N elements from array AA$, starting with element K, to be sorted into ascending order. The sort occurs in place; there is no additional memory overhead. N and K can be constants or variables, and any character array name can be substituted for AA$.

Before RUNning the sort, though, it must be LOADed by BASIC. The appropriate loader is supplied in Programs 1-3. The tradeoff for the increased speed of Ultrasort is increased complexity, especially in machine language. The sort program runs from $7C00 to $7F8B (908 bytes) on the PET. The increased size, of course, creates a greater possibility of errors when you enter the numbers. In order to minimize this, the PET loader


32 pages of 63 × 254 cells

# the foxiest

## PLANNING and FORECASTING TOOL AVAILABLE

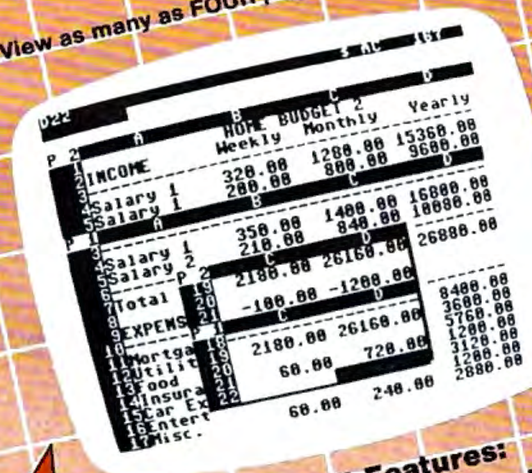



GRAPHIC DISPLAY on screen and printer

# Calc Result™

## For the COMMODORE 64™ with disk drive

View as many as FOUR pages at one time




We started with the best that standard spreadsheet programs could offer.

Then we added features designed to create a planning tool more useful than ever before.

The Result?

CALC RESULT! The most powerful, understandable and economical spreadsheet program on the market today.

Thirty-two pages, graphics and the ability to view up to four pages (spreadsheets) at once make CALC RESULT the spreadsheet program that outfoxes them all!

### Calc Result Features:

- A three dimensional spreadsheet with 32 pages of 63 × 254 cells, offering unrivaled flexibility
- Graphic display on screen and printer
- The ability to view as many as four pages at once through a window and split screen. This allows you to compare spreadsheets!
- Help function on-line to make Calc Result's features easy to use
- Color coordinated cells that speed calculations
- IF-THEN-ELSE with AND, OR and NOT-ELSE functions in each cell give you unlimited possibilities for decision making
- Timesaving full function editing
- Flexible column width printouts for formatting reports
- Utilization of memory only in cells that are active
- Replicate, copy and move commands that save time
- Consolidation of spreadsheets to get the "bottom line"
- Protection of cells containing formulas
- Ability to load VisiCalc™ files

Distributed by:

**COMPUTER MARKETING SERVICES INC.**

300 W. Marlton Pike • Cherry Hill, N.J. 08002 • 609-795-9480

"Our products outfox them all!"

(Program 1) is written to be self-checking to a degree. The DATA statements are grouped in blocks of 20 lines or 140 numbers (except for the first and last blocks), each of which is supplied with a checksum. If all the numbers in a block do not add up to the checksum, an error message is printed, giving you an indication of which block is in error. VIC and 64 owners should check their typing carefully, as there is no checksum.

Notice that the first thing the loader programs do is reset the top of memory pointer. This is very important – you must use the BASIC loader before RUNning the sort program.

Once Program 1 is loaded into upper memory of the PET, you should save it to disk by entering the monitor (SYS 4) and typing:

**S"0:ULTRASORT",08,7C00,7F8C**

VIC and 64 owners should use a monitor program or cartridge (e.g., VICMON, Supermon 64) or a routine such as "Machine Language Saver" (COMPUTE!, June 1983, p. 216) to save a copy of the Ultrasort machine language.

To load your copy of Ultrasort from the PET monitor, reset the top of memory and type:

**L"0:ULTRASORT",08**

From PET, VIC, or 64 BASIC type:

**LOAD"ULTRASORT",8,1** for disk, or
**LOAD"ULTRASORT",1,1** for tape.

You can use Program 4 to watch the action with the PET, VIC, or 64 versions of Ultrasort.

## Program 1: Ultrasort For PET

```
1 REM ULTRASORT-LOADER
10 POKE 52,0 : POKE 53,124 : CLR
20 FOR IB=1 TO 7
30 READ N,NL,CC:CS=0 : IF NL<>0 THEN L=NL
40 FORI=1 TO N : READ X : CS=CS+X : POKE
   L,X
50 L=L+1 : NEXT I
60 IF CS<>CC THEN PRINT"ERROR IN BLOCK"IB
    : END
70 PRINT"BLOCK"IB"OK"
80 NEXT IB
90 END
199 REM ... BLOCK 1 ...
200 DATA 3,31744,300
205 DATA 76,100,124
206 REM ... BLOCK 2 ...
207 DATA 140,31844,14808
210 DATA 32,245,190,32,152,189,32
215 DATA 45,201,165,17,141,12,124
220 DATA 165,18,141,13,124,32,245
225 DATA 190,32,152,189,56,165,68
230 DATA 233,3,133,84,165,69,233
235 DATA 0,133,85,162,1,173,12
240 DATA 124,157,20,124,173,13,124
245 DATA 157,40,124,169,1,157,60
250 DATA 124,169,0,157,80,124,189
255 DATA 60,124,141,16,124,189,80
260 DATA 124,141,17,124,189,20,124
265 DATA 141,18,124,189,40,124,141
270 DATA 19,124,32,47,127,173,11
275 DATA 124,48,4,202,208,221,96
280 DATA 189,60,124,141,16,124,189
285 DATA 80,124,141,17,124,169,1
290 DATA 141,18,124,169,0,141,19
295 DATA 124,32,101,127,189,20,124
300 DATA 141,18,124,141,14,124,189
305 DATA 40,124,141,19,124,141,15
306 REM ... BLOCK 3 ...
307 DATA 140,0,13385
310 DATA 124,32,47,127,173,11,124
315 DATA 48,3,76,167,125,32,131
320 DATA 127,173,16,124,141,3,124
325 DATA 173,17,124,141,4,124,173
330 DATA 14,124,141,5,124,173,15
335 DATA 124,141,6,124,32,132,126
340 DATA 32,180,126,173,11,124,48
345 DATA 218,173,16,124,141,3,124
350 DATA 173,17,124,141,4,124,173
355 DATA 18,124,141,16,124,173,19
360 DATA 124,141,17,124,169,1,141
365 DATA 18,124,169,0,141,19,124
370 DATA 32,101,127,173,16,124,141
375 DATA 18,124,173,17,124,141,19
380 DATA 124,173,3,124,141,16,124
385 DATA 173,4,124,141,17,124,32
390 DATA 47,127,173,11,124,16,35
395 DATA 173,14,124,141,3,124,173
400 DATA 15,124,141,4,124,173,18
405 DATA 124,141,5,124,173,19,124
406 REM ... BLOCK 4 ...
407 DATA 140,0,13499
410 DATA 141,6,124,32,132,126,32
415 DATA 180,126,173,11,124,48,152
420 DATA 32,47,127,173,11,124,16
425 DATA 18,173,16,124,141,3,124
430 DATA 173,17,124,141,4,124,32
435 DATA 132,126,32,31,127,76,241
440 DATA 124,234,189,20,124,141,3
445 DATA 124,189,40,124,141,4,124
450 DATA 173,16,124,141,5,124,173
455 DATA 17,124,141,6,124,32,132
460 DATA 126,32,31,127,173,16,124
465 DATA 141,18,124,141,3,124,173
470 DATA 17,124,141,19,124,141,4
475 DATA 124,32,81,127,189,20,124
480 DATA 141,18,124,189,40,124,141
485 DATA 19,124,32,101,127,173,11
490 DATA 124,48,15,189,60,124,141
495 DATA 18,124,189,80,124,141,19
500 DATA 124,32,101,127,169,1,141
505 DATA 18,124,169,0,141,19,124
506 REM ... BLOCK 5 ...
507 DATA 140,0,15957
510 DATA 173,3,124,141,16,124,173
515 DATA 4,124,141,17,124,173,11
520 DATA 124,16,52,189,60,124,232
525 DATA 157,60,124,202,189,80,124
530 DATA 232,157,80,124,32,101,127
535 DATA 173,16,124,157,20,124,173
540 DATA 17,124,157,40,124,32,131
545 DATA 127,32,131,127,202,173,16
550 DATA 124,157,60,124,173,17,124
555 DATA 157,80,124,76,128,126,32
560 DATA 131,127,232,173,16,124,157
565 DATA 60,124,173,17,124,157,80
570 DATA 124,202,189,20,124,232,157
575 DATA 20,124,202,189,40,124,232
580 DATA 157,40,124,202,32,101,127
585 DATA 32,101,127,173,16,124,157
590 DATA 20,124,173,17,124,157,40
```

```
595 DATA 124,232,76,162,124,160,3
600 DATA 165,84,133,88,133,90,165
605 DATA 85,133,89,133,91,24,165
606 REM ... BLOCK 6 ...
607 DATA 140,0,15683
610 DATA 88,109,3,124,133,88,165
615 DATA 89,109,4,124,133,89,24
620 DATA 165,90,109,5,124,133,90
625 DATA 165,91,109,6,124,133,91
630 DATA 136,208,223,96,160,0,140
635 DATA 11,124,177,88,141,7,124
640 DATA 177,90,141,8,124,200,152
645 DATA 205,7,124,240,2,176,13
650 DATA 205,8,124,240,21,144,19
655 DATA 238,11,124,76,30,127,205
660 DATA 8,124,240,2,176,62,206
665 DATA 11,124,76,30,127,140,9
670 DATA 124,160,1,177,88,133,86
675 DATA 200,177,88,133,87,172,9
680 DATA 124,136,177,86,141,10,124
685 DATA 140,9,124,160,1,177,90
690 DATA 133,86,200,177,90,133,87
695 DATA 172,9,124,177,86,200,205
700 DATA 10,124,208,3,76,195,126
705 DATA 144,184,76,224,126,96,160
706 REM ... BLOCK 7 ...
707 DATA 108,0,11613
710 DATA 2,177,88,72,177,90,145
715 DATA 88,104,145,90,136,16,243
720 DATA 96,169,0,141,11,124,173
725 DATA 17,124,205,19,124,144,6
730 DATA 240,8,238,11,124,96,206
735 DATA 11,124,96,173,16,124,205
740 DATA 18,124,144,244,208,238,96
745 DATA 173,16,124,24,109,18,124
750 DATA 141,16,124,173,17,124,109
755 DATA 19,124,141,17,124,96,169
760 DATA 0,141,11,124,56,173,16
765 DATA 124,237,18,124,141,16,124
770 DATA 173,17,124,237,19,124,141
775 DATA 17,124,176,3,206,11,124
780 DATA 96,238,16,124,208,3,238
785 DATA 17,124,96
```

## Program 2: Ultrasort For VIC

```
5 I1=PEEK(56)*256-1024
6 POKE 55,0:HI=INT(I1/256):POKE 56,HI:CL
  R
7 I1=PEEK(55)+PEEK(56)*256
8 HI=INT(I1/256)
10 I=I1
20 READ A:IF A=256 THEN PRINT"TO RUN SOR
   T, USE: SYS"I1:END
22 IF A<0 THEN A=ABS(A)-26+HI
25 IF A=257 THEN I=I1+100:GOTO 20
30 POKE I,A:I=I+1:GOTO 20
6656 DATA 76,100,-26,257
6768 DATA 32,253,206,32,158
6776 DATA 205,32,247,215,165,20,141
6784 DATA 12,-26,165,21,141,13,-26
6792 DATA 32,253,206,32,158,205,56
6800 DATA 165,71,233,3,133,75,165
6808 DATA 72,233,0,133,76,162,1
6816 DATA 173,12,-26,157,20,-26,173
6824 DATA 13,-26,157,40,-26,169,1
6832 DATA 157,60,-26,169,0,157,80
6840 DATA -26,189,60,-26,141,16,-26
6848 DATA 189,80,-26,141,17,-26,189
6856 DATA 20,-26,141,18,-26,189,40
6864 DATA -26,141,19,-26,32,47,-29
6872 DATA 173,11,-26,48,4,202,208
6880 DATA 221,96,189,60,-26,141,16
6888 DATA -26,189,80,-26,141,17,-26
6896 DATA 169,1,141,18,-26,169,0
6904 DATA 141,19,-26,32,101,-29,189
6912 DATA 20,-26,141,18,-26,141,14
6920 DATA -26,189,40,-26,141,19,-26
6928 DATA 141,15,-26,32,47,-29,173
6936 DATA 11,-26,48,3,76,167,-27
6944 DATA 32,131,-29,173,16,-26,141
6952 DATA 3,-26,173,17,-26,141,4
6960 DATA -26,173,14,-26,141,5,-26
6968 DATA 173,15,-26,141,6,-26,32
6976 DATA 132,-28,32,180,-28,173,11
6984 DATA -26,48,218,173,16,-26,141
6992 DATA 3,-26,173,17,-26,141,4
7000 DATA -26,173,18,-26,141,16,-26
7008 DATA 173,19,-26,141,17,-26,169
7016 DATA 1,141,18,-26,169,0,141
7024 DATA 19,-26,32,101,-29,173,16
7032 DATA -26,141,18,-26,173,17,-26
7040 DATA 141,19,-26,173,3,-26,141
7048 DATA 16,-26,173,4,-26,141,17
7056 DATA -26,32,47,-29,173,11,-26
7064 DATA 16,35,173,14,-26,141,3
7072 DATA -26,173,15,-26,141,4,-26
7080 DATA 173,18,-26,141,5,-26,173
7088 DATA 19,-26,141,6,-26,32,132
7096 DATA -28,32,180,-28,173,11,-26
7104 DATA 48,152,32,47,-29,173,11
7112 DATA -26,16,18,173,16,-26,141
7120 DATA 3,-26,173,17,-26,141,4
7128 DATA -26,32,132,-28,32,31,-29
7136 DATA 76,241,-26,234,189,20,-26
7144 DATA 141,3,-26,189,40,-26,141
7152 DATA 4,-26,173,16,-26,141,5
7160 DATA -26,173,17,-26,141,6,-26
7168 DATA 32,132,-28,32,31,-29,173
7176 DATA 16,-26,141,18,-26,141,3
7184 DATA -26,173,17,-26,141,19,-26
7192 DATA 141,4,-26,32,81,-29,189
7200 DATA 20,-26,141,18,-26,189,40
7208 DATA -26,141,19,-26,32,101,-29
7216 DATA 173,11,-26,48,15,189,60
7224 DATA -26,141,18,-26,189,80,-26
7232 DATA 141,19,-26,32,101,-29,169
7240 DATA 1,141,18,-26,169,0,141
7248 DATA 19,-26,173,3,-26,141,16
7256 DATA -26,173,4,-26,141,17,-26
7264 DATA 173,11,-26,16,52,189,60
7272 DATA -26,232,157,60,-26,202,189
7280 DATA 80,-26,232,157,80,-26,32
7288 DATA 101,-29,173,16,-26,157,20
7296 DATA -26,173,17,-26,157,40,-26
7304 DATA 32,131,-29,32,131,-29,202
7312 DATA 173,16,-26,157,60,-26,173
7320 DATA 17,-26,157,80,-26,76,128
7328 DATA -28,32,131,-29,232,173,16
7336 DATA -26,157,60,-26,173,17,-26
7344 DATA 157,80,-26,202,189,20,-26
7352 DATA 232,157,20,-26,202,189,40
7360 DATA -26,232,157,40,-26,202,32
7368 DATA 101,-29,32,101,-29,173,16
7376 DATA -26,157,20,-26,173,17,-26
7384 DATA 157,40,-26,232,76,162,-26
7392 DATA 160,3,165,75,133,79,133
7400 DATA 81,165,76,133,80,133,82
7408 DATA 24,165,79,109,3,-26,133
7416 DATA 79,165,80,109,4,-26,133
```

# BUSIWRITER

## BUSIWRITER A Honey of a Word Processor

**Why word processors?**

Word processors allow the user to quickly and easily create letters, memos, notes, reports, term papers, manuals, poetry and any other written information using the memory of the computer as a pencil and paper. The computer display or terminal acts as a window through which the user views the information as it is entered. The outstanding advantage of using BUSIWRITER is that it acts not only as a pencil and paper but as a perfect eraser and automatic typewriter.


**For Commodore CBM-64**
**Commodore 1515, 1525, Epson, C. Itoh, Qume, Diablo, NEC Spinwriter, Starwriter, Prowriter, Okidata, Microline, Gemini-10**
**And many more printers**

## BUSIWRITER The Queen Bee of Word Processors

BUSIWRITER allows the user to quickly and easily make any number of alterations to the text. BUSIWRITER will instantly reformat your text and show you exactly and continuously how the final output will appear. BUSIWRITER has more functions than any other known microcomputer word processor. With BUSIWRITER assisting in the entry of text, providing a **20 page memory** and performing an enormous number of editing/composing functions, the preparation of written data is far faster and outstandingly more accurate than if it were prepared by hand.

## BUSIWRITER With the Sting Removed from the Prices

**BUSIWRITER 64** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . only **$99.00** for the CBM 64

BUSIWRITER AVAILABLE NOW FROM YOUR LOCAL DEALER
(800) 227-9998
FOR THE NAME OF YOUR NEAREST DEALER

California, Canada, Alaska and Hawaii please call (415) 965-1735

**Skyles Electric Works**
231G South Whisman Road
Mountain View, CA 94041

Europe please contact Supersoft, Winchester House, Harrow Wealdstone, England HA3 7SJ, Tel. 01 861 1166

```
7424 DATA 80,24,165,81,109,5,-26
7432 DATA 133,81,165,82,109,6,-26
7440 DATA 133,82,136,208,223,96,160
7448 DATA 0,140,11,-26,177,79,141
7456 DATA 7,-26,177,81,141,8,-26
7464 DATA 200,152,205,7,-26,240,2
7472 DATA 176,13,205,8,-26,240,21
7480 DATA 144,19,238,11,-26,76,30
7488 DATA -29,205,8,-26,240,2,176
7496 DATA 62,206,11,-26,76,30,-29
7504 DATA 140,9,-26,160,1,177,79
7512 DATA 133,77,200,177,79,133,78
7520 DATA 172,9,-26,136,177,77,141
7528 DATA 10,-26,140,9,-26,160,1
7536 DATA 177,81,133,77,200,177,81
7544 DATA 133,78,172,9,-26,177,77
7552 DATA 200,205,10,-26,208,3,76
7560 DATA 195,-28,144,184,76,224,-28
7568 DATA 96,160,2,177,79,72,177
7576 DATA 81,145,79,104,145,81,136
7584 DATA 16,243,96,169,0,141,11
7592 DATA -26,173,17,-26,205,19,-26
7600 DATA 144,6,240,8,238,11,-26
7608 DATA 96,206,11,-26,96,173,16
7616 DATA -26,205,18,-26,144,244,208
7624 DATA 238,96,173,16,-26,24,109
7632 DATA 18,-26,141,16,-26,173,17
7640 DATA -26,109,19,-26,141,17,-26
7648 DATA 96,169,0,141,11,-26,56
7656 DATA 173,16,-26,237,18,-26,141
7664 DATA 16,-26,173,17,-26,237,19
7672 DATA -26,141,17,-26,176,3,206
7680 DATA 11,-26,96,238,16,-26,208
7688 DATA 3,238,17,-26,96,256
```

## **Program 3:** Ultrasort For 64

```
10 I=49152
20 READ A:IF A=256 THEN END
30 POKE I,A:I=I+1:GOTO 20
49152 DATA 76,100,192,170,170,170,170
49159 DATA 170,170,170,170,170,170,170
49166 DATA 170,170,170,170,170,170,170
49173 DATA 170,170,170,170,170,170,170
49180 DATA 170,170,170,170,170,170,170
49187 DATA 170,170,170,170,170,170,170
49194 DATA 170,170,170,170,170,170,170
49201 DATA 170,170,170,170,170,170,170
49208 DATA 170,170,170,170,170,170,170
49215 DATA 170,170,170,170,170,170,170
49222 DATA 170,170,170,170,170,170,170
49229 DATA 170,170,170,170,170,170,170
49236 DATA 170,170,170,170,170,170,170
49243 DATA 170,170,170,170,170,170,170
49250 DATA 170,170,32,253,174,32,158
49257 DATA 173,32,247,183,165,20,141
49264 DATA 12,192,165,21,141,13,192
49271 DATA 32,253,174,32,158,173,56
49278 DATA 165,71,233,3,133,75,165
49285 DATA 72,233,0,133,76,162,1
49292 DATA 173,12,192,157,20,192,173
49299 DATA 13,192,157,40,192,169,1
49306 DATA 157,60,192,169,0,157,80
49313 DATA 192,189,60,192,141,16,192
49320 DATA 189,80,192,141,17,192,189
49327 DATA 20,192,141,18,192,189,40
49334 DATA 192,141,19,192,32,47,195
49341 DATA 173,11,192,48,4,202,208
49348 DATA 221,96,189,60,192,141,16
49355 DATA 192,189,80,192,141,17,192
49362 DATA 169,1,141,18,192,169,0
49369 DATA 141,19,192,32,101,195,189
49376 DATA 20,192,141,18,192,141,14
49383 DATA 192,189,40,192,141,19,192
49390 DATA 141,15,192,32,47,195,173
49397 DATA 11,192,48,3,76,167,193
49404 DATA 32,131,195,173,16,192,141
49411 DATA 3,192,173,17,192,141,4
49418 DATA 192,173,14,192,141,5,192
49425 DATA 173,15,192,141,6,192,32
49432 DATA 132,194,32,180,194,173,11
49439 DATA 192,48,218,173,16,192,141
49446 DATA 3,192,173,17,192,141,4
49453 DATA 192,173,18,192,141,16,192
49460 DATA 173,19,192,141,17,192,169
49467 DATA 1,141,18,192,169,0,141
49474 DATA 19,192,32,101,195,173,16
49481 DATA 192,141,18,192,173,17,192
49488 DATA 141,19,192,173,3,192,141
49495 DATA 16,192,173,4,192,141,17
49502 DATA 192,32,47,195,173,11,192
49509 DATA 16,35,173,14,192,141,3
49516 DATA 192,173,15,192,141,4,192
49523 DATA 173,18,192,141,5,192,173
49530 DATA 19,192,141,6,192,32,132
49537 DATA 194,32,180,194,173,11,192
49544 DATA 48,152,32,47,195,173,11
49551 DATA 192,16,18,173,16,192,141
49558 DATA 3,192,173,17,192,141,4
49565 DATA 192,32,132,194,32,31,195
49572 DATA 76,241,192,234,189,20,192
49579 DATA 141,3,192,189,40,192,141
49586 DATA 4,192,173,16,192,141,5
49593 DATA 192,173,17,192,141,6,192
49600 DATA 32,132,194,32,31,195,173
49607 DATA 16,192,141,18,192,141,3
49614 DATA 192,173,17,192,141,19,192
49621 DATA 141,4,192,32,81,195,189
49628 DATA 20,192,141,18,192,189,40
49635 DATA 192,141,19,192,32,101,195
49642 DATA 173,11,192,48,15,189,60
49649 DATA 192,141,18,192,189,80,192
49656 DATA 141,19,192,32,101,195,169
49663 DATA 1,141,18,192,169,0,141
49670 DATA 19,192,173,3,192,141,16
49677 DATA 192,173,4,192,141,17,192
49684 DATA 173,11,192,16,52,189,60
49691 DATA 192,232,157,60,192,202,189
49698 DATA 80,192,232,157,80,192,32
49705 DATA 101,195,173,16,192,157,20
49712 DATA 192,173,17,192,157,40,192
49719 DATA 32,131,195,32,131,195,202
49726 DATA 173,16,192,157,60,192,173
49733 DATA 17,192,157,80,192,76,128
49740 DATA 194,32,131,195,232,173,16
49747 DATA 192,157,60,192,173,17,192
49754 DATA 157,80,192,202,189,20,192
49761 DATA 232,157,20,192,202,189,40
49768 DATA 192,232,157,40,192,202,32
49775 DATA 101,195,32,101,195,173,16
49782 DATA 192,157,20,192,173,17,192
49789 DATA 157,40,192,232,76,162,192
49796 DATA 160,3,165,75,133,79,133
```

```
49803 DATA 81,165,76,133,80,133,82
49810 DATA 24,165,79,109,3,192,133
49817 DATA 79,165,80,109,4,192,133
49824 DATA 80,24,165,81,109,5,192
49831 DATA 133,81,165,82,109,6,192
49838 DATA 133,82,136,208,223,96,160
49845 DATA 0,140,11,192,177,79,141
49852 DATA 7,192,177,81,141,8,192
49859 DATA 200,152,205,7,192,240,2
49866 DATA 176,13,205,8,192,240,21
49873 DATA 144,19,238,11,192,76,30
49880 DATA 195,205,8,192,240,2,176
49887 DATA 62,206,11,192,76,30,195
49894 DATA 140,9,192,160,1,177,79
49901 DATA 133,77,200,177,79,133,78
49908 DATA 172,9,192,136,177,77,141
49915 DATA 10,192,140,9,192,160,1
49922 DATA 177,81,133,77,200,177,81
49929 DATA 133,78,172,9,192,177,77
49936 DATA 200,205,10,192,208,3,76
49943 DATA 195,194,144,184,76,224,194
49950 DATA 96,160,2,177,79,72,177
49957 DATA 81,145,79,104,145,81,136
49964 DATA 16,243,96,169,0,141,11
49971 DATA 192,173,17,192,205,19,192
49978 DATA 144,6,240,8,238,11,192
49985 DATA 96,206,11,192,96,173,16
49992 DATA 192,205,18,192,144,244,208
49999 DATA 238,96,173,16,192,24,109
50006 DATA 18,192,141,16,192,173,17
50013 DATA 192,109,19,192,141,17,192
50020 DATA 96,169,0,141,11,192,56
50027 DATA 173,16,192,237,18,192,141
50034 DATA 16,192,173,17,192,237,19
50041 DATA 192,141,17,192,176,3,206
50048 DATA 11,192,96,238,16,192,208
50055 DATA 3,238,17,192,96,170,170
50062 DATA 170,170,170,170,170,170,170
50069 DATA 170,170,170,170,170,170,170
50076 DATA 170,170,170,170,170,170,170
50083 DATA 170,170,170,170,170,170,170
50090 DATA 170,170,170,170,170,170,170
50097 DATA 170,170,170,170,170,170,170
50104 DATA 170,170,170,170,170,170,170
50111 DATA 170,170,170,170,170,81,85
50118 DATA 73,67,75,83,79,82,84
50125 DATA 32,76,79,65,42,32,32
50132 DATA 3,255,50,48,44,82,69
50139 DATA 65,68,32,69,82,82,79
50146 DATA 82,44,49,56,44,48,48
50153 DATA 0,170,170,170,170,81,85
50160 DATA 73,67,75,83,79,82,84
50167 DATA 32,76,79,65,68,69,82
50174 DATA 16,255,256
```

## Program 4: Sort Test Program

```
100 PRINT "{CLR}"
110 N=1000
120 DIM AA$(N)
130 PRINT "CREATING"N" RANDOM STRINGS"
140 SD=-TI : A=RND(SD)
150 FOR I=1 TO N
160 PRINT I"{UP}"
170 N1=INT(RND(1)*10+1)
180 A$=""
190 FOR J=1TO N1
200 B$=CHR$(INT(RND(1)*26+65))
210 A$=A$+B$
220 NEXT J
230 AA$(I)=A$
240 NEXTI
250 PRINT "HIT ANY KEY TO START SORT"
260 GET A$:IF A$="" THEN 260
270 PRINT "SORTING..."
280 T1=TI
290 REM SYS 31744,N,AA$(1) FOR PET/CBM
291 REM SYS 49154,N,AA$(1) FOR 64
292 REM USE SYS VALUE GENERATED BY THE
    LOADER FOR VIC
300 SYS 31744,N,AA$(1)
310 T2=TI
320 PRINT "DONE"
330 PRINT "HIT ANY KEY TO PRINT SORTED S
    TRINGS"
340 GET A$:IF A$="" THEN 340
350 FORI=1TON:PRINT I,AA$(I):NEXT
360 PRINT:PRINT N" ELEMENTS SORTED IN"(T
    2-T1)/60"SECONDS"
```

©

## Special PET Version Note

PETs with BASIC 4.0 do not have the problem of lengthy *garbage collection times* (this occurs when the computer finds that it has run out of memory, and must eliminate all strings that are no longer "active"). The price of this convenience is that all dynamic strings are now two bytes longer. Those two bytes are a "back-pointer" from the top of the memory (where the actual data contained in the string is kept) to the bottom of memory where the variable keeps a pointer to that data.

This sort does *not* modify the back-pointers. So, if after sorting you continue using the new data, it will eventually be garbled.

There is a solution. Immediately after sorting, write the data to disk as a file. Then issue a CLR command. This will remove all your variables. Then read the data back off the disk into a new array.

This problem does not occur on the VIC-20 or the Commodore 64.

# INSIGHT: Atari

Bill Wilkinson

The new 600XL and 1400XL computers were exactly what I expected (except that Atari goofed and changed the number on the 1201 XL – and that's a joke until you study the case designs of the 1200XL and 1400XL). The 800XL was a little bit of a surprise, but kind of a logical step now that I have the benefit of hindsight. The 1450XLD was a pure delight.

I really could envision a 1450XLD doing some nice, small business work. Especially if you put one of the new three-inch hard disk drives (that's over four megabytes of disk space) into that empty space supposedly designed for a second floppy.

If Atari has any problems at all with the XL line of computers, it may be simply that they are priced too close together. After all, an 800XL is essentially a 600XL with 64K of RAM, and the already announced RAM-pack for the 600XL ends up producing an equivalent machine for the same price. Redundancy.

The 1400XL suffers a little, also. After all, if the rumored price of the 1450XLD holds up ($800-$900 retail), why would you buy a 1400XL and then add a snail's-pace 1050 drive when you can have the much faster XLD for less money? And who but the more sophisticated user will buy a 1400XL when the 600XL (even with expansion to 64K) is so much less? Will the modem and speech synthesizer really prove attractive to a first-time user? Atari marketing obviously thinks so. I think that people who know they want those features will also know enough to want a disk drive.

Anyway, all of that is crystal-balling and nit-picking on my part. The new lineup of computers is one that any company could be proud of. Atari should be doubly complimented after the fiasco with the 1200.

## The New Disk Drives

Before I stop making observations about Atari, though, I would like to carp a bit about one thing: the new Atari disk drives and DOS III (or is it DOS 3?). When I first heard that Atari was going to throw away a potential 50K per disk drive, I thought there was an almost-good excuse. After all, Atari DOS 2.0S could, with absolutely minimum modifications, utilize all the sectors of the one-and-one-third density 1050 drive, so the change, though inefficient when compared to true double-density drives, would allow many current programs to work without modification.

It is not to be. Atari DOS III is just as different from DOS 2.0S as our own Version 4 OS/A+ is. Which means many, many programs (including data base programs, etc.) simply will not work without modification. I do *not* feel this is inherently bad. Let's face it: DOS 2.0S is not a particularly good DOS and it is totally inadequate for larger disk drives. DOS III is actually a very nice DOS for small drives (say up to 128K per drive). It goes downhill rapidly when used on larger drives. This means that if you convert your programs and data files from DOS 2.0S to DOS III this year, you will have to convert to some other DOS again next year, when you move to one of those nice little hard disks I mentioned.

Anyway, when the 1050 finally appears, watch here (I hope) for instructions for using DOS 2.0S (or OS/A+ Version 2) in one-and-one-third density mode, so you won't have to convert all your programs. (You'll still have to convert the diskettes themselves, which won't be easy or fast if you only have one drive, but the same holds true of DOS III – and, to be fair, OS/A+ Version 4 – so you won't have lost anything.)

## Self-Relocatable Machine Language, Part III

This month, I will discuss some more techniques which can be used to make your machine language self-relocatable. Last month, we noted which kinds of instructions were implicitly "safe" (register-only instructions, branches, etc.). There was also a list of "Safe Relocatable Techniques." To summarize, the safe techniques mentioned were:

**1.** Change JMPs to branches.

**2.** Save register values in the stack, not in fixed memory.

**3.** From BASIC, pass the address of a string as a location (or series of locations) to load from or store to.

**4.** Move code from relocatable memory to fixed memory temporarily.

I also promised to discuss two points this month: (1) where the "safe" locations in Atari memory are; and (2) some special techniques usable only with Atari BASIC. Let me fulfill my promise.

## Safe Locations

There are none. Next topic.

Oh, all right, I admit that is a bit of an exaggeration, but it is dismayingly close to the truth. When I write machine language routines, I really do prefer that they be usable with as many products as possible. Just as a start – and *not* as a comprehensive list – I would hope that they would work with the following software: Atari BASIC, Atari DOS, OS/A +, BASIC A +, Atari Microsoft BASIC, *Atariwriter*, Atari Assembler Editor Cartridge, MAC/65, AMAC, and a few more.

Okay. Not too long a list. How many zero page locations are not used by any of those? None. How many Page Six locations ($600 through $6FF) are not used by any of those? None. How many.... But I think you get the idea. Is all this strictly true?

Actually, there are quite a few bytes which can be used for *your* temporary storage. And I suggest you consult your *Atari Technical User's Notes* or *Mapping the Atari* (from COMPUTE! Publications) to find where they are. (Caution: Watch out for changes in the new XL computers.) But even these locations are suspect. What happens if I write this neat new printer-spooler routine which uses location $00 (believe it or not, that's free in almost all the above programs), and then you come along and add a driver for graphics mode 27 and you use location $00?

Perhaps I am being a bit of a purist here. Certainly very little of my own programming is this clean, this free of conflict with other potential programs. And yet it really does require only a little more work to write a program "correctly" (by my definition), so why not do it right? Let's try.

So, we must assume that *no* location outside our own, self-relocatable, properly-loaded-at-LOMEM program is safe at all times. Unpleasant. However, that does *not* say that we can't use some almost-safe locations while our routine has control. In particular, you should be able to use several reserved locations in zero page (for indirect-Y pointers, etc.) by, if necessary, moving values into them from within your relocatable block; using and/or changing the zero page locations in your program; and then moving the values back into your relocatable block.

Sounds complicated? It is. And yet you might be surprised at how seldom you really need to go through all that.

So what zero page locations are safe, even as temporaries? Probably the safest spots, as long as you aren't writing an interrupt handler, are those locations used as temporaries by the DOS File Manager. Locations $43 through $49, inclusive, are always reinitialized by FMS every time it gets control. FMS does *not* presume the locations have maintained their contents from one call to the next. (In fact, the locations should properly be called "Device Driver Zero Page Temporaries," since that is what they were intended for.)

And one more comment before I leave you with the impression that absolutely nothing is safe to do on the Atari computers. If you are writing routines specifically designed to be used with Atari BASIC (as I suspect the majority of you are), there are several safe temporaries. First, you can always use the floating point work area, $D4 through $EF, whenever BASIC calls either a USR routine or an I/O routine. Also, BASIC does not use locations $CB through $CF (only four bytes!) at all. Again, let me give you the caution about adding your routine to a system which already has a custom routine. Be sure there is no conflict.

## A Built-In Relocatable Pointer

It's true. There really is such a thing. There are some *if*s though: *if* you are using Atari BASIC or OSS BASIC A + or OSS BASIC XL; *if* you have placed your relocatable program in a string and are calling a machine language routine via USR( ADR( STRING$ ) ) or USR( "...machine-language-string..."); *if* you don't mind a small trick.

First, the trick. It's really quite simple. Whenever BASIC calls a USR routine, it calls the routine by placing the routine's calculated address in location $D4-$D5 (which just happens to be the first two bytes of floating point register zero). It then JSRs to a routine which simply does a "JMP ($D4)", a jump indirect to the USR routine.

But why can't we take advantage of that pointer? It already points to our relocatable program, so why can't it point to our relocatable data? Perhaps a demonstration is in order.

```
FRØ     =       $D4
USRROUTINE
        CLC
        BCC     START           ; branches are ok
;
SAVEBYTE
        .BYTE Ø                 ; some data
;
; begin actual code
;
START
        LDY     #SAVEBYTE-USRROUTINE  ; index
        PLA                     ; count of parameters
        CMP     #1              ; how many?
        BNE     NOPARAMS        ; none, we presume
; the user is passing a byte to us
        PLA                     ; high byte...ignored
        PLA                     ; low byte...stored
        STA     (FRØ),Y         ; thusly
; we join here, whether a byte is passed or not
NOPARAMS
        LDA     (FRØ),Y         ; get the byte
        STA     FRØ             ; to be returned
        LDA     #Ø
        STA     FRØ+1           ; high byte zero
        RTS
```

This program is a *very* dumb one, for demonstration purposes only. If you call it from BASIC via, for example, "PRINT USR(routine)", your program will print the byte value saved in location SAVEBYTE (zero, initially). On the other hand, if you use "JUNK = USR(routine, 97)", the routine will store the second parameter (97) in location

# THE PROGRAM STORE

*Guarantee:* We will ship your order for items in this ad within 48 working hours or you get 20% OFF! (Books/non-released items excluded. Gift certificate issued when shipped. Offer expires September 10, 1983)

## FINAL FLIGHT

*by Mark Chasin from MMG*

Experience the thrill and tension of piloting a small plane with this real-time flight and landing simulator. Look out of the cockpit, view the runway or fly by instrument only in foggy weather. It's joystick controlled with 4 levels and multiple screen updates per second. Don't crash or it will be your FINAL FLIGHT!

**20912 Atari 24K Tape**
**10900 Atari 24K Disk**
**~~$29.95~~ $25.46**
**Now 15% off til Sept 10**

## SUSPENDED

*from Infocom*

Suspended between the agony of a dying planet and the fun of a great adventure, you are frozen in the planet's underground complex. Wired to computers, you control robots to solve the challenge of survival. Features a multiple save feature to guard the secrets of your mind!

**25186 Atari 32 Disk $49.95**

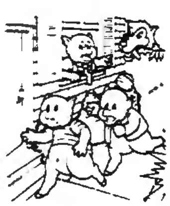
***Also available for C-64, Apple, IBM and TRS-80***

## TIME TRIALS

*from CBS*

It's a timed road rally that gives kids 5-10 years practice in addition and multiplication. Driver chooses the shortest course to reach the number goal. Levels of difficulty based on math abilities. Really makes you think ahead.

**31015 Atari Rom Cartridge $49.95**



## MAGIC STORYBOOK

*Three Little Pigs from Amulet*

For young and old. See three little pigs unfold before you in a beautiful scrolling storybook. Enjoy animation and a delightful musical narrative sound track. Includes the script and coloring insert.

**30261 Atari 16K Tape $29.95**

## SCRAPER CAPER

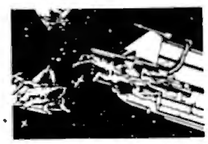

*by Bill Hogue from Big 5*

The legendary mountie, Bounty Bob returns! He's taken a fireman's job in the city to fund the chase for Yubon Yohan. You can help Bob rescue Miss Betty's pet poodle from the burning skyscraper. Watch out for collapsing stairways, crashing elevators and water hazards!

**31206 Atari Rom Cartridge**
**~~$49.95~~ $42.46**
**Now 15% off til Sept 10**

***Also Miner 2049'er, the original adventures of Bounty Bob!***

**39204 Atari Rom Cartridge $49.95**

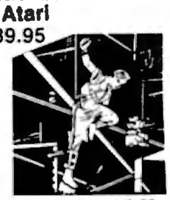


## DEMON ATTACK

*by Dave Johnson from Imagic*

Marooned on the ice planet Krybor, watch eerie creatures stream overhead and hover ominously. Attack and destroy them—or be destroyed! For 1-2 players with ever increasing dangers. Blast'em and survive! **Atari**

**41656 ROM Cartridge $39.95**
**Also available for VIC 20**



## JUMPMAN

*by Randy Glover from Epyx*

New science fiction game! Jumpman must save all 30 levels of Jupiter Headquarters. Scale ladders, girders and perilous ropes while fighting off demonic destroyers. Joystick controlled, 5 game variations, 8 speed, music, sound effects, and graphics. The ultimate test of reflexes for 1-4 players. **Atari**

**41713 32K Disk**
**$39.95**

## BURIED BUCKS

*from Analog*

Fast action "daredevil" game! Equipping your helicopter with explosives, you intend to blast the gold. But your arch-enemy plans to keep the treasure you buried by dropping loads of dirt from his World War II bomber! Incredible 99 levels of joystick controlled action for 1-2 players. It's a race for the bucks as you avoid the falling dirt! **Atari**

**32364 16K Tape or 35873 16K Disk $29.95**

## BASIC COMMANDER

*by Robert Martin from MMG Micro Software*

Atari Basic was made for 8K, so many features were left out. Now you can add capabilities including Single keystroke, LIST, SAVE, ENTER, LOAD, RUN and DOS functions. Look at directory, RENAME, DELETE, LOCK, UNLOCK, NUMBER and FORMAT from Basic. This version is updated to include system reset and convert for double density. The most powerful programming aid for Atari. Requires Basic cartridge.

