Serving the CoCo Community for

The RAINBOW 12 YEARS

THE COLOR COMPUTER MONTHLY MAGAZINE

August 1992 Vol. XII No. 1 Canada $4.95 U.S. $3.95

The First Issue of Our 12th Year

Feature Program

Make Forms the Easy Way

Have you ever needed to create a custom form for personal or business use? If so, you probably pulled out paper, a pencil and a ruler, then got busy drawing all sorts of lines and boxes. Then when you got ready to add the text, you likely discovered it would have been much easier had you started with the text and added the graphics to that — especially if you planned to use the CoCo to add the text. *Forms* takes most of the tedium out of creating good-looking forms.

Forms, a program designed for the CoCo 3, is a sort of word processor for forms that allows you to easily mix text with graphics; it gives you full-screen control over placement of text and graphic elements. The program does not require an RGB monitor, but it is important that the monitor you use be capable of showing all the columns on a 40-column screen. To get started, enter the program as shown in the listing and save it to tape or disk. If you have an RGB monitor, enter RGB before running the program.

When you run *Forms*, you'll first be asked if you want to read the onscreen instructions. If so, press Y. Otherwise press any other key, and you'll be prompted to adjust the paper in your printer. Knowing just where to position the paper will take a little trial and error the first few times —

practice with a few "dummy" forms before setting out to build your masterpiece.

Use the arrow keys to move the cursor around the design screen. Rather than displaying a block cursor, *Forms* uses dashes at the top and left side of the screen to show you where the cursor is positioned — it is at the intersection of these short lines. Hold the