**12182 Atari 16K Disk ~~$34.95~~ $29.71**
**NOW 15% off til Sept 10**

## BANK STREET WRITER

*from Brøderbund*

The first word processor designed for the whole family! Every function and command is fully and clearly prompted on the screen. You'll want to try all of the powerful features! Requires basic cartridge (for tutorial).

**43850 Atari 48K Disk $69.95**
**Also available for C-64 and Apple II**

## Preparing for the SAT

*from Program Design, Inc.*

Learn how to take the SAT! Features analogies, vocabulary, number series, quantitative comparisons and a unit of time strategy. Includes six complete lessons, manual and booklet Making the Grade. Score higher—get into the best colleges! **Atari**

**16K Tape $99.95**
**24K Disk $99.95**

***New Stores Now Open In***

Rt. 422
Rt. 276 Pa Turnpike
**Plymouth Meeting Mall**
**Plymouth Meeting, PA**

Rt. 130
Rt. 38
Rt. 70
Rt. 295
**Cherry Hill Mall**
**Cherry Hill, NJ**

**Over 2500 Programs for TRS-80, ATARI 400/800, APPLE, IBM & VIC 20.**

***Visit our other stores:***

829 Bethel Rd., Columbus, Ohio
Seven Corners Center, Falls Church, VA
W. Bell Plaza, 6600 Security Blvd., Baltimore, MD
White Flint Mall, Rockville Pike, Rockville, MD
Harvard Square, 13 Dunster St., Cambridge, MA

**Opening Soon in Tulsa and Oklahoma City**

Westmoreland Mall, Rte. 30 East, Greensburg, PA
Plymouth Meeting Mall, 500 Germantown Pike, Plymouth Meeting, PA
Cherry Hill Mall, Rte. 38 & Haddonfield Rd., Cherry Hill, NJ
Nanuet Mall, Rte. 59 & Middletown Road, Nanuet, NY
Monmouth Mall, Rte. 35 & Wyckoff Road, Eatontown, NJ

***FRANCHISE OPENINGS IN SELECTED CITIES***



**To Order Call Toll Free 800-424-2738**
**For Information Call (703) 556-9778**

*Mailorders:*
Send check or M.O. for total purchase plus $2.00 shipping and handling. VA add 4% sales tax.
Charge cards— include all embossed information

THE PROGRAM STORE T.M.

THE PROGRAM STORE • Dept. 10-09-3 • Box 9582 • 4200 Wisconsin Avenue, NW • Washington, D.C. 20016

Item # Title Tape/Disk/Rom/Book Price

Postage $2.00
Total ______
☐ CHECK ☐ VISA
☐ MASTERCARD
Computer ______

Name ______
Address ______
City ______ State ______ Zip ______
Card # ______ Exp ______

For the ATARI 400/800

SAVEBYTE. Presumably, you could then later recover the 97.

The point to be made, however, is that this program is completely self-relocatable and yet is able to load and store data from within its own relocatable block! The secret is the "LDY #SAVEBYTE-USRROUTINE" line directly after the label START. Since location FR0 contains the address of USRROUTINE, loading the Y-register with the proper offset (SAVEBYTE-USRROUTINE, which happens to be 3 in our example) will allow us to do indirect loads and stores to any location within 255 bytes following USRROUTINE.

Can I put that more clearly? Since, when we do either the "LDA (FR0),Y" or the "STA (FR0),Y", the Y register contains the value 3 and location FR0 points to the location USRROUTINE, the LDA and STA instructions will reference the third byte after USRROUTINE. Which just happens to be SAVEBYTE.

And just a reminder if you don't know or remember what the PLA instructions in this program are for. Whenever BASIC calls a USR routine, it pushes all the parameters it is given onto the CPU stack (after first converting them to 16-bit integers, of course). Then, the last thing it does before the call is to push a count of the number of parameters (presumed to be 1 or 0 in our example) onto the same stack. Thus, the first PLA lets us discover how many parameters were passed. The other two PLAs are necessary if a parameter is passed; otherwise the RTS instruction will return to an unknown location and will likely crash the system. (Note that in our simple-minded example you can probably crash BASIC by calling the routine with two parameters, since no check is made for more than one parameter.)

Next month we're going to take this technique a couple of steps further. We will discover how to have more than 255 bytes of relocatable storage (which may or may not be useful to you) and how to generate similar self-pointers when the routine in question has not been called from BASIC. ©

# Easy Atari Page Flipping

Chris Allen

*Here's a short program that lets you display one screen creation while drawing another offscreen. Put them together and you've got page flipping.*

Have you ever wished that you could just POKE a couple of locations and have a complicated picture appear on your Atari? This demo program will show you how to use *page flipping* – changing the addresses that tell the Atari where screen memory is. Page flipping will allow you to show one picture and, at the same time, draw another picture offscreen. You don't see it drawn: it just "appears" instantly.

Page flipping allows you to draw offscreen using the normal graphics commands (PLOT, DRAWTO, etc.), or, if you use a text mode, to PRINT normally. You don't have to do any spectacular POKEing.

The method is simple. The Atari keeps two separate two-byte registers for the address of screen memory. The first register, locations 88 and 89 (decimal), is used *solely* for PRINTing, PLOTting, etc.; it is not concerned with display. The second register, bytes five and six of the display list (located by PEEK(560) + PEEK(561)*256), is used *only for display*. Having two locations simplifies matters – changing the first register allows you to draw offscreen, while changing the display list register will "flip" your new screen into view.

A few cautions are in order. First, page flipping uses a lot of memory. Since one GRAPHICS 7 screen uses 3200 bytes, two such pictures are impossible on an 8K machine. However, GRAPHICS 5 uses only 800 bytes, ideal for computers with limited memory. Second, be sure to clear any garbage from the area you have reserved for your new screen. Third, if you modify the display list, be aware that your new display list may not have the screen address register in the same location as a normal list. (If you can change the display list, you should be able to handle this minor problem.)

Now that the warnings are out of the way, let's do some page flipping. First, type in this short program:

```
10 GRAPHICS 5
20 GOSUB 200
60 END
200 COLOR 1:FOR I=0 TO 79:PLOT I,0:D
    RAWTO I,39:NEXT I:RETURN
```

When you run it, notice that you can see the screen being filled in. Now add these lines to enable page flipping:

```
5 POKE 106,PEEK(106)-4:SCREEN2=PEEK(
  106)
15 SCREEN1=PEEK(89):POKE 89,SCREEN2
25 B=PEEK(560)+PEEK(561)*256
30 FOR I=1 TO 100
35 POKE B+5,SCREEN2
40 FOR J=1 TO 200:NEXT J
45 POKE B+5,SCREEN1
50 FOR J=1 TO 200:NEXT J
55 NEXT I
```

The picture is drawn offscreen, where you can't see it. By switching values in the display register (B + 5 is the sixth, or high byte), you can alternate or "flip" between screens. Here's a line-by-line explanation:

**Line 5** reserves memory for the second screen and sets up a pointer to the reserved area.

**Line 15** sets a pointer to the present screen, then flips the draw register over to screen two.

**Line 25** finds the start of the display list.

**Lines 30-55** simply loop 100 times, alternating the screen displayed each time.

Although we changed only the high byte, the low byte (88 or B + 4) can also be changed. (Try changing just B + 4 – and you're screen scrolling.)

# How To Create A Data Filing System

## Part III. Planning The Input

Jim Fowler

*A little foresight in planning your input can save a lot of time and frustration. In Part III, the author tells how to handle some common input problems and offers some advice on how to prevent problems down the road.*

In the first two installments we discussed setting goals for the kind of system you want, the types of files, and what kind of output is best. For most cases, a relative file structure with index files gives flexibility and speed. The index files will be composed of index words which will be either shortened versions of data in the records themselves or bytes encoded with some kind of bitmapping.

Before discussing input strategies, let's review some of the ideas from Part II in a bit more detail. We discussed setting up a buffer for inputting keys or index words. This buffer can be any free area of unused RAM memory. It must be large enough to accommodate the record or field to be compared. For example, if your index word is the first eight letters of the author's name, create an eight-byte buffer for your comparison.

### A Closer Look At Indexing

Another technique we discussed was building your index file into your record format. For example:

| AUTHOR | SUBJECT | TITLE | YEAR | INDEX | AUTHOR |
|---|---|---|---|---|---|

←——— Record 1 ———→←— Record 2 ——

After entering your first record – author, subject, title, year – you can reserve several bytes at the end of that record to create an index file. If you choose to bitmap here, as illustrated in last month's installment, you gain search efficiency, although it may at first seem tedious when creating the index this way.

If you use one byte in the index for each field, you then have 256 possibilities for each field, which in most cases would be more than adequate. Using last month's illustration, a bit configuration of 1000 0000 would indicate a subject on computers. Since the integer equivalent of a binary 1000 0000 is 128, you can use this with an AND for compare. Let's say you've chosen the variable SU (for subject field). The appropriate line would be:

```
IF SU AND 128 THEN GOTO n
```

where n is a line that will direct a PRINT to screen or printer.

When using an AND, the computer will test individual bits. The value in SU, 1000 0000, is compared to 128:

```
1000 0000  (SU)
1000 0000  (128)
---------
1000 0000  (result)
```

The Boolean truth table, remember, makes this compare result "true," thus a "hit" is made in your search.

In some cases, depending on the total number of subjects you want to index, it might be practical to assign variable names to the binary equivalents:

```
A=1    E=16
B=2    F=32
C=4    G=64
D=8    H=128
```

Then, IF SU AND H THEN n.

Let's say you're searching for a more specific

subject, computers in education. We'll assign the subject of education a binary 0100 0000 (or integer 64). A computer subject, remember, was assigned 1000 0000 (128). A book dealing with computers and education would then be 1100 0000 (192). Our search statement would be:

```
IF SU AND 192 THEN n
```

Obviously, if you use this method, you'll have to be very thorough in creating your index. No matter what method of indexing you choose, do it carefully – your search speed and accuracy depend on it.

If you choose not to use the bitmapping method, a word of caution is in order: be sure to write the data that makes up your index file(s) also in the *records themselves*. You may later decide to change the format of an index file to rewrite a search routine. Maybe you will be forced to do this to accommodate an index file you found you needed. The easiest way to create the new index file is to read it item by item from the disk and assemble the index that way, rather than to type it in by hand. The accuracy will be much greater. Remember that one wrong bit in an index makes the record it refers to "invisible" to a search.

## System Input Problems

Now for the problems with input. You want a system which is easy to use. This means giving cues that tell the user what is going on. One way is to use the top one or two lines on the screen to indicate what the program is doing or expecting at all times. Another important feature is to make the screen format logical and easy to understand.

Finally, when inputting new records, there should be ample opportunity to edit, erase, change, or abort without disturbing or crashing the program.

Some computers, including my CBM, cannot handle a string input containing commas. The operating system looks for these delimiters in an input string. When I input titles of publications, commas are important punctuation. That means I have to use a roundabout way of getting the string in without having it cut off at the comma. There are several ways of doing this. You can use GET and assemble the string byte by byte.

I have used a nice routine for Commodore equipment written by Jerry Dunmire (COMPUTE!, December 1981). This routine takes up to 80 characters in a string which can contain any symbols you want. If the 80-character limit is exceeded, you can tell by the value of ST, a status byte in the operating system. Problems like this should be handled at the outset. Make the system easy to use. A little frustration becomes a big one when you are typing in data. Having to substitute something else for commas would be very frustrating.

One thing to remember in connection with input is that the program must "know" at all times the number of records on the disk and the length of each index file. When you enter a new record, it must go into the very next empty location on the disk. The new record's index words must be put at the end of the appropriate index files. The way to save this information from one run to the next is to have a register pointing to the next record number. Inputting a new record will cause the register to be incremented by one. When you SAVE the index files, you should also SAVE this register and if the register is adjacent to the index files, you can save them all at once.

## Writing The Input

Any writing of data should be done as it is input. For example, if there is to be a change from ASCII letters (or in my case, PETSCII), then that ought to be done when the time delay is not objectionable. After you type a name, and after you have a chance to edit it, you should be asked to give a final approval. Once this is given, the program ought to translate parts of the input before writing (sending the input) to the disk. This might take a few seconds, but if you are typing records from a list or card file, you will be reading the next item or moving the pointer on the copy stand while this goes on.

For example, this is how I handle my index file of authors. On the disk, the author's name is in capital and lowercase, last name first, with commas and periods after initials. In the index file all letters are written as pseudo-ASCII caps, and the index word ends with the eighth letter of the last name. To make pseudo-ASCII, all you need to do is shorten each ASCII byte to five bits with "AND 31" (or AND #$1F). If the last name is shorter than eight letters, I let the following comma and initials appear, too. The key used in searching for an author is also changed to pseudo-ASCII caps. After the last letter, the extra bytes, if any, are nulls. As mentioned, the search program then considers it a match when the next byte of the key is a null. That way you can search for SMITH,J. or SMITH, or even all the S's. That's very helpful when you aren't sure about the spelling of a name. Program 1 in the previous article illustrates this search technique.

Bitmapping is not hard. You can do it in machine language, but there is no particular advantage in doing so, except saving program space. The byte in question is zeroed and then the *n*th power of two is added to it whenever you want the bit in the *n*th position set. You can clear the same bit by subtracting. Be sure the bit is set before you do any subtracting and vice versa, and be sure it is clear before setting it. You must arrange it so the user cannot inadvertently set a bit twice

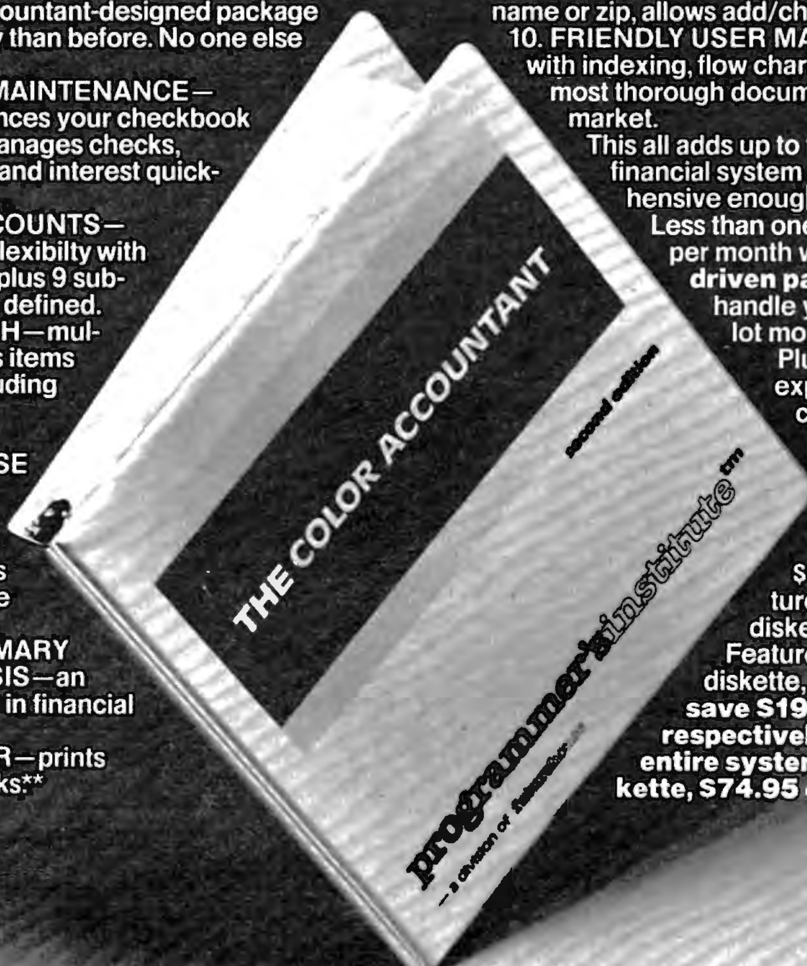
# Complete Personal Accountant: we've made the best much more friendly.

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

**Routine To Set Bits In An Index Word.**

*(This routine is based on Y/N response with cursor moving down list on screen. You must arrange a stop or wraparound when N gets to maximum and P=7. Same at N=0:P=0.)*

1. DIM the array IW(x) to nr of bytes in index word.
   Zero IW if not already done initially.
2. Print subjects in list on screen.
   N=0;P=0 Zero byte nr and bit nr.
   Print "cursor" opposite zeroth subject.
3. GET loop.
4. If SPACE, move cursor down: P=P+1.
   IF P=8, then N=N+1;P=0.
5. If SHIFT-SPACE, move cursor up.
   IF P=-1, then N=N-1;P=7.
6. If Y, then move cursor down.
   If subject is marked, then GOTO 3.
   Else, add 2 P to IW(N); mark subject.
7. If N, move cursor down.
   If subject not marked, GOTO 3.
   Else, subtract 2 P from IW(N); clear mark.
8. Other inputs invalid: GOTO 3.

or clear a bit that isn't set. The table shows a routine for inputting subjects by bitmapping.

Particularly sticky situations can always be handled with a table. An array with the existing value for each value of the input is one way of doing this: A(N) contains the value used for N, the input value.

## Editing The Files

By all means, make it easy to display a record entered some time ago, edit the display, and write the newly changed data in place of the original record. If you use subroutines for inputting each kind of data, this is easy to program.

For example, I have a subroutine that takes as input an author's name, then when it's acknowledged to be correct, writes it in the correct place on record "n" and also puts a corrected entry in the author index file in the right place. The record "n" may be an old one or the one we are writing for the first time. All you need to do is branch to such routines as one of the options given on a menu at the top of the program. Some errors will inevitably get by in your initial input. You need a way to correct errors both at the input and later as well.

Next issue we will outline the main program and talk about other techniques. ©

# Mixing Graphics Modes On The 64

## Part II

Sheldon Leemon

*The two programs in the first part of this article (last month's COMPUTE!) showed you how to have different graphics modes simultaneously on the 64 screen. To conclude this discussion of mixing modes, here is a machine language program which uses a mixed graphics mode display to demonstrate the raster interrupt.*

The interrupt uses a table of values that are POKEd into four key locations during each of the three interrupts, as well as values to determine at what scan lines the interrupts will occur. The locations affected are Control Register 1, Control Register 2, the Memory Control Register, and Background Color 0.

Control Register 1 (at location 53265) allows the selection of extended background color text mode, bitmap mode, screen blanking, and 24 or 25 rows of text. Control Register 2, at 53270, controls the selection of multicolor mode, and of a 38- or 40-column display. The Memory Control Register (53272) allows you to select which portion of VIC memory will be used for the video display (screen memory), and which for the data that defines the shape of text characters. Background Color Register 0 (53281) controls the background color in text mode. More detailed information about the bit assignments of these locations can be found in Appendix O of the *Commodore 64 User's Guide* and in the *Programmer's Reference Guide*.

The data for the interrupt routine is contained in lines 49152-49276. Each of these line numbers corresponds to the location where the first data byte in the statement is POKEd into memory. If you look at lines 49264-49276 of the BASIC program, you will see REMark statements that explain which VIC-II registers are affected by the DATA statements in each line. The numbers in these DATA statements appear in the reverse order in which they are put into the VIC register. For example, line 49273 holds the data that will go into Control Register 2. The last number, 8, is the one that will be placed into Control Register 2 while the top part of the screen is displayed. The first number, 24, is placed into Control Register 2 during the bottom part of the screen display and changes that portion of the display to multicolor mode.

The only tricky part in determining which data byte affects which interrupt comes in line 49264, which holds the data that determines the scan line at which each interrupt will occur. Each DATA statement entry reflects the scan line at which the *next* interrupt will occur. The first item in line 49264 is 49. Even though this is the entry for the third interrupt, this number corresponds to the top of the screen (only scan lines 50-249 are visible on the display). That is because after the third interrupt, the next to be generated is the first interrupt, which occurs at the top of the screen. Likewise, the last data item of 129 is used during the first interrupt to start the next one at scan line 129, in the middle of the screen. Try experimenting with these values to see what results you come up with. For example, if you change the number 169 to 209, you will increase the text area by five lines (40 scan lines).

### Changing Effects

By changing the values in the data tables, you can alter the effect of each interrupt. Change the 20 in line 340 to 22, for example, and you will get lowercase text in the middle of the screen. Change the first 8 in line 49276 to 24, and you will get multicolor text in the center window. Each of these table items may be used in exactly the same way that you would use the corresponding register, in order to change background color, to obtain text or bitmap graphics, regular or multicolor modes, screen blanking, or extended background color mode.

```
0000            ; VIC CHIP EQUATES
0000            ;
0000            SCROLY = $D011        ;CONTROL REGISTER 1
0000            RASTER = $D012        ;RASTER LOCATION
0000            SCROLX = $D016        ;CONTROL REGISTER 2
0000            VMCSB  = $D018        ;V/C BASE ADDRESS
0000            VICIRQ = $D019        ;LATCHES ON IRQ FROM VIC
0000            IRQMSK = $D01A        ;VIC IRQ ENABLE
0000            BGCOL0 = $D021        ;BACKGROUND COLOR 0
0000            CIAICR = $DC0D        ;INTERRUPT CONTROL
0000            ;
0000            INTNO  = $FB          ;INTERRUPT COUNTER
0000            ;
0000                   * = $C000
C000            ;
C000 78         SETIRQ SEI            ;DISABLE ALL INTERRUPTS
C001 A9 7F             LDA #$7F
C003 8D 0D DC          STA CIAICR     ;DISABLE CIA INTERRUPTS
C006 A9 01             LDA #01
C008 8D 1A D0          STA IRQMSK     ;ENABLE RASTER IRQ
C00B A9 03             LDA #03
C00D 85 FB             STA INTNO      ;INITIALIZE INTERRUT NO.
C00F AD 70 C0          LDA RASTBL
C012 8D 12 D0          STA RASTER     ;SET SCAN LINE OF TOP RIRQ
C015 A9 18             LDA #24
C017 8D 11 D0          STA $D011      ;SET HIGH BIT OF RIRQ SCAN LINE
C01A            ;
C01A AD 14 03          LDA $314       ;SAVE OLD IRQ VECTOR AND
C01D 8D 6E C0          STA OLDIRQ+1   ;MODIFY OLDIRQ TARGET ADDRESS TO
C020 AD 15 03          LDA $315       ;INSURE AGAINST CHANGE IN ADDRESS
C023 8D 6F C0          STA OLDIRQ+2   ;OF NORMAL INTERRUPT ROUTINE
C026            ;
C026 A9 32             LDA #<RASIRQ   ;SET IRQ VECTOR
C028 8D 14 03          STA $314       ;TO USER ROUTINE
C02B A9 C0             LDA #>RASIRQ
C02D 8D 15 03          STA $315
C030 58                CLI            ;RE-ENABLE INTERRUPTS
C031 60                RTS
C032            ;
C032            ;
C032 AD 19 D0   RASIRQ LDA VICIRQ
C035 8D 19 D0          STA VICIRQ     ;CLEAR VIC INTERRUPTS
C038 29 01             AND #01        ;IS RASTER THE SOURCE OF IRQ?
C03A F0 2B             BEQ INTRT      ;NO, EXIT
C03C C6 FB             DEC INTNO      ;NEXT INTERRUPT
C03E 10 04             BPL RAS1       ;NOT LAST INTERRUPT
C040 A9 02             LDA #2         ;LAST INTERRUPT, RESET COUNTER
C042 85 FB             STA INTNO
C044            ;
C044 A6 FB      RAS1   LDX INTNO
C046 BD 73 C0          LDA COLTBL,X   ;SET BACKGROUND COLOR
C049 8D 21 D0          STA BGCOL0
C04C BD 76 C0          LDA CR1TBL,X   ;SET CONTROL REG 1
C04F 8D 11 D0          STA SCROLY
C052 BD 79 C0          LDA CR2TBL,X   ;SET CONTROL REG 2
C055 8D 16 D0          STA SCROLX
C058 BD 7C C0          LDA MEMTBL,X   ;SET MEMORY CONTROL
C05B 8D 18 D0          STA VMCSB
C05E BD 70 C0          LDA RASTBL,X   ;RESET INTERRUPT SCAN LINE
C061 8D 12 D0          STA RASTER
C064 8A                TXA            ;LAST INTERRUPT EXITS
C065 F0 06             BEQ OLDIRQ     ;THROUGH OLD VECTOR
C067            ;
C067 68         INTRT  PLA            ;RESTORE STACK
C068 A8                TAY
```

COMMODORE 64

# SHOOTING GALLERY

- *100% MACHINE CODE*
- *REAL TIME MUSIC*
- *COLORFUL SCROLLING GRAPHICS*
- *GOOD PLAY ABILITY*

CASSETTE DISK **$23.95**

# KOOKY CLIMBER

- *100% MACHINE CODE*
- *ARCADE QUALITY GRAPHICS*
- *EXTENDED REAL TIME MUSIC*
- *EXCITING ANIMATED SPRITES*

DISK **$24.95**


*SPECIFY CASSETTE OR DISK*
*N.Y. RESIDENTS PLEASE ADD 7¼% SALES TAX*
*PLEASE ADD $2.00 SHIPPING AND HANDLING.*
*PREPAID (CHECK OR MONEY ORDER) OR C.O.D. ONLY*

## RENSOFT SOFTWARE SYSTEMS

P.O.Box 170 Coram
New York 11727

*SOFTWARE AUTHORS WANTED*
*(516) 732-3390*
DEALER INQUIRIES INVITED

---

TIMEWORKS

# ATTENTION C-64 PROGRAMMERS!

If you have a professional quality unique computer program ready to go for the Commodore 64, we would like to discuss with you the possibility of publishing your entertainment, educational or home small business program for distribution on a national basis.

Timeworks is a national publisher and distributor of Commodore 64 (& other) Software with over 100 field Sales Representatives servicing mass merchants and large retail chains in all 50 States and Canada. Our products are sold in over 1000 retail outlets, computer stores and chain stores. (And that doesn't even include mail order.)

Timeworks prides itself on quality software and effective merchandising. Our programs are both User smart and fulfill a specific need. Entertainment software always includes that special element to capture the player's interest. Our software packaging is accepted as some of the most creative in the industry and we keep our overall product quality at the highest level.

There is a great demand for Commodore 64 Software. If you feel your program meets our standards, we'll pay you generous royalties and fringe benefits for accepted works. Let Timeworks put your program in major stores across the country.

For more information, feel free to contact me on our Programmers' Hot Line, or write to Vic Schiller, Vice President, Development

**Programmers' Hot Line 800-323-9755**

TIMEWORKS, INC.
405 Lake Cook Road Deerfield, IL 60015

---

# the *foxiest* WORDPROCESSOR FOR THE COMMODORE 64™ ALSO CHECKS YOUR SPELLING!

**"REALLY FOXY IS BEING LETTER PERFECT"**

# SCRIPT 64™

Suggested Retail: $99.95

**Contact Your Nearest Commodore Dealer Today . . .**
**You'll Be So Glad You Did!**

**Distributed By:**

**COMPUTER MARKETING SERVICES INC.**

300 W. Marlton Pike
Cherry Hill, New Jersey 08002
(609) 795-9480

Commodore 64 is a trademark of Commodore Electronics Limited
Script 64 is a trademark of Richvale Telecommunications

```
00077 C069 68              PLA
00078 C06A AA              TAX
00079 C06B 68              PLA
00080 C06C 40              RTI
00081 C06D          ;
00082 C06D 4C 31 EA OLDIRQ JMP $EA31        ;OLD IRQ--ADDRESS MODIFIED ABOVE
00083 C070          ;
00084 C070 31       RASTBL .BYT 49,170,129;SCAN LINE OF NEXT INTERRUPT
00084 C071 AA
00084 C072 81
00085 C073 00       COLTBL .BYT 0,6,0       ;BACKGROUND COLORS
00085 C074 06
00085 C075 00
00086 C076 3B       CR1TBL .BYT 59,27,59    ;CONTROL REGISTER 1 VALUES
00086 C077 1B
00086 C078 3B
00087 C079 18       CR2TBL .BYT 24,8,8      ;CONTROL REGISTER 2 VALUES
00087 C07A 08
00087 C07B 08
00088 C07C 18       MEMTBL .BYT 24,20,24    ;MEMORY CONTROL REGISTER VALUES
00088 C07D 14
00088 C07E 18
00089 C07F
00090 C07F                 .END
```

©

C. Regena

# Subscripted Variables

TI BASIC allows variable names to be *subscripted*, or used in arrays of up to three dimensions. Examples of subscripted variables are A(1), ING$(2,6), and N(7,2,8).

Both numeric and string variables may use subscripts, which are written as numbers in parentheses after the variable name. The subscript itself may be a numeric variable or numeric expression. One constraint is that you cannot use the same variable name both with and without a subscript; that is, you cannot use the variable N and the variable N(3).

## Just Like Mailboxes

I often think of variables as a mailbox system in memory:

| A | B |
|---|---|

Here are two variables, named A and B. Initially, they each have the value of zero. As your program runs, you may assign values to these boxes. Suppose you have the statements:

```
100 B=7
150 A=A+1
```

The computer will put the value 7 in B's mailbox, then any later statement using B will simply use 7 in the formula instead of B. Line 150 says to add 1 to the value that is currently in A, then place the new value in A.

Some mailboxes are larger than others, and I compare these to subscripted variables. You might think of it as a big box for the Smith family – the first part of the box for John, the second part for James, and the third part for Jeremy. Here is our mailbox again:

| A | B |
|---|---|
| $C_1$ | $C_2$ |

The C box actually holds two values, which are written in TI BASIC as C(1) and C(2).

Boxes can be even larger – representing 1, 2, or 3 "dimensions," or using 1, 2, or 3 numbers in the subscripts. C(2) is the second element in the one-dimensional array of C above. N(2,4) would be an element in a two-dimensional array. X(3,4,2) would be an element in a three-dimensional array.

## Arrays Are Workhorses

Arrays or subscripted variables can make a computer program more efficient in many cases. If you use a process several times, it may be worth using a variable with a subscript rather than several variables.

For example, suppose you are using your computer to sort a list of 25 students with their scores on a particular test. You could use the following method:

```
200 INPUT A$,A (FIRST STUDENT, SCORE)
210 INPUT B$,B (SECOND STUDENT, SCORE)
220 INPUT C$,C (THIRD STUDENT, SCORE)

ETC., FOR 25 STUDENTS

.
.
.
(SORT ROUTINE USING 25 VARIABLES)

.
.
.
600 PRINT A,A$
610 PRINT B,B$
620 PRINT C,C$

ETC., FOR 25 SORTED SCORES AND STUDENTS.
```

Using arrays or subscripted variables, you could INPUT the names as the N$ array and the corresponding scores in the SC array, sort, and then print using this method:

```
200 FOR C=1 TO 25
210 INPUT N$(C),SC(C)
220 NEXT C
(SORT ROUTINE)
600 FOR C=1 TO 25
610 PRINT SC(C),N$(C)
620 NEXT C
```

Here is another example program that would be considerably longer if you did not use subscripted variables. Lines 110-130 READ from DATA a subject, a verb, and a phrase and put them in the S$, V$, and P$ arrays. Lines 140-190 contain the data (you could combine data lines if you wish). For the first time through the loop, S$(1) would be "I"., V$(1) would be "RAN", and P$(1) would be "TO OUR HOUSE." S$(2) is "HE", V$(2) is "WALKED", and P$(2) is "TO THE STORE."; and so forth.

Line 200 uses the DEF function to define R6 as a random integer from 1 to 6. Each time R6 is used in the program, the computer will choose a random number from 1 to 6.

Line 210 clears the screen, and line 220 prints a title. Lines 230-240 choose a random S$, a random V$, and a random P$ to make up a sentence and print it. Line 250 returns to line 230 to repeat the process until you press CLEAR.

```
100 REM RANDOM SENTENCES
110 FOR C=1 TO 6
120 READ S$(C),V$(C),P$(C)
130 NEXT C
140 DATA I,RAN,TO OUR HOUSE.
150 DATA HE,WALKED,TO THE STORE.
160 DATA SHE,HOPPED,AROUND THE ROOM.
170 DATA IT,SPED,UP THE HILL.
180 DATA WE,ZOOMED,ACROSS THE GRASS.
190 DATA YOU,JUMPED,ALONG THE PATH.
200 DEF R6=INT(6*RND)+1
210 CALL CLEAR
220 PRINT "** RANDOM SENTENCES **":::
230 RANDOMIZE
240 PRINT :S$(R6);" ";V$(R6);" ";P$(R6)
250 GOTO 230
260 END
```

## Memory Reserved

As soon as you specify a variable name with a subscript, the computer automatically reserves memory for an array with that name. If you use a variable D(3), the computer will automatically reserve elements up to D(10). In two-dimensional arrays, the computer will reserve up to N(10,10); and in three-dimensional arrays, the computer will reserve up to X(10,10,10).

If you need more than ten elements, use a DIMension statement to clear enough space. For example, for our 25 students and 25 scores in the program discussed previously, we would need a DIMension statement:

```
100 DIM N$(25),SC(25)
```

If your program is running nearly full memory and you do not need all the elements automatically reserved, you may save memory by dimensioning the array for the exact number you need:

```
100 DIM N(6)
```

The DIMension statement must appear before any reference to the array. I usually put my DIMension statements near the beginning of the program. You may specify several variables in one DIMension statement.

The computer actually starts all subscripts with the zero element, N(0). Thus, the automatic dimensioning includes 11 elements in arrays. If you prefer to use only elements numbered 1 and above, you may use the OPTION BASE statement to avoid reserving space for the zero elements:

```
100 OPTION BASE 1
110 DIM D(25,6)
```

*Note:* The OPTION BASE 1 statement must precede the DIM statement.

## Combining The Ingredients

Following is an educational program which illustrates the use of subscripted variables. The program prints a recipe conversion problem for a math competency test. First, one of three recipes is printed. A random ingredient is chosen, and a random multiplication factor is chosen to print the problem. The student must choose from four possible answers.

Line 140 DIMensions the R$ array and the R array so the first subscript may go up to 3 and the second subscript may go up to 6. The first subscript will actually be 1, 2, or 3, which will correspond to the first, second, or third recipe. R$(C,0) will contain the title of the recipe for each of the three recipes. R(C,0) will be the number of servings each of the three recipes will make. R(C,I) and R$(C,I) contain the amount and the ingredient, where C is the recipe number and I is from 1 to 6. The values are read in as DATA in lines 150-230.

Lines 410-440 define values for the elements of the J array. These elements are multiplication factors for the conversion problem. These variables are used first to choose a factor for the problem, then to calculate the multiple-choice answers.

**Program Structure**

| <u>Lines</u> | |
|---|---|
| 100-130 | Print title screen. |
| 140 | DIMension arrays for recipe elements. |
| 150-200 | READ from DATA the values for the R$ and R arrays. |
| 210-230 | DATA for recipes (please be careful while copying these lines – watch the commas and decimals). |
| 240 | Branch around subroutines. |
| 250-390 | Subroutines to convert decimals to fractions for printing the recipes and the multiple-choice answers. |
| 400 | Clear screen for problem. |
| 410-440 | Define multiplication factors. |

450-460 Randomly choose Recipe 1, Recipe 2, or Recipe 3. The variable C refers to the recipe number.
470-480 Print title of recipe and number of servings.
490-530 Print amount, measure, and ingredient six times. One of the recipes contains only five ingredients, so line 500 checks for a zero value. Line 510 converts the amount from a decimal to a fraction if necessary.
540-560 Randomly choose a multiplication factor for the problem. If F = 1 then J(1) = 1 which indicates no recipe conversion, and another number is chosen.
570-590 Draw a horizontal line of a random color under the given recipe.
600-640 Print the question, where A is the randomly chosen ingredient.
650 Calculate correct answer as N1.
660-750 Randomly print multiple-choice answers.
760-780 Sound a "beep" then wait for answer.
790-820 If answer is incorrect, play "uh-oh" and return for another answer.
830-870 Indicate correct answer and play arpeggio.
880-910 Print option to try another problem and branch appropriately.
920-930 Clear screen and END.

### Math Competency Recipe Conversion

```
100 CALL CLEAR
110 PRINT TAB(6);"MATH COMPETENCY"
120 PRINT :::TAB(5);"RECIPE CONVERS
    ION"
130 PRINT
140 DIM R$(3,6),R(3,6)
150 FOR C=1 TO 3
160 READ R$(C,0),R(C,0)
170 FOR I=1 TO 6
180 READ R(C,I),R$(C,I)
190 NEXT I
200 NEXT C
210 DATA CHEESE SOUFFLE,2,2,TBSP BU
    TTER,2,TBSP FLOUR,1,C. MILK,.75
    ,C. GRATED CHEESE,2,EGGS,.5,TSP
     SALT
220 DATA DUMPLINGS,4,1,C. FLOUR,2,T
    SP BAKING POWDER,.5,TSP SALT,.5
    ,C. MILK,2,TBSP SALAD OIL,0,""
230 DATA PRONTO PUPS,6,2,EGGS,.5,C.
     MILK,.75,C. FLOUR,1,TSP BAKING
     POWDER,1,TSP SALT,.5,C. CORN M
    EAL
240 GOTO 400
250 N=R(C,I)
260 IF N<1 THEN 290
270 N$=STR$(N)
280 RETURN
290 IF N<>.75 THEN 320
300 N$="3/4"
310 RETURN
320 IF N<>.5 THEN 350
330 N$="1/2"
340 RETURN
350 IF N<>.375 THEN 380
360 N$="3/8"
370 RETURN
380 N$="1/4"
390 RETURN
400 CALL CLEAR
410 J(0)=.5
420 J(1)=1
430 J(2)=2
440 J(3)=4
450 RANDOMIZE
460 C=INT(RND*3)+1
470 PRINT TAB(7);R$(C,0)
480 PRINT :"SERVES";R(C,0)::
490 FOR I=1 TO 6
500 IF R(C,I)=0 THEN 530
510 GOSUB 250
520 PRINT N$;TAB(5);R$(C,I)
530 NEXT I
540 F=INT(RND*4)
550 IF F=1 THEN 540
560 F=J(F)
570 H=INT(RND*12)+5
580 CALL COLOR(13,H,H)
590 CALL HCHAR(24,1,128,32)
600 PRINT :::"IF YOU WANTED TO MAKE
    "
610 PRINT R$(C,0);" TO SERVE";F*R(C
    ,0)
620 A=INT(RND*5)+1
630 PRINT "HOW MANY ";R$(C,A)
640 PRINT "WOULD YOU NEED?"::
650 N1=F*R(C,A)
660 FOR CH=1 TO 4
670 X=INT(RND*4)
680 IF J(X)=-1 THEN 670
690 N=J(X)*R(C,A)
700 IF N1<>N THEN 720
710 ANS=CH
720 GOSUB 260
730 PRINT TAB(6);CHR$(64+CH);"  "&N$
740 J(X)=-1
750 NEXT CH
760 CALL SOUND(150,1497,2)
770 CALL KEY(0,K,S)
780 IF S<1 THEN 770
790 IF K=ANS+64 THEN 830
800 CALL SOUND(100,330,2)
810 CALL SOUND(100,262,2)
820 GOTO 770
830 CALL HCHAR(19+ANS,7,42)
840 CALL SOUND(100,262,2)
850 CALL SOUND(100,330,2)
860 CALL SOUND(100,392,2)
870 CALL SOUND(200,523,2)
880 PRINT :"ANOTHER PROBLEM? (Y/N)"
    ;
890 CALL KEY(0,K,S)
900 IF K=89 THEN 400
910 IF K<>78 THEN 890
920 CALL CLEAR
930 END
```

©

Jim Butterfield, Associate Editor

# Bagel Break, Part 2

Last month we outlined the logic of a simple machine language program to play "Bagels," a well-known guessing game. Let's pause and look at the various ways we can change our planned program into a real machine language program.

You may have a tiny assembler that is built into your monitor system. This type of simple assembler is often called a *nonsymbolic* assembler for reasons we'll discuss in a moment. If so, you'll work out all the addresses yourself and write them in as you jot down the program coding. The type of outline you write will be similar to that in Program 2. You'll need to guess at some of the "forward" branches; at the time you write the branch instruction, you won't know what the exact destination address will be. No matter, as long as you remember to put the correct addresses in later.

You may have purchased a full-scale assembler, in which case you'll write the program as shown in Program 1. It's still the same logic flow, but now we can give a name (or "symbolic address") to the various parts of the program. We'll let the assembler figure out where these locations are and compute the correct branch for us. This type of assembler, where we can name locations with symbolic names, or "labels," is often called a "symbolic assembler" to distinguish it from the simple assemblers mentioned previously.

Symbolic names, or labels, seem like a convenience feature at first: not too important, but handy. In fact, they change the nature of the work in a couple of ways. First, we now have the freedom to give meaningful names to our program and work locations. The program is easier to read. Second – and this can be very powerful – we can move the logic to an entirely new part of memory with very little work; the assembler will figure everything out for us. Perhaps most important of all: if we wish to change or correct the program, we can do so without needing to type everything in again; the "source" coding will be saved on a file and may be recalled and corrected as desired.

In whatever fashion we write our program outline, we'll still need to change it into machine language. We may use an assembler – symbolic or nonsymbolic – or we might do the job by hand. Program 3 shows the output from a typical assembly. It's full of information, but the only data that really count are the two-digit hexadecimal numbers found to the left of the printout. (The four-digit hex numbers at the extreme left are addresses, to help you know where the code is located.)

An assembly listing is a rich source of information, especially if it's well commented. But the business end – the two-digit hex numbers – is all that is needed to do the job. Those numbers are all that we need to put into the computer. Program 4 shows a hexadecimal dump of memory with the program in place. All the pretty trimmings from the assembly listing are gone. All that we have are the instructions, ready to go to work. That's probably the way we would type them in, using the screen-editing feature of the machine language monitor to change memory until it looked like Program 4.

But our game isn't completed yet. We need to generate the mystery numbers from BASIC, and tie all the pieces together. Next time....

### Program 1:
**Code As Prepared For A Full Assembler**

```
NGUESS =$0240      ;number of guesses
EXACT  =$0241      ;exact count
MATCH  =$0242      ;other match count
INCHAR =$0243      ;input character count
SECRET =$0244      ;secret code
SCOPY  =$0248      ;copy of secret code
UGUESS =$024C      ;user's guess
       *= $033C    ;start program here
       ; start game
START  LDA #$00         ; guesses to zero
       STA NGUESS
       ; accept next guess
GUESS  INC NGUESS       ; count the guess
       LDA NGUESS       ; look at it
       CMP #10          ; over nine?
       BEQ QUIT         ; yup, quit
       JSR PLAY         ; take guess
       BNE GUESS        ; not finished?  back
```

```
QUIT    RTS              ; end of game
        ; get guess & play
PLAY    ORA #$30         ;ascii numeric
        JSR $FFD2        ; .. print
        LDA #$20         ;ascii space
        JSR $FFD2        ; print it
        ; set counts to zero
        LDX #0
        STX EXACT
        STX MATCH
        STX INCHAR
        ; get 4 character guess
INLOOP  JSR $FFE4        ; get char
        CMP #$41         ; less than A
        BCC INLOOP
        CMP #$47         ; over F
        BCS INLOOP       ; reject it
        JSR $FFD2        ; OK, print it
        LDX INCHAR       ; get position
        INC INCHAR       ; bump position
        STA UGUESS,X     ; store character
        LDA SECRET,X     ; copy secret char
        STA SCOPY,X      ; .. to copy area
        CPX #3           ; four chars?
        BNE INLOOP       ; nope, go back
        ; check guess for exact matches
COMPAR  LDA SCOPY,X      ; test character
        CMP UGUESS,X     ; against guess
        BNE SKIP         ; nope, try next
        INC EXACT        ; yes, count it
        LDA #0           ; and wipe out..
        STA SCOPY,X      ; .. matching ..
        STA UGUESS,X     ; .. characters
SKIP    DEX
        BPL COMPAR
        ; check for out-of-place matches
        LDY #$00         ; first secret char
RETRY   LDX #$00         ; first guessed char
CHECK   LDA SCOPY,Y      ; is character wiped?
        BEQ PASS         ; yes, ignore next bit

        CMP UGUESS,X     ; compare it to guess
        BNE PASS         ; nope, move on
        INC MATCH        ; yup, count it
        LDA #$00         ; and wipe out..
        STA SCOPY,Y      ; .. matching ..
        STA UGUESS,X     ; .. characters
PASS    INX              ; next guessed character
        CPX #4           ; tried them all?
        BCC CHECK        ; no, try next one
        INY              ; next secret character
        CPY #4           ; tried them all?
        BCC RETRY        ; no, keep going
        ; print results
        LDX #0           ; start at 'exact'
PLOOP   LDA #$20         ; print a space
        JSR $FFD2
        LDA EXACT,X      ; get the number
        ORA #$30         ; to ascii numeric
        JSR $FFD2        ; and print
        INX              ; move to 'match'
        CPX #$02         ; too far?
        BCC PLOOP        ; nope, keep printing
        LDA #$0D         ; print 'return'
        JSR $FFD2
        LDA EXACT        ; four exact?
        CMP #4           ; if so, set z flag
        RTS
```

## Program 2: Code As Prepared For A Tiny Assembler

```
(033C)  LDA #$00
        STA $0240
(0341)  INC $0240
        LDA $0240
        CMP #10
        BEQ $0350
        JSR $0351
        BNE $0341
(0350)  RTS
(0351)  ORA #$30
        JSR $FFD2
        LDA #$20
        JSR $FFD2
        LDX #$00
        STX $0241
        STX $0242
        STX $0243
(0366)  JSR $FFE4
        CMP #$41
        BCC $0366
        CMP #$47
        BCS $0366
        JSR $FFD2
        LDX $0243
```

## Program 3: Code As Assembled By A Full Assembler

```
                       NGUESS =$0240     ;number of guesses
                       EXACT  =$0241     ;exact count
                       MATCH  =$0242     ;other match count
                       INCHAR =$0243     ;input char count
                       SECRET =$0244     ;secret code
                       SCOPY  =$0248     ;copy secret code
                       UGUESS =$024C     ;user's guess
                              *= $033C   ;start program here
                                            ; start game
033C A9 00      START   LDA #$00            ; guesses, zero
033E 8D 40 02           STA NGUESS
                                            ; accept next guess
0341 EE 40 02   GUESS   INC NGUESS          ; count guess
0344 AD 40 02           LDA NGUESS          ; get it
0347 C9 0A              CMP #10             ; over nine?
0349 F0 05              BEQ QUIT            ; yup, quit
034B 20 51 03           JSR PLAY            ; take guess
034E D0 F1              BNE GUESS           ; not finished?
0350 60         QUIT    RTS                 ; end of game
                                            ; get guess & play
0351 09 30      PLAY    ORA #$30            ;ascii numeric
0353 20 D2 FF           JSR $FFD2           ; .. print
0356 A9 20              LDA #$20            ;ascii space
0358 20 D2 FF           JSR $FFD2           ; print it
                                            ; set counts to zero
035B A2 00              LDX #0
035D 8E 41 02           STX EXACT
0360 8E 42 02           STX MATCH
0363 8E 43 02           STX INCHAR         ; get 4 character
                                             guess
```

SAVE

commodore
VIC 64
$229.00

**VIC 20 SPECIAL $89.00 INCLUDES GORTEK & MICROCHIPS VL110 Cassette program (retail value $19.95).**

| | |
|---|---|
| Commodore VIC20/64 1541 Single Disk Drive | $239.00 |
| Commodore VIC20/64 1530 Datassette | 65.00 |
| Commodore VIC20/64 1525 Printer | 229.00 |
| Commodore VIC20/64 1701 Color Monitor | 249.00 |
| Commodore VIC20/64 1600 Vicmoden (Telephone Interface) | 55.00 |

## Everything you need to support your COMMODORE VIC20 AND VIC64

Unbelievable prices for game and business software for your VIC20 and VIC64.

.... *TOO MANY TO LIST BUT WE GOT IT*.....

*Micro Software International Inc*

**PractiCalc 20**
**For the Commodore VIC-20—**
**16K RAM required**

| | |
|---|---|
| Tape (CV2T10) | $28.00 |
| Disk (CV2D10) | 33.00 |

**PractiCalc 64 For the Commodore 64**

| | |
|---|---|
| Tape (C60T11) | $37.00 |
| Disk (C60D11) | 39.00 |

**Cardco, Creative Software, EPYX, HES, UMI, WICO, Romox.**

## Texas Instruments TI-99/4A Home Computer

*ONLY* **$109.00***

*After TI manufacturer's rebate of $50.00.

SAVE

## *MORE GOOD DEALS*

| | |
|---|---|
| PHA 4100 Color Monitor | $349 95 |
| PHP 2400 TI Impact Printer | 540.95 |
| PHP 1100 Wired Remote Controllers | 27 95 |
| PHP 2700 Program Recorder | 49 95 |

**FREE EXPANSION BOX** ($183. VALUE) with the purchase of any three of the following low price items:

| | |
|---|---|
| PHP 1220 RS 232 Card | $135.95 |
| PHP 1240 Disk Controller Card | 189.95 |
| PHP 1250 Expansion System Disk Drive | 300.95 |
| PHP 1260 Memory Expansion Card 32K | 227.95 |
| PHP 1270 P-Code Card | 189.95 |
| PHM 3111 TI Writer | 75.95 |
| PHM 3113 Microsoft Multiplan | 75.95 |

**FREE SPEECH SYNTHESIZER** when you buy six solid state software, command cartridges.

***QUANTITIES ARE LIMITED!***

**We carry in stock all hardware and software for TI home computers.**

| | |
|---|---|
| PHM 3026 Extended Basic | $ 75.95 |
| PHM 3035 Terminal Emulator II. | 38.95 |
| PHM 3058 Mini-Memory | 75.95 |
| PHM 3055 Editor/Assembler | 37.95 |
| PHM 3109 TI Logo II. | 74.95 |
| PHM 3112 Parsec | 30.95 |
| PHM 3053 TI Invaders | 44.95 |
| PHM 3041-T Adventure (Pirate Adv.) | 30.95 |
| PHM 3057 Munch Man | 30.95 |
| PHM 3006 Home Financial Decisions | 23.95 |
| PHM 3044 Personal Report Generator | 30.95 |
| PHM 3002 Early Learning Fun | 22.95 |
| PHM 3090-97 Milliken Home Math Series K-8th grade | 30.95 |
| PHM 3059-62 Scholastic Spelling Level 3-6 | 41.95 |
| PHM 3046-48 Scott. Foresman. Reading Pkgs. | 41.95 |
| PHM 3064 Touch Typing Tutor | 30 95 |
| PHM 7008 Speaking Scholastic Spelling | 165 95 |
| PHM 3122 Plato Interpreter | 37.95 |
| PHM 3083-88 Addison Wesley Computer Math Games I-VI | 30.95 |

SAVE

SOUTHERN AUDIO VIDEO ELECTRONICS, INC.
1782 Marietta Blvd., N.W., Atlanta, Georgia 30318

**Order Toll Free 1-800-241-2682**
**In Georgia (404)-351-8459**

Get the best prices on TI Hardware and software. For a complete listing of all SAVE's products, send $5.00 for our catalogue (refundable with your first order).

Enjoy the convenience of in-home shopping. Call our toll free number today for orders only.

Use your American Express, VISA, Mastercard, check or money order. Minimum order of $50. Outside continental U.S.A. minimum order $250 U.S. currency. Shipping and handling charges are extra. All prices are subject to change without notice. Allow 2-4 weeks for delivery. Prices good through November, 1983. *$50.00 Rebate expires January, 1984.

```
        INC $Ø243
        STA $Ø24C,X
        LDA $Ø244,X
        STA $Ø248,X
        CPX #$Ø3
        BNE $Ø366
(Ø387)  LDA $Ø248,X
        CMP $Ø24C,X
        BNE $Ø394
        INC $Ø241
        LDA #$ØØ
        STA $Ø248,X
        STA $Ø24C,X
(Ø39A)  DEX
        BPL $Ø381
        LDY #$ØØ
(Ø39F)  LDX #$ØØ
(Ø3A1)  LDA $Ø248,Y
        BEQ $Ø3BØ
        CMP $Ø24C,X
        BNE $Ø3BØ
        INC $Ø242
        LDA #$ØØ
        STA $Ø248,Y
        STA $Ø24C,X
(Ø3B6)  INX
        CPX #$Ø4
        BCC $Ø39B
        INY
        CPY #$Ø4
        BCC $Ø399
        LDX #$ØØ
(Ø3C2)  LDA #$2Ø
        JSR $FFD2
        LDA $Ø241,X
        ORA #$3Ø
        JSR $FFD2
        INX
        CPX #$Ø2
        BCC $Ø3BC
        LDA #$ØD
        JSR $FFD2
        LDA $Ø241
        CMP #$Ø4
        RTS
```



```
Ø366 2Ø E4 FF INLOOP JSR $FFE4       ; get char
Ø369 C9 41           CMP #$41        ; less than A
Ø36B 9Ø F9           BCC INLOOP
Ø36D C9 47           CMP #$47        ; over F
Ø36F BØ F5           BCS INLOOP      ; reject it
Ø371 2Ø D2 FF        JSR $FFD2       ; OK, print it
Ø374 AE 43 Ø2        LDX INCHAR      ; get position
Ø377 EE 43 Ø2        INC INCHAR      ; bump position
Ø37A 9D 4C Ø2        STA UGUESS,X    ; store
                                       character
Ø37D BD 44 Ø2        LDA SECRET,X    ; copy secret
                                       ch
Ø38Ø 9D 48 Ø2        STA SCOPY,X     ; .. to copy
area
Ø383 EØ Ø3           CPX #3          ; four chars?
Ø385 DØ DF           BNE INLOOP      ; nope, go back
                                     ; check guess for
                                       exact matches
Ø387 BD 48 Ø2 COMPAR LDA SCOPY,X     ; test
                                       character
Ø38A DD 4C Ø2        CMP UGUESS,X    ; against guess
Ø38D DØ ØB           BNE SKIP        ; nope, try
                                       next
Ø38F EE 41 Ø2        INC EXACT       ; yes, count it
Ø392 A9 ØØ           LDA #Ø          ; and wipe
                                       out..
Ø394 9D 48 Ø2        STA SCOPY,X     ; .. matching
                                       ..
Ø397 9D 4C Ø2        STA UGUESS,X    ; .. characters
Ø39A CA       SKIP   DEX
Ø39B 1Ø EA           BPL COMPAR
                                     ; check for matches
Ø39D AØ ØØ           LDY #$ØØ        ; first secret
Ø39F A2 ØØ    RETRY  LDX #$ØØ        ; first guessed
Ø3A1 B9 48 Ø2 CHECK  LDA SCOPY,Y     ; char wiped?
Ø3A4 FØ 1Ø           BEQ PASS        ; yes, ignore
Ø3A6 DD 4C Ø2        CMP UGUESS,X    ; compare
Ø3A9 DØ ØB           BNE PASS        ; nope, move on
Ø3AB EE 42 Ø2        INC MATCH       ; yup, count it
Ø3AE A9 ØØ           LDA #$ØØ        ; and wipe
                                       out..
Ø3BØ 99 48 Ø2        STA SCOPY,Y     ; .. matching
                                       ..
Ø3B3 9D 4C Ø2        STA UGUESS,X    ; .. characters
Ø3B6 E8       PASS   INX             ; next guess
Ø3B7 EØ Ø4           CPX #4          ; tried all?
Ø3B9 9Ø E6           BCC CHECK       ; no, try next
Ø3BB C8              INY             ; next char
Ø3BC CØ Ø4           CPY #4          ; tried all?
Ø3BE 9Ø DF           BCC RETRY       ; no, keep on
                                     ; print results
Ø3CØ A2 ØØ           LDX #Ø          ; first numbr
Ø3C2 A9 2Ø    PLOOP  LDA #$2Ø        ; print space
Ø3C4 2Ø D2 FF        JSR $FFD2
Ø3C7 BD 41 Ø2        LDA EXACT,X     ; get number
Ø3CA Ø9 3Ø           ORA #$3Ø        ; to ascii num
Ø3CC 2Ø D2 FF        JSR $FFD2       ; and print
Ø3CF E8              INX             ; move on
Ø3DØ EØ Ø2           CPX #$Ø2        ; too far?
Ø3D2 9Ø EE           BCC PLOOP       ; nope, loop
Ø3D4 A9 ØD           LDA #$ØD        ; print return
Ø3D6 2Ø D2 FF        JSR $FFD2
Ø3D9 AD 41 Ø2        LDA EXACT       ; four exact?
Ø3DB C9 Ø4           CMP #4          ;   z flag
Ø3DD 6Ø              RTS
```

## Program 4: Hexadecimal Dump Of Memory

```
C*
      PC   IRQ  SR AC XR YR SP
.;   B780 E455 2C 34 3A 9D F8
.
.:   033C A9 00 8D 40 02 EE 40 02
.:   0344 AD 40 02 C9 0A F0 05 20
.:   034C 51 03 D0 F1 60 09 30 20
.:   0354 D2 FF A9 20 20 D2 FF A2
.:   035C 00 8E 41 02 8E 42 02 8E
.:   0364 43 02 20 E4 FF C9 41 90
.:   036C F9 C9 47 B0 F5 20 D2 FF
.:   0374 AE 43 02 EE 43 02 9D 4C
.:   037C 02 BD 44 02 9D 48 02 E0
.:   0384 03 D0 DF BD 48 02 DD 4C
.:   038C 02 D0 0B EE 41 02 A9 00
.:   0394 9D 48 02 9D 4C 02 CA 10
.:   039C EA A0 00 A2 00 B9 48 02
.:   03A4 F0 10 DD 4C 02 D0 0B EE
.:   03AC 42 02 A9 00 99 48 02 9D
.:   03B4 4C 02 E8 E0 04 90 E6 C8
.:   03BC C0 04 90 DF A2 00 A9 20
.:   03C4 20 D2 FF BD 41 02 09 30
.:   03CC 20 D2 FF E8 E0 02 90 EE
.:   03D4 A9 0D 20 D2 FF AD 41 02
.:   03DC C9 04 60
.
```

©

# ISAM

# Building Your Own Random File Manager

Michael D Lipay

*There are several approaches to handling computer files (collections of data). Among the fastest and best is the random access disk file which uses special techniques to quickly locate any piece of information from anywhere within the entire file.*

*This tutorial explains how random access can be achieved and examines alternative ways to process data files. It includes a sample program, written in Applesoft BASIC, but which can easily be adapted to work on other computers using Microsoft BASIC.*

Besides protecting earth from aliens, a main purpose of a computer is processing information. This data processing can be anything from keeping track of your stamp collection to maintaining a running inventory for your business. When it becomes necessary to retain the information long after the computer has been turned off, tape or disk storage is used.

Magnetic storage devices are capable of storing information indefinitely (provided they are kept clean and away from magnetic fields). Basically, there are two types of magnetic storage devices available to the micro computer user – tape and disk. Both devices are capable of storing large amounts of information, and do so in groups called files. A file is a collection of related information, and the user has three primary types of files to select from:

I) Sequential Tape Files
II) Sequential Disk Files
III) Random Access Disk Files

Which of the three you decide to use for a given program will depend on many factors. Each has its own advantages and disadvantages; they are discussed here in an effort to help you select the best one for your needs.

## Sequential Tape Files

If you have large amounts of data which you do not need to process frequently, then tape files should be considered. Tapes can store vast amounts of data in a relatively compact space, and at a very low price. Tapes serve as an excellent medium to keep a backup of disk programs and files. The big drawback to using tapes is that they are slow, so make sure you have plenty of time.

## Sequential Disk Files

Sequential disk files are best if you have small amounts of data to process. The files have the advantages of being faster than tape and more space conservative than random access files. Probably the only disadvantage of sequential disk files is the slowness of updating large files. In order to change a single record on a sequential file, you must copy all records to a work file, changing any records desired along the way, then delete the old file and rename the work file. This could be as time consuming as tape files, were it not for the speed of the disk.

## Random Access Disk Files

Large volumes of data which must be updated with any frequency should be held in random access files. This type of file lets you easily update any given record without having to process or read through any other record on the file. It also has disadvantages such as requiring all records to be of the same, fixed length and needing to know where on the file a particular record is located.

There are several methods available to help

# 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 Fig-FORTH derivative and readily accessible 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.

## 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 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.*

## 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

you to determine where a particular record is located on a random access file. John Hudson covered the HASH/LINK method in the March 1982 issue of COMPUTE!. He did an excellent job; and if you desire to learn more about it, I suggest that you read this article. The HASH/LINK method does have some problems. For example:

I) If you fill the overflow area, you will have to reorganize the file again.

II) As soon as you initialize the random file, you take up more space than you may need.

III) Successive "collisions" can greatly increase access time (rec 100 links to rec 212, rec 212 links to rec 487, rec 487 links to...).

IV) Expanding the main and overflow areas of the file may require major program revisions (deciding the main area should be 2000 recs instead of 1000 recs will require changes to your hashing logic), as well as requiring you to reload the file.

V) Sequential (ascending or descending) processing is almost impossible.

VI) If you need to "key" on an alphabetic field (such as a name), you must first convert it to a numeric value.

VII) Once the file has been created, it is impossible to select an alternate key (e.g., a file is hashed on the last name, but you need a report in social security number order).

VIII) Deleting a record requires several Read/Write steps to keep the link field updated. Once a record has been deleted, the position that it occupied on the file is unusable, since all adds occur at the end of the file.

In the rest of this article I will cover an alternate method known as Indexed-Sequential Access Method (ISAM).

## ISAM

ISAM can solve all the problems associated with HASH/LINK files, but it has some problems of its own. ISAM works on the principle that it is faster to search memory than a disk. Unfortunately, before you can search memory, you must have something in it, and this is the problem with ISAM.

ISAM works by loading the desired "key" field of each record in a file into an array. This is done by placing the key field of the first record into the first position of the array, the key from the second record into the second position of the array, etc. Once the array has been loaded, you simply search the array for the desired key; its position in the array is the record number for the random access file. Described below are the procedures necessary for the most common types of file processing:

I) ADD A RECORD
- a) Search the array to determine if the record already exists.
- b) Move the new "key" to the end of the array, or to the first "open" position in the array.
- c) Use the position number of the array to write the record to the file.

II) DELETE A RECORD
- a) Find the key in the array.
- b) "Open" the entry in the array by moving a "dummy" key into the array (such as zeros).
- c) Write the dummy values to the file.

III) CHANGE A RECORD
- a) Find the key in the array.
- b) Use the position number to read in the record.
- c) Make your change to the record (even change the key).
- d) Write the new record to the file using the position number.
- e) If you changed the key, move the new key into the array.

IV) PROCESS SEQUENTIALLY BY KEY
- a) Sort the array into the desired order (ascending or descending).
- b) Process the records sequentially through the array.

V) PROCESS BY A DIFFERENT KEY
- a) Load the array with the new keys from the file.
- b) Process normally using the new array.

Listed below are sample programs, written in Applesoft, which illustrate ISAM programming techniques. The programs are shells which can easily be modified to suit your own purposes. Note that all branch instructions bypass the REM statements; thus, if you want to key the program in without remarks, no line numbers will have to be changed. Variables used in the programs are:

**D$** – Control-D (disk access)

**IA** – Index Array

**IE** – Index End (last entry used)

**IP** – Index Pointer (entry number for the part searched for)

**IO** – Index Open (entry number for first "open" or empty record)

**FOUND** – Switch to indicate if part searched for is in the index:

0 – part not in index
1 – part in index

**PART** – Part number being searched for

**10-13** This section goes to a one-time routine to load the index array with the desired key field (in this case a part number).

**100-114** Display the options available in a menu format.

**120-122** This gets the option into a string. Then, using the VAL command, goes to the appropriate routine. Note that if zero, a non-numeric character, or a number greater than five is entered, the menu is displayed again.

**200-215** The index array is searched sequentially in this section. If the key is found, the following values are returned:

FOUND =1
IP =Entry in array for desired key
IO =First open entry in array (entry with key of zero)

If the key is not found, the following values are returned:

FOUND =0
IO =First open entry in array

Note on lines 212 and 213 the method used to exit from the FOR/NEXT loop. This is the method suggested by Apple to exit the loop from other than normal completion. Its purpose is to prevent ?OUT OF MEMORY errors from occurring as a result of too many "open" loops.

**300-324** ADD A PART

**310** Accepts the part number to be added to the file.

**311** Goes to the routine to search the index. If the part already exists (FOUND=1), an error message is displayed and control is returned to the menu.

**321-322** The new part is written to the master file using the open entry pointer (IO) as the record number.

**323** If the new part is added to the end of the file, the number of the last entry (IE) is updated.

**324** Returns to the menu.

**400-424** DELETE A PART

**410** Accepts the part to be deleted.

**411** Goes to the search routine. If the part is not on file (FOUND=0), an error message is displayed and control is transferred to the menu.

**420** The part is removed from the index by making the entry zero.

**421-422** The part is removed from the master file.

**423** If the part was the last one in the array, the ending pointer (IE) is reduced by one.

**424** Return to the menu.

**800-813** UPDATE INDEX POINTER

**810-811** Write the number of the last entry in the index to record zero of the master file.

**812** Closes the master file.

**813** Stops the program.

**900-930** LOAD THE INDEX ARRAY

**910** Initially sets up variables.

**911** Sets up an error routine to handle end-of-data and not-found conditions.

**912** Opens the master file.

**913-914** Read the number of the last record on the master file.

**915** Turns off the error routine, dimensions the index array to allow up to ten records to be added to the end of the array (this can be changed to allow for more expansion).

**916** If no records exist on the master, control goes to the menu.

**920** Sets up the error routine.

**921-924** Load the key field (part number) into the array.

**930** Turns the error routine off; returns to the menu.

The second program offers a different method of handling the index. Type in lines 10-630 from Program 1, then add the lines from Program 2. In this program the index is kept on a sequential disk file, for speed of loading the array.

**800-833** Save the index array.

**810** Check the index change switch; if it is zero, the index has not changed and does not have to be rewritten. Control goes to 832.

**811** Deletes the index file.

**820-823** Write the array to the index file.

**830-831** Write the number of the last entry in the index to record zero of the master file.

**832** Closes the master file.

**833** Stops the program.

**900-940** LOAD THE INDEX ARRAY

**910** Initially sets up variables.

**920** Opens the master file.

**921** Sets up the error routine.

**922-923** Read the number of entries in the index file.

**930** Sets up a new error routine.

**931** Dimensions the index array (with expansion of 10).

**932-934** Read the index file into the array.

**935** Turns the error routine off and closes the index.

**940** Turns control over to the menu.

## Program 1: ISAM

```
10  REM
11  REM  CALL INDEX LOAD ROUTINE
12  REM
13  GOTO 910
100  REM
101  REM  SELECT OPTION
```

```
102  REM
110  HOME : PRINT "1) ADD PART"
111  PRINT "2) DELETE PART"
112  PRINT "3) CHANGE PART"
113  PRINT "4) DISPLAY PART"
114  PRINT "5) STOP"
120  PRINT : INPUT "SELECT OPTION: ";OPT$
121  ON  VAL (OPT$) + 1 GOTO 110,310,410,510
     ,610,810
122  GOTO 110
200  REM
201  REM  SEARCH INDEX ARRAY
202  REM
210 IO = IE + 1: IF IE = 0 THEN FOUND = 0:
     RETURN
211  FOR I = 1 TO IE
212  IF IA(I) = PART THEN IP = I:I = IE + 1:
      NEXT :FOUND = 1: RETURN
213  IF IA(I) = 0 AND IO = IE + 1 THEN IO =
     I: NEXT
214  NEXT I
215 FOUND = 0: RETURN
300  REM
301  REM  ADD A PART
302  REM
310  INPUT "ENTER NEW PART NUMBER: ";PART
311  GOSUB 210: IF FOUND = 1 THEN  PRINT "PA
     RT ALREADY ON FILE": GOTO 110
320 IA(IO) = PART
321  PRINT D$;"WRITE MASTER,R";IO
322  PRINT PART: PRINT D$
323  IF IO > IE THEN IE = IO
324  GOTO 110
400  REM
401  REM  DELETE A PART
402  REM
410  INPUT "ENTER PART TO BE DELETED: ";PART

411  GOSUB 210: IF FOUND = 0 THEN  PRINT "PA
     RT IS NOT ON FILE": GOTO 110
420 IA(IP) = 0
421  PRINT D$;"WRITE MASTER,R";IP
422  PRINT 0: PRINT D$
423  IF IP = IE THEN IE = IE - 1
424  GOTO 110
500  REM
501  REM  CHANGE A PART
502  REM
510  INPUT "ENTER PART TO BE CHANGED: ";PART

511  GOSUB 210: IF FOUND = 0 THEN  PRINT "PA
     RT IS NOT ON FILE": GOTO 110
520  PRINT D$;"READ MASTER,R";IP
521  INPUT PART: PRINT D$
530  REM  CODING TO CHANGE PART
540 IA(IP) = PART
541  PRINT D$;"WRITE MASTER,R";IP
542  PRINT PART: PRINT D$
543  GOTO 110
600  REM
601  REM  DISPLAY PART
602  REM
610  INPUT "ENTER PART NUMBER: ";PART
611  GOSUB 210: IF FOUND = 0 THEN  PRINT "PA
     RT IS NOT ON FILE": GOTO 110
612  PRINT D$;"READ MASTER,R";IP
613  INPUT PART: PRINT D$
620  REM  CODING TO DISPLAY PART
630  GOTO 110
800  REM
801  REM  UPDATE INDEX POINTER
802  REM
810  PRINT D$;"WRITE MASTER,R0"
811  PRINT IE
812  PRINT D$;"CLOSE MASTER"
813  END
900  REM
901  REM  LOAD INDEX ARRAY
902  REM
910 D$ =  CHR$ (4):IE = 0:IP = 0:IO = 0:FOUN
     D = 0:PART = 0
911  ONERR  GOTO 915
912  PRINT D$;"OPEN MASTER,L25"
913  PRINT D$;"READ MASTER,R0"
914  INPUT IE: PRINT D$
915  POKE 216,0: DIM IA(IE + 10)
916  IF IE = 0 GOTO 110
920  ONERR  GOTO 924
921  FOR I = 1 TO IE
922  PRINT D$;"READ MASTER,R";I
923  INPUT IA(I)
924  NEXT I: PRINT D$
930  POKE 216,0: GOTO 110
```

## Program 2: Index Array Routine

```
800  REM
801  REM  SAVE INDEX
802  REM
810  IF IC = 0 GOTO 832
811  PRINT D$;"DELETE INDEX"
820  PRINT D$;"OPEN INDEX"
821  PRINT D$;"WRITE INDEX"
822  FOR I = 1 TO IE: PRINT IA(I): NEXT I
823  PRINT D$;"CLOSE INDEX"
830  PRINT D$;"WRITE MASTER,R0"
831  PRINT IE
832  PRINT D$;"CLOSE MASTER"
833  END
900  REM
901  REM  LOAD INDEX ARRAY
902  REM
910 D$ =  CHR$ (4):IE = 0:IP = 0:IC = 0:IO =
     0:FOUND = 0:PART = 0
920  PRINT D$;"OPEN MASTER,L25"
921  ONERR  GOTO 930
922  PRINT D$;"READ MASTER,R0"
923  INPUT IE
930  ONERR  GOTO 935
931  DIM IA(IE + 10)
932  PRINT D$;"OPEN INDEX"
933  PRINT D$;"READ INDEX"
934  FOR I = 1 TO IE: INPUT IA(I): NEXT I
935  POKE 216,0: PRINT D$;"CLOSE INDEX"
940  GOTO 110
```

©

# TI Cadette: Computer Aided Design

Bradley Rogers

*This clever program should provide hours of amusement for children who enjoy creating pictures. Similar to coloring or cut-and-paste, the computer screen becomes a magic window allowing easy design, color selection, and erasure. Requires Extended BASIC and joysticks.*

"Cadette" is for children. Based on a scaled-down version of CAD (the Computer Aided Design), it transforms your TV screen into an electronic easel on which children can "draw" tropical birds, planes, surreal landscapes, or any number of other fascinating pictures. Joysticks and fire buttons are used instead of conventional pens and brushes.

Using these simple instruments, children can create intricate designs from a basic stockpile of 16 different shapes. Each shape can assume five different colors chosen at the start of the program. Cadette calls upon the imagination, but does not require highly developed motor skills. Most children over five should be able to manage it nicely.

Cadette is simple to use, with only four basic activities required:

1. Choosing a page (screen) color;
2. Choosing five brush (shape) colors;
3. Moving joysticks to position the shapes or the eraser; and
4. Pressing fire buttons to print or to erase.

The process is the electronic equivalent to pasting cutouts on construction paper. However, the program involves considerably less frustration than conventional craft activities. It permits children to erase neatly or to change their minds at any point without having to start over with a clean sheet.

## Running The Program

Once the RUN command has been entered, a brief message appears, instructing you to select a page color. The page in this case is, of course, the TV screen. Next, you are confronted by a display of 12 colors, each identified by a number from 1 to 12. From this menu you select a screen color by pressing the appropriate number key and then the ENTER key. If you enter anything other than numbers 1 to 12, the computer waits patiently for you to reconsider.

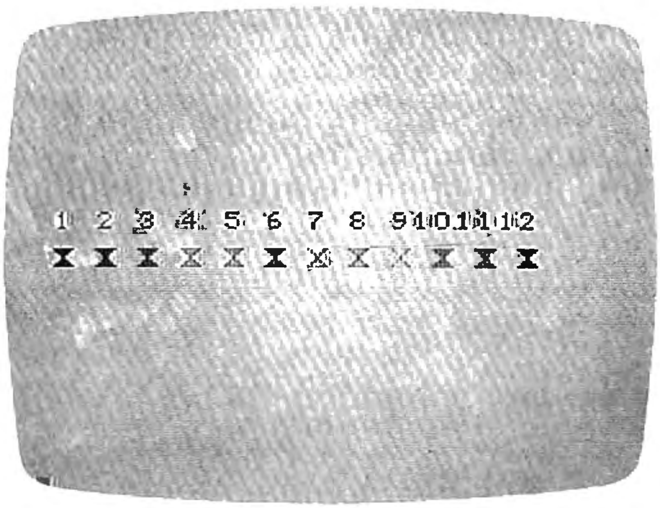
A second message now appears on the background color you chose. You are to select five brush colors. This message disappears, and you are asked to choose five from among twelve brush colors. The brushes in this case represent the colors of the shapes you will eventually use to create your design. Simply enter your five choices and remember to press ENTER after each selection.

After the color choices, the screen will blank and 16 geometric shapes will appear, eight across the top of the screen, and eight across the bottom. They consist of a circle, a square, assorted lines, triangles, and semicircles. Every few seconds the color of all 16 shapes changes, running through a cycle of five color changes, and then repeating.

Near the center of the screen is a small hollow box, which is the cursor. By using either of the joysticks, you move the cursor to capture and transport the colored shapes. After deciding which shape you want to capture, move the cursor to a position immediately adjacent to the shape. Once the shape turns the desired color, position the cursor on the shape.

The cursor will then disappear, and a duplicate of the colored shape you chose will appear immediately above or below the original, depending upon whether you selected from the top or bottom row. This duplicate may now be moved with the joysticks to any desired location. It will maintain its shape and color no matter what else happens on the screen. The original from which it was copied will remain in its display row and continue to undergo color transformations.

The duplicate shape, which now represents the cursor, can be placed at any position on the screen. Move it to the location you want and simply press the fire button. You will hear a low tone indicating that the button has done its job. If you have picked up the right joystick, the shape will "lock" at that screen location. Even if you move the cursor, the shape will remain fixed as



*Twelve page and twelve brush colors are available in "Cadette" from the TI-99.*

long as the program runs. If you have picked up the wrong joystick, the shape will be erased.

Assuming you have the "lock" joystick, you now have two options. You can move the cursor shape to a new position and print it again, or you can select another shape of the same color or the same shape of another color. If you choose a new shape, repeat the initial capture procedure. Remember, however, that the cursor no longer appears as a hollow box, but in the shape of your previous selection. But once it is placed on a new colored shape, it will automatically assume the new shape and color.

The "lock" joystick locks your selection at the location you want. The other joystick also controls the cursor, but is used to erase. To erase a "locked" shape, simply move the cursor on top of that shape and press the fire button. A higher tone will sound, the shape will disappear, and you can make another selection. To avoid confusion, you might label one joystick "lock" or "print" and the other "erase."

## Extensions And Modifications

If you want to alter the shapes, you can change lines 540, 560, 580, and 600, which are DATA statements that contain the hexadecimal representations of the shapes. Each shape is defined by a string of 16 hexadecimal numbers.

Some children may find that the cursor moves too quickly, rushing past the space in which they wanted to print a shape. You can change the cursor's speed in line 920 by adjusting the limit (4) in the FOR/NEXT loop.

One interesting modification to the program would make it more versatile without requiring a great deal of extra programming. For example, a larger menu of shapes could be shown initially, and 16 could then be chosen from it. This would not be a terribly complicated program adjustment as long as you remember that the shapes must be read into S$. It is better to present the shape menu before the color menus; once you start fooling with color statements, all kinds of unexpected complications develop. In considering such modifications, just remember that often there is a tradeoff between versatility and user-convenience. The program could become less fun to use if a child has to make too many decisions.

### Cadette

```
100 CALL CLEAR
110 CALL SCREEN(15)
120 DISPLAY AT(5,6):"SELECT PAGE CO
    LOR, 1 TO 12."
130 FOR I=1 TO 800 :: NEXT I :: CAL
    L CLEAR
140 DIM Z(5):: DIM S$(16)
150 A$="3C7EFFFFFFFF7E3C" :: X=4
160 FOR I=62 TO 142 STEP 8 :: CALL
    CHAR(I,A$):: CALL COLOR(X,X-1,1
    ):: X=X+1 :: NEXT I
170 CALL CHAR(40,A$):: CALL COLOR(2
    ,16,1):: CALL COLOR(9,14,1):: C
    ALL VCHAR(12,5,40):: X=3
180 FOR I=62 TO 142 STEP 8 :: CALL
    VCHAR(12,X*2+1,I):: X=X+1 :: NE
    XT I
190 CALL CHARPAT(56,Z$):: CALL CHAR
    PAT(57,W$):: CALL CHAR(33,Z$)::
     CALL CHAR(34,W$)
200 FOR I=1 TO 12
210 IF I=8 THEN CALL VCHAR(10,19,33
    ):: GOTO 240
220 IF I=9 THEN CALL VCHAR(10,21,34
    ):: GO TO 240
230 DISPLAY AT(10,I*2):USING "##":I
240 NEXT I
250 ACCEPT AT(24,1)VALIDATE(DIGIT)B
    EEP:Y
260 IF Y<1 OR Y>12 THEN 250
270 IF Y=7 THEN Y=13
280 IF Y=1 THEN Y=15
290 CALL SCREEN(Y+1):: CALL CLEAR
300 FOR I=1 TO 14 :: CALL COLOR(I,2
    ,1):: NEXT I :: DISPLAY AT(6,4)
    :"SELECT 5 BRUSH COLORS,{6 SPACE
    S}1 TO 12."
310 FOR I=1 TO 800 :: NEXT I :: CAL
    L CLEAR
320 FOR I=4 TO 14 :: CALL COLOR(I,I
    -1,1):: NEXT I :: CALL COLOR(2,
    16,1):: CALL COLOR(9,14,1)
330 A$="FF7E3C18183C7EFF"
340 FOR I=62 TO 142 STEP 8 :: CALL
    CHAR(I,A$):: NEXT I :: CALL CHA
    R(40,A$)
350 IF Y=15 THEN Y=0
360 IF Y=13 THEN Y=7
370 CALL COLOR(Y+2,2,1):: X=6 :: CA
    LL VCHAR(12,4,40)
380 FOR I=62 TO 142 STEP 8 :: CALL
    VCHAR(12,X,I):: X=X+2 :: NEXT I
390 FOR I=1 TO 12
400 IF I=8 THEN CALL VCHAR(10,18,33
    ):: GOTO 430
410 IF I=9 THEN CALL VCHAR(10,20,34
    ):: GOTO 430
420 DISPLAY AT(10,I*2-1):USING "##"
    :I
```

```
430 NEXT I
440 FOR I=1 TO 5
450 ACCEPT AT(24,1)VALIDATE(DIGIT)B
    EEP:Z(I)
460 IF Z(I)<1 OR Z(I)>12 THEN 450
470 IF Z(I)=1 THEN CALL VCHAR(18,I*
    2+2,40)ELSE 490
480 GO TO 500
490 CALL VCHAR(18,I*2+2,46+8*Z(I))
500 NEXT I
510 FOR I=1 TO 500 :: AA=8*8 :: NEX
    T I
520 CALL CLEAR
530 FOR I=1 TO 16 :: READ S$(I):: N
    EXT I
540 DATA 187E7EFFFF7E7E18,0107070F0
    F070701,80E0E0F0F0E0E080,FF7E7E
    1800000000
560 DATA 00000000187E7EFF,FFFFFFFFF
    FFFFFFF,FFFEFCF8F0E0C080,000103
    070F1F3F7F
580 DATA 80C0E0F0F8FCFEFF,7F3F1F0F0
    7030100,8080808080808080,010101
    0101010101
600 DATA FF00000000000000,000000000
    00000FF,8040201008040201,010204
    0810204080
610 X=40
620 FOR I=1 TO 5
630 P=Z(I)+1
640 IF Z(I)=1 AND Y=0 THEN CALL COL
    OR(I*2,2,1):: CALL COLOR(I*2+1,
    2,1):: GO TO 690
650 IF Z(I)>1 AND Z(I)=Y THEN CALL
    COLOR(I*2,2,1):: CALL COLOR(I*2
    +1,2,1):: GO TO 690
660 IF Z(I)=1 AND Y>0 THEN P=16
670 IF Z(I)=7 AND Z(I)<>Y THEN P=14
680 CALL COLOR(I*2,P,1):: CALL COLO
    R(I*2+1,P,1)
690 FOR J=1 TO 16 :: CALL CHAR(X,S$
    (J)):: X=X+1 :: NEXT J
700 NEXT I
710 G2=32 :: H1=12 :: F1=16
720 CALL CHAR(37,"FF818181818181FF"
    ):: CALL VCHAR(H1,F1,37):: J=39
730 QW=1 :: UU=1
740 FOR I=1 TO 8 :: CALL VCHAR(24,I
    *2+4,J+I):: NEXT I
750 FOR I=9 TO 16 :: CALL VCHAR(1,I
    *2-12,J+I):: NEXT I
760 X=1
770 CALL JOYST(UU,F2,H2)
780 CALL KEY(UU,RV,SV)
790 IF (H1=1 AND H2=4)OR(H1=24 AND
    H2=-4)OR(F1=2 AND F2=-4)OR(F1=3
    0 AND F2=4)THEN F2=0 :: H2=0 ::
    GO TO 900
800 IF F2=0 AND H2=0 AND SV=0 THEN
    900
810 H3=H1-H2/4 :: F3=F1+F2/4
820 CALL GCHAR(H3,F3,G3):: CALL GCH
    AR(H1,F1,G1)
830 IF RV+QW=19 AND(H1=24 OR H1=1)T
    HEN 890
840 IF H3=24 AND G3<>32 THEN H3=23
    :: CALL VCHAR(H1,F1,32):: CALL
    VCHAR(H3,F3,G3):: G2=32 :: GO
    TO 870
850 IF H3=1 AND(G3<>32)THEN H3=2 ::
     CALL VCHAR(H1,F1,32):: CALL VC
    HAR(H3,F3,G3):: G2=32 :: GO TO
    870
860 IF (H3<>24 AND H3<>1)OR(H3=24 A
    ND G3=32)OR(H3=1 AND G3=32)THEN
     CALL VCHAR(H3,F3,G1):: CALL VC
     HAR(H1,F1,G2):: G2=G3
870 IF RV+QW=19 THEN CALL VCHAR(H3,
    F3,G1):: G2=G3 :: CALL SOUND(10
    0,110,2):: GOTO 890
880 IF RV+QW=17 THEN CALL VCHAR(H3,
    F3,37):: G2=32 :: CALL SOUND(10
    0,220,2)
890 H1=H3 :: F1=F3
900 QW=-QW
910 IF QW=1 THEN UU=1 ELSE UU=2
920 FOR AA=1 TO 4 :: NEXT AA
930 X=X+1 :: IF X=10 THEN 940 ELSE
    770
940 J=J+16
950 IF J>103 THEN J=39
960 GO TO 740
```

# ATARI FONTBYTER

Orson Scott Card

*It's hard to tell, when you're using "Fontbyter," whether this is a utility or a game. You can easily create graphics displays many times the size of the screen and save them to disk, using the ROM character set – or character sets you have designed yourself. And because Fontbyter allows you to use two "hidden" character modes, ANTIC modes 4 and 5, you get all the high-resolution color of Graphics 7 with the convenience and memory usage of Graphics 0.*

Once you have a character set designed and a picture drawn on the screen using "Fontbyter," changing an 8-by-8-pixel character block takes only one POKE. This allows easy, almost instant animation; your programs can be shorter than they would be if you tried to get the same effect with Graphics 7; and you have more memory available to you because the screen displays take up less room.

The problem is creating the actual display. In ANTIC 4, you have 24 lines of 40 characters; in ANTIC 5, 12 lines of 40 characters. Laying out the screen display and writing the DATA statements can be a long, tedious, painful process. You have to remember what each character looks like and make sure that the characters are in the right order in the DATA statements you create. And when you want to change a display, you have to go back and find the right DATA statement and alter it.

Fontbyter lets you create and edit in ANTIC 4 or 5 right on the screen. You don't have to write down the number of the character and POKE it into memory; you only have to press a key or combination of keys, and your character will be displayed exactly where you want it on the screen. Simple commands allow you to fill large areas with a single character, insert or delete lines, scroll around the screen to view large areas quickly, or change the colors on the screen. And Fontbyter will scroll horizontally and vertically, so you can use the screen as a window onto a very large display – up to 4K.

Best of all, you can save your screen to disk at any point and return to continue editing it. Using a simple subroutine, you can then load your screen into memory in your own program. The first eight bytes of every file Fontbyter creates contain the mode number, the display width, the display height, and the five colors of the screen display.

## Starting Fontbyter

*Character set.* When you RUN Fontbyter, the program accesses your disk and shows you a directory of all the files with the filetype ".SET". Fontbyter assumes that these are all character sets. The program then asks you to choose which one you want to use. Or, if you wish to use the built-in ROM character set, enter the character "@" as the filename.

There is only one custom character set included with Fontbyter, but by using a character editor you can create as many different sets as you want.

If the character set you ask for is not on the disk in drive 1, the program will prompt you to either insert the correct disk or ask for a different set. Also, whenever Fontbyter asks you for a filename, you don't need to enter more than the eight-character name – Fontbyter always supplies the device name "D1:" and the extender ".SET" or ".SCR". If you use an illegal name, Fontbyter will ask you to try again.

*Screen files.* When you have chosen your character set, Fontbyter displays a directory of all the files with the filetype ".SCR". Fontbyter assumes that these files contain screen displays created and saved by Fontbyter. If no directory is displayed, it means that there are no files with the filetype ".SCR" on the disk.

At the end of the directory, you will be told the number of sectors left on the disk. Be sure that the disk you use for saving screens has enough room for the screen you intend to save. A maximum-size display is almost 4K, which will create a file of 33 sectors. Disks can fill up pretty fast at that rate.

*Save file.* The program asks you what name your saved screen file should have. When you are through editing and want to save your finished screen, this is the filename that Fontbyter will use to create the save file. You can use a filename that you used before, but saving the new file will erase

the old one.
*Load file.* The program asks you if you want to edit a screen that was previously saved. If you do, you will be asked the name of the file you want to load from.

Notice that this system allows you to load from a file and then save your edited version back to the same file, erasing the old version; or you can choose to save the file under a different filename, so that both versions will exist. There is an added safeguard, too. When you save the screen display, it is first saved under the name "D1.TEMPFILE.SCR". Then Fontbyter asks you if you want to save it under the name you chose at the beginning of the program. If you change your mind about the save filename then, you can exit Fontbyter and use DOS to change "D: TEMPFILE.SCR" to whatever name you want.

*Load file parameters.* If your load file is found, Fontbyter immediately opens it and reads the first three bytes. Then it reminds you of the ANTIC mode, width (in characters), and height (in lines) of the file as it was saved. If you don't want to change those parameters, you can proceed directly to the final check; if you do want to change them, Fontbyter will ask you to choose the mode, width, and height of the file as if you were creating a new screen.

*ANTIC mode.* Fontbyter asks you to choose which ANTIC mode you want. The only choices are 2 (Graphics 0), 4, or 5. Mode 4 has shorter, squarer characters, and fits 24 lines on a screen. Mode 5 has tall, thin characters and fits only 12 lines on a screen. This means that a display file a hundred lines from top to bottom will give you more than eight distinct screen displays in ANTIC 5, but only just over four distinct displays in ANTIC 4. ANTIC 2 (Graphics 0) is included, even though it is not a four-color mode, so that you can use Fontbyter to create displays using the built-in ROM character set.

If you own an XL model (600XL, 800XL, 1200XL, 1400XL, or 1450XLD), ANTIC 4 and 5 correspond to Graphics 12 and 13.

*Display width.* The minimum width of a line is 40 characters. If you enter a number less than 40, Fontbyter will change it to 40. The maximum width depends on the mode. The limiting factor here is that all screen displays must fit within 4K. Because of this, the wider a screen display you choose, the fewer vertical lines you can have. You cannot have a line so wide that it would not allow the minimum number of lines. Since you will not be allowed any fewer than 24 screen lines in ANTIC 2 or 4, you naturally can't have as wide a screen as in mode 5, which has a minimum of 12 lines per screen.

*Display height.* The minimum height, in number of lines, is 12 lines for ANTIC 5 and 24 lines for ANTIC 2 and 4. The maximum height depends on the line width you chose. If you ask for more lines than the allowable maximum, Fontbyter will change the figure to the maximum.

*Final check.* Fontbyter clears the screen and then displays what your choices were: the character set, the file in which to save your screen, the file (if any) to load from, the mode, the width (in characters), and the height (in lines). If you want to make any changes, press OPTION. If you are satisfied with your choices, press START.

Fontbyter will display a wait message for a few moments, and then the screen will go completely blank. This is so that the setup operations will run faster. When Fontbyter is ready to go on – and it won't be long – either the load screen you asked for will appear or a cursor will appear in the upper-left-hand corner of a blank screen. The cursor is whatever the ESCAPE character looks like in the character set you chose.

Also, part of the character set will be displayed on the bottom four lines of the screen. The characters are arranged in the same order as the computer keyboard, so that you can easily figure out which key to press in order to display a particular character.

## Editing Features

*To use the keyboard.* The character set is divided into three groups: regular, shifted, and control. You can change from one to another using the CAPS/LOWR key. To get the regular character group, press CAPS/LOWR. To get the shifted character group, press SHIFT and CAPS/LOWR at the same time. To get the control character group, press CONTROL and CAPS/LOWR at the same time. As soon as you make the change, the character keyboard display at the bottom of the screen will change to show you the characters now available.

Instead of the usual computer keyboard system of locking only the alphabetic keys into shifted and control functions, Fontbyter shifts the entire keyboard. After you press SHIFT and CAPS/LOWR, you can press any key on the keyboard and get the shifted character – without pressing SHIFT again. The same applies to CONTROL with CAPS/LOWR.

Some keys, of course, don't have a shifted or control value (ESC, DEL, and RETURN, for instance), and others usually display only the inverse of another character (SHIFT-TAB, for instance). Since these don't display a separate character, pressing them only produces the same character that you would get if you pressed the space bar – a blank. In addition, if your character set redefines the space bar character, that character will fill your display when it first comes up, and will appear on the screen whenever you enter a

nonprinting character.

The keys do not produce their normal clicking sound, except for the command keys, which are described next.

*Command keys.* No matter which character group you are using, there are some key combinations that Fontbyter interprets as commands. Pressing INSERT and SHIFT together will insert a blank line on the screen. Pressing DELETE and SHIFT together will delete a line. Pressing CONTROL and an ARROW key together will cause the cursor to move.

Remember, to print the *character* represented by the CONTROL-ARROW combination, press only the ARROW key while the control group is locked in. To move the cursor, press CONTROL and ARROW at the same time, regardless of which group is locked in.

*Inverse video (Atari logo) key.* This key is a *toggle*. Pressing it switches between inverse and regular video. In ANTIC 2 (Graphics 0), this will cause all the characters you enter to be reversed, as the computer normally does. In ANTIC 4 and 5, however, this will cause Color 3 to take its value from color register 4 (memory location 711 instead of 710). It will affect, therefore, only one of the colors, and if a character does not contain any dots of Color 3, inverse mode won't have any effect at all.

*CONTROL-ESC.* This key combination is a toggle. It will switch between Still and Auto-Advance modes. In Still Mode, pressing noncommand keys will display a new character in the same place on the screen. In Auto-Advance Mode, pressing noncommand keys will display a new character and then advance the cursor to the next position to the right, unless doing so would take the cursor beyond the edge of the display.

*To move the cursor.* Either move the joystick in the direction you want to move, or press the appropriate CONTROL-ARROW key combination. Only the joystick allows diagonal movement.

When the cursor reaches the edge of the screen, the display will begin to scroll until it reaches the limits of display height and width you specified during start-up. If you are at the edge of the display, the cursor simply won't move any farther that direction.

*Fast-fill function.* Sometimes you will have large areas or lines to fill with the same character. Instead of entering the character by typing it in each space where it is to appear, you can use the joystick and fire button. First maneuver the cursor until it is on top of the character you want to copy, or move it to the place where you want to begin the fast-fill operation and enter the character from the keyboard. Then press down the joystick button and hold it down while you use the joystick to move the cursor. From then on, until you let up on the button, wherever you move the cursor using the joystick, a trail made up of that character will be left behind.

You can also use this function to erase areas of the screen fairly quickly. Just move the cursor to a blank, press down the button, and the cursor will leave blanks behind it wherever you make it go.

*Clear screen function.* To erase the entire display, press CONTROL-SHIFT-CLEAR.

*Delete line function.* To delete an entire line of your screen, move your cursor to the line you want to delete and press SHIFT-DELETE. The line will vanish, and the entire display below that line will move upward one line on the screen. Whether the very bottom of your display is visible on the screen or not, a line of blanks will be inserted as the last line in your display.

*Insert line function.* To insert a blank line in your display, move the cursor to the position where you want the new line. Then press SHIFT-INSERT. The line that the cursor was on will move down, as will all the other lines below it in the display, and the cursor will now be on a blank line. At the bottom of the display, whether it is visible on the screen or not, the last line of your display will be deleted completely.

With both the delete and insert line functions, the line that disappears is irrecoverably lost. To get it back, you will have to enter all the characters just as you did before. So take care when using these two functions.

By using the delete and insert functions in succession, you can quickly blank large areas of the screen, a line at a time. Simply move to the top of the area you want to blank out, and press SHIFT-DELETE as often as it takes to erase all the lines you wanted to get rid of. Then press SHIFT-INSERT until the desired number of blank lines appears.

You can also use these functions to move the entire picture up or down in the display. For instance, suppose you loaded a display that had been created and saved with only 24 lines, and you want to add another 24-line picture above it. At the beginning of the editing session, simply specify 48 lines as the height of the display. Fontbyter will put the 24 new blank lines at the end of the display. To move the old picture down into that blank area, start at the top of the screen and press SHIFT-INSERT 24 times.

*Three joystick modes.* We've already gone over the use of the joystick in Cursor Mode. The joystick can also be toggled into two other modes. If you press the START button while in Cursor Mode, the joystick will change to Scroll Mode. If you press the START button in Scroll Mode, the joystick will shift to Color Mode. And pressing the START button in Color Mode will shift you back to Cursor Mode again.

# ComputAbility™

## The Discount Software Company That Pays You A Dividend! !

ATARI

| | |
|---|---|
| **ATARI 800 48K** | **CALL** |
| ATARI 800XL 64K | CALL |
| ATARI 810 DISK DRIVE | CALL |
| ATARI 1050 DISK DRIVE | CALL |
| ATARI 850 INTERFACE | CALL |
| ATARI 1010 RECORDER | 77.00 |
| HOME MANAGER KIT | 64.95 |
| ATARI 1027 PRINTER | 289.00 |

Call for **PERCOM** Disk Drive Prices.

**MOSAIC**

| | |
|---|---|
| 64K Ram/400 | 149.00 |
| 64K Ram + Cable Kit /400/800 | 169.00 |
| 48K Ram Kit | 94.00 |
| 16/32 Expander | 64.95 |
| 32K Ram | 77.95 |
| Mosaic Adaptor | 49.95 |

B-KEY 400 .......... 39.95
**AMDEK COLOR I .......... 289.00**

**Suncom Joysensor...34.95**
30 day money back guarantee

**PRINTERS**
PROWRITER 8510 AP ... CALL
**Seikosha**
**W/Parallel Port .......... 289.00**

## SUPER SPECIALS

| FROGGER | ZAXXON | REPTON | PREPPIE II | CHOPLIFTER |
|---|---|---|---|---|
| $22.95 DISK/TAPE | $26.95 DISK/TAPE | $26.95 DISK | $22.95 DISK/TAPE | $21.49 DISK |

PRICES EFFECTIVE NOW THROUGH SEPTEMBER 30, 1983

## FREE* SOFTWARE AND ACCESSORIES

### ATARI

| | |
|---|---|
| Conversational Languages - T | 43.95 |
| Programming 2 & 3 - T | 21.95 |
| Music Composer - C | 32.95 |
| My First Alphabet - D | 26.95 |
| Touch Typing - T | 19.95 |
| Home Filing Manager - D | 37.95 |
| Mailing List - T | 19.95 |
| Asteroids - C | 26.95 |
| Caverns Of Mars - D | 28.95 |
| Computer Chess - C | 26.95 |
| Missile Command - C | 26.95 |
| Super Breakout - C | 26.95 |
| Star Raiders - C | 32.95 |
| Assembly Editor - C | 44.95 |
| Basic - C | 39.95 |
| Macroassembler - D | 65.95 |
| Microsoft Basic - D | 65.95 |
| Pilot (Home Package) - C | 58.95 |
| Invitation To Programming I - T | 18.95 |
| Speed Reading - T | 55.95 |
| Basketball - C | 26.95 |
| Graph-It - T | 15.95 |
| Juggles House - D/T | 22.95 |
| Pilot (Educator) - C | 97.95 |
| Video Easel - C | 26.95 |
| Defender - C | 32.95 |
| Galaxian - C | 32.95 |
| Qix - C | 32.95 |
| Dig Dug - C | 32.95 |
| ET - C | 37.95 |
| Timewise - D | 23.95 |
| Atariwriter - C | 74.95 |
| Donkey Kong - C | 37.95 |
| Ms. Pac-Man - C | 39.95 |
| Tennis - C | 35.95 |
| Eastern Front - C | 32.95 |
| Donkey Kong Jr. - C | 39.95 |
| Pengo - C | 35.95 |
| Logo - C | 79.95 |
| Robitron - C | 35.95 |
| Pole Position - C | 39.95 |
| Microsoft Basic II - C | 65.95 |
| Paint - D | 33.95 |
| Caverns of Mars - C | 32.95 |

### FIRST STAR

| | |
|---|---|
| Astro Chase - D/T | 23.95 |
| Bristles - D/T | 23.95 |
| Flip Flop - D/T | 23.95 |
| Rent Wars - D/T | 23.95 |

### BIG FIVE

| | |
|---|---|
| Miner 2049'er - Cart | 39.95 |
| Scraper Caper - C | 39.95 |

### COMPUTABILITY DIVIDEND COUPON PREMIUMS

| | |
|---|---|
| 3 M Blank Disk - Box of 10 | 27.00 or 10 Coupons |
| Elephant Disks - Box of 10 | 22.00 or 8 Coupons |
| Flip 'n File Disk Holder | 20.95 or 6 Coupons |
| Flip 'n File Cartridge Holder | 20.95 or 6 Coupons |
| Monitor Stand | 27.95 or 10 Coupons |
| VU - Case Disk Holder W/Lock | 24.00 or 10 Coupons |
| Heavy Plastic Dust Cover 800 or 810 | 9.95 or 3 Coupons |
| First Book of Atari Software 1983 | 17.95 or 6 Coupons |
| Kids & The Atari | 17.95 or 6 Coupons |
| The Atari Assembler | 12.95 or 5 Coupons |
| Atari Games & Recreation | 14.95 or 6 Coupons |
| Atari Pilot for Beginners | 14.95 or 6 Coupons |
| The Visicalc Book Atari | 14.95 or 6 Coupons |
| Atari Basic | 12.95 or 5 Coupons |
| Atari Sound & Graphics | 9.95 or 3 Coupons |
| Starfighter | 12.99 or 5 Coupons |
| Slik Stik | 9.95 or 3 Coupons |
| Extension Cable - 6 ft. | 6.95 or 3 Coupons |
| Lefty Adaptor | 9.95 or 3 Coupons |
| Stik Stand | 6.99 or 3 Coupons |
| Suncom Tac II | 19.95 or 8 Coupons |
| Suncom Game Switch | 6.95 or 3 Coupons |
| TI Adaptor | 12.95 or 5 Coupons |
| Kraft Switch Hitter Joystick | 17.95 or 7 Coupons |
| Kraft Joystick | 15.95 or 6 Coupons |
| Maxell 2 Pack Disks | 8.95 or 3 Coupons |
| Inside Atari Basic | 12.95 or 5 Coupons |
| Compute's First Book of Atari | 12.95 or 5 Coupons |
| Compute's Second Book of Atari | 12.95 or 5 Coupons |
| De Re Atari | 17.95 or 6 Coupons |
| Wico Joystick | 24.00 or 10 Coupons |
| $24.00 Worth of Software | 10 Coupons |

### *COUPON PROGRAM

The purchase of each program (with the exception of Super Specials, Atari and APX) will earn you 1 COMPUTABILITY DIVIDEND COUPON. Save 10 coupons and redeem them for your choice of any program we sell for $24.00 or less. Less than 10 coupons may be redeemed for premium items as indicated. You pay only a $2.50 shipping & handling charge.

### DATASOFT

| | |
|---|---|
| Pooyan - D/T | 23.95 |
| Moon Shuttle - D/T | 23.95 |
| Sands of Egypt - D | 23.95 |
| Text Wizard - D | 37.95 |
| Spell Wizard - D | 37.95 |
| Basic Compiler - D | 59.95 |
| Teletalk - D | 37.95 |
| Canyon Climber - D/T | 20.95 |

### SIRIUS

| | |
|---|---|
| Bandits - D | 27.95 |
| Way Out - D | 31.95 |
| Repton - D | 31.95 |
| Twerps - D | 27.95 |
| Blade of Blackpoole - D | 31.95 |
| Type Attack - D | 31.95 |
| Beer Run - D | 27.95 |
| Critical Mass - D | 31.95 |

### ADVENTURE INTERNATIONAL

| | |
|---|---|
| Preppie - D/T | 23.95 |
| S.A.G.A. Adventures - D | 31.95 |
| Sea Dragon - D/T | 27.95 |
| Stratos - D/T | 27.95 |
| Bugoff - D/T | 23.95 |
| Preppie II - D/C | 27.95 |
| Stone of Sisyphus - D | 31.95 |
| Eliminator - D/T | 20.95 |
| Rally Speedway - Cart | 37.95 |
| Ultra Disassembler - D | 39.95 |
| Diskey - D | 39.95 |
| Triad - D/T | 27.95 |

### SYNAPSE

| | |
|---|---|
| Dimension X - D/T | 27.95 |
| Shadow World - D/T | 27.95 |
| Rainbow Walker - D/T | 27.95 |
| Drelbs - D/T | 27.95 |
| Shamus II - D/T | 27.95 |
| Blue Max - D/T | 27.95 |
| Fort Apocalypse - D/T | 27.95 |
| Necromancer - D/T | 27.95 |
| Pharoah's Curse - D/T | 27.95 |
| Survivor - D/T | 27.95 |

### PARKER BROS.

| | |
|---|---|
| Q-Bert | |
| Frogger | Please |
| Astro Chase | call |
| Popeye | for |
| Tutankhamen | Prices |
| Risk | |
| Chess | |

### MISCELLANEOUS

| | |
|---|---|
| Night Mission - D/T | 23.95 |
| Jumpman - D/T | 31.95 |
| Juice - D/T | 23.95 |
| Castle-Wolfenstein - D | 23.95 |
| Financial Wizard - D | 44.95 |
| PM Animator - D | 27.95 |
| Teletari - D | 31.95 |
| Starbowl Football - D/T | 25.50 |
| Ali Baba - D | 26.50 |
| Ultima II - D | 44.95 |
| Crisis Mountain - D | 27.95 |
| Star League Baseball - D/T | 25.50 |
| Home Accountant - D | 54.95 |
| Mountain King - Cart | 31.95 |

**D - Disk T - Cassette C - Cartridge**

ATARI is a trademark of ATARI, Inc.

**We Carry Hundreds of Items for ATARI 400/800, Ask for Our FREE CATALOG.**

**Mastercard/VISA Order Toll Free** **800-558-0003** No surcharge for credit cards **In Wisc. Call 414/351-2007**

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

**ORDERING INFORMATION:** To order by mail, send money order, certified check, or personal check (allow 14 days to clear) to COMPUTABILITY. Include $2.00 shipping on all software orders. Include 3% shipping on all hardware orders, minimum $2.50. Mastercard and Visa please include card number and expiration date. WI residents please add 5% sales tax. Canadian, APO & FPO software orders include 5% shipping minimum $5.00. All other foreign software, please add 15% shipping, minimum $7.00 (US Funds only). Prices subject to change without notice.

ORDER HOURS
Mon-Fri 12-9 PM (CST)
Sat 12-5 PM (CST)

**1.** Scroll Mode. This mode enables you to scroll the TV screen window around the entire display by moving the joystick in the appropriate direction. When you move, the cursor character will disappear. When you return to Cursor Mode, the cursor will come back to the middle of the screen.

**2.** Color Mode. In this mode, the joystick controls the color registers as follows:

- Forward and back: Color register 1 (Memory location 708)
- Left and right: Color register 2 (709)
- Forward and back with joystick button depressed: Color register 3 (710)
- Left and right with joystick button depressed: Background color register (712)
- Forward and back with SELECT depressed: Inverse color register (711)

**Summary Of Command Keys**

**START.** Cycle from Cursor Mode to Scroll Mode to Color Mode and back to Cursor Mode.

**SELECT.** Save the current display without interrupting the edit. In Color Mode, moving the joystick forward and back with SELECT pressed will change the inverse color.

**OPTION.** Save the current display and stop the editing session.

**CONTROL-ARROW.** Move the cursor.

**SHIFT-INSERT.** Insert a blank line where the cursor is, and delete the bottom line of the display.

**SHIFT-DELETE.** Delete the cursor line, and add a blank line at the bottom of the display.

**Atari logo key.** Toggle back and forth between inverse and regular characters.

**SHIFT-CAPS/LOWR.** Select the shifted character group.

**CONTROL-CAPS/LOWR.** Select the control character group.

**CONTROL-ESC.** Toggle between Still and Auto-Advance modes.

**CONTROL-SHIFT-CLEAR.** Erase the entire display.

As you press the joystick forward or to the right in Color Mode, that particular color will get brighter and brighter until it reaches maximum brightness; then it will jump to the next color at its darkest value and get brighter and brighter with that color. Pushing leftward or backward cycles through the colors from bright to dark. There are 16 colors, each with eight levels of brightness. You can cycle through the colors endlessly in either direction.

When you start editing with a new display, the colors are the system default colors. When you load a previously saved display, however, you start with the colors saved with that display. You can change the colors however you like, and whatever the colors are when you save your display, those values will be saved with it.

## Ending And Saving

There are two ways to save a screen.

**1.** You can press the SELECT button when the joystick is in Cursor Mode, and the display will be saved as "D1:TEMPFILE.SCR". The screen is not changed, and you can resume editing as soon as the joystick or keyboard respond again.

**2.** You can press the OPTION button. Fontbyter will save the entire display in a file named "D1:TEMPFILE.SCR". The screen then clears, and Fontbyter asks if you want to save the display in the save file you asked for at the beginning of the edit. If you answer yes, "TEMPFILE" is renamed with the save filename you chose at the beginning. If a file with the same name already exists on the disk, it will be erased at this time.

If you are merely saving a half-done file to make sure some catastrophe doesn't lose it for you, then "TEMPFILE.SCR" should be security enough – if the system crashes, you'll know that the screen as you last saved it is in that file.

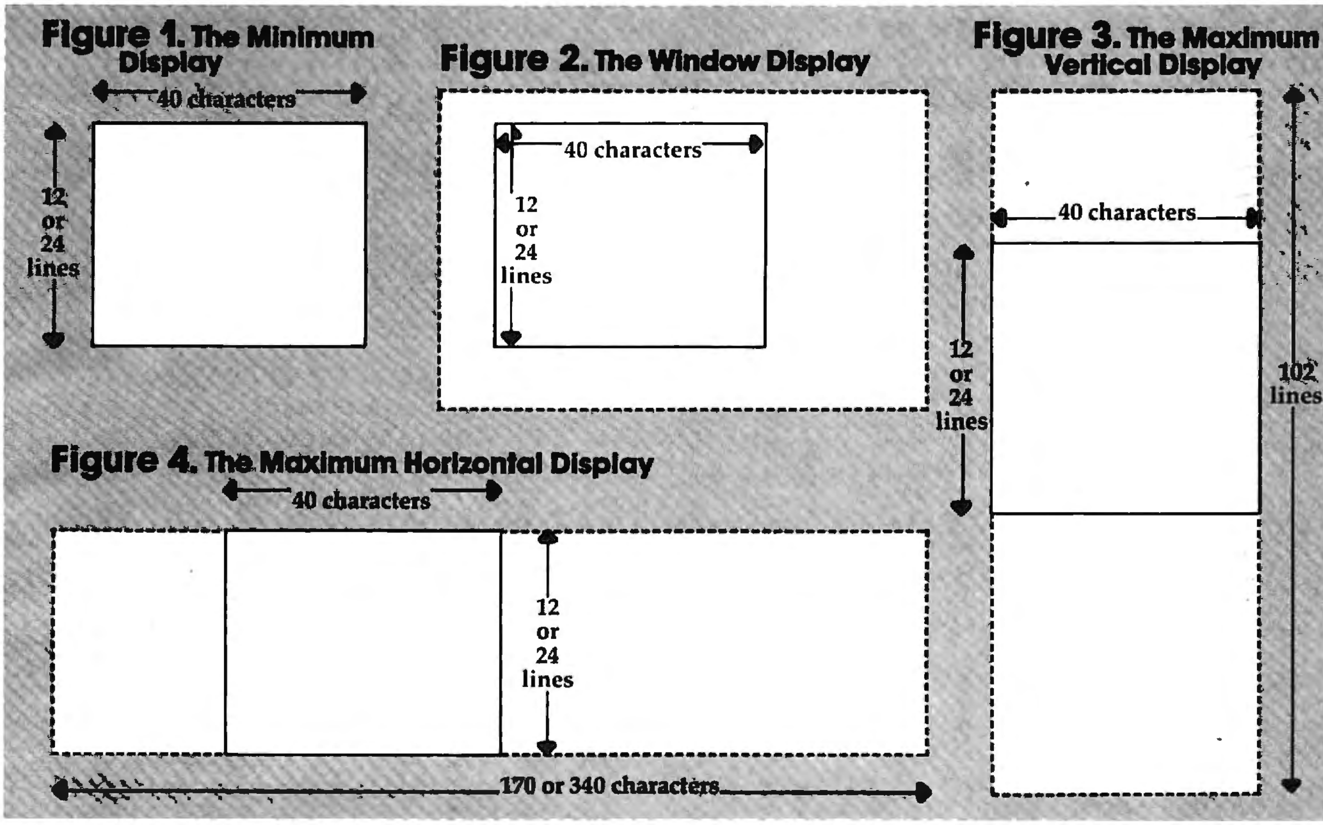
You will then be asked if you want to return to edit the same screen. If you say yes, your saved screen will quickly be reloaded into memory, and the program will reinitialize. If you say no, you will be asked whether you want to quit or start Fontbyter over again. If you choose the quit option and change your mind, don't worry. Just give the direct command RUN, and Fontbyter will begin again with the setup prompts.

## Using Fontbyter Screens In Your Programs

Just because Fontbyter scrolls doesn't mean you have to make one continuous scrolling display. You can create many different screen displays in one file, "stacking" them vertically, and then use page flipping in your own program to move instantly from one to another. The advantage of using Fontbyter is that while you are creating them, you can scroll from one to the other to compare them and make sure that any animation effects you are trying for are working properly.

The diagrams will show you the variety of display configurations you can choose.

Here are subroutines you can include in your own programs to use the displays you have created with Fontbyter.




*Loading files.* To use Fontbyter displays, your program will need to load a character set and the display file. Subroutine 1 loads slowly, entirely from BASIC. Subroutine 2 loads quickly, using a machine language routine that accesses an operating system fast-load program.

**Subroutine 1. Slow Load**

```
10 REM Slow load (character set)
100 OPEN #1,4,0,"D1:CHARACT.SET":FOR
    I=0 TO 1023:GET #1,N:POKE CH,N:
   NEXT I:CLOSE #1:RETURN
190 REM Slow load (display file)
200 OPEN #1,4,0,"D1:DRAWING.SCR":GET
    #1,MD:GET #1,WD:GET #1,LN:IF MD
   >5 THEN MD=MD-10:WD=WD+256
205 FOR I=708 TO 712:GET #1,N:POKE I
   ,N:NEXT I:FOR I=0 TO WD*LN-1:GET
    #1,N:POKE SC+I,N:NEXT I:CLOSE #
   1:RETURN
```

**Subroutine 2. Fast Load**

```
10 REM Set up variables
20 X=16:ICCOM=834:ICBADR=836:ICBLEN=
   840:REM See text for meaning of v
   ariables SP and CHBAS
90 REM Fast load (display file)
100 OPEN #1,4,0,"D1:DRAWING.SCR":GET
    #1,MD:GET #1,WD:GET #1,LN:IF MD
   >5 THEN WD=WD+256:MD=MD-10
110 SZ=WD*LN:FOR I=708 TO 712:GET #1
   ,N:POKE I,N:NEXT I
120 POKE ICBADR+X+1,SP:POKE ICBADR+X
   ,0:POKE ICBLEN+X+1,1+INT(SZ/256)
   :POKE ICBLEN+X,0
130 POKE ICCOM+X,7:I=USR(ADR("hhh■LV
   ■"),X):CLOSE #1:RETURN
190 REM Fast load (character set)
200 OPEN #1,4,0,"D1:CHARACT.SET":POK
   E ICBADR+X+1,CHBAS:POKE ICBADR+X
   ,0:POKE ICBLEN+X+1,4:POKE ICBLEN
   +X,0
210 POKE ICCOM+X,7:C=USR(ADR("hhh■LV
   ■"),X):CLOSE #1:POKE 756,CHBAS:R
   ETURN
```

*Display list setup.* Subroutine 3 sets up an ANTIC 2 or 4 display list that can be horizontally or vertically scrolled. Subroutine 4 sets up an ANTIC 5 display list that can be horizontally or vertically scrolled. Subroutines 5 and 6 set up display lists that *cannot* be horizontally scrolled – use these only to load displays that were created with the minimum line width.

On XL models, the simple display list can be set up with the BASIC statements Graphics 12+16 and Graphics 13+16, making the non-scrolling display list subroutine unnecessary. For horizontally scrolling displays, however, these subroutines are still needed.

**Subroutine 3.**
**Horizontal Scroll Display List, ANTIC 2 or 4**

```
10 REM Lines 20 and 30 are just a de
   monstration.  Change the value of
    SC and see what happens!
20 DL=PEEK(88)+256*PEEK(89):SC=DL:MO
   DE=4:WIDE=40:GOSUB 100
30 FOR I=0 TO 1000:NEXT I:SC=0:MODE=
   2:GOSUB 100:FOR I=0 TO 1000:NEXT
   I:GOTO 20
90 REM This ANTIC 2 or 4 display lis
   t can be horizontally scrolled.
   Just set the values of SC,DL,MODE
   , and WIDE.
```

```
100 FOR I=0 TO 2:POKE DL+I,112:NEXT
    I:N=0:M=MODE+64
110 FOR I=DL+3 TO DL+72 STEP 3:C=SC+
    N:POKE I,M:POKE I+2,INT(C/256):P
    OKE I+1,C-256*PEEK(I+2):N=N+WIDE
    :NEXT I
120 POKE I,65:DLHI=INT(DL/256):DLLO=
    DL-DLHI*256:POKE I+1,DLLO:POKE I
    +2,DLHI:POKE 561,DLHI:POKE 560,D
    LLO:RETURN
```

**Subroutine 4. Horizontal Scroll Display List, ANTIC 5**

```
10 REM Lines 20 and 30 are just a de
   monstration.  Change the value of
    SC and see what happens!
20 DL=PEEK(88)+256*PEEK(89):SC=PEEK(
   106)*256:MODE=5:WIDE=40:GOSUB 100
30 FOR I=0 TO 1000:NEXT I:SC=0:GOSUB
    100:FOR I=0 TO 1000:NEXT I:GOTO
   20
90 REM This ANTIC 5 display list can
    be horizontally scrolled.  Just
   set the values of SC,DL,MODE, and
    WIDE.
100 FOR I=0 TO 2:POKE DL+I,112:NEXT
    I:N=0:M=MODE+64
110 FOR I=DL+3 TO DL+36 STEP 3:C=SC+
    N:POKE I,M:POKE I+2,INT(C/256):P
    OKE I+1,C-256*PEEK(I+2):N=N+WIDE
    :NEXT I
120 POKE I,65:DLHI=INT(DL/256):DLLO=
    DL-DLHI*256:POKE I+1,DLLO:POKE I
    +2,DLHI:POKE 561,DLHI:POKE 560,D
    LLO:RETURN
```

**Subroutine 5. Regular Display List, ANTIC 2 or 4**

```
10 REM The actual subroutine is line
   s 100-120.  You set the value of
   DL,SC,MODE,and WIDE.
20 DL=PEEK(88)+256*PEEK(89):MODE=2:W
   IDE=40
30 SC=0:MODE=2+2*(MODE=2):GOSUB 100
40 TRAP 30:ON PEEK(753)<>3 GOTO 40:S
   C=SC+480:SP=INT(SC/256):POKE DL5,
   SP:POKE DL4,SC-256*SP
50 FOR I=0 TO 30:NEXT I:GOTO 40
90 REM This ANTIC 2 and 4 display li
   st can be page flipped from BASIC
   .  POKE the screen address into D
   L4 and DL5.
100 FOR I=0 TO 2:POKE DL+I,112:NEXT
    I:DL4=DL+4:DL5=DL+5
110 POKE DL+3,64+MODE:POKE DL5,INT(S
    C/256):POKE DL4,SC-256*PEEK(DL5)
    :FOR I=DL+6 TO DL+28:POKE I,MODE
    :NEXT I
120 POKE I,65:DLHI=INT(DL/256):DLLO=
    DL-DLHI*256:POKE I+1,DLLO:POKE I
    +2,DLHI:POKE 561,DLHI:POKE 560,D
    LLO:RETURN
```

**Subroutine 6. Regular Display List, ANTIC 5**

```
10 REM The actual subroutine is line
   s 100-120.  You set the value of
   DL,SC,MODE,and WIDE.
20 DL=PEEK(88)+256*PEEK(89):MODE=5:W
   IDE=40:GOSUB 100
30 SC=0
40 TRAP 30:ON PEEK(753)<>3 GOTO 40:S
   C=SC+480:SP=INT(SC/256):POKE DL5,
   SP:POKE DL4,SC-256*SP
50 FOR I=0 TO 30:NEXT I:GOTO 40
90 REM This ANTIC 5 display list can
    be page flipped from BASIC.  Jus
   t POKE the screen address into DL
   4 and DL5.
100 FOR I=0 TO 2:POKE DL+I,112:NEXT
    I:DL4=DL+4:DL5=DL+5
110 POKE DL+3,64+MODE:POKE DL5,INT(S
    C/256):POKE DL4,SC-256*PEEK(DL5)
    :FOR I=DL+6 TO DL+16:POKE I,MODE
    :NEXT I
120 POKE I,65:DLHI=INT(DL/256):DLLO=
    DL-DLHI*256:POKE I+1,DLLO:POKE I
    +2,DLHI:POKE 561,DLHI:POKE 560,D
    LLO:RETURN
```

These routines use the following variables:

TOP is the page number of the top of memory. The Atari will not touch anything located above the top of memory – but anything below it is fair game. The display list, character set, screen memory, and machine language routines should all be placed above SP. So the load routines find out where the top of memory is and move it down enough pages to leave room for all the protected program areas. SC is the absolute address of the top of memory (SP*256); it is also the start of screen memory, so that it is POKEd into both the display list and location 106.

How much room should you leave? The character set takes 1K (four pages) and must start on a 1K boundary. Screen memory will never take more than 4K (16 pages), and should start on a 4K boundary, since ANTIC has problems when screen memory crosses that line. If your display is less than 2K, you can probably skip back from the top of memory a mere 4K (16 pages, or PEEK(106)-16), place screen memory at the new top of memory, and put the display list, machine language routines, and character set above it. If your display list is 3K or more, you should probably skip back 6K (24 pages, or PEEK(106)-24), place the character set at the new top of memory, followed by the display list, machine language subroutines, and then screen memory beginning at the 4K boundary line, 16 pages before the old top of memory. This routine assumes that arrangement.

SP is the page number of the start of screen memory, and SC is the absolute address of the start of screen memory (SP*256).

DL is the start of the display list. For page flipping, DL3 is DL+3, and DL4 is DL+4. These will contain the low byte and high byte of screen memory, and POKEing new values into these locations will flip screen memory.

CHBAS is the page number of the character set, and CH is the absolute address (CHBAS* 256).

MODE is the ANTIC mode number – either 2, 4, or 5. Adding 64 to MODE each time it is POKEd in tells the computer to look for a new screen memory address in the next two bytes in the display list.

WIDE is the width, in characters, of the entire horizontal line, not just the 40-character portion visible on screen at any one time. Thus, every MODE instruction is followed by a two-byte address, C, which tells it where to find the start of the next horizontal line.

POKEing 560 and 561 with 0 and DL/256 is what actually makes the display list start working. Until that moment, the display list is just a series of numbers in memory. But once 560 and 561 contain the address of the start of your display list, the TV screen is under your program's control.

ICBADR, ICBLEN, and ICCOM are the addresses of key locations in the IOCB handler. ICCOM must contain the number of the operation to be performed (7 to load, 11 to save). ICBADR must contain the low byte of the starting address of the area in memory to be saved from or loaded to (ICBADR+1 will contain the high byte). ICBLEN must contain the low byte of the length of the file to load (ICBLEN+1 will contain the high byte). The variable X represents the offset into the IOCB area. If you OPEN #1, then X=16. If you OPEN #2, then X=32. And so on, in multiples of 16. You might not get good results using OPEN #0 or OPEN #6 – those are reserved for system use.

With screen files created by Fontbyter, remember that the first eight bytes always contain the following information:

- ANTIC mode number (plus 10, if width is greater than 255 characters)
- width, or number of characters per line (low byte only, if width is greater than 255 characters)
- display height, or number of lines in the entire display
- colors to POKE into locations 708 through 712

To calculate the number of bytes in the whole screen display (SZ), multiply the height by the width. The number of bytes in the file is that number plus eight.

## Typing The Program

The bulk of the program is written in BASIC. The shortest machine language routines are included as string constants. The longer routines, however, DISPLAY, EXPAND, and DELETE, and two data files, MENU.DAT and CHARDATA.DAT, are listed after the main program. These should be entered using the BASIC loader program provided and saved on disk with exactly the filename specified. Fontbyter will look for these files and load them into strings or particular areas or memory during the run of the program.

Since Fontbyter works most efficiently with a disk drive, the program as written assumes a disk drive. However, a patient cassette user can remove all the routines related to choosing and testing filenames, and simply assign the value "C:" to all filename variables. All machine language routines could be added as DATA statements. You may also want to add prompts to tell the user what file the program is asking for. The biggest problem arises with load files during initialization, when the program tests the saved screen file once, then loads it again later. If you decide not to revise the program, make sure that you rewind the cassette containing the screen file after that initial test, so the file will be complete when it is loaded by the screen load subroutine.

## Program 1: Fontbyter

```
5 DIM F$(20),FSAVE$(20),FLOAD$(20),F
  L$(40),FLL$(20),DELETE$(124),EXPAN
  D$(124),CLEAR$(33),C(255)
10 GRAPHICS 0:X=16:ICCOM=834:ICBADR=
   836:ICBLEN=840
15 COL1=708:COL2=709:COL3=710:COL4=7
   11:COL5=712:SHIF=64:SCON=PEEK(559
   ):POKE 16,112:GOTO 440
20 OPEN #1,4,0,FL$:GET #1,MD:GET #1,
   WD:GET #1,LN:IF MD>5 THEN WD=WD+2
   56:MD=MD-10
25 SZ=WD*LN:FOR I=COL1 TO COL5:GET #
   1,N:POKE I,N:NEXT I
30 SC=SP*256:POKE ICBADR+X+1,SP:POKE
    ICBADR+X,0:POKE ICBLEN+X+1,1+INT
   (SZ/256):POKE ICBLEN+X,0
35 POKE ICCOM+X,7:I=USR(ADR("hhh■LV■
   "),X):CLOSE #1:RETURN
40 OPEN #1,8,0,"D1:TEMPFILE.SCR":WD=
   WIDE:MD=MODE:IF WIDE>255 THEN WD=
   WIDE-256:MD=MODE+10
45 PUT #1,MD:PUT #1,WD:PUT #1,LINE:F
   OR I=COL1 TO COL5:PUT #1,PEEK(I):
   NEXT I
50 POKE ICBADR+X+1,SP:POKE ICBADR+X,
   0:POKE ICBLEN+X+1,1+INT((LINE*WID
   E)/256):POKE ICBLEN+X,0
55 POKE ICCOM+X,11:I=USR(ADR("hhh■LV
   ■"),X):CLOSE #1:RETURN
60 IF ((LINE*WIDE-PIX)<WIDE) THEN RE
   TURN
65 LOWAD=SC+WIDE*INT(PIX/WIDE)-1:HIA
   DD=LOWAD+WIDE:POKE 206,INT(HIADD/
   256):POKE 205,HIADD-PEEK(206)*256
70 POKE 204,INT(LOWAD/256):POKE 203,
   LOWAD-PEEK(204)*256:POKE SC+PIX,O
   LD
75 POKE 207,INT((LINE*WIDE-PIX)/WIDE
   ):POKE 208,WLO:POKE 209,WHI
80 C=USR(ADR(DELETE$))
85 OLD=PEEK(SC+PIX):POKE SC+PIX,91:R
   ETURN
90 IF ((LINE*WIDE-PIX)<WIDE) THEN RE
   TURN
95 HIADD=SC+WIDE*(LINE-1)-1:LOWAD=HI
   ADD-WIDE:POKE 206,INT(HIADD/256):
   POKE 205,HIADD-PEEK(206)*256
100 POKE 204,INT(LOWAD/256):POKE 203
    ,LOWAD-PEEK(204)*256:POKE SC+PIX
    ,OLD
105 POKE 207,INT((LINE*WIDE-PIX)/WID
    E):POKE 208,WLO:POKE 209,WHI
110 C=USR(ADR(EXPAND$))
115 OLD=0:POKE SC+PIX,91:RETURN
120 OLD=PEEK(SC+PIX):POKE SC+PIX,91:
    POKE 559,SCON:POKE 16,112
```

```
125 MV=0:V=0:H=0:OPT=PEEK(53279):DI=
    PEEK(632):T=PEEK(644):E=0
130 IF OPT=6 THEN GOSUB 870:GOSUB 26
    0:GOTO 125
135 IF DI<15 THEN GOSUB 155:GOTO 125
140 IF PEEK(753)=3 THEN GOSUB 220:ON
     MV GOSUB 165:GOTO 125
145 ON OPT=3 GOTO 705:IF OPT=5 THEN
    GOSUB 40:POKE SC+PIX,91:GOTO 125
150 GOTO 125
155 V=WIDE*((DI=9 OR DI=13 OR DI=5)-
    (DI=10 OR DI=14 OR DI=6)):POKE 7
    7,0
160 H=(DI=6 OR DI=7 OR DI=5)-(DI=10
    OR DI=11 OR DI=9)
165 UD=INT(PIX/WIDE):IF UD-(V<0)<0 O
    R UD+(V>0)=LINE THEN V=0
170 LR=PIX-WIDE*UD:IF LR+H<0 OR LR+H
    >WIDE-1 THEN H=0
175 IF H=0 AND V=0 THEN 215
180 WH=0:WV=0:W=PEEK(DL4)+256*PEEK(D
    L5)-SC
185 U=INT(W/WIDE):IF V<>0 THEN WV=(U
    D-U-(V<0)<0)+2*(UD-U+(V>0)>8+12*
    (MODE<>5))
190 IF H<>0 THEN L=W-U*WIDE:WH=(LR+H
    -L<0)+2*(LR+H-L>39)
195 IF WH>0 OR WV>0 THEN POKE DL+114
    ,WH:POKE DL+115,WV:C=USR(DISPLAY)
200 POKE SC+PIX,OLD:PIX=PIX+H+V:POKE
     53279,1
205 IF T=1 THEN OLD=PEEK(SC+PIX):POK
    E SC+PIX,91:GOTO 215
210 POKE SC+PIX,OLD
215 RETURN
220 GOSUB 785:ON (C=134)+(C=135)+(C=
    142)+(C=143)+2*(C=116)+3*(C=119)
    +4*(C=246) GOTO 250,60,90,645
225 IF C=156 THEN AV=1*(AV=0):GOTO 9
    20
230 IF N=60 THEN SHIF=4+C-64:POKE 53
    279,4:GOSUB 930:RETURN
235 IF N=39 THEN VERS=128*(VERS=0):G
    OTO 920
240 OLD=C(N+SHIF)+VERS:POKE SC+PIX,O
    LD:ON AV GOTO 245:RETURN
245 C=135
250 V=WIDE*((C=143)-(C=142)):H=(C=13
    5)-(C=134):MV=1:RETURN
255 GOSUB 920:POKE SC+PIX,91:RETURN
260 GOSUB 920
265 DI=PEEK(632):T=PEEK(644):DI=DI+5
    *(DI=7):DI=DI-10:OPT=PEEK(53279)
    :IV=(OPT=5):IF OPT=6 THEN 255
270 IF DI<1 OR DI>4 THEN 265
275 ON (4*T)+DI GOSUB 280,285,290,29
    5,300,305,310,320:GOTO 265
280 POKE COL5,PEEK(COL5)-2+256*(PEEK
    (COL5)<2):RETURN
285 POKE COL5,PEEK(COL5)+2-256*(PEEK
    (COL5)>253):RETURN
290 POKE COL3,PEEK(COL3)-2+256*(PEEK
    (COL3)<2):RETURN
295 POKE COL3,PEEK(COL3)+2-256*(PEEK
    (COL3)>253):RETURN
300 POKE COL2,PEEK(COL2)-2+256*(PEEK
    (COL2)<2):RETURN
305 POKE COL2,PEEK(COL2)+2-256*(PEEK
    (COL2)>253):RETURN
310 IF IV THEN POKE COL4,PEEK(COL4)-
    2+256*(PEEK(COL4)<2):RETURN
315 POKE COL1,PEEK(COL1)-2+256*(PEEK
    (COL1)<2):RETURN
320 IF IV THEN POKE COL4,PEEK(COL4)+
    2-256*(PEEK(COL4)>253):RETURN
325 POKE COL1,PEEK(COL1)+2-256*(PEEK
    (COL1)>253):RETURN
330 FLL$=FL$:FOR I=1 TO LEN(FL$):N=A
    SC(FL$(I,I)):ON N=58 GOSUB 370:N
    EXT I:FL$=FLL$
335 FLL$=FL$:FOR I=1 TO LEN(FL$):N=A
    SC(FL$(I,I)):ON N=46 GOSUB 375:N
    EXT I:FL$=FLL$
340 IF LEN(FL$)>8 THEN FL$=FL$(1,8)
345 IF LEN(FL$)<1 THEN 390
350 N=ASC(FL$(1,1)):IF N>90 OR N<65
    THEN 385
355 IF LEN(FL$)<2 THEN GOTO 365
360 FOR I=2 TO LEN(FL$):N=ASC(FL$(I,
    I)):ON (N>90 OR N<65) AND (N>57
    OR N<48) GOTO 380:NEXT I
365 FLL$="D1:":FLL$(4)=FL$:N=0:RETURN
370 FLL$=FL$(I+1,LEN(FL$)):RETURN
375 FLL$=FL$(1,I-1):RETURN
380 POP :? "{CLEAR}":? "Illegal char
    acters in ";FL$:GOTO 390
385 ? "{CLEAR}":? FL$;" must start w
    ith a capital":? "letter.":GOTO
    390
390 ? "Let's try that name again.":N
    =1:RETURN
395 TRAP 400:OPEN #1,4,0,FL$:N=0:CLO
    SE #1:RETURN
400 ? :? FL$;" isn't on disk in":? "
    drive 1":? "Insert disk with ";F
    L$;"and":? "press RETURN.":CLOSE
     #1
405 ? "Or to try another file name,
    press anyother key."
410 ON PEEK(753)<>3 GOTO 410:GOSUB 7
    85:ON N=12 GOTO 395:N=1:RETURN
415 TRAP 435:OPEN #1,4,0,FL$:? FL$;"
     is already on disk.":? "Unless
    you change the name, the old"
420 ? "file will be lost.  To change
     the namepress RETURN":? "Or pre
    ss any other key to continue.":C
    LOSE #1
425 ON PEEK(753)<>3 GOTO 425:GOSUB 7
    85:ON N=12 GOTO 430:N=0:RETURN
430 N=1:RETURN
435 CLOSE #1:N=0:RETURN
440 ? "{13 SPACES}Fontbyter":? :? :?
    :GOSUB 905
445 GOSUB 850:? "What is the name of
     your character{4 SPACES}set? (E
    nter '@' for ROM set)":POKE 764,
    255:INPUT F$
450 IF F$="@" THEN 465
455 FL$=F$:GOSUB 330:ON N GOTO 445:F
    $=FLL$:F$(LEN(FLL$)+1)=".SET"
460 FL$=F$:GOSUB 395:ON N GOTO 445
465 GOSUB 840:? :? "What file should
     hold your finished{3 SPACES}scr
    een? (Eight characters)":POKE 76
    4,255:INPUT FSAVE$
470 FL$=FSAVE$:GOSUB 330:ON N GOTO 4
    65:FSAVE$=FLL$:FSAVE$(LEN(FLL$)+
    1)=".SCR"
475 FL$=FSAVE$:GOSUB 415:ON N GOTO 4
    65
480 FLOAD$="":? :? "Would you like t
    o edit a screen you{3 SPACES}hav
    e already saved? (Y or N) "
485 GOSUB 785:ON N=35 GOTO 535:ON N=
    43 GOTO 490:GOTO 485
490 ? :? "What is the name of the sa
    ved screen  file? ":POKE 764,255
```

```
    :INPUT FLOAD$
495 FL$=FLOAD$:GOSUB 330:ON N=0 GOTO
     500:GOTO 480
500 FLOAD$=FLL$:FLOAD$(LEN(FLL$)+1)=
    ".SCR"
505 FL$=FLOAD$:GOSUB 395:ON N GOTO 4
    80:OPEN #1,4,0,FLOAD$:GET #1,MD:
    GET #1,WD:GET #1,LN:CLOSE #1:FLO
    AD=1
510 IF MD>5 THEN MD=MD-10:WD=WD+256
515 ? :? FLOAD$;" was saved as:":? "
    Mode ";MD;",":? "with ";LN;" lin
    es":? "of ";WD;" characters per
    line."
520 ? "If you wish to Change these p
    arameterspress RETURN.":? "To le
    ave them unchanged press any
    {5 SPACES}other key."
525 ON PEEK(753)<>3 GOTO 525:GOSUB 7
    85:IF N=12 THEN 540
530 MODE=MD:WIDE=WD:LINE=LN:GOTO 585
535 FLOAD=0
540 ? :? "What Antic mode will you w
    ork in?":? "(Antic 2, 4, OR 5) "
    :POKE 764,255
545 GOSUB 785:ON N<>30 AND N<>24 AND
     N<>29 GOTO 545
550 MODE=C(N)-16
555 ? :? "How wide a line?":? "  (Mi
    nimum 40 characters":? "
    {3 SPACES}maximum ";170+170*(MOD
    E=5);" characters)"
560 POKE 764,255:TRAP 560:INPUT WIDE
    :WIDE=INT(WIDE):ON WIDE<40 OR WI
    DE>170 GOSUB 790
565 ? :? "How many lines do you want
     to edit?{5 SPACES}(Minimum ";12
    +12*(MODE<>5);:? "{3 SPACES}Maxi
    mum ";INT(4096/WIDE);")"
570 TRAP 570:INPUT LINE
575 LINE=INT(LINE):IF LINE>INT(4096/
    WIDE) THEN LINE=INT(4096/WIDE)
580 IF LINE<12+12*(MODE=4) THEN LINE
    =12+12*(MODE=4)
585 ? "{CLEAR}":? "You have chosen:"
    :? "Character set--";F$:? "Save
    file--";FSAVE$:? "Load file--";F
    LOAD$
590 SZ=LINE*WIDE-1:? "Mode ";MODE:?
    LINE;" lines of ";WIDE;" charact
    ers"
595 ? "If this is right, press START
    {9 SPACES}To make changes, press
     OPTION"
600 ON (PEEK(53279)=6)+(2*(PEEK(5327
    9)=3)) GOTO 605,440:GOTO 600
605 A=PEEK(106):TOP=A-24:CHBAS=TOP:C
    H=CHBAS*256:SP=TOP+8:SC=SP*256:P
    OKE 106,TOP:OLDCHBAS=224:GRAPHIC
    S 0
610 ? "Just a minute while I get mys
    elf{6 SPACES}together . . ."
615 IF F$="@" THEN CHBAS=224:CH=CHBA
    S*256:GOTO 630
620 OPEN #1,4,0,F$:POKE ICBADR+X+1,C
    HBAS:POKE ICBADR+X,0:POKE ICBLEN
    +X+1,4:POKE ICBLEN+X,0
625 POKE ICCOM+X,7:C=USR(ADR("hhh*LV
    d"),X):CLOSE #1
630 POKE 559,0:GOSUB 640:GOSUB 655:G
    OSUB 810:GOSUB 635:ON FLOAD GOSU
    B 650:GOSUB 925:GOTO 120
635 POKE 756,CHBAS:RETURN
640 OPEN #1,4,0,"D1:CLEAR.SUB":FOR I
    =1 TO 33:GET #1,N:CLEAR$(I,I)=CH
    R$(N):NEXT I:CLOSE #1
645 C=USR(ADR(CLEAR$),SP,X):RETURN
650 T=SZ:FL$=FLOAD$:GOSUB 20:SZ=T:RE
    TURN
655 DL=256*(TOP+4):DL4=DL+4:DL5=DL+5
    :FOR I=0 TO 2:POKE DL+I,112:NEXT
     I:PIX=0:N=0
660 FOR I=DL+3 TO DL+27+36*(MODE<>5)
     STEP 3:C=SC+N*WIDE:POKE I,64+MO
    DE:POKE I+2,INT(C/256)
665 POKE I+1,C-256*PEEK(I+2):N=N+1:N
    EXT I
670 N=0:MENU=256*(TOP+5)+64:DLMEN=DL
    +32+36*(MODE<>5):POKE DLMEN-2,MO
    DE+64:POKE DLMEN,INT(MENU/256)
675 POKE DLMEN-1,MENU-256*PEEK(DLMEN
    ):FOR I=DLMEN+1 TO DLMEN+3:POKE
    I,MODE:NEXT I
680 POKE I,65:POKE I+1,0:POKE I+2,DL
    /256:OPEN #1,4,0,"D:DISPLAY.SUB"
685 DISPLAY=DL+128:TRAP 690:FOR I=0
    TO 186:GET #1,N:POKE DISPLAY+I,N
    :NEXT I:GOTO 695
690 POP
695 WHI=INT(WIDE/256):WLO=WIDE-256*W
    HI:POKE DL+112,WLO:POKE DL+113,W
    HI
700 POKE 560,0:POKE 561,DL/256:CLOSE
     #1:RETURN
705 POKE SC+PIX,OLD:GOSUB 40:POKE 75
    6,OLDCHBAS:GRAPHICS 0:POKE 764,2
    55
710 ? "Screen is saved as D1:TEMPFIL
    E.SCR":? :? "Do you want to save
     the screen as":? FSAVE$;"? (Y o
    r N)"
715 GOSUB 785:ON N<>43 AND N<>35 GOT
    O 715:IF N=43 THEN GOSUB 765:GOT
    O 725
720 FSAVE=0
725 ? :? "Do you want to quit? (Y or
     N)":POKE 764,255
730 GOSUB 785:ON N<>43 AND N<>35 GOT
    O 730:ON N=35 GOTO 735:ON N=43 G
    OTO 760
735 ? :? "To return to edit the same
     screen,{4 SPACES}press OPTION":
    ? :? "To start FONTBYTER over, p
    ress START"
740 OPT=PEEK(53279):ON ((OPT=6)+(2*(
    OPT=3))) GOTO 745,750:GOTO 740
745 POKE 106,A:GRAPHICS 0:GOTO 10
750 POKE 106,TOP:GOSUB 635:FL$="D1:T
    EMPFILE.SCR":IF FSAVE=1 THEN FL$
    =FSAVE$
755 GOSUB 20:GOSUB 655:GOTO 120
760 POKE 106,A:POKE 764,255:GRAPHICS
     0:END
765 FSAVE=1:TRAP 770:OPEN #2,4,0,FSA
    VE$:CLOSE #2:XIO 36,#2,0,0,FSAVE
    $:XIO 33,#2,0,0,FSAVE$:GOTO 775
770 CLOSE #2
775 FL$="D1:TEMPFILE.SCR,":FLL$=FSAV
    E$(4,LEN(FSAVE$)):FL$(17)=FLL$
780 XIO 32,#1,0,0,FL$:RETURN
785 C=PEEK(764):N=C-64*INT(C/64):RET
    URN
790 IF WIDE<40 THEN WIDE=40:RETURN
795 IF WIDE>170 AND MODE<>5 THEN WID
    E=170:RETURN
800 IF WIDE<340 THEN RETURN
805 WIDE=340:RETURN
810 TRAP 815:OPEN #1,4,0,"D:DELETE.S
```

```
    UB":FOR I=1 TO 124:GET #1,N:DELE
    TE$(I,I)=CHR$(N):NEXT I:GOTO 820
815 POP
820 CLOSE #1:WHI=INT(WIDE/256):WLO=W
    IDE-256*WHI
825 TRAP 830:OPEN #1,4,0,"D:EXPAND.S
    UB":FOR I=1 TO 124:GET #1,N:EXPA
    ND$(I,I)=CHR$(N):NEXT I:GOTO 835
830 POP
835 CLOSE #1:RETURN
840 TRAP 865:XIO 36,#1,0,0,"D:*.SCR"
845 ? :? "Currently saved screen fil
    es:":FLL$="SCR":GOTO 860
850 TRAP 865:XIO 35,#1,0,0,"D:*.SET"
855 ? :? "Currently available charac
    ter sets:":FLL$="SET"
860 FL$="D1:*.":FL$(LEN(FL$)+1)=FLL$
    :OPEN #1,6,0,FL$:FOR I=0 TO 50:I
    NPUT #1,FLL$:? FLL$:NEXT I
865 CLOSE #1:RETURN
870 GOSUB 920:POKE SC+PIX,OLD:GOTO 8
    95
875 WV=2*((DI=5)+(DI=13)+(DI=9))+(DI
    =10)+(DI=6)+(DI=14):WH=2*(DI<8 A
    ND DI>4)+(DI<12 AND DI>8)
880 W=(PEEK(DL4)+256*PEEK(DL5))-SC:U
    =INT(W/WIDE):WV=WV-(U=0 AND WV=1
    )-2*((U+7+12*(MODE<>5)=LINE-2) A
    ND WV=2)
885 L=W-(U*WIDE):WH=WH-(L=0 AND WH=1
    )-2*((L+40)=WIDE AND WH=2)
890 POKE DL+114,WH:POKE DL+115,WV:C=
    USR(DISPLAY)
895 IF PEEK(53279)<>6 THEN DI=PEEK(6
    32):ON DI<>15 GOTO 875:GOTO 895
900 PIX=PEEK(DL4)+256*PEEK(DL5)+(6+6
    *(MODE<>5))*WIDE+20:OLD=PEEK(PIX
    ):PIX=PIX-SC:RETURN
905 OPEN #4,4,0,"D:CHARDATA.DAT"
910 FOR I=0 TO 255:GET #4,N:C(I)=N:N
    EXT I
915 CLOSE #4:RETURN
920 FOR I=0 TO 10:POKE 53279,4:NEXT
    I:RETURN
925 OPEN #1,4,0,"D:MENU.DAT":FOR I=4
     TO 483:GET #1,N:POKE MENU+I,N:N
    EXT I:CLOSE #1
930 MENSH=MENU+160*INT(SHIF/64):POKE
     DLMEN,INT(MENSH/256):POKE DLMEN
    -1,MENSH-256*PEEK(DLMEN):RETURN
```

## Program 2: DISPLAY. SUB
### Machine Language Scrolling Subroutine

```
900 OPEN #1,8,0,"D1:DISPLAY.SUB"
910 FOR I=1 TO 186:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1000 DATA 104,173,49,2,133,206,133,2
     13
1008 DATA 173,48,2,105,3,133,205,105
1016 DATA 109,133,212,162,0,161,205,
     41
1024 DATA 191,133,207,230,205,161,20
     5,133
1032 DATA 203,160,1,177,205,133,204,
     200
1040 DATA 177,212,240,34,201,2,208,1
     6
1048 DATA 24,165,203,105,1,133,203,1
     65
1056 DATA 204,105,0,133,204,24,144,1
     4
1064 DATA 56,165,203,233,1,133,203,1
     65
1072 DATA 204,233,0,133,204,24,160,3
1080 DATA 177,212,240,42,201,2,208,1
     9
1088 DATA 24,165,203,160,0,113,212,1
     33
1096 DATA 203,200,165,204,113,212,13
     3,204
1104 DATA 24,144,19,56,165,203,160,0
1112 DATA 241,212,133,203,165,204,20
     0,241
1120 DATA 212,133,204,24,144,0,160,8
1128 DATA 165,207,201,5,240,2,160,20
1136 DATA 162,0,165,203,129,205,230,
     205
1144 DATA 165,204,129,205,132,207,24
     ,165
1152 DATA 203,160,0,113,212,133,203,
     165
1160 DATA 204,200,113,212,133,204,23
     0,205
1168 DATA 230,205,164,207,136,208,21
     9,165
1176 DATA 203,129,205,230,205,165,20
     4,129
1184 DATA 205,96
```

## Program 3: EXPAND. SUB
### Machine Language Line Insert Subroutine

```
900 OPEN #1,8,0,"D1:EXPAND.SUB"
910 FOR I=1 TO 122:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1000 DATA 104,166,207,169,0,165,209,
     240
1008 DATA 29,160,255,177,203,145,205
     ,136
1016 DATA 208,249,230,204,230,206,16
     4,208
1024 DATA 177,203,145,205,136,208,24
     9,198
1032 DATA 204,198,206,24,144,9,164,2
     08
1040 DATA 177,203,145,205,136,208,24
     9,202
1048 DATA 240,29,56,165,205,229,208,
     133
1056 DATA 205,165,206,229,209,133,20
     6,56
1064 DATA 165,203,229,208,133,203,16
     5,204
1072 DATA 229,209,133,204,24,144,182
     ,165
1080 DATA 209,240,27,160,255,169,0,1
     45
1088 DATA 203,136,208,251,230,206,23
     0,204
1096 DATA 164,208,145,203,136,208,25
     1,198
1104 DATA 206,198,204,24,144,11,164,
     208
1112 DATA 240,7,169,0,145,203,136,20
     8
1120 DATA 251,96
```

## Program 4: DELETE. SUB
### Machine Language Line Delete Subroutine

```
900 OPEN #1,8,0,"D1:DELETE.SUB"
910 FOR I=1 TO 122:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1000 DATA 104,166,207,169,0,165,209,
     240
1008 DATA 29,160,255,177,205,145,203
     ,136
```

```
1016 DATA 208,249,230,204,230,206,16
4,208
1024 DATA 177,205,145,203,136,208,24
9,198
1032 DATA 204,198,206,24,144,9,164,2
08
1040 DATA 177,205,145,203,136,208,24
9,202
1048 DATA 240,29,24,165,205,101,208,
133
1056 DATA 205,165,206,101,209,133,20
6,24
1064 DATA 165,203,101,208,133,203,16
5,204
1072 DATA 101,209,133,204,24,144,182
,165
1080 DATA 209,240,27,160,255,169,0,1
45
1088 DATA 205,136,208,251,230,206,23
0,204
1096 DATA 164,208,145,205,136,208,25
1,198
1104 DATA 206,198,204,24,144,11,164,
208
1112 DATA 240,7,169,0,145,205,136,20
8
1120 DATA 251,96
```

## Program 5: MENU. DAT

```
900 OPEN #1,8,0,"D1:MENU.DAT"
910 FOR I=1 TO 482:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1000 DATA 0,0,91,0,17,0,18,0
1008 DATA 19,0,20,0,21,0,22,0
1016 DATA 23,0,24,0,25,0,16,0
1024 DATA 28,0,30,0,126,0,0,0
1032 DATA 0,0,0,0,0,0,0,0
1040 DATA 0,0,0,127,0,113,0,119
1048 DATA 0,101,0,114,0,116,0,121
1056 DATA 0,117,0,105,0,111,0,112
1064 DATA 0,13,0,29,0,0,0,0
1072 DATA 0,0,0,0,0,0,0,0
1080 DATA 0,0,0,0,0,0,97,0
1088 DATA 115,0,100,0,102,0,103,0
1096 DATA 104,0,106,0,107,0,108,0
1104 DATA 27,0,11,0,10,0,0,0
1112 DATA 0,0,0,0,0,0,0,0
1120 DATA 0,0,0,0,0,0,0,122
1128 DATA 0,120,0,99,0,118,0,98
1136 DATA 0,110,0,109,0,12,0,14
1144 DATA 0,15,0,0,0,0,0,0
1152 DATA 0,0,0,0,0,0,0,0
1160 DATA 0,0,0,0,1,0,2,0
1168 DATA 3,0,4,0,5,0,6,0
1176 DATA 7,0,32,0,8,0,9,0
1184 DATA 125,0,0,0,0,0,0,0
1192 DATA 0,0,0,0,0,0,0,0
1200 DATA 0,0,0,0,0,49,0,55
1208 DATA 0,37,0,50,0,52,0,57
1216 DATA 0,53,0,41,0,47,0,48
1224 DATA 0,63,0,124,0,0,0,0
1232 DATA 0,0,0,0,0,0,0,0
1240 DATA 0,0,0,0,0,0,33,0
1248 DATA 51,0,36,0,38,0,39,0
1256 DATA 40,0,42,0,43,0,44,0
1264 DATA 26,0,60,0,62,0,0,0
1272 DATA 0,0,0,0,0,0,0,0
1280 DATA 0,0,0,0,0,0,0,58
1288 DATA 0,56,0,35,0,54,0,34
1296 DATA 0,46,0,45,0,59,0,61
1304 DATA 0,31,0,0,0,0,0,0
1312 DATA 0,0,0,0,0,0,0,0
1320 DATA 0,0,0,0,0,0,0,0
1328 DATA 0,0,0,0,0,0,0,0
1336 DATA 0,0,0,0,0,0,0,0
1344 DATA 0,0,0,0,0,0,0,0
1352 DATA 0,0,0,0,0,0,0,0
1360 DATA 0,0,0,0,0,81,0,87
1368 DATA 0,69,0,82,0,84,0,89
1376 DATA 0,85,0,73,0,79,0,80
1384 DATA 0,92,0,93,0,0,0,0
1392 DATA 0,0,0,0,0,0,0,0
1400 DATA 0,0,0,0,0,0,65,0
1408 DATA 83,0,68,0,70,0,71,0
1416 DATA 72,0,74,0,75,0,76,0
1424 DATA 123,0,94,0,95,0,0,0
1432 DATA 0,0,0,0,0,0,0,0
1440 DATA 0,0,0,0,0,0,0,90
1448 DATA 0,88,0,67,0,86,0,66
1456 DATA 0,78,0,77,0,64,0,96
1464 DATA 0,0,0,0,0,0,0,0
1472 DATA 0,0,0,0,0,0,0,0
1480 DATA 0,0
```

## Program 6: CHARDATA. DAT

```
900 OPEN #1,8,0,"D1:CHARDATA.DAT"
910 FOR I=1 TO 256:READ N:PUT #1,N:N
    EXT I:CLOSE #1:? I:END
1000 DATA 108,106,27,0,0,107,11,10
1008 DATA 111,0,112,117,0,105,13,29
1016 DATA 118,0,99,0,0,98,120,122
1024 DATA 20,0,19,22,91,21,18,17
1032 DATA 12,0,14,110,0,109,15,0
1040 DATA 114,0,101,121,127,116,119,
     113
1048 DATA 25,0,16,23,126,24,28,30
1056 DATA 102,104,100,0,0,103,115,97
1064 DATA 44,42,26,0,0,43,60,62
1072 DATA 47,0,48,53,0,41,63,124
1080 DATA 54,0,35,0,0,34,56,58
1088 DATA 4,0,3,6,0,5,2,1
1096 DATA 59,0,61,46,0,45,31,0
1104 DATA 50,0,37,57,0,52,55,49
1112 DATA 8,0,9,7,0,32,125,0
1120 DATA 38,40,36,0,0,39,51,33
1128 DATA 76,74,123,0,0,75,94,95
1136 DATA 79,0,80,85,0,73,92,93
1144 DATA 86,0,67,0,0,66,88,90
1152 DATA 0,0,0,0,0,0,0,0
1160 DATA 64,0,96,78,0,77,0,0
1168 DATA 82,0,69,89,0,84,87,81
1176 DATA 0,0,0,0,0,0,0,0
1184 DATA 70,72,68,0,0,71,83,65
1192 DATA 0,0,0,0,0,0,0,0
1200 DATA 0,0,0,0,0,0,0,0
1208 DATA 0,0,0,0,0,0,0,0
1216 DATA 0,0,0,0,0,0,0,0
1224 DATA 0,0,0,0,0,0,0,0
1232 DATA 0,0,0,0,0,0,0,0
1240 DATA 0,0,0,0,0,0,0,0
1248 DATA 0,0,0,0,0,0,0,0
```

## Program 7: CLEAR. SUB

### Machine Language Screen Clear Subroutine

```
900 OPEN #1,8,0,"D1:CLEAR.SUB"
910 FOR I=1 TO 33:READ N:PUT #1,N:NE
    XT I:CLOSE #1:? I:END
1000 DATA 104,104,104,133,208,104,10
     4,101
1008 DATA 208,133,209,169,0,133,207,
     160
1016 DATA 255,145,207,136,208,251,14
     5,207
1024 DATA 230,208,165,208,197,209,20
     8,235
1032 DATA 96
```

©

# Timex/Sinclair
# Making Change

Michael B. Williams

*This game is an excellent educational tool for children and is based on a previously published COMPUTE! article. The author also includes conversion tips for T/S users who want to translate programs originally written for other computers.*

Converting a program written for one computer to another computer can get difficult if the program contains machine-dependent features (graphics commands, for instance) or a lot of POKEs. If a program has many such features, it would probably be easier to rewrite it from scratch, once you understand it.

With modification, many programs published in COMPUTE! and other computer magazines can be converted to run on the Timex/Sinclair. One of these is "Making Change," by Myron Miller (COMPUTE!, February 1983), a program written to help children learn how to count money, divide money, and make change.

## Program Conversion

When you transfer any program, your conversion should make up for deficiencies in one area by compensating for them in another area. If your computer has limited graphics, liven it up with sound, and vice versa. The Sinclair has no sound, no color, and limited graphics. But that does not mean we cannot make the program interesting for the user.

First of all, I decided which version of "Making Change" to convert. I chose the Atari version and went to work. I made multiple statements into individual lines and added STR$ when printing numbers so there would be no pause before they were displayed. Congratulating myself on my persistence, I soon found I wasn't done yet – I had the huge task of debugging ahead. I eliminated one bug only to find several more.

Finally, I concluded it would be easier to rewrite the program using the listing as a guide. In doing so, I just went about program conversion in a different way. Instead of interpreting the listing line by line, I first tried to understand what it did as a whole. The result was a bug-free program that made it fun to learn about money.

Each problem is a question about money: how to count it, how to divide it – in short, how to make change. It addresses the child by name and asks him or her directly. If the child answers incorrectly, the program encourages the child to try once more, and the question is repeated. By the third attempt, if the child has not correctly solved the problem, he or she is given the answer.

A correct answer causes the program to call the machine language routine in line 1. This routine is very important to the program, but you can modify it as you wish.

Regardless of whether or not the question was answered correctly, the child is given the option of receiving his or her score, continuing, or stopping the program. In this way the child's progress may be evaluated at any stage of the program.

## Special Program Notes

Program 1 POKEs a machine language routine into the REM statement in line 1. After typing in Program 1, proofread it carefully. Type RUN and ENTER. Test it by using RAND USR 16514. The screen should fill up very fast with inverse spaces. List Program 1 and note that the REM statement in line 1 has been altered. It now contains a machine language subroutine.

Now, delete lines 10-60 and type in Program 2 with line 1 still in memory. When typing in Program 2, use the following instructions where graphics characters appear:

**60: Graphics, SHIFT-S, and SHIFT-D**
**5010: Underlined letters are inverse video**
**5115: Graphics and SHIFT-D**

## **Program 1:** Machine Language Loader

```
1 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
10 LET ZZ=16514
20 LET A$="2A0C40017618237EB92604C680770410F5C9"
30 POKE ZZ,16*CODE A$+CODE A$(2)-476
40 LET ZZ=ZZ+1
50 LET A$=A$(3 TO )
60 GOTO 30
```

## **Program 2:** Making Change

```
50 SLOW
60 PRINT "            MAKING CHANGE                "
70 PRINT ,,"      TIMEX/SINCLAIR VERSION  "
80 PRINT ,,"HELLO. PLEASE TELL ME YOUR NAME."
90 INPUT N$
100 GOSUB 2000
1000 REM PROBLEMS
1010 REM
1020 LET TP=TP+1
1030 LET CT=INT (100*RND)+1
1040 LET TR=0
1050 LET PT=NOT PT
1060 LET QU=INT (CT/25)
1070 LET DI=INT ((CT-QU*25)/10)
1080 LET NI=INT ((CT-QU*25-DI*10)/5)
1090 LET PE=INT CT-QU*25-DI*10-NI*5
1100 GOTO PT*1000+3000
2000 REM VARIABLES
2010 LET TP=0
2020 LET TC=0
2030 LET TW=0
2040 LET ME=0
2050 LET ML=0
2060 LET FI=0
2070 LET PT=0
2080 LET SCORE=5000
2100 DIM R$(5,10)
2110 FOR X=1 TO 5
2120 LET R$(X)="FANTASTIC.GREAT.    RIGHT.    TERRIFIC. VERY GOOD."(X*10-9 TO X*10)
2130 NEXT X
2999 RETURN
3000 REM COUNT CHANGE
3010 CLS
3020 PRINT TP;") ";N$;", IF I HAVE..."
3040 PRINT ,,TAB 5;QU;" QUARTER"+("S" AND QU<>1)
3050 PRINT ,,TAB 5;DI;" DIME"+("S" AND DI<>1)
3060 PRINT ,,TAB 5;NI;" NICKEL"+("S" AND NI<>1);", AND "
3070 PRINT ,,TAB 5;PE;" PENN"+("Y" AND PE=1)+("IES" AND PE<>1);", THEN"
3080 PRINT ,,"HOW MUCH CHANGE DO I HAVE? ";
3110 INPUT CH
3120 PRINT CH
3130 LET TR=TR+1
3140 IF CH=CT THEN GOTO 3500
3150 PRINT ,,"NOPE. THAT'S "+STR$ ABS (CT-CH)+" CENT"+("S" AND ABS (CH-CT)<>1)+" TOO "+("MUCH." AND CH>CT)+("LITTLE." AND CH<CT)
3160 LET ML=ML+1
3165 IT TR=1 THEN LET TW=TW+1
3170 IF TR=3 THEN GOTO 3210
3180 PRINT ,,"PRESS ENTER TO TRY AGAIN."
3190 IF INKEY$="" THEN GOTO 3190
3200 GOTO 3000
3210 PRINT ,,"THE ANSWER IS "+STR$ CT+", "+N$+"."
3220 GOTO SCORE
3500 LET ZZ=USR 16514
3510 PRINT
3520 PRINT N$+", "+"THATS "+R$(INT (5*RND)+1)
3530 LET ZZ=USR 16514
3540 IF TR=1 THEN LET TC=TC+1
3550 LET ME=ME+1
3560 GOTO SCORE
4000 REM GIVE CHANGE
4010 CLS
4020 PRINT STR$ TP+") "+N$+", IF I HAVE "+STR$ CT+" CENT"+("S" AND CT<>1)+","
4030 PRINT ,,"HOW MANY QUARTERS DO I HAVE? ";
4040 INPUT Q
4050 PRINT Q,,TAB 9;"DIMES? ";
4060 INPUT D
4070 PRINT D,,TAB 9;"NICKELS? ";
4080 INPUT N
4090 PRINT N,,TAB 9;"PENNIES? ";
4100 INPUT P
4110 PRINT P
4115 LET TR=TR+1
4120 IF Q=INT Q AND N=INT N AND D=INT D AND P=INT P THEN GOTO 4200
4130 PRINT ,,"THATS NOT FAIR, USING DECIMALS. YOU MUST PAY A FINE, "+N$+"."
4140 LET FI=FI+1
4150 GOTO 4220
4200 IF 25*Q+10*D+5*N+P=CT THEN GOTO 4400
4205 PRINT ,,"NOT QUITE--THAT MAKES "+STR$ (25*Q+10*D+5*N+P)+" CENTS."
4210 LET ML=ML+1
4215 IF TR=1 THEN LET TW=TW+1
4220 IF TR=3 THEN GOTO 4300
4230 PRINT ,,"PRESS ENTER TO TRY ONCE MORE."
4240 IF INKEY$="" THEN GOTO 4240
4250 GOTO 4000
4300 PRINT ,,"A THOUSAND TIMES NO, "+N$+"."
4310 PRINT ,,"I WOULD HAVE "+STR$ QU;" QUARTER"+("S" AND QU<>1);TAB13;STR$ DI+" DIME"+("S" AND DI<>1)
4320 PRINT TAB 13;STR$ NI+" NICKEL"+("S" AND NI<>1);", AND ";TAB 13;STR$ PE+" PENN"+("Y" AND PE=1)+("IES" AND PE<>1);"."
4330 GOTO SCORE
4400 IF Q=QU AND D=DI AND N=NI AND P=PE THEN GOTO 4500
4410 PRINT ,,"TRUE, BUT YOU COULD HAVE USED   FEWER COINS, "+N$+"."
4420 GOTO 4210
4500 LET ZZ=USR 16514
4510 PRINT
4520 PRINT "HEY..."+R$(INT (5*RND)+1)
4530 LET ZZ=USR 16514
4540 LET ME=ME+1
```

```
4550 IF TR=1 THEN LET TC=TC+1
5000 REM SCORE
5010 PRINT ,,"PRESS C TO CONTINUE, S FOR
      YOUR SCORE, OR ENTER TO STOP."
5020 LET I$=INKEY$
5030 IF I$="" THEN GOTO 5020
5040 IF I$="C" THEN GOTO 1000
5050 IF I$<>"S" AND I$<>CHR$ 118 THEN GO
     TO 5020
5100 CLS
5110 PRINT N$+"S "+("FINAL " AND I$=CHR$
      118)+"SCORE:"
5115 PRINT "
     "
5120 PRINT ,,"NUMBER OF PROBLEMS: "+STR$
      TP
5130 PRINT ,,"TOTAL CORRECT: "+STR$ TC
5140 PRINT ,,"TOTAL WRONG: "+STR$ TW
5150 PRINT ,,"PERCENT CORRECT: "+STR$ IN
     T ((TC/TP)*100)
5160 PRINT ,,,,"MONEY EARNED: "+STR$ ME
5170 PRINT ,,"MONEY LOST: "+STR$ ML
5180 PRINT ,,"FINES: "+STR$ FI
5190 LET X=ME-ML-FI
5200 PRINT ,,("I OWE YOU" AND X>0)+("YOU
      OWE ME" AND X<0)+" "+STR$ ABS X+"
     CENT"+("S" AND ABS X<>1);"."
5210 IF I$=CHR$ 118 THEN GOTO 5250
5220 PRINT ,,"PRESS ENTER TO CONTINUE "
5230 IF INKEY$="" THEN GOTO 5230
5240 GOTO 1000
5250 PRINT ,,"BYE, "+N$+". I HAD FUN."
9997 STOP
9998 SAVE "MAKING CHANGE"
9999 RUN
```

©

# Relative Files For VIC-20 And Commodore 64
## Part I

Jim Butterfield, Associate Editor

*You can use relative files with your VIC or 64 computer and 1540/1541 disk drive. If you have the appropriate IEEE interface, you can do the same job on the 4040, 8050, or other recent Commodore disk units. It takes a little more work, and careful programming. But it can be done.*

*All the examples given here will work on all PETs and CBMs. On 4.0 BASIC, there are easier ways, but this will work.*

### Binary Numbers: High And Low

We'll be talking about some numbers packed into ASCII characters. In the expression CHR$(N), we can't use a value of N greater than 255. Sometimes we will want to send larger numbers. For example, if we want to select record number 1000 of a relative file, we'll need to split it into two parts. The "high" part would be the number divided by 256; the low part would be the remainder. So a value of 1000 would split up into a high part of 3 and a low part of 232, since 1000 divided by 256 gives 3 with 232 remainder.

When we send a number this way, we almost always send the low part first. So to send 1000, we'll eventually send to the disk:

CHR$(232);CHR$(3).

In Part II, we'll indicate a number that is split in this manner with the terms "High" and "Low."

### Creating A Relative File

Decide how long you want a record to be. For example, you might have a file that will contain a name, a set of initials, and a date. You could allow 15 characters for the name; two characters for the initials; and seven characters for the date. Additionally, you'd need two extra characters as "separators" between the three data fields, giving a total of up to 26 characters in a record. You can go as high as 254, but no higher.

When we create a relative file, we must give the record length. After it is created, we don't need to specify the length: the disk will remember.

Let's open a relative file using direct statements. You can do this in a program, of course; but you may find it interesting to see things happening. First, however, we'll set up a program to allow us to check for errors on the command channel. Enter this program:

```
100 INPUT#15,E,E$,E1,E2
110 PRINT E;E$;E1,E2
```

Now type, as a direct command: OPEN 15,8,15. This will open the command channel for us. Anytime we want to look at a disk error condition, we can type GOTO 100, and the error will be printed.

We're ready to open our relative file. Type:

```
OPEN 1,8,2,"RANDFIL,L,"+CHR$(26)
```

That does the job. The name of the new file is RANDFIL. The L stands for length: don't forget to put a comma both before and after. Finally, the CHR$(26) gives the length of our record. We don't need to use all 26 characters, but we must not exceed this value when we write a record.

### Positioning To A Specific Record

We've created the file, but we have not written any records yet. It's a good idea to bring enough records into existence to fill more than one disk sector, which takes up 254 bytes. In the case of 26-character records, this means that we should create at least ten records.

We could do this with ten PRINT#1 statements, but I'd like to show you another way. Let's position directly to record number 10 and write something there. Automatically, all missing records (in this case, 1 to 9) will come into existence. So we'd better learn how to position a relative file.

Now, we send our "position" command down the command channel. To identify to the disk which file we want to position, we use the secondary address. For our relative file in progress, that would be 2. That's important: secondary address, not logical file number. Now, another thing about the disk: it likes to see you add 96 to the secondary address, so we should send 98.

We have said that we want to go to record number 10. We must split this number up into high and low byte: we get 0 high and 10 low. Finally, we want to choose the start of the record, or position 1. Let's put it all together and type in:

```
PRINT#15,"P"+CHR$(96+2)+CHR$(10)+CHR$
  (0)+CHR$(1)
```

You'll see the disk error light go on – we'll account for this in a moment. To review: P for position; 96+2 for secondary address 2; 10 and 0 for record number 10; and 1 for the start of the record.

Why did the error light go on? Because there is no record number ten – yet. You may type GOTO 100 and look at the error notice: you'll see RECORD NOT PRESENT, which makes sense. The moment we write something, we'll bring this missing record into existence. Let's do that:

```
PRINT#1,"DOAKES"+CHR$(13)+"J"+CHR$(13)+
  "AUG15";
```

We have just written record number ten. Note that we are using a Return character to separate the fields (name, initial, date), and note that the PRINT statement ends with a semicolon. This seems puzzling: it doesn't work that way on sequential files. Let's give the golden rule for writing relative files:

*Rule: One PRINT statement writes one and only one record to a relative file.*

So the semicolon at the end does not change anything: we've written a complete record. And the Return characters in the middle do not change anything: we've written only one record.

Let's tie up this file for the moment. Close it with:

```
CLOSE 1
```

No need to close the command channel.

## After Creation

Let's write a program to read and write this little file that we have created. Here we go:

```
100 OPEN 4,8,5,"RANDFIL"
```

I've changed the logical file number and secondary address just to prove that it doesn't matter. Note that we don't need to specify the length, once the file exists.

```
110 OPEN 15,8,15
```

Now for the main user interface. We'll ask for a record number, and quit if the user types zero:

```
200 INPUT"RECORD NUMBER";R
210 IF R<1 GOTO 600
```

Let's position to the record:

```
220 R0=INT(R/256):R1=R-R0*256
```

R0 is the high part of the number, and R1 is low. Now we can position:

```
230 PRINT#15,"P"+CHR$(101)+CHR$(R1)+CHR$
    (R0)+CHR$(1)
```

We remember that 101 is 96 plus 5. Let's look for an error:

```
240 INPUT#15,E,E$,E1,E1
250 IF E<>0 THEN PRINT E$:GOTO 200
```

We've positioned at a valid record. Now let's ask if we want to read or write:

```
300 INPUT"R OR W";C$
310 IF C$="R" GOTO 400
320 IF C$="W" GOTO 500
330 GOTO 200
```

Now for the reading part. We are already positioned, but first we must learn another golden rule, this one for reading records:

*Rule: Variable ST signals end of record with value 64.*

This, too, is different from sequential files, where ST signals end-of-file. Now we can read however many fields are in the record, since we'll detect the end of record in ST:

```
400 F=0
410 F=F+1:INPUT#4,F$
420 PRINT "FIELD";F;":";F$
430 IF ST=0 GOTO 410
440 GOTO 200
```

Thus, we keep reading until we have gathered all the fields within the record. Now, for the writing part. We've been here before:

```
500 FOR F=1 TO 3
510 PRINT"FIELD";F;
520 INPUT F$(F)
530 NEXT F
540 PRINT#4,F$(1)+CHR$(13)+F$(2)+CHR$(13)+
    F$(3);
550 GOTO 200
```

That's it except for quitting. We must remember to close our file:

```
600 CLOSE 4
```

Try playing with this program. You might like to try for nonexistent records, or writing records that are too large to fit. You'll quickly see how it all works. Note the curiosity: the character "pi," or CHR$(255), is stored in unused records.

It's not too hard and can be very useful. With relative records, you can go directly to any chosen part of your file. You can read or write as desired.

It's another tool for effective use of your computer.

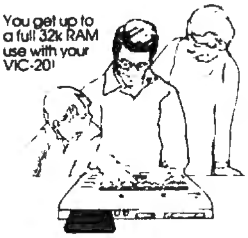
# Maximum Memory=Maximum Application!

Maximum Memory allows you to use more powerful programs for:
- EDUCATION ● ENTERTAINMENT ● MAIL LISTS
- BUSINESS APPLICATIONS ● FINANCIAL RECORDS

APROPOS TECHNOLOGY

## RAMAX™


You get up to a full 32k RAM use with your VIC-20!



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**
**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.

**NEW! 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 compliment of memory!

**SPECIAL $109.95** Shipping Included

To equal the total memory of RAMAX™ you would have to buy Commodore's 16k Memory Expansion, PLUS their 8k Expansion, PLUS their 3k Expansion for a total list price of over $200.00! THEN you would need a "mother board" which would run the total up to OVER $400! 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.
- Built-in switch allows User selection of any combination of 5 areas of RAM memory*:
  - BLK 1 (8k:Adr 8192 16383)
  - BLK 2 (8k Adr 16383-24575)
  - BLK 3 (8k Adr 24576-32767)
  - BLK 5 (Adr 40960-49152, 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!

**Contact Your Local Dealer.**

Or Send Check or Money Order For the Total
Calif. residents add 6% tax.

Phone orders Call **(805) 482-3604**

Foreign orders, add $15.00

All items shipped from stock.

**DEALER INQUIRIES WELCOME**

**WE SERVICE WHAT WE SELL!**

VIC-20 & Commodore-64 are registered trademarks of Commodore International.

**350 N. Lantana Ave., Suite 821**
**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**

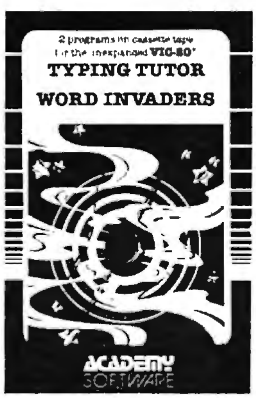
**APROPOS TECHNOLOGY**

---

**FOR THE VIC 20™**

## JOIN THE COMPUTER REVOLUTION WITH A MASTERY OF THE KEYBOARD!

In the age of the computer, *everyone* from the school child to the Chairman of the Board should be at home at the computer keyboard. Soon there will be a computer terminal on every desk and in every home. Learn how to use it right ...and have some fun at the same time!




***Rated THE BEST educational program for the VIC 20™ by Creative Computing Magazine***

## TYPING TUTOR PLUS WORD INVADERS — $21.95

***(2 programs on one cassette tape for the unexpanded VIC 20™)***

Typing Tutor plus Word Invaders makes learning the keyboard easy and fun! Typing Tutor teaches the keyboard in easy steps. Word Invaders makes typing practice an entertaining game. Highly praised by customers:

*"Typing Tutor is great!", "Fantastic", "Excellent", High quality", "A source of great joy and learning for our children", "Even my little sister likes it", "Word Invaders is sensational!"*

**Customer comment says it all . . .**

*". . . and it was everything you advertised it would be. In three weeks, my 13 year old son, who had never typed before, was typing 35 w.p.m. I had improved my typing speed 15 w.p.m. and my husband was able to keep up with his college typing class by practicing at home."*

**FOR THE COMMODORE 64™**

## SPECIAL VERSION OF TYPING TUTOR PLUS WORD INVADERS NOW AVAILABLE FOR THE COMMODORE 64™... $21.95 *(Tape)* $24.95 *(Disk)*

***All of the features of the VIC 20™ Version and more***

**SPECIAL!**

## SPRITE DESIGNER

***by Dr. Lee T. Hill***


**$16.95** *(Tape)* **$21.95** *(Disk)*

Create and then transform sprites automatically. We have the other sprite making programs, but this is the one *we use* to make sprites. The automatic transformations are great!

Shipping and handling $1.00 per order. California residents add 6% sales tax. VISA and Mastercard orders must include full name as shown on card, card number, and expiration date. Free catalog sent with order and on request.

**ACADEMY SOFTWARE**

**P.O. Box 9403, San Rafael, CA 94912 (415) 499-0850**

**Programmers: Write to our New Program Manager concerning any exceptional VIC 20™ or Commodore 64™ game or other program you have developed.**

# Sprite Editor For TI

Larry Long

*Here's a way to get maximum use of sprites on the TI-99/4A – and a program that generates listings for your sprite creations.*

A very powerful yet often unused feature of the TI-99/4A is its ability to display and control sprites. With the 99/4A and the Extended BASIC Module, it is possible to generate 28 sprites for display and independent simultaneous movement. Program 1 should convince any doubters that this can be done. Although a lot of colored letters floating around the screen are a bit pointless, if we can modify and control the sprites, we will have a most useful feature.

Sprites can be designed by drawing on a piece of graph paper and then converting the on/off pixels to a hexadecimal number. If the two largest sizes of sprites are used, the hexadecimal number describing the shape of the sprite would be 64 characters long. A solution is a sprite editor that will allow us to draw the pattern we want on the screen and then have the computer create the program we need to make that sprite pattern. Program 2 will do exactly that, and more. It will allow us to edit the sprite pattern. Then, when we press the L key, it will display a complete listing that would, if copied on paper and then entered into the computer, provide a sprite and the necessary routine to control its movement.

## Your Options

When you run the program, the first display screen will be a design grid with a box-shaped cursor. The area under the cursor will initially be white (signifying an "off" pixel). Press 1 to change the color beneath the cursor to black (representing an "on" pixel) or to move the cursor about the grid with the arrow keys. To turn off a particular pixel, press 0 and the background color will be returned to white. When you have completed your design, press the P key to see it displayed as a sprite.

At this point, you are given several options. You can magnify your newly constructed sprite (M key), change its color (C key), change its background color (B key), or set it in motion (E, S, D, X keys). If you are not pleased with the sprite's shape, you can modify it by striking the T key or (if the changes required are quite drastic) simply press the A key to start with a fresh grid. On the other hand, if you are satisfied with your sprite and its color and directional parameters, press the L key to create the BASIC statements needed to achieve these effects.

If using the sprite editor is your only concern, then skip the rest of this article and go straight to Program 2 and enjoy this easy access to sprites.

## How The Editor Works

To understand what makes the editor work, let's take a general overview of the program:

| Lines | |
|---|---|
| 100-260 | Set up screen display. |
| 270-460 | Are the main loop of the designing portion of the program. |
| 470-680 | Evaluate the design, put its values in an array, read the values in the array, convert them to hexadecimal numbers, and then build a 64-character string to describe the sprite pattern. |
| 690-770 | Put the sprite on the screen and display new program instructions. |
| 780-930 | Main loop of the implementation portion of the program. |
| 940-980 | Change size of sprite. |
| 1000-1150 | Display a listing of the sprite program. |
| 1160-1220 | Change the color of the sprite and screen. |

A cursor is needed to indicate where you are located on the design grid. I chose to use a sprite (line 220) because I could move it around freely without disturbing the display under it. Repositioning the cursor is accomplished in line 380 with a CALL LOCATE. The arrow keys reposition the cursor, and the ENTER key changes the area under the cursor.

What makes "Sprite Editor" so valuable is its ability to generate the hexadecimal pattern for the sprite. The loop from line 500 through line 560 determines the character in each position of the design grid and stores that value in the array B (R,C). Line 570 provides a string with all of the possible hexadecimal digits placed in ascending order. Line 580 sets M$ to be "null." The loop from line 590 to line 630 evaluates the array elements and converts each row in the left half of the design grid to a pair of hexadecimal digits and concatenates them to M$. Line 620 is probably the most significant line in this loop, as it provides the hexadecimal numbers. It causes the computer to look at a particular digit (element) in HEX$ determined by the values calculated for HIGH and LOW. Lines 630-680 perform the same operation as 590-630, only for the right half of the design grid.

Line 690 assigns the hexadecimal numbers to ASCII characters 104, 105, 106, and 107. It is necessary to specify only the first character number in the CALL CHAR statement. When this feature is used, it is required that you start with a character that is evenly divisible by 4. Line 730 actually displays the sprite.

Lines 740-770 provide instructions for the implementation portion of the program. Lines 780-830 check for specific key presses and provide appropriate branching to list the program; end the program; start from the beginning; change the background color; modify the existing sprite; change sprite size; or change sprite color. Lines 840-920 check for arrow key presses and then increment or decrement sprite speed.

Lines 940-980 change sprite size. Lines 1000-1150 display a program listing that would generate a sprite like the one designed by the Sprite Editor. One problem with listing the program is displaying the quote character. The computer interprets it to mean that you want to end the PRINT statement. The solution is to redefine an unused character (I chose the lowercase "n") to look like the quote character.

Finally, lines 1160-1220 allow you to change the color of the sprite and screen.

## **Program 1:** Sprite Generation

```
100 CALL MAGNIFY(2)::FOR X=1 TO 28::
    CALL SPRITE(#X,64+X,X/2,96,128,I
    NT(RND*100)-50,INT(RND*100)-50):
    :NEXT X::GOTO 100
```

## **Program 2:** Sprite Editor

```
100 REM SPRITE EDITOR
110 DIM B(16,16):: SC=1
130 C1=7
140 CALL CHAR(100,"")
150 CALL CHAR(101,"FFFFFFFFFFFFFFFF
    ")
160 CALL CHAR(102,"FFFFC3C3C3C3FFFF
    ")
170 CALL COLOR(9,2,16)
180 CALL CLEAR
190 DISPLAY AT(1,10):"SPRITE EDITOR"
200 FOR R=1 TO 16 :: CALL HCHAR(4+R
    ,2,100,16):: NEXT R
210 CALL MAGNIFY(1)
212 IF K=84 THEN GOTO 217
215 CALL SCREEN(8)
217 CALL DELSPRITE(ALL)
220 CALL SPRITE(#28,102,14,32,8)
225 CALL HCHAR(21,1,32,31):: CALL H
    CHAR(22,1,32,31)
230 DISPLAY AT(22,2):"E=UP X=DOWN S
    =LEFT D=RIGHT"
240 DISPLAY AT(23,2):"PRESS 1 - PIX
    EL ON ,0 - OFF"
250 DISPLAY AT(24,2):"PRESS P TO DI
    SPLAY SPRITE"
260 R=1 :: C=1
270 CALL KEY(0,K,S)
271 IF S=0 THEN 270
272 IF K=48 THEN KHAR=100
274 IF K=49 THEN KHAR=101
280 IF K=83 THEN C=C-1 :: GOTO 320
290 IF K=68 THEN C=C+1 :: GOTO 320
300 IF K=69 THEN R=R-1 :: GOTO 320
310 IF K=88 THEN R=R+1 :: GOTO 320
312 IF K=80 THEN 470
320 IF C<1 THEN C=16
330 IF C>16 THEN C=1
340 IF R<1 THEN R=16
350 IF R>16 THEN R=1
380 CALL LOCATE(#28,(8*R)+25,8*C+1)
420 CALL HCHAR(4+R,1+C,KHAR)
430 CALL SOUND(20,200,5)
460 GOTO 270
470 CALL DELSPRITE(ALL)
480 CALL HCHAR(21,1,32,128)
490 DISPLAY AT(22,2):"PLEASE WAIT W
    HILE I THINK."
500 FOR R=1 TO 16
510 FOR C=1 TO 16
520 CALL GCHAR(4+R,1+C,GC)
530 GC=GC-100
540 B(R,C)=GC
550 NEXT C
560 NEXT R
570 HEX$="0123456789ABCDEF"
580 M$=""
590 FOR R=1 TO 16
600 LOW=B(R,5)*8+B(R,6)*4+B(R,7)*2+
    B(R,8)+1
610 HIGH=B(R,1)*8+B(R,2)*4+B(R,3)*2
    +B(R,4)+1
620 M$=M$&SEG$(HEX$,HIGH,1)&SEG$(HE
    X$,LOW,1)
630 NEXT R
640 FOR R=1 TO 16
650 LOW=B(R,13)*8+B(R,14)*4+B(R,15)
    *2+B(R,16)+1
660 HIGH=B(R,9)*8+B(R,10)*4+B(R,11)
    *2+B(R,12)+1
670 M$=M$&SEG$(HEX$,HIGH,1)&SEG$(HE
    X$,LOW,1)
680 NEXT R
690 CALL CHAR(104,M$)
700 CALL MAGNIFY(3)
710 MM=3
720 M=4
730 CALL SPRITE(#1,104,C1,50,170,0,
    0)
740 DISPLAY AT(21,2):"C COLOR  M MA
    GNIFY  T EDIT"
750 DISPLAY AT(22,2):"A ERASE  Q QU
    IT  B BACKGRD"
```

```
760 DISPLAY AT(23,2):"E=UP X=DOWN S=LEFT D=RIGHT"
770 DISPLAY AT(24,8):"L LISTS PROGRAM"
780 CALL KEY(0,K,S)
790 IF K=76 THEN GOTO 1000
800 IF K=81 THEN GOTO 990
810 IF K=65 THEN GOTO 100
812 IF K=66 THEN GOSUB 1200
815 IF K=84 THEN GOTO 210
820 IF K=77 THEN GOTO 940
830 IF K=67 THEN GOTO 1160
840 IF K=83 THEN H=H-2
850 IF K=68 THEN H=H+2
860 IF K=69 THEN V=V-2
870 IF K=88 THEN V=V+2
880 IF V>120 THEN V=120
890 IF V<-120 THEN V=-120
900 IF H>120 THEN H=120
910 IF H<-120 THEN H=-120
920 CALL MOTION(#1,V,H)
930 GOTO 780
940 CALL MAGNIFY(M)
950 MM=M
960 IF M=3 THEN M=4 ELSE M=3
970 FOR D=1 TO 20 :: NEXT D
980 GOTO 780
990 STOP
1000 REM PROGRAM LISTER
1010 CALL CHAR(110,"002424")
1020 CALL CLEAR
1030 PRINT "{6 SPACES}PROGRAM LISTING"
1035 CALL DELSPRITE(ALL)
1040 PRINT
1050 PRINT ">100 CALL CHAR(104,n";::: FOR W=1 TO 64 :: PRINT SEG$(M$,W,1);:: NEXT W :: PRINT "n)"
1055 PRINT ">105 CALL SCREEN(";SC;")"
1060 PRINT ">110 CALL MAGNIFY(";MM;")"
1070 PRINT ">120 CALL SPRITE(#1,104,";C1;",150,150,";V;",";H;")"
1080 PRINT ">130 CALL KEY(0,K,S)"
1090 PRINT ">140 IF K=68 THEN H=H+2"
1100 PRINT ">150 IF K=83 THEN H=H-2"
1110 PRINT ">160 IF K=88 THEN V=V+2"
1120 PRINT ">170 IF K=69 THEN V=V-2"
1130 PRINT ">180 CALL MOTION(#1,V,H)"
1140 PRINT ">190 GOTO 130"
1150 PRINT :: PRINT :: PRINT :: PRINT :: PRINT
1155 DISPLAY AT(21,3):"A - ERASE{3 SPACES}Q - QUIT"
1156 CALL KEY(0,K,ST):: IF ST=0 THEN 1156
1157 IF K=81 THEN GOTO 990
1158 IF K=65 THEN GOTO 100
1159 GOTO 1156
1160 C1=C1+1 :: IF C1>16 THEN 1180
1170 CALL COLOR(#1,C1):: GOTO 780
1180 C1=2 :: CALL COLOR(#1,C1):: GOTO 780
1200 REM SCREEN COLOR CHANGE
1210 SC=SC+1 :: IF SC=17 THEN SC=2
1220 CALL SCREEN(SC):: RETURN
```

# Atari Menu Buttons

Joseph D. Korman

*This utility streamlines the menu selection process by using the OPTION, SELECT, and START keys. The resulting program can be stored on tape or disk and can then be used as the beginning of new programs requiring menu selections.*

---

After catching the programming bug and purchasing an Atari 800, I began to write custom programs for home use. These included checking account, household inventory, telephone book, and the like. In all the programs, the menus ended with an input statement requesting the code for the desired choice. For example:

D. ENTER DEPOSIT
C. ENTER CHECKS
L. LIST CHECKS
S. SAVE REVISED DATA
ENTER NEXT FUNCTION:?

After input of the variable, the program would run a series of IF tests to determine the choice and proceed to the indicated line number for execution. Although the programs worked well, I felt that something was missing to streamline the selection process. I found the answer in James Brunn's article in *COMPUTE!'s First Book of Atari*. The article included information about using the OPTION / SELECT / START buttons on the 800 keyboard.

The menu create utility is actually a skeleton of a program designed to let the user move the cursor to each of the menu options by pressing the OPTION key. Once the cursor is at the line of the desired option; the SELECT key is used to move the program execution to the appropriate line. After the skeleton is loaded, the titles and option names should be changed to reflect the requirements of the new program. After this is done, the programmer need only enter the logic of the options and commands to return to the menu after their execution.

The following program provides ten options starting on line five (5) and printing on each odd line. The user may add more selections to this column and may add a second column to the right side of the screen. If this is done, some changes in the cursor movement logic will be required. This will allow the user to make truly custom menus for the Atari 400/800 programs.

The -] is printed in reverse mode and moves down each time the OPTION button is pressed.

The menu created by this program looks like this on the screen:

"TITLE OF MENU"
-] ITEM 1
ITEM 2
ITEM 3
ITEM 4
ITEM 5
ITEM 6
ITEM 7
ITEM 8
ITEM 9
ITEM 10

Each time the OPTION button is pressed, the arrow moves down the menu one position. Holding the OPTION button causes the arrow to continuously move from top to bottom and jump back to the top. The operator releases the button when the arrow is adjacent to the desired option. The SELECT button is then pressed to execute that part of the program.

### Menu Buttons

```
1 REM MENU CREATE UTILITY
10 GRAPHICS Ø:SETCOLOR 2,2,8:SETCOLO
   R 1,2,Ø
11 POKE 752,1
20 POSITION 12,1:? "TITLE OF MENU"
21 POSITION 2,3:? "{3 SPACES}ITEM 1"
22 POSITION 2,5:? "{3 SPACES}ITEM 2"
23 POSITION 2,7:? "{3 SPACES}ITEM 3"
24 POSITION 2,9:? "{3 SPACES}ITEM 4"
25 POSITION 2,11:? "{3 SPACES}ITEM 5
   "
26 POSITION 2,13:? "{3 SPACES}ITEM 6
   "
27 POSITION 2,15:? "{3 SPACES}ITEM 7
   "
28 POSITION 2,17:? "{3 SPACES}ITEM 8
   "
29 POSITION 2,19:? "{3 SPACES}ITEM 9
   "
30 POSITION 2,21:? "{3 SPACES}ITEM 1
   Ø"
31 REM POSITION ENTRIES ON ALL LINES
    TO INCREASE THE NUMBER OF SELECT
   IONS
35 POSITION 2,3:L=3:? "->"
40 IF PEEK(53279)=3 THEN 5Ø
41 IF PEEK(53279)=5 THEN 6Ø
42 IF PEEK(53279)=6 THEN RUN
43 GOTO 4Ø
50 REM MENU SCROLL
51 POSITION 2,L
52 ? "  "
```

```
53 L=L+2:REM USE L+1 IF ENTRIES ARE
   ON ALL LINES
54 IF L=23 THEN L=3
55 POSITION 2,L:REM FOR TWO COLUMN M
   ENU POSITION 21,L AND ADD LOGIC T
   O RETURN TO LEFT COLUMN FROM BOT
   RT
56 ? "=>":FOR T=1 TO 40:NEXT T:REM U
   SE HIGHER NUMBER TO SLOW =>
57 GOTO 40
60 REM SELECT OPTION
61 IF L=3 THEN 100
62 IF L=5 THEN 200
63 IF L=7 THEN 300
64 IF L=9 THEN 400
65 IF L=11 THEN 500
66 IF L=13 THEN 600
67 IF L=15 THEN 700
68 IF L=17 THEN 800
69 IF L=19 THEN 900
70 IF L=21 THEN 1000
71 IF L=23 THEN L=3:GOTO 61
72 REM ADJUST THE ABOVE LOGIC FOR SI
   NGLE LINE SELECTIONS AND DUAL COL
   UMN MENUS
100 GRAPHICS 0:SETCOLOR 2,1,2:SETCOL
    OR 1,1,8
110 POSITION 12,1:? "ITEM NUMBER 1"
115 REM PUT LOGIC FOR THE SELECTION
    HERE
116 REM DON'T FORGET LOGIC TO RETURN
     TO THE MAIN MENU AFTER THE SELE
    CTION IS COMPLETED
117 REM CONTINUE FOR ALL OTHER SELEC
    TIONS
120 FOR T=1 TO 500:NEXT T:RUN
200 GRAPHICS 0:SETCOLOR 2,8,2:SETCOL
    OR 1,8,8
210 POSITION 12,1:? "ITEM NUMBER 2"
220 FOR T=1 TO 500:NEXT T:RUN
300 GRAPHICS 0:SETCOLOR 2,8,8:SETCOL
    OR 1,8,2
310 POSITION 12,1:? "ITEM NUMBER 3"
320 FOR T=1 TO 500:NEXT T:RUN
400 GRAPHICS 0:SETCOLOR 2,4,8:SETCOL
    OR 2,4,2
410 POSITION 12,1:? "ITEM NUMBER 4"
420 FOR T=1 TO 500:NEXT T:RUN
500 GRAPHICS 0:SETCOLOR 2,11,8:SETCO
    LOR 2,11,2
510 POSITION 12,1:? "ITEM NUMBER 5"
520 FOR T=1 TO 500:NEXT T:RUN
600 GRAPHICS 0:SETCOLOR 2,1,2:SETCOL
    OR 1,1,8
610 POSITION 12,1:? "ITEM NUMBER 6"
620 FOR T=1 TO 500:NEXT T:RUN
700 GRAPHICS 0:SETCOLOR 2,8,2:SETCOL
    OR 1,8,8
710 POSITION 12,1:? "ITEM NUMBER 7"
715 IF L=23 THEN L=3:GOTO 61
720 FOR T=1 TO 500:NEXT T:RUN
800 GRAPHICS 0:SETCOLOR 2,8,8:SETCOL
    OR 1,8,2
810 POSITION 12,1:? "ITEM NUMBER 8"
820 FOR T=1 TO 500:NEXT T:RUN
900 GRAPHICS 0:SETCOLOR 2,4,8:SETCOL
    OR 2,4,2
910 POSITION 12,1:? "ITEM NUMBER 9"
920 FOR T=1 TO 500:NEXT T:RUN
1000 GRAPHICS 0:SETCOLOR 2,11,8:SETC
     OLOR 2,11,2
1010 POSITION 12,1:? "ITEM NUMBER 10
     "
1020 FOR T=1 TO 500:NEXT T:RUN
```

©

# All About The Hardware Interrupt

Peter Marcotty

*Using the hardware interrupt vector is not something that you can learn by reading a user's manual. This article defines it and discusses how to use it in your machine language programs.*

An interrupt is a hardware event. Every 60th of a second, a clock inside the computer causes a change in voltage on one of the pins of the 6502 chip (6510 if you have a 64). This change tells the 6502 to stop (interrupt) whatever it is doing, remember how to get back to it, and go to the machine language program pointed to by the hardware interrupt vector (an address inside the computer which *points to* the address of a machine language program that normally "services" the interruption).

Usually the vector sends the computer to a program that updates the screen, looks at the keyboard, and changes the value of TI$. (This is the "servicing.") No matter what you are doing in BASIC or machine language, the interrupt will happen 60 times a second unless you specifically turn it off.

Perhaps the most effective use of the interrupt is that you can wedge a routine of your own into the process, before it goes off to its regular housekeeping chores. Simply point the interrupt vector to the beginning of your routine, do whatever you want to do, and then send the computer to where it usually goes.

In order for us to change the interrupt vector, we must stop the hardware interrupt action altogether. If it tried to jump to the location pointed to by the interrupt vector, and we had changed only one byte of the two-byte vector (remember, interrupts can happen at any time), we'd get some very undesirable results.

## Implementing The Interrupt

It will be helpful if you refer to the program for your machine while reading this section.

The first line of your program should be the SEI command. SEI stands for SEt Interrupt mask, and it will stop the computer from interrupting until you let it. After an SEI, you have about 0.009 seconds to change the interrupt vector before the computer gets impatient and crashes. Fortunately, this is plenty of time for our purposes. The next four lines take the address of our program (both the low and high byte) and put them in the hardware interrupt vector. Next we have a CLI (CLear Interrupt mask) which tells the computer it can start performing interrupts again. Finally, we have an RTS command which returns us to BASIC.

The program does not finish running with the RTS command; in fact, it's only just beginning. Since the hardware interrupt vector now points to our own routine, every 60th of a second our main program will be run, almost without any delay in whatever else we might be doing.

At the end of the routine that does the actual work, we cannot return from wherever we were called with a simple RTS. The screen has yet to be updated, and the keyboard hasn't been checked to see if any keys are down. We must JMP to the location where the vector usually points. That's where the servicing routine resides. The locations of the hardware interrupt vector for various computers are given in the table.

The sample program should help you understand how your interrupt routines must be set up.

To turn off your interrupt-driven program, you can change the pointer back to its original value, or on the VIC and 64, simply hit RUN/STOP and RESTORE.

The example programs simply take a look at the contents of the memory location that shows what key is currently being pressed and puts it in the top left corner of the screen.

Two programs are given for each machine. The first can be typed in with an assembler, and the second is a hexadecimal dump to be entered with a monitor. Both have exactly the same effects. To RUN the programs on a PET, type SYS 826; on a VIC or 64, SYS 828. The programs are located in the second cassette buffer, a 192-byte-area of memory that is usually safe for small machine language programs.

Note that interrupt-driven programs will interfere with the normal operation of LOAD and SAVE commands.

### Interrupt Memory Locations

*The hardware interrupt on the 64 and VIC works in exactly the same way as on the PET, although memory locations will be different.*

*This table shows all the differences:*

| | Location of Hardware Interrupt Vector | | Points to |
|---|---|---|---|
| Upgrade PETs | 144-145 | ($90-$91) | $E62E |
| 4.0 PETs | 144-145 | ($90-$91) | $E455 |
| 64 | 788-789 | ($314-$315) | $EA31 |
| VIC | 788-789 | ($315-$315) | $EABF |

### Program 4: Hardware Interrupt Routine – VIC Version

```
2
4:      033C                    .OPT P4,00
6:      033C                    *=   $033C
                        ;VIC VERSION
10:     033C 78                 SEI           ;DISABLE INTERRUPTS
20:     033D A9 49              LDA  #$49     ;LOAD LOW BYTE OF ROUTINE IN LINE 80
30:     033F 8D 14 03           STA  $314     ;STORE LO BYTE OF INTERRUPT VECTOR
40:     0342 A9 03              LDA  #$03     ;LOAD HI BYTE OF ROUTINE IN LINE 80
50:     0344 8D 15 03           STA  $315     ;STORE HI BYTE OF INTERRUPT VECTOR
60:     0347 58                 CLI           ;REENABLE INTERRUPT
70:     0348 60                 RTS           ;RETURN
80:     0349 A5 CB              LDA  203      ;LOAD CURRENT KEY PRESSED
90:     034B 8D 00 1E           STA  $1E00    ;STORE IT ON THE SCREEN
95:     034E A9 00              LDA  #0
97:     0350 8D 00 96           STA  $9600    ;SET COLOR TO BLACK
1000:   0353 4C BF EA           JMP  $EABF

C*
     PC   IRQ  SR AC XR YR SP
.;  B780 E455 2C 34 3A 9D FA
.
.:  033C 78 A9 49 8D 14 03 A9 03
.:  0344 8D 15 03 58 60 A5 CB 8D
.:  034C 00 1E A9 00 8D 00 96 4C
.:  0354 BF EA 49 56 2E 36 34 2E
```

# Interrupt Applications

Eric Brandon, Editorial Programmer

Interrupts can be used in many different applications, but the two most common are within utilities and games.

Because an interrupt-driven program is in the "background" of whatever the user is doing, it is ideal for applications where we want to do something concurrent with the normal operation of the computer. Good examples of this are found in "Marquee" (COMPUTE!, February 1981), which displays a message across the top of the screen as a sort of electronic "string around your finger," and "Realtime Clock" (COMPUTE!, January 1982), which displays the time in a corner of the screen to remind you to stop playing Alien Zap and go to bed.

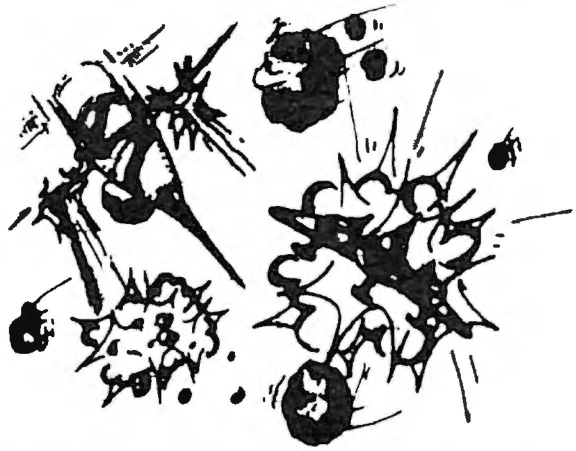
Other uses for interrupt-driven utilities are programs which constantly check which keys are pressed and act accordingly. My favorite from this class is "Keyprint" (COMPUTE!, November/December 1980). Whenever you hit the & and the shift simultaneously, the computer freezes and sends whatever is on the screen to the printer.

In games, interrupts can be used for convenience or smoothness. Suppose you want to write a space game which has a moving starfield in the background. You could worry about writing a program which simultaneously moves your spaceship, the mutant ants, *and* the starfield around, or you could use interrupts. It is a simple matter to write a routine which moves some stars around and to point the interrupt vector to it. Now, you can write your game safe in the knowledge that whatever is going on in your program, those stars will keep floating by.

Best of all, when something holds up your main program for a second or two, such as a sound effect or an explosion, the background won't freeze up but will keep moving, making your game look "smoother" and more professional.

# 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

---

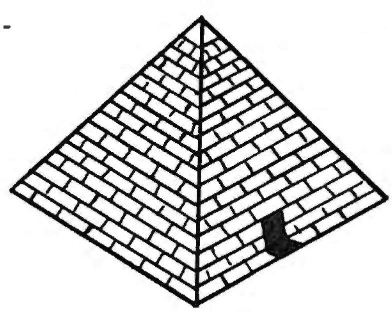
## SHROOMS

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
D E S
Data Equipment Supply Corp.
8315 Firestone Blvd., Downey, CA 90241
(213) 923-9361

## **Program 1:** Hardware Interrupt Routine – 4.0 BASIC Version (4032, 8032)

```
2
4:      033A                      .OPT P4,OO
6:      033A                      *=   $033A
                  ;PET 4.0 VERSION
10:     033A 78                   SEI        ;DISABLE INTERRUPTS
20:     033B A9 45                LDA  #$45  ;LOAD LOW BYTE OF ROUTINE IN LINE 80
30:     033D 85 90                STA  $90   ;STORE LO BYTE OF INTERRUPT VECTOR
40:     033F A9 03                LDA  #$03  ;LOAD HI BYTE OF ROUTINE IN LINE 80
50:     0341 85 91                STA  $91   ;STORE HI BYTE OF INTERRUPT VECTOR
60:     0343 58                   CLI        ;REENABLE INTERRUPT
70:     0344 60                   RTS        ;RETURN
80:     0345 A5 97                LDA  151   ;LOAD CURRENT KEY PRESSED
90:     0347 8D 00 80             STA  $8000 ;STORE IT ON THE SCREEN
1000:   034A 4C 55 E4             JMP  $E455

C*
    PC   IRQ  SR AC XR YR SP
.;  B780 E455 2C 34 3A 9D F8
.
.:  033A 78 A9 45 85 90 A9 03 85
.:  0342 91 58 60 A5 97 8D 00 80
.:  034A 4C 55 E4 30 2C 30 30 3F
```

## **Program 2:** Hardware Interrupt Routine – Upgrade ROM Version (3016, 3032)

```
2
4:      033A                      .OPT P4,OO
6:      033A                      *=   $033A
                  ;PET UPGRADE (2.0) VERSION
10:     033A 78                   SEI        ;DISABLE INTERRUPTS
20:     033B A9 45                LDA  #$45  ;LOAD LOW BYTE OF ROUTINE IN LINE 80
30:     033D 85 90                STA  $90   ;STORE LO BYTE OF INTERRUPT VECTOR
40:     033F A9 03                LDA  #$03  ;LOAD HI BYTE OF ROUTINE IN LINE 80
50:     0341 85 91                STA  $91   ;STORE HI BYTE OF INTERRUPT VECTOR
60:     0343 58                   CLI        ;REENABLE INTERRUPT
70:     0344 60                   RTS        ;RETURN
80:     0345 A5 97                LDA  151   ;LOAD CURRENT KEY PRESSED
90:     0347 8D 00 80             STA  $8000 ;STORE IT ON THE SCREEN
1000:   034A 4C 2E E6             JMP  $E62E

C*
    PC   IRQ  SR AC XR YR SP
.;  B780 E455 2C 34 3A 9D FA
.
.:  033A 78 A9 45 85 90 A9 03 84
.:  0342 91 58 60 A5 97 8D 00 80
.:  034A 4C 2E E6 30 2C 30 30 3F
```

## **Program 3:** Hardware Interrupt Routine – 64 Version

```
2
4:      033C                      .OPT P4,OO
6:      033C                      *=   $033C
                  ;64 VERSION
10:     033C 78                   SEI        ;DISABLE INTERRUPTS
20:     033D A9 49                LDA  #$49  ;LOAD LOW BYTE OF ROUTINE IN LINE 80
30:     033F 8D 14 03             STA  $314  ;STORE LO BYTE OF INTERRUPT VECTOR
40:     0342 A9 03                LDA  #$03  ;LOAD HI BYTE OF ROUTINE IN LINE 80
50:     0344 8D 15 03             STA  $315  ;STORE HI BYTE OF INTERRUPT VECTOR
60:     0347 58                   CLI        ;REENABLE INTERRUPT
70:     0348 60                   RTS        ;RETURN
80:     0349 A5 CB                LDA  203   ;LOAD CURRENT KEY PRESSED
90:     034B 8D 00 04             STA  $0400 ;STORE IT ON THE SCREEN
95:     034E A9 01                LDA  #1
97:     0350 8D 00 D8             STA  $D800 ;SET COLOR TO WHITE
1000:   0353 4C 31 EA             JMP  $EA31

C*
    PC   IRQ  SR AC XR YR SP
.;  B780 E455 2C 34 3A 9D FA
.
.:  033C 78 A9 49 8D 14 03 A9 03
.:  0344 8D 15 03 58 60 A5 CB 8D
.:  034C 00 04 A9 01 8D 00 D8 4C
.:  0354 31 EA 49 56 2E 34 2E 30
```

# Cracking The Kernal

Peter Marcotty

*What is the 64 Kernal? How is it available and how do you use it? This article answers these questions and summarizes each of the Kernal's routines – a real machine language programmer's aid.*

What if you want to write a machine language (ML) program for the Commodore 64 that uses the disk drive? Or what if you would like to have your ML program print out to the printer? Where do you begin?

First of all, when you're writing ML programs, it is often helpful to use the routines that are already part of the computer's operating system. But sometimes these routines are buried in ROM among countless other things and they can seem impossible to find. For Commodore 64 users, the Kernal simplifies the search. The Kernal *is* the 64 operating system and contains a collection of extremely useful subroutines that are often quite easy to use.

The wonderful thing about these routines is the incredibly simple way to communicate with them and the powerful results of such brief programming. Often all that is necessary to utilize the subroutine is to load the accumulator (LDA) with one number. Occasionally, a routine will call for another preparatory subroutine to be called first, but these setup routines are just as easy to use.

Using the Kernal involves just these three simple steps: 1) setting up, 2) calling the routine, and 3) handling any errors.

**User Callable Kernal Routines**

| Name | Address Hex | Address Decimal | Function |
|---|---|---|---|
| ACPTR | $FFA5 | 65445 | Input byte from serial port. |
| CHKIN | $FFC6 | 65478 | Open channel for input. |
| CHKOUT | $FFC9 | 65481 | Open channel for output. |
| CHRIN | $FFCF | 65487 | Input character from channel. |
| CHROUT | $FFD2 | 65490 | Output character to channel. |
| CIOUT | $FFA8 | 65448 | Output byte to serial port. |
| CINT | $FF81 | 65409 | Initialize screen editor. |
| CLALL | $FFE7 | 65511 | Close all channels and files. |
| CLOSE | $FFC3 | 65475 | Close a specified logical file. |
| CLRCHN | $FFCC | 65484 | Close input and output channels. |
| GETIN | $FFE4 | 65508 | Get character from keyboard buffer. |
| IOBASE | $FFF3 | 65523 | Return base address of I/O devices. |
| IOINIT | $FF84 | 65412 | Initialize input/output. |
| LISTEN | $FFB1 | 65457 | Command devices on serial bus to LISTEN. |
| LOAD | $FFD5 | 65493 | Load RAM from a device. |
| MEMBOT | $FF9C | 65436 | Read/set bottom of memory. |
| MEMTOP | $FF99 | 65433 | Read/set top of memory. |
| OPEN | $FFC0 | 65472 | Open a logical file. |
| PLOT | $FFF0 | 65520 | Read/set X,Y cursor position. |
| RAMTAS | $FF87 | 65415 | Initialize RAM, reset tape buffer. |
| RDTIM | $FFDE | 65502 | Read realtime clock. |
| READST | $FFB7 | 65463 | Read I/O status word. |
| RESTOR | $FF8A | 65418 | Restore I/O default vectors. |
| SAVE | $FFD8 | 65496 | Save RAM to device. |
| SCNKEY | $FF9F | 65439 | Scan keyboard. |
| SCREEN | $FFED | 65517 | Return X,Y organization of screen. |
| SECOND | $FF93 | 65427 | Send secondary address after LISTEN. |
| SETLFS | $FFBA | 65466 | Set logical, first, and second address. |
| SETMSG | $FF90 | 65424 | Control Kernal messages. |
| SETNAM | $FFBD | 65469 | Set filename. |
| SETTIM | $FFDB | 65499 | Set realtime clock. |
| SETTMO | $FFA2 | 65442 | Set time-out on serial bus. |
| STOP | $FFE1 | 65505 | Check for STOP key. |
| TALK | $FFB4 | 65460 | Command serial bus device to TALK. |
| TKSA | $FF96 | 65430 | Send secondary address after TALK. |
| UDTIM | $FFEA | 65514 | Increment realtime clock. |
| UNLSN | $FFAE | 65454 | Command serial bus to UNLISTEN. |
| UNTLK | $FFAB | 65451 | Command serial bus to UNTALK. |
| VECTOR | $FF8D | 65421 | Read/set vectored I/O. |

Here is a brief summary of each routine with examples:

**ACPTR** is used to get data off the serial bus. TALK and TKSA must be called first.

```
; Get a byte from the serial bus.
JSR  ACPTR
STA $0800
;This example only shows the end result; call TALK
   and TKSA first.
```

**CHKIN** is used to define any OPENed file as an input file. OPEN must be called first.

```
; Define logical file #2 as an input channel.
LDX #2
JSR  CHKIN
; The X register designates which file #.
```

**CHKOUT.** Just like CHKIN, but it defines the file for output. OPEN must be called first.

```
; Define logical file #4 as an output file.
LDX #4
JSR  CHKOUT
; Once again the X register defines the file #.
```

**CHRIN** will get a character from the current input device. Calling OPEN and CHKIN can change the input device.

```
 ; Store a typed string to the screen.
        LDY #$00
LOOP JSR CHRIN
        STA $0800,Y
        INY
        CMP #$0D
        BNE LOOP
        RTS
 ; This example is like an INPUT statement. Try
   running it.
```

**CHROUT.** Load the accumulator with your number and call. OPEN and CHKOUT will change the output device.

```
; Duplicate the command of CMD 4:PRINT "A";
LDX #4
JSR  CHKOUT
LDA #'A
JSR  CHROUT
RTS
; The letter A is printed to the screen; call OPEN first
for the printer.
```

**CIOUT** will send data to the serial bus. LISTEN and SECOND must be called first. Call UNLSN to finish up neatly.

```
; Send the letter X to the serial bus.
LDA #'X
JSR  CIOUT
RTS
; The accumulator is used to transfer the data.
```

**CINT** resets the 6567 video controller chip and the screen editor.

```
; Reset the 6567 chip and the 6566 VIC chip.
JSR CINT
RTS
; Basically, just like pressing the STOP and RESTORE
  keys.
```

**CLALL** really does what its name implies – it closes all files and resets all channels.

```
; Close all files.
JSR CLALL
RTS
; The CLRCHN routine is called automatically.
```

**CLOSE.** This routine will CLOSE any logical file that has been OPENed.

```
Close logical file #2.
LDA #2
JSR  CLOSE
; The accumulator designates the file #.
```

**CLRCHN** resets all channels and I/O registers – the input to keyboard and the output to screen.

```
; Restore default values to I/O devices.
JSR CLRCHN
RTS
; The accumulator and the X register are altered.
```

**GETIN** will get one piece of data from the input device. OPEN and CHKIN can be used to change the input device.

```
  ; Wait for a key to be pressed.
WAIT JSR GETIN
       CMP #0
       BEQ WAIT
  ; If the serial bus is used, then all registers are altered.
```

**IOBASE** returns the low and high bytes of the starting address of the I/O devices in the X and Y registers.

```
; Set the Data Direction Register of the user port to 0
  (input).
JSR  IOBASE
STX  POINT
STY  POINT+1
LDY  #2
LDA  #0
STA  (POINT),Y
; POINT is a zero-page address used to access the DDR
  indirectly.
```

**IOINIT** initializes all I/O devices and routines. It is part of the system's powering-up routine.

```
; Initialize all I/O devices.
JSR  IOINIT
RTS
; All registers are altered.
```

**LISTEN** will command any device on the serial bus to receive data.

```
; Command device #8 to listen.
LDA #8
JSR
LISTEN
; The accumulator designates the device #.
```

**LOAD.** The computer will perform either the LOAD or the VERIFY command. If the accumulator is a 1, then LOAD; if 0, then verify.

```
; Load a program into memory.
LDA  #$08
LDX  #$02
LDY  #$00
JSR  SETLFS
LDA  #$04
LDX  #L,NAME
LDY  #H,NAME
JSR  SETNAM
LDA  #$00
LDY  #$20
JSR  LOAD
RTS
```

```
      NAME .BY 'FILE'
      ; Program 'FILE' will be loaded into memory starting at
      8192 decimal, X being the low byte and Y being the
      high byte for the load.
```

**MEMBOT.** If the carry bit is set, then the low byte and the high byte of RAM are returned in the X and Y registers. If the carry bit is clear, the bottom of RAM is set to the X and Y registers.

```
; Move bottom of memory up one page.
SEC
JSR   MEMBOT
INY
CLC
JSR   MEMBOT
RTS
; The accumulator is left alone.
```

**MEMTOP.** Same principle as MEMBOT, except the top of RAM is affected.

```
; Protect 1K of memory from BASIC.
SEC
JSR   MEMTOP
DEY
CLC
JSR   MEMTOP
; The accumulator is left alone.
```

**OPEN.** After SETLFS and SETNAM have been called, you can OPEN a logical file.

```
 ; Duplicate the command OPEN 15,8,15,'I/O'
 LDA  #3
 LDX  #L,NAME
 LDY  #H,NAME
 JSR  SETNAM
 LDA  #15
 LDX  #8
 LDY  #15
 JSR  SETLFS
 JSR  OPEN
 RTS
NAME .BY 'I/O'
 ; OPEN opens the current name file with the current LFS.
```

**PLOT.** If the carry bit of the accumulator is set, then the cursor X,Y is returned in the Y and X registers. If the carry bit is clear, then the cursor is moved to X,Y as determined by the Y and X registers.

```
; Move cursor to row 12, column 20 (12,20).
LDX  #12
LDY  #20
CLC
JSR  PLOT
; The cursor is now in the middle of the screen.
```

**RAMTAS** is used to test RAM, reset the top and bottom of memory pointers, clear $0000 to $0101 and $0200 to $03FF, and set the screen memory to $0400.

```
; Do RAM test.
JSR   RAMTAS
RTS
; All registers are altered.
```

**RDTIM.** Locations 160-162 are transferred, in order, to the Y and X registers and the accumulator.

```
; Store system clock to screen.
JSR   RDTIM
STA   1026
STX   1025
STY   1024
; The system clock can be translated as hours/minutes/
seconds.
```

**READST.** When called, READST returns the status of the I/O devices. Any error code can be translated as operator error.

```
; Check for read error.
JSR   READST
CMP  #16
BEQ  ERROR
; In this case, if the accumulator is 16, a read error
  occurred.
```

**SCREEN** returns the number of columns and rows the screen has in the X and Y registers.

```
; Determine the screen size.
JSR   SCREEN
STX   MAXCOL
STY   MAXROW
RTS
; SCREEN allows further compatibility between the 64,
  the VIC-20, and future versions of the 64.
```

**SECOND.** After LISTEN has been called, a SECONDary address may be sent.

```
; Address device #8 with secondary address #15.
LDA  #8
JSR   LISTEN
LDA  #15
JSR   SECOND
; The accumulator designates the address number.
```

**SETLFS** stands for SET Logical address, File address, and Secondary address. After SETLFS is called, OPEN may be called.

```
; Set logical file #1, device #8, secondary address of 15.
LDA   #1
LDX   #8
LDY   #15
JSR   SETLFS
; If OPEN is called, the command will be OPEN 1,8,15.
```

**SETMSG.** Depending on the accumulator, either error messages, control messages, or neither is printed.

```
; Turn on control messages.
LDA  #$40
JSR   SETMSG
RTS
; A 128 is for error messages; a zero, for turning both
  off.
```

**SETNAM.** In order to access the OPEN, LOAD, or SAVE routines, SETNAM must be called first.

```
 ; SETNAM will prepare the disk drive for 'FILE#1'.
 LDA  #6
 LDX  #L,NAME
 LDY  #H,NAME
 JSR  SETNAM
NAME .BY 'FILE#1'
 ; Accumulator is file length, X is low byte, and Y is high
  byte.
```

# Using The Kernal From BASIC

Charles Brannon, Program Editor

Surprisingly, the BASIC programmer will find little use for the Commodore 64's powerful Kernal structure. The Kernal is a collection of machine language modules. Kernal routines exist for OPENing files, reading or writing data, checking the keyboard, testing memory, and reading system variables. All these routines are easily available as BASIC commands, such as OPEN, PRINT, INPUT, GET, FRE(0), etc. You, as a BASIC programmer, have a wealth of such powerful and easy-to-use commands.

When you begin to work with machine language, however, you'll discover that there are no built-in "commands" in the 6502 microprocessor for doing all these tasks. The 6502 (the Commodore 64's 6510 processor is functionally identical) deals with very small tasks, no more complicated than the BASIC statement A=2, or POKE 100+X,3. That's why a library of ready-to-use routines such as the Kernal is so valuable.

However, you can replicate almost anything the Kernal does in BASIC. In fact, the BASIC interpreter, which lets you edit and run BASIC programs, is just a large machine language program that itself calls the same Kernal routines.

You can do almost everything machine language and the Kernal does in BASIC, assisted by POKE and PEEK, just more slowly (since BASIC has to be interpreted, command by command, instead of directly executed like machine language). Using the Kernal, it is easy to write very short machine language routines which do things faster and more efficiently than BASIC.

**SETTIM** is the opposite of RDTIM: it SETs the system clock instead of ReaDing it.

```
; Set system clock to 10 minutes=3600 jiffies.
LDA  #0
LDX  #L,3600
LDY  #H,3600
JSR  SETTIM
; This allows very accurate timing for many things.
```

**SETTMO** is used only with an IEEE add-on card to access the serial bus.

```
; Disable time-outs on serial bus.
LDA  #0
JSR  SETTMO
; To enable time-outs, set the accumulator to a 128 and
  call SETTMO.
```

**STOP** will set the Z flag of the accumulator if the STOP key was pressed.

```
 ; Check for STOP key being pressed.
WAIT JSR  STOP
     BNE  WAIT
     RTS
 ; STOP must be called if the STOP key is to remain
   functional.
```

**TALK.** This routine will command a device on the serial bus to send data.

```
; Command device #8 to TALK.
LDA  #8
JSR  TALK
RTS
; The accumulator designates the file number.
```

**TKSA** is used to send a secondary address for a TALK device. TALK must be called first.

```
; Signal device #4 to talk with command #7.
LDA  #4
JSR  TALK
LDA  #7
JSR  TKSA
RTS
; This example will tell the printer to print in uppercase.
```

**UDTIM.** If you are using your own interrupt system, you can update the system clock by calling UDTIM.

```
; Update the system clock.
JSR  UDTIM
RTS
; It is useful to call UDTIM before calling STOP.
```

**UNLSN** commands all devices on the serial bus to stop receiving data.

```
; Command the serial bus to UNLiSteN.
JSR  UNLSN
RTS
; The serial bus can now be used for other things.
```

**UNTLK.** All devices previously set to TALK will stop sending data.

```
; Command serial bus to stop sending data.
JSR  UNTLK
RTS
; Sending UNTLK commands all talking devices to get
  off the serial bus.
```

**VECTOR.** If the carry bit of the accumulator is set, the start of a list of the current contents of the RAM vectors is returned in the X and Y registers. If the carry bit is clear, then the user list pointed to by the X and Y registers is transferred to the system RAM vectors.

```
; Change the input routines to new system.
SEC
JSR  VECTOR
LDA  #L,MYINP
STA  USER+10
LDA  #H,MYINP
```

```
        STA   USER+11
        LDX   #L,USER
        LDY   #H,USER
        CLC
        JSR   VECTOR
        RTS
USER .DE 26
        ; The new input list can start anywhere. USER is the
          location for temporary strings, and 35-36 is the utility
          pointer area.
```

## Error Codes

If an error occurs during a Kernal routine, then the carry bit of the accumulator is set and the error code is returned in the accumulator.

| Number | Meaning |
|---|---|
| 0 | Routine ended by the STOP key. |
| 1 | Too many files open. |
| 2 | File already open. |
| 3 | File not open. |
| 4 | File not found. |
| 5 | Device not present. |
| 6 | File is not an input file. |
| 7 | File is not an output file. |
| 8 | File name is missing. |
| 9 | Illegal device number. |
| 240 | Top-of-memory change RS-232 buffer allocation. © |

# Mastermaze Update For The Atari

David Butler

*In the February 1983 issue of COMPUTE!, there was an excellent, multilevel maze game called "Mastermaze" by Kenneth S. Szajda. Here's a machine language routine for the Atari version that greatly speeds up the maze generator.*

A game written in machine language can run far faster than a BASIC version. However, in the case of this "Mastermaze" update, the speed of the game is not affected, only the maze generator which starts things off.

Before you can even begin to play Mastermaze, your computer must generate from 1 to 32 levels of mazes. This process can take several minutes. By incorporating this new machine language routine, you should find that the maze generating part of the program runs more than ten times faster. Thirty-two levels can be generated in under 90 seconds.

To use this routine to update your version of Mastermaze, first LIST the original version of the program. Delete lines 50, 90, 100, 110, and 111. Then, substitute the lines in this new program listing where they appear in the original. Also add the new lines that did not appear in Mastermaze.

### Adding An ML Maze Generator

```
8 DIM A$(37):SW=0
26 POKE 752,1:? "{CLEAR}":POSITION 4
   ,10:? "CONSTRUCTING MAZE...WAIT F
   OR START"
27 RESTORE 1000:FOR I=1536 TO 1690:R
   EAD A:POKE I,A:NEXT I:POKE 755,1
80 G=USR(1536,1675,A):GOSUB 500
1000 DATA 104,104,133,208,104,133,20
     7,104,133,204,104,133,203,173,1
     0,210,41,3,133,212
1010 DATA 133,213,24,10,168,165,203,
     113,207,133,205,165,204,200,113
     ,207,133,206,160,0
1020 DATA 177,205,201,128,208,40,165
     ,212,24,105,1,145,205,105,3,10,
     168,165,203,113
1030 DATA 207,133,203,200,165,204,11
     3,207,133,204,169,0,168,145,203
     ,165,205,133,203,165
1040 DATA 206,133,204,24,144,183,230
     ,212,165,212,41,3,133,212,197,2
     13,208,180,160,0
1050 DATA 177,203,133,212,152,145,20
     3,169,251,24,101,212,176,24,198
     ,212,165,212,24,10
1060 DATA 168,56,165,203,241,207,133
     ,203,200,165,204,241,207,133,20
     4,24,144,131,96,2
1070 DATA 0,176,255,254,255,80,0,1,0
     ,216,255,255,255,40,0
```

©

# COMPUTE! Back Issues

Here are some of the applications, tutorials, and games from available back issues of **COMPUTE!**. Each issue contains much, much more than there's space here to list, but here are some highlights:

**May 1981:** Named GOSUB/GOTO in Applesoft, Generating Lower Case Text on Apple II, Copy Atari Screens to the Printer, Disk Directory Printer for Atari, Realtime Clock on Atari, PET BASIC Delete Utility, PET Calculated Bar Graphs, Running 40 Column Programs on a CBM 8032.

**June 1981:** Computer Using Educators (CUE) on Software Pricing, Apple II Hires Character Generator, Ever- expanding Apple Power, Color Burst for Atari, Mixing Atari Graphics Modes 0 and 8, Relocating PET BASIC Programs, An Assembler In BASIC for PET, QuadraPET: Multitasking?

**July 1981:** Home Heating and Cooling, Animating Integer BASIC Lores Graphics, The Apple Hires Shape Writer, Adding a Voice Track to Atari Programs, Machine Language Atari Joystick Driver, Four Screen Utilities for the PET, Saving Machine Language Programs on PET Tape Headers, Commodore ROM Systems, The Voracious Butterfly on OSI.

**August 1981:** Minimize Code and Maximize Speed, Apple Disk Motor Control, A Cassette Tape Monitor for the Apple, Easy Reading of the Atari Joystick, Blockade Game for the Atari, Atari Sound Utility, The CBM "Fat 40," Keyword for PET, CBM/PET Loading, Chaining, and Overlaying.

**October 1981:** Automatic DATA Statements for CBM and Atari, VIC News, Undeletable Lines on Apple, PET, VIC, Budgeting on the Apple, Switching Cleanly from Text to Graphics on Apple, Atari Cassette Boot-tapes, Atari Variable Name Utility, Atari Program Library, Train your PET to Run VIC Programs, Interface a BSR Remote Control System to PET, A General Purpose BCD to Binary Routine, Converting to Fat-40 PET.

**December 1981:** Saving Fuel $$ (multiple computers: versions for Apple, PET, and Atari), Unscramble Game (multiple computers), Maze Generator (multiple computers), Animating Applesoft Graphics, A Simple Printer Interface for the Apple II, A Simple Atari Wordprocessor, Adding High Speed Vertical Positioning to Atari P/M Graphics, OSI Supercursor, A Look At SuperPET, Supermon for PET/CBM, PET Mine Maze Game.

**January 1982:** Invest (multiple computers), Developing a Business Algorithm (multiple computers), Apple Addresses, Lowercase with Unmodified Apple, Cryptogram Game for Atari, Superfont: Design Special Character Sets on Atari, PET Repairs for the Amateur, Micromon for PET, Self-modifying Programs in PET BASIC, Tinymon: a VIC Monitor, Vic Color Tips, VIC Memory Map, ZAP: A VIC Game.

**May 1982:** VIC Meteor Maze Game, Atari Disk Drive Speed Check, Modifying Apple's Floating Point BASIC, Fast Sort For PET/CBM, Extra Atari Colors Through Artifacting, Life Insurance Estimator (multiple computers), PET Screen Input, Getting The Most Out Of VIC's 5000 Bytes.

**August 1982:** The New Wave Of Personal Computers, Household Budget Manager (multiple computers), Word Games (multiple computers), Color Computer Home Energy Monitor, Intelligent Apple Filing Cabinet, Guess That Animal (multiple computers), PET/CBM Inner BASIC, VIC Communications, Keyprint Compendium, Animation With Atari, VIC Curiosities, Atari Substring Search, PET and VIC Electric Eraser.

**September 1982:** Apple and Atari and the Sounds of *TRON*, Commodore Automatic Disk Boot, VIC Joysticks, Three Atari GTIA Articles, Color Computer Graphics, The Apple Pilot Language, Sprites and Sound on the Commodore 64, Peripheral Vision Exerciser (multiple computers), Banish INPUT Statements (multiple computers), Charades (multiple computers), PET Pointer Sort, VIC Pause, Mapping Machine Language, Editing Atari BASIC With the Assembler Cartridge, Process Any Apple Disk File.

**January 1983:** Sound Synthesis And The Personal Computer, Juggler And Thunderbird Games (multiple computers), Music And Sound Programs (multiple computers), Writing Transportable BASIC, Home Energy Calculator (multiple computers), All About Commodore WAIT, Supermon64, Perfect Commodore INPUTs, Atari Autonumber, Copy VIC Disk Files, Commodore 64 Architecture.

**February 1983:** How The Pros Write Computer Games, 12 Joysticks Compared, Slalom (a game in 3-D for multiple computers), Super Shell Sort For PET, Atari SuperFont Plus, Creating Graphics On The VIC, Joysticks And Sprites On The 64, Bi-Directional VIC Scrolling, Commodore 64 Video: A Guided Tour, The Atari Cruncher, Easy Apple Editing, VIC Custom Characters For Games.

**March 1983:** An Introduction To Data Storage (multiple computers), Mass Memory Now And In The Future, Games: Closeout, Boggler, Fighter Aces, Letter And Number Play (all for multiple computers), VIC Music, Direct Atari Disk Access, TRS-80 Color Computer Data Base, Apple Subroutine Capture, PET Quickplot, TI Graphics Made Easy, VIC and Atari Memory Management.

**April 1983:** Selecting The Right Word Processor, Air Defense (multiple computers), Scriptor: An Atari Word Processor, Retirement Planner (multiple computers), TI-99 Match-Em, Dr. Video For Commodore, Atari Filefixer, Video 80: 80 Columns For The Atari, VICword, Magic Commodore BASIC.

**May 1983:** The New Low Cost Printer/Plotters, Jumping Jack (multiple computers), Deflector (multiple computers), VIC Kaleidoscope, Graphics on the Sinclair/Timex, Bootmaker For VIC, PET and 64, VICSTATION: A "Paperless Office," The Atari Musician, Apple Fast Sort, TI BASIC One-liners.

**June 1983:** How To Buy The Right Printer, The New, Low-cost Printers, Astrostorm (multiple computers), The Hawkmen Of Dindrin (multiple computers), MusicMaster For The Commodore 64, Commodore Data Searcher, Atari Player/Missile Graphics Simplified, TI Structured BASIC, UnNEW For The VIC and 64, Atari Fast Shuffle, VIC Contractor, Sinclair/Timex Screen Splitter.

**Home and Educational COMPUTING!** (Fall 1981 and Summer 1981 – count as one back issue): Exploring The Rainbow Machine, VIC As Super Calculator, Custom Characters, Alternate Screens, Automatic Line Numbers, Using The Joystick (Spacewar Game), Fast Tape Locater, Window, VIC Memory Map.

Back issues are $3 each or six for $15. Price includes freight in the US. Outside the US add $1 per magazine ordered for surface postage. $4 per magazine for air mail postage. All back issues subject to availability.

**In the Continental US call**
**TOLL FREE 800-334-0868**
**(In NC Call 919-275-9809)**

Or write to **COMPUTE!** Back Issues, P.O. Box 5406, Greensboro, NC 27403 USA. Prepayment required in US funds. MasterCard, Visa and American Express accepted. North Carolina Residents add 4% sales tax.

# COMPUTE!'s Second Book Of Atari

After only three years on the market, the Atari 400/800 microcomputers have become among the most popular personal computers ever made. So it was no surprise when *COMPUTE!'s First Book of Atari*, a collection of the best Atari articles published during 1980-81 in **COMPUTE!** Magazine, also became a "bestseller" with Atari enthusiasts. The first printing sold out in just a few months.

That's why we've followed up with *COMPUTE!'s Second Book of Atari*. Available immediately, the *Second Book of Atari* continues **COMPUTE!**'s tradition for personal computer users.

But the *Second Book of Atari* differs from the *First Book* in one important respect – all the articles are totally new and previously unpublished. The *Second Book of Atari* includes such interesting articles as "Page Flipping," "Fun With Scrolling," "Perfect Pitch," "Player-Missile Drawing Editor," and "TextPlot Makes a Game." Whole chapters are devoted to subjects such as "Advanced Graphics and Game Utilities," "Programming Techniques," and "Beyond BASIC." With 250 pages – more than 25 percent thicker than the *First Book* at the same price – the *Second Book of Atari* is crammed with information and ready-to-type program listings. And the book is spiral-bound to lie flat and is fully indexed for quick reference.

Best of all, *COMPUTE!'s Second Book of Atari*, like **COMPUTE!** Magazine itself, is written and edited to appeal to all computer enthusiasts – beginners and experts alike. Priced at only $12.95.



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 $2.00 shipping and handling. Outside the U.S. add $5.00 for air mail, $2.00 for surface mail. All orders prepaid. U.S. funds only.

# COMPUTE!'s Programmer's Reference Guide to the TI-99/4A

**Author:** C. Regena
**Price:** $14.95
**On Sale:** Now

Just about the best way to learn how to program a computer is to sit down with a patient friend who already knows how, and ask questions while you experiment with the computer. Owners of the popular Texas Instruments home computer will find that C. Regena is that kind of friend, and *Programmer's Reference Guide to the TI-99/4A* is that kind of book.

Regena carefully explains every BASIC command and function, and all the techniques needed to program TI graphics, sound, and speech. It's hard to think of a question that she doesn't answer simply and clearly, with hints about ways to write programs that do exactly what you want.

The book also provides dozens and dozens of programs, ranging from very short examples to full-length commercial-quality software. In effect, readers can look over Regena's shoulder as she goes through the programming process step by step, explaining what she's doing as she goes along. Not to mention the fact that the finished programs are valuable in their own right.

Even readers who are familiar with the computer will find this book valuable as a reference, where they can look up information they need and find the answers to particular questions.

Above all, *Programmer's Reference Guide to the TI-99/4A* is a book that lets readers use it however they like. You don't have to start at page one and read through, following someone else's plan for what you should learn first and what can wait until later. Instead, you can explore this book from any point of view, to solve almost any programming problem, and find the answer quickly and easily.

C. Regena is **COMPUTE!** Magazine's regular columnist on the TI-99/4A. She's an experienced and resourceful programmer. Like most of her readers, she taught herself how to program, and she hasn't forgotten what it's like to be a beginner, just starting out with the computer. And with *Programmer's Reference Guide*, TI users now have Regena to help them learn how to make their computer do exactly what they want it to do.

*Programmer's Reference Guide to the TI-99/4A* is available from **COMPUTE!** Publications, the leading publisher of books and magazines for home, educational, and recreational computing.

## Table of Contents



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

# 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.

# 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.*

# 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: ▬▬▬▬▬▬▬▬▬ 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, S would mean to type the S key while holding the shift key. If you find an underlined key enclosed in braces (e.g., {10 N}), 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.

# CAPUTE!

Modifications Or Corrections To Previous Articles

### RATS! For The 64

The 64 version of this game from the July issue is in two parts. Sue Roberts suggests a simple addition which will cause the first part, the setup program, to make the second part, the game itself, LOAD and RUN automatically. Disk users should SAVE the main game program with the filename RATMAZE, then add the following line to the setup program (Program 2, p. 60):

```
160 LOAD"RATMAZE",8:RUN
```

### Astrostorm For TI

In the TI-99/4A version of "Astrostorm" (June 1983, p. 82), line 780 should read:

```
780 IF CSHIP>0 THEN 810
```

### Hawkmen Of Dindrin, VIC And 64 Versions

In the second part of the VIC version of this game from the June 1983 issue (Program 2, p. 92), the { 06 LEFT }(six cursor lefts) in line 58 should be omitted. If you happen to be pushing the joystick when you lose your last player, the game ends. Bruce Stevenson and others suggest the following additional line to give you time to release the joystick or fire button:

```
1024 FOR X=1 TO 700:NEXT X
```

In the 64 version, the misplaced line 288 should be omitted.

### Checkers For The 64

Arnold J. Schmeling suggests the following addition and correction for this game from the May 1983 issue (p. 90), which prevent the computer from allowing illegal moves:

```
545 IF S(E,H)=1 AND B-H<1 THEN 1040
550 IF ABS(E-A)=2 AND S((E+A)/2,(H+B)/2)=
    >0 THEN 1040
```

### UnNEW For The 64

Under most conditions, this utility program from the June 1983 issue (p. 213) works equally well on either the VIC or 64. However, to guarantee proper operation on the 64, reader Don Lewis suggests that the existing line 330 be replaced with:

```
330 116,164
```

and a new tape be prepared in accordance with the original instructions. ©

# NEWS & PRODUCTS

## Drawing By Touch

Koala Technologies has introduced a touch tablet that allows computer users to draw directly on the video display screen, bypassing the keyboard.

The Koalapad Touch Tablet, available in versions for the Apple, Atari, Commodore 64, and IBM computers, weighs about a pound, and connects to the computer through a joystick port.

It can be used as a sketch pad, as a custom keypad, or as a game controller.

Though other applications are available, the Touch Tablet is packaged with *Micro Illustrator* from Island Graphics. This combination allows the touch pad to be used as a drawing board. Images can be drawn with a finger or stylus, and shapes, colors, shadings, and various "paintbrushes" can be selected from a menu.

*The Koalapad Touch Tablet can be used for drawing or as an auxiliary keyboard.*

The touch tablet and *Micro Illustrator* package sells for $125. Additional application programs will sell for about $50.

*Koala Technologies Corp.*
*4962 El Camino Real, Suite 125*
*Los Altos, CA 94022*

## Atari 400 Expansion

A 48K memory expansion kit, designed to upgrade the 8K or 16K Atari 400, is available from Atari.

The board is available for $130 installed at Atari Regional Repair Centers, or, for those who prefer to install the board themselves, it will be available through the Atari Program Exchange for $110.

*Atari Inc.*
*1265 Borregas Ave.*
*P.O. Box 427*
*Sunnyvale, CA 94086*
*(800) 538-8543*

## Checkbook System

T & F Software has produced a check register program for the 8K VIC-20, the Commodore 64, and Atari computers.

*CheckEase!* allows maintenance of multiple accounts, in-

16K ATARI RAM $49.75
32K RAM .......$69.75
48K RAM .......$99.75
64K RAM ......$129.75
128K RAM ....$399.75
TECH NOTES...$29.75

BOOK of ATARI SOFTWARE 1983
346 pages..... $16.75

CX415 HOME FILING MANAGER ... $41.75
CXL4007 MUSIC COMP..$33.75
CXL4002 ATARI BASIC ..$45.75
CX8126 MICROSOFT ....$65.75
CX4119 FRENCH ........$45.00
CX4118 GERMAN........$45.00
CX4120 SPANISH .. ....$45.00
CX4108 HANGMAN......$12.75
CX4102 KINGDOM ......$12.75
CX4112 STATES .......$12.75
CXL4003 ASSEMBLER ..$45.75

EASTERN HOUSE
Monkey Wrench 2 .......$52.75
INHOME
Baseball..................$29.95
IDSI
Speedway Blast......... $29.95
Pool 1.5.................. $26.95

GALAXIAN ................$29.75
DEFENDER ................$29.75
DIG DUG ..................$29.75
SPEED READING .........$53.75
ATARI WRITER............$54.75
BOOKKEEPER...........$102.75
CX4018 PILOT HOME ....$54.75
CX 405 PILOT EDU.......$91.75
CX404 WORD PRO ........$99.75
CXL4013 ASTEROID.... $25.75
CXL4020 CENTIPEDE .. $29.75
CXL4022 PACMAN ..... $29.75
CXL4011 STAR RAIDER $29.75
CXL4004 BASKETBALL $25.75
CXL4006 SUPER BREAK.$25.75
CXL4008 SPACE INVAD..$25.75
CX8130 CAVERNS OF M.$27.75

APX
Eastern Ft.41 ............. $25.50
DeRay Atari ............. $19.95
Math-Tic-Tac.............. $15.95
Pres of US................. $15.95
3R Math.................... $19.95
Typo Attack ............. $24.95
Family Budget ............ $19.95
F. Cash Flow.............. $19.95

BRODERBUND
Bank Street Writer D......$44.75
AE D.........................$24.75
Apple Panic D.............$23.75
Choplifter ROM ...........$32.75
David's Midnight..........$24.75
Stellar Shuttle C/D........$18.75
Ft. Apocalypse............$24.75

HES
Gridrunner R............ $27.75
Sword Point D............ $24.75

INFOCOM
Zork 1/2/or 3 ............. $29.95

FIRST STAR
Astro Chase............... $22.95

RANA DISK DRIVE
1000.....$295.00

ALIEN GROUP
Voice Box 2. $99.75

DON'T ASK
Sam .........$41.75
Abuse .......$15.95
Teleatri......$27.95
Poker Sam ..$24.95

Amulet
Nuke Sub ................$16.75
Magic Story Book.......$24.75
Thunder Island..........$13.95

ARTWORX
Hazard Run ..............$24.95
Hodge Podge ............$16.95
S. Poker..................$26.95
Bridge 3.0...............$18.95

EPYX
Jumpman ...............$26.97
Temple of Apashi......$27.75
Star Warrior............$27.75

ATARI
Computers for people.

## MODEMS

ANCHOR MARK I ..$79.00
ANCHOR MARK II..$79.00
HAYES SMART ...$239.00
HAYES MICRO II $309.00
Micro Bit
MPP-1000.........$159.00
NOVATION
CAT ..............$144.00
D-CAT............ $155.00
J-CAT ............ $115.00
APPLE CAT II .... $279.00
212 APPLE CAT . $589.00

## MONITORS

NEC JB1260............$125.00
NEC JB1201 ............$155.00
NEC TC1201 ............$315.00

Amdek Color I ..........$275.00
Amdek 300 Green......$149.00
Amdek 300 Amber .... $149.00
Gorilla Green ............$99.00

## WICO JOYSTICKS

APPLE - VIC - ATARI - TI
Command Control .......$20.95
Redball....................$22.75
Apple Trackball ..........$58.00
TRACK BALL ..............$52.75
EXTENSION CORD ....... $9.75
APPLE ADAPTOR .......$18.95
T.I. ADAPTOR ............ $9.95
POINT MASTER......... $12.75

apple

APPLE DUMPLING GX ........$99.75
APPLE DUMPLING 64...(16 Buffer)$179.75

INFOCOM
Zork I, II, or III.............. $26.75
Deadline ...................... $33.75

## BLANK DISKETTES

ELEPHANT SS/SD.......$18.25
ELEPHANT SS/DD.......$21.75
KANGAROO SS/DD with storage case..........$24.75
MAXELL MD I ............$29.75
MAXELL MD II ...........$39.75
DISK CASE (holds 10) $4 95
DISK CASE (holds 50) $19 75
ROM CASE (holds 10) $19 75

## SANYO

MBC 1000..............$1549.00
(with micropro software package)
MBC 1250..............$2195.00
MBC4050 ..............$2749.00
EFD 160 Disk............$499.00
5500 Letter Q. Printer...$649.00

## PRINTER INTERFACING AVAILABLE

APPLE DUMPLING GX..........$99.75
APPLE DUMPLING 64...(16 Buffer)...$179.75

| | ATARI | APPLE | VIC |
|---|---|---|---|
| EPSON | .....$35.00 | .......$79.00 | .....$65.00 |
| CITOH | .....$35.00 | .......$79.00 | .....$65.00 |
| NEC | .....$35.00 | .......$79.00 | .....$65.00 |
| OKIDATA | .....$35.00 | .......$79.00 | .....$65.00 |
| GEMINI | .....$35.00 | .......$79.00 | .....$65.00 |
| SMITH CORONA | ......$35.00 | .......$79.00 | .....$65.00 |

Atari parallel Printer cable (6') ...$39.95
Microbits Parallel Cable ........$89.95

## commodore

HES VIC-20
Torg C...........................$14.75
HES Games I C .................$14.75
HES Games II C ................$14.75
VIC Fortit Rom..................$42.75
HES MON Rom ..................$28.75
Turtle Graphics Rom ...........$28.75
HES Writer Rom ................$28.75
Shamus Rom.....................$28.75
Protector Rom ..................$31.75
Robot Panic Rom ...............$28.75

## DISKETTE SALE

BULK DISKS (qty 150) .......$1.50 ea.
Certron CC-10(3 blank cassettes)...$3.75

PRINTER RIBBONS AVAILABLE

PROWRITER 2P.....$699.00
GEMINI 15..........$449.00
PRINTMASTER....$1589.00
SMITH CORONA TP1...$549.00
CITOH 8600B.............$1025.00
STARWRITER.............$1099.00
OKIDATA 82 ..... SAVE
OKIDATA 83 ..... LOWEST
OKIDATA 84P.... PRICE AVAILABLE
OKIDATA 93 ..... SAVE
TRACTOR......... $49.75
OKIDATA 92................ $SAVE

# SAVE on these in-stock PRINTERS

GORILLA GX-100 ..............$199.00
EPSON ............... $CALL
GEMINI 10 ....... $299.00
PROWRITER...... $365.00
NEC 8023 ..................... $399.00

MasterCard

## TO ORDER

VISA

CALL TOLL FREE
800-233-8760
In PA 1-717-327-1824

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 1774C

POLICY
In-stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash, add 4% for Master Card or Visa. DEALER INQUIRIES INVITED.

cludes checkbook reconciliation and budget categories, and allows changes or deletions in any check or deposit with automatic updating of all balance figures.

*CheckEase!* is compatible with Commodore's *Personal Finance* program. The VIC version sells for $24.95. The Atari and 64 versions are $29.95 for tape, and $34.95 for disk.

Among other new products available from T & F are *Space Sentinel* and *Slot Trivia*. *Space Sentinel* is an arcade-type game for the Commodore 64. The object is to protect Earth from alien attackers who hurl heat missiles at our polar ice caps. The game sells for $29.95.

*Slot Trivia* is a trivia question-and-answer game in a slot-machine format. The game, available on disk for Atari computers, includes more than 500 questions in 11 categories.

*T & F Software*
*10902 Riverside Drive*
*North Hollywood, CA 91602*
*(213) 501-3856*

## 80-Column Thermal Printer

Alphacom has produced an 80-column thermal printer with graphics capabilities that is designed to be compatible with several computers.

The Alphacom 81 printer includes upper- and lowercase letters and wraparound for lines longer than 80 characters. It prints at 80 characters per second, recognizes standard ASCII control codes, and can print bit-mapped graphics.

Alphacom also has produced a series of interface cables that make the printer compatible with the TRS-80 Color Computer as well as the Commodore 64, VIC-20, Atari, and Apple computers.

The Alphacom 81 will sell for $169.95. Interface cables start at $44.95.

*Alphacom, Inc.*
*2323 South Bascom Ave.*
*Campbell, CA 95008*
*(408)559-8000*

*Alphacom 81 is an 80-column thermal printer with graphics capabilities.*

# COMPU SENSE

| CS1 | QUICK BROWN FOX | $55.00 |
|---|---|---|

The Word Processor of this decade! For the VIC-20 and C-64.

## COMMODORE 64®
## $435.00

**Plus you receive a free QBF Word Processor valued at $55.00**

**Write for FREE Catalog!**

## VIC-20®
## $88.00

**When you buy our 6 Game Pac or 6 Finance Pack $43.00**

### C-64 Software

| | | |
|---|---|---|
| Pet Emulator | | $27.95 |
| Editor Pac | | 67.00 |
| File Pac | | 32.36 |
| Account Pac | | 57.00 |
| Farm Management I (Agricultural Software) | | 47.25 |
| Home Budget | | 29.95 |
| Stock Investments | | 76.95 |
| Calc Result | | 140.00 |
| 650Z Professional Development System | | 27.95 |
| Mail List | | 34.95 |
| Vic Easy Lesson & Easy Quiz | | 35.97 |
| Loan Calculator | | 15.95 |
| Data Files | | 14.95 |
| Research Assistant 2.0 | | 28.00 |
| Total Label 2.6 (Mailing Lables) | Tape | 19.95 |
| | Disk | 22.00 |
| Total Time Manager 2.6 | | 37.00 |
| Total Text (Word Processor) 2.6 | | 40.00 |

### C-64 Games

| | | |
|---|---|---|
| Flight 64 (Flight Simulator) | Tape | $13.95 |
| | Disk | 15.95 |
| Gunslinger | | 13.95 |
| Spellathon | | 16.95 |
| Motor Mania | | 25.95 |
| Renassance | | 25.95 |
| Vic Clowns | | 25.00 |
| Radar Rat Race | | 25.00 |
| Jupiter Lander | | 25.00 |
| Temple of Apshai | | 39.95 |
| Upper Reaches of Apshai | | 16.95 |
| Curse of Ra | | 26.95 |
| Sword of Fargoal | | 19.95 |
| Jump Man | | 26.95 |

### VIC-20 & C-64 Hardware

| | | | |
|---|---|---|---|
| VIC-1541 | Disk Drive | | 347.00 |
| VIC-1530 | Datasette | | 67.50 |
| VIC-1515 | Printer | | 334.95 |
| VIC-1010 | Expansion Module | | 139.95 |
| VIC-1311 | Joystick | | 9.95 |
| | Wico Joystick | | 28.00 |
| VIC-1312 | Game Paddles | | 19.95 |
| | Telephone Modem | | 99.95 |
| | Terminal Emulator (64) | Tape | 9.95 |
| | | Disk | 15.95 |
| | 40x25 Terminal Emulator VIC | | 40.95 |

**VIC-1210 VIC 3K Memory Expander Cart.** 34.95
Plugs directly into the VIC's expansion port. Expands to 8K RAM total.

**VIC-1110 VIC 8K Memory Expander Cart.** 52.50
8K RAM expansion cartridge plugs directly into the VI.

**CM102 24K Memory Expander Cart.** 119.95

**VIC-1011A RS232C Terminal Interface** 39.95
Provides interface between the VIC-20 and RS232 telecommunications modems. Connects to VIC's user port.

| | |
|---|---|
| **PETSPEED** — Basic Compiler for Commodore | 140.00 |
| **Vic Rabbit Cartridge** | 35.00 |
| **CBM 64 Rabbit** | 35.00 |
| **Star G-10 Printer** | 360.00 |
| **Mura Modem** | 120.00 |
| **Smith Corona TP-1 Printer** | 650.00 |

### CARDCO HARDWARE

**CARDBOARD 6** $87.50
An expansion interface for the VIC-20. Allows expansion to 40K or accepts up to six games. May be daisy-chained for more versatility.

**CARDBOARD 3** $35.95
Economy expansion interface for the VIC-20

**CARD "?" CARD/PRINT** $76.00
Universal Centronics Parallel Printer Interface for the VIC-20 or CBM-64. Use an Epson MX-80 or OKIDATA or TANDY or just about any other

**CARDETTE** $30.95
Use any standard cassette player/recorder with your VIC-20 or CBM-64.

**LIGHT PEN** $29.95
A light pen with six good programs to use with your VIC-20 or CBM-64

**16K Memory Expander** $50.50

All CARDCO Products have a lifetime warranty.

### COMMODORE SOFTWARE

**VIC-1211A VIC-20 Super Expander** $55.00
Everything Commodore could pack into one cartridge — 3K RAM memory expansion, high resolution graphics plotting, color, paint and sound commands. Graphic, text, multicolor and music modes. 1024x1024 dot screen plotting. All commands may be typed as new BASIC commands or accessed by hitting one of the VIC's special function keys. Includes tutorial instruction book. Excellent for all programming levels.

**VIC-1212 Programmer's Aid Cartridge** $45.99
More than 20 new BASIC commands help new and experienced programmers renumber, trace and edit BASIC programs. Trace any program line-by-line as it executes, pause to edit. Special KEY command lets programmers redefine function keys as BASIC commands, subroutines or new commands

**VIC-1213 VICMON Machine Language Monitor** $48.99
Helps machine code programmers write fast, efficient 6502 assembly language programs. Includes one line assembler/disassembler.

### VIC-20 Software

for Business & Home Applications

| | | |
|---|---|---|
| 6502 Professional Development System | | $25.00 |
| Vic Forth (Advance Computer Language) | | 49.95 |
| Hess Mon (Machine Language Monitor) | | 34.95 |
| Hess Writer (Word Processor) | | 34.95 |
| Turtle Graphics | | 34.95 |
| Total Label 2.1 | Tape | 17.95 |
| | Disk | 22.00 |
| Total Time Manager 2.1 | Tape | 28.00 |
| | Disk | 32.00 |
| Research Assistant 2.0 | Tape | 28.00 |
| | Disk | 32.00 |
| Total Text 2.5 | Tape | 30.00 |
| Encoder | | 34.95 |
| Acct. Payable & Receivable | Tape | 29.00 |
| | Disk | 35.00 |

### VIC-20 Games

| | | |
|---|---|---|
| Exterminator Plus (Excellent) | | $17.95 |
| Anti Matter Splatter (Disaster) | | 17.95 |
| Rescue From Nufon (Great) | | 12.95 |
| Tank Wars (War Game) | | 15.95 |
| Simon (Great for kids) | | 13.45 |
| Dam Bomber (Avoid the enemy) | | 13.45 |
| Breakout | | 7.95 |
| Snack Man (Pac Man) | | 17.95 |
| Defender on Tri | | 17.95 |
| Amok | Cart. | 23.95 |
| | Cassette | 19.94 |
| Starfighter | | 17.95 |
| Torg | | 15.95 |
| Gridrunner | | 34.95 |
| Invasion Orion | | 20.95 |
| 8K Backgammon | | 19.95 |

TO ORDER:
P.O. BOX 768
WICHITA, KS 67201
(316) 263-1095

Handling charges $2.00
C.O.D. (Add $2.00)
Personal checks allow 3 week delivery
VIC-20® is a registered trademark of Commodore
Prices subject to change

## Timex Tutorial

*The Programming Kit I,* a BASIC tutorial program for Timex/ Sinclair users, is among three new programs produced by Timeworks. *The Programming Kit,* a how-to learning approach to programming, includes an explanation of an eight-step approach to program design.

*Family Pak* is a set of five 2K programs designed for day-to-day home tasks. The programs are Check Book Balancer, Recipe Recorder, Mini-Money Manager, Homework Helper and Memo-board, a family message center.

Timeworks' third new program, *Scyon's Revenge,* is a deep-space combat game that includes 3-D simulation. The game is provided with a keyboard overlay to give you the feeling of punching command buttons rather than hunting and pecking on a keyboard.

*Timeworks, Inc.*
*405 Lake Cook Road*
*Building A*
*Deerfield, IL 60015*
*(312)291-9200*

## Kindergarten Gallery

Midwest Software has developed a series of computer programs designed for kindergarten children.

The *Kinder Koncepts* software addresses reading awareness, math concepts, pattern recognition, letters, numbers, colors, and shapes.

All programs follow the same general format, operate with a single keystroke, and keep the necessity for reading to a minimum.

In each program, ten problems are presented. A correct answer is rewarded with a smiling face and a tune. An incorrect answer results in a frown and the chance to try again. Each program has a built-in graph so that progress can be monitored at a glance.

The programs are available on cassette or disk for all Commodore computers except the VIC-20. Disk versions also are available for the Apple II+. The cost is $7.95 per program for cassette or $69.50 for a disk with ten programs.

*Midwest Software*
*Box 214*
*Farmington, MI 48024*
*(313)477-0897*

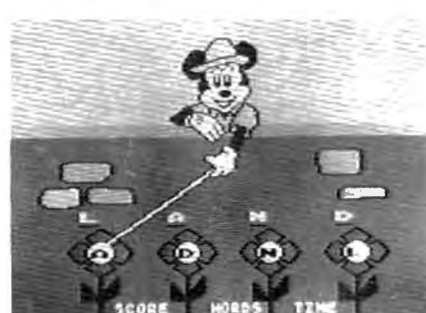
## Mickey's New Adventure

Walt Disney Productions has entered the computer software market, and it's making its debut with the help of Mickey Mouse.

*Mickey in the Great Outdoors* is a pair of interactive adventure games for children seven to ten years old. Mickey Goes Hiking develops grammar and spelling skills by requiring players to finish sentences and unscramble words to help Mickey through his adventure. Mickey Goes Exploring is a similar game, but is based on math skills and equation solving.

*Mickey in the Great Outdoors,* is being offered only for Atari computers, and distributed through Atari. This program, however, is just the tip of Walt Disney's software iceberg, according to the company.

Plans call for as many as 50 additional Walt Disney programs to be released this year, sup-



*Mickey Mouse helps unscramble a word in* Mickey in the Great Outdoors.

# 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.

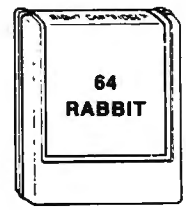
The MONKEY WRENCH provides 18 direct mode commands. They are: AUTO LINE NUMBERING — Provides new line numbers when entering BASIC program lines. RENUMBER — Renumbers BASIC's line numbers including internal references. DELETE LINE NUMBERS — Removes a range BASIC line numbers.

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

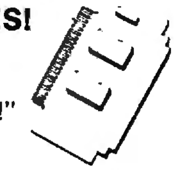



**NEW FEATURE!**
**DATA FILES!**

"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 Mainframe, 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.

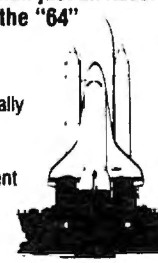
## ATARI and VIC Cartridges
EHS can supply large quantities of ATARI and VIC Cartridges for software developers. If you need cartridges, call for pricing.

## More than just an Assembler/Editor!
**Now for the "64"**

# MAE
for PET APPLE ATARI

~~$169.95~~ New Price $99.95

It's a Professionally Designed Software Development System

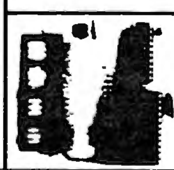


Blast off with the software used on the space shuttle project!

- 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!

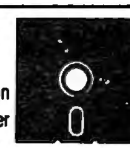
## ATARI AND PET EPROM PROGRAMMER

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

porting Atari, Radio Shack, Texas Instruments, NEC, and Panasonic computers, and distributed through those companies.

*Walt Disney Telecommunications*
*500 South Buena Vista St.*
*Burbank, CA 91521*
*(213)840-1111*

## Add 64K To Timex/Sinclair

Sunflower Systems has produced a 64K RAM pack for the Timex/Sinclair computers.

The self-contained memory expander plugs into the computer just like the Timex 16K RAM. No additional equipment is needed.

The unit, housed in metal to eliminate radio frequency interference, sells for $119.95 plus $5 for shipping.

*Sunflower Systems*
*718 East Avenue B*
*Hutchinson, KS 67501*
*(316)662-2134*

## Science Fiction Text Adventure

*Cyborg* is a science fiction text adventure that has no treasures to collect and no score to tally.

The adventure is available from Sentient Software for $34.95 in versions for Atari, Commodore 64, Apple, and IBM.

The game includes character development, animals to talk to, opinions from the Cyborg, and a command structure that allows full sentences.

*Sentient Software, Inc.*
*P.O. Box 4929*
*Aspen, CO 81612*
*(303) 925-9293*

## Turn On The Juice

Tronix, a young company that made its first splash in the VIC-20 market, has added the Commodore 64 to its repertoire.

The company's latest creation is *Juice*, a fast-paced strategy game for the 64 and Atari computers. The hero in *Juice* is Edison, whose job is to complete circuit boards in the face of all the troubles his adversary – Kilerwatt – can throw his way.

The game includes six play levels, each with three rounds plus a bonus round. The 32K Atari version sells for $29.95, and the Commodore 64 version sells for $34.95.

Another Tronix offering for the 64 is *Kid Grid*, which previously had been released in an Atari version. In the game, "the Kid" darts around a grid trying to connect all the dots while eluding four bullies. *Kid Grid* sells for $34.95.

In addition to branching into the 64 market, Tronix has bolstered its VIC-20 lineup with the addition of three new cartridge games, *Deadly Skies, Scorpion*, and *Gold Fever!*.

*Deadly Skies* is a shoot-em-up

game in which the player, equipped with a squadron of helicopters, takes on a military base. *Gold Fever!* is a maze game in which a prospector faces runaway boxcars, boulders, claim jumpers, and a limited supply of oxygen. The object of *Scorpion* is to keep the snake alive and fed in the midst of a world filled with dragons, frogs, Venus's-flytraps, stalkers, worms, and pods.

Each of the VIC-20 games sells for $39.95.

*Tronix Publishing, Inc.*
*8295 S. La Cienega*
*Inglewood, CA 90301*
*(213)215-0529*

## A Program To Remember

*Memory Trainer*, an interactive program to teach memory improvement, is available from Einstein.

The program, which is available for the Apple, Atari 800, and Commodore 64, is based on memory improvement research from the past 100 years.

*MemoryTrainer* includes five lessons in a three-disk package that sells for $89.95. The lessons teach the ability to remember faces, dates, telephone numbers, lists, and quotations, and to use association as a memory tool.

The package also includes Memory Mix, a game that provides practice for each memory skill.

*The Einstein Corporation*
*11340 W. Olympic Blvd.*
*Los Angeles, CA 90064*
*(213) 477-6733*

## VIC Wafer Storage

A low-cost micro-wafer storage device for the VIC-20 will be available later this year from Unitronics, through a licensing agreement with Vadem, the unit's builder.

The V-20 Expander is described as an inexpensive alternative to floppy disk storage for low-end computers. It reads or writes data to small tape cassettes at a speed approaching that of disks.

The device, which measures 5x6x7 inches, plugs into the VIC's cartridge expansion slot. It includes a 10K RAM memory expansion board, a 64K data wafer and high-speed micro-wafer drive, a filing system, and VWOS – the Vadem Wafer Operating System.

Because VWOS is able to access the computer's memory bus directly, rather than through a serial port, the V-20 is able to improve on the data transfer rates of existing micro-wafer devices.

The expander is expected to sell for about $100.

*Vadem*
*3517 Ryder Street*
*Santa Clara, CA 95051*
*(408) 738-0571*

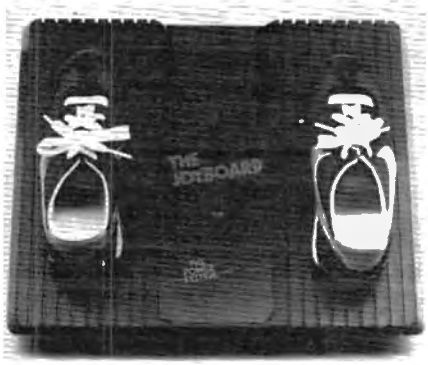
## Stand-On Game Controller

The Joyboard, a game controller that involves the whole body rather than just the hands, has been introduced by Amiga for the VIC-20 and Atari computers.

The Joyboard, which comes with *Mogul Maniac*, a skiing simulation game, will sell for about $50. Other games designed for use with the Joyboard – *Surf's Up* and *Off Your Rocker* – will cost about $20.

The Joyboard also can be used with many existing maze-type games to provide a different challenge, or, for shoot-em-up games, a conventional joystick can be plugged into the Joyboard to control firing, while your feet control direction.

Amiga also has produced a version of its Power-Stick joystick for the TI-99/4A. This includes two controllers hard-



*The Joyboard from Amiga transfers videogame control from your hands to your entire body.*

wired into a single plug to fit the TI's single jack configuration. The pair will sell for about $20.

*Amiga Corporation*
*3350 Scott Boulevard, Building 7*
*Santa Clara, CA 95051*
*(408) 748-0222*

## Add-On Adventure

Infocom, the company that produced the *Zork*, the popular underground fantasy series, is launching a new trilogy that will take the adventurer into the world of magical powers and perilous predicaments.

The first in the new series, *Enchanter*, scheduled to be available by mid-September, is a prose adventure that takes place in an abandoned castle. The passage of time plays an important role in the game: you must eat, drink, and sleep regularly, or your powers will fail.

The game, which will be available in versions for most popular microcomputers, will retail for $49.95 to $59.95.

*Infocom, Inc.*
*55 Wheeler St.*
*Cambridge, MA 02138*
*(617) 492-1031*

## Battle Games

Thorn EMI Home Video has released a pair of new battle games, *Orc Attack* and *Fourth Encounter*.

In *Orc Attack*, the player must defend his castle against the Orcs, who erect ladders and scale the castle walls under cover of a volley of crossbow bolts from their archers. The game, which is available for the Atari 400 and 800, sells for $39.95.

*Fourth Encounter* is a cartridge game for the VIC-20. The challenge here is to save a planet from an invasion of aliens, who bring with them slavery, death, and destruction. *Fourth Encounter* is available for $39.95. In addition to these two new games, Thorn EMI has converted a couple of other games into new formats. *Submarine Commander*, previously released as an Atari game, is now available for the VIC, and *River Rescue* now can be played on the Atari.

*Thorn EMI Home Video*
*1370 Avenue of the Americas*
*New York, NY 10019*

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 |
| DISKNIZ | 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 |

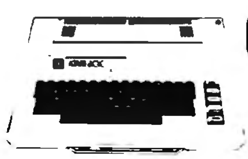
| | | |
|---|---|---|
| AMDEK COLOR I PLUS MONITOR | | 319.00 |
| APPLE EMULATOR | CBM64 | 79.50 |
| CARDBOARD (6 CART + RESET) | VIC-20 | 70.00 |
| THE CARD? (FOR PARALLEL PRTR) | CBM64&VIC-20 | 58.00 |
| CARDETTE (CASSETTE INTFCE) | CBM64&VIC-20 | 22.00 |
| CARDAPTER/1 ATARI 2600 INTFCE | VIC-20 | 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 · FRANKLIN · NEC · IBM · MOSAIC 64K RAM SELECT · EAGLE · TEXAS INSTRUMENTS · COLUMBIA DATA PRODUCTS INC · 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! (619) 765-0239

P.O. Box 1099,
Julian, Calif. 92036

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.

## Ancient Game On Computer

A computer version of the ancient strategy game, *GO*, is available from Hayden Software.

The program is designed to teach the novice as well as hone the skills of the experienced *GO* player.

The object of *GO* is to encircle, trap, and capture the computer's playing pieces while defending against the computer's attempts to trap you.

Versions of the game are available for $34.95 for Apple and $29.95 for Atari.

Also available from Hayden is an action-packed maze game, *Wargle*. In the game, priced at $34.95 for Apple and Atari, the player must take evasive action while using a laser beam to eliminate the Wargles. The game includes seven mazes and six levels.

*Hayden Software Company*
*600 Suffolk St.*
*Lowell, MA 01853*
*(617) 937-0200*

## Printer Buffer

The Microbuffer In-Line, a printer buffer with a memory expandable to 256K, is available from Practical Peripherals.

The buffer is compatible with almost any serial or parallel printer, modem, word processor, or computer equipped with an RS-232 serial output device.

The buffer includes a COPY feature that allows printing up to 255 copies of any document with the touch of a button, and the PASS feature allows data to be routed around the buffer when appropriate.

Microbuffer In-Line with 32K memory is available for $299. A 64K version sells for $349, and additional memory expansion is available for $179 per 64K.

*Practical Peripherals, Inc.*
*31245 La Baya Drive*
*Westlake Village, CA 91362*

## Apple In Space

*Mission: Escape!* is an arcade-type space game for the 64K Apple II computer.

To play, you pilot your shuttlecraft through asteroids and meteors to save the inhabitants of the 12 planets in the Galaxy of Appel, which is about to self-destruct because of violent volcanic activity.

The hazards increase with each planet you attempt to evacuate.

*MicroSparc, Inc.*
*10 Lewis St.*
*Lincoln, MA 01773*
*(617) 259-9710*

## T/S Text Editor, Input Utility

An input utility program and a text editor for the Timex/Sinclair are available from SyncMaster. Each program sells for $14.95 plus $1 for shipping.

The *Screen Machine* is a 1.5K machine language utility that allows inputs anywhere on the screen. The routine performs length verification of responses, compacts numbers, and allows dates in MMYY or MMDDYY formats.

The *Vu-Write Text Editor* is a menu-driven program for machines with at least 16K RAM.

The program includes insert, delete, change, and save functions. It leaves 11K available for text and allows line length to be set by the user.

*Vu-Write Text Editor* is written to be compatible with the ZX81 printer, but the program is listable and can be modified for any printer.

*SyncMaster*
*P.O. Box 511*
*Oak Ridge, NC 27310*
*(800) 334-0854, or*
*(919) 643-7120 in North Carolina*

## Alphabet Zoo

Spinnaker Software is scheduled to release another game in its early learning series this fall.

The game, *Alphabet Zoo*, is designed to teach three- to eight-year-olds the relationship between letters and sounds. It incorporates two maze games, colorful graphics, and sound.

*Alphabet Zoo* will be available on disk for Apple, Atari, IBM, and Commodore 64 computers. Cartridge versions will be available for the 64 and Atari.

Another fall offering from Spinnaker is *Cosmic Life*, a computer learning game in the style of checkers and Go. It is designed to strengthen planning, strategy, and pattern recognition skills.

*Cosmic Life* will be available in cartridge for the Atari and Commodore 64.

*Spinnaker Software*
*215 First St.*
*Cambridge, MA 02142*
*(617) 868-4700*

## EPROM Programmer

Gloucester Computer has produced a Commodore 64 version of its VIC Promqueen EPROM programmer.

The PQ/64 cartridge includes a 28 pin Textool ZIF socket, a matrix switch EPROM type selection that accommodates all JEDEC pinout devices that work on 5 volts, RS-232 communications software, faster burn process, a burn test procedure, and a 24K workspace.

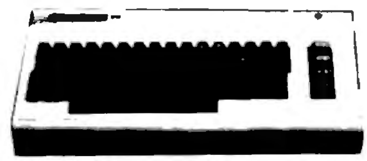
The PQ/64 is expected to retail for $299.

*Gloucester Computer, Inc.*
*One Blackburn Center*
*Gloucester, MA 01930*
*(617) 283-7719*

commodore

# VIC-20
## $99

# COM-64
## $427



MEMORY EXPANSION

# 16K RAM
## $69

# 8K RAM
## $39

| | |
|---|---|
| **SOFTWARE FOR THE VIC-20** | |
| WORD PROCESSING | $23.00 |
| ADV WORD PROCESSING | $32.00 |
| MAILING LISTS | $20.00 |
| **SOFTWARE FOR THE COM-64** | |
| WORD PROCESSING | $38.00 |
| MAILING LISTS | $20.00 |
| *Call for other software items.* | |
| VIC 1541 DISC DRIVE | $339.00 |
| VIC 1530 1530 DATASSETTE | $ 64.95 |
| VIC 1525 GRAPHIC PRINTER | $339.00 |
| 2Kx8 STATIC RAM CHIPS (200 NSEC) QTY. | ea. $ 7.95 |

VISA — ORDER FORM — MasterCard

(Circle Above Items)

NAME ____________________

STREET ____________________

CITY ____________________

STATE ____________________ ZIP __________

PHONE ____________________

Personal checks accepted
(Allow 3 weeks extra)

CHECK ONE:

☐ VISA ☐ MASTERCARD

☐ Check Enclosed ☐ C.O.D.

Credit Card # __________

Expiration Date __________

Credit Cards add 3%
Add 3% Shipping Charge
COD's add $1.50 plus 20% Deposit
Required CA Res. 6½% Tax

# U.S. TECHNOLOGIES

1625 W. OLYMPIC SUITE 800
LOS ANGELES, CA 90015
(213) 383-8127
(Information & Orders)

CREDIT CARD ORDERS ONLY CALL TOLL FREE:
**1-800-824-7888★** *(48 states)* **1-800-824-7919 ★** *(Alaska & Hawaii)* **★ Ask for Operator #649**

# CALENDAR

**September 15-18, Minneapolis Auditorium, Minneapolis, MN.** The Second Annual Twin Cities Computer Show and Software Exposition. Features microcomputers, software, accessories and peripherals, publications and other services. Hours are 10:30 a.m. to 5:30 p.m. daily. Admission is $5 for adults, $3 for children. For more information, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167. (800) 841-7000 or (617) 739-2000 (within Massachusetts).

**September 29 - October 1, Hynes Auditorium, Boston, MA.** CP/M '83 East. A dual-purpose international event for the CP/M industry and its users. An exposition, featuring what is called the largest presentation of CP/M based hardware and software ever assembled: manufacturers, independent software developers, venture capitalists, software publishers, distributors, and dealers. The Conference Program, with nearly 100 sessions, includes noted leaders from the software industry. Admission is $10 for a one-day exhibit pass, or $25 for a three-day exhibits and conference ticket. For more information, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167. (800) 841-7000 or (617) 729-2000 (inside Massachusetts).

**September 22-24, Denver Merchandise Mart, Denver, CO.** The Second Annual Rocky Mountain Computer Show. Features microcomputers, personal, educational, business and entertainment software, accessories and peripherals, publications, and other services. Hours are 10:30 a.m. to 5:30 p.m. daily. Admission is $5 for adults, $3 for children. For more information, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167. (800)841-7000 or (617)739-2000 (within Massachusetts).

**October 13-15, Philadelphia Civic Center, Philadelphia, PA.** EduTech/East '83. A national educational computer conference and exposition. Presentations, by nationally recognized speakers, include CAI, classroom management, programming, research applications, and literacy. Presentations will be in the form of workshops, seminars, demonstrations, and MicroCourses. Hardware, software, and publishing companies will showcase their products. For further information, contact Carol Houts, Judco Computer Expos, Inc., 2629 North Scottsdale Road, Scottsdale, AZ 85257. (800) 528-2355 or (602) 990-1715 (in Arizona).

**October 14-15, Ball State University, Muncie, IN.** Third Annual Computer Conference for Educators. Sponsored by Indiana Computer Educators, this conference will cover areas of educational computing from preschool to college level uses, from instructional to administrative applications. For more information, contact Dave Flowers, 1230 South Clinton Street, Fort Wayne, IN 46802. (219) 425-7228.

**October 14-15, Dallas, TX.** Computers & Reading/Learning Difficulties. Sponsored by Computers, Reading and Language Arts. Sessions on the use of microcomputers in education, specifically in reading, language arts, and learning disabilities. Open to anyone involved with computers in education, both novice and experienced. For brochure on program, faculty and registration, contact Frost Conference Management, Department I, 1070 Crows Nest Way, Richmond, CA 94803. (415) 222-1249.

**October 18-20, Silicon Valley, CA.** EdCompCon '83: "Applying Technology to Education in the Next Ten Years." Primary focus of this educational conference will be on application of the latest technology in computer-related areas, hardware and software, to education. Sponsored by the IEEE Computer Society.

**October 27-30, Washington Convention Center, Washington, D.C.** Mid-Atlantic Computer Show and Office Equipment Exposition. Produced by Computer Expositions, Inc., P.O. Box 3315, Annapolis, MD 21403. (800) 368-2066 or (800) 492-0192 (within Maryland).

**October 28-30, Moscone Center, San Francisco, CA.** Applefest/San Francisco, the largest Apple-specific computer show in the United States. Seminars, tutorials, application workshops, advanced user workshops, and software/hardware displays and booths. Highlights: an open forum with Steve Wozniak of Apple Computers, Inc., and a panel discussion with industry leaders. Show hours are 10:30 a.m. to 5:30 p.m. daily. Ticket prices are $25 for a three-day exhibits and conference ticket or $10 for a one-day exhibits-only ticket. For more information, call or write Northeast Exhibitions, 822 Boylston Street, Chestnut Hill, MA 02167. (800) 841-7000 or (617) 739-2000 (within 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, their 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.*

# PRODUCT MART

# Advertisers Index

# TRUNKS FOR THE MEMORIES.

Introducing the most logical place to store Elephant Memory Systems® (or lesser brands of disks): The Trunk.

With its alphabetized library index, you can file or retrieve up to 60 disks, instantly.

The Trunk is made of durable molded plastic with a hinged, one-piece lid, to keep disks safe from dust, dirt, and other detriments which disks despise.

And, it's portable. Because the lid doubles as a carrying handle so your Elephant Memory Systems® disks can go anywhere you do.

There's a model for 5¼" and 8" floppies, as well as a cassette-and-game file and a special Atari® version.

So if you're looking for the best disk storage system on the market . . .

The Trunk is an open-and-shut case.

# THE TRUNK. ENDORSED BY ELEPHANTS.

**Elephant Memory Systems® Disks**

*A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a life-time of heavy-duty use).*

*Marketed exclusively by Leading Edge Products, Inc., Information Systems and Supplies Division, 55 Providence Highway, Norwood, Massachusetts 02062. Call: toll-free 1-800-343-8413; or in Massachusetts call collect (617) 769-8150. Telex 951-624.*

# 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.

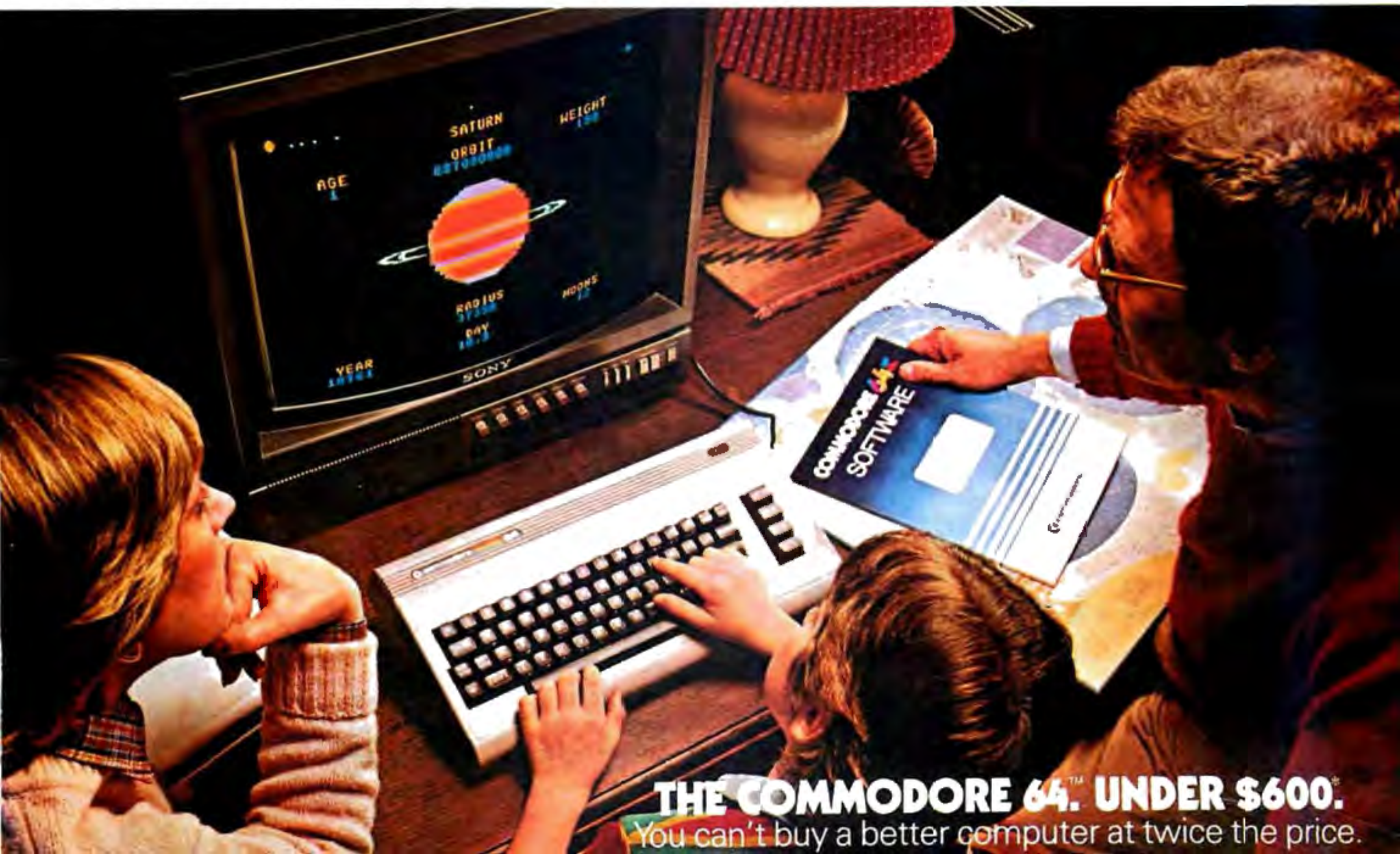
So while other companies are trying to take advantage of the computer revolution, it seems to us they're really taking advantage of something else:

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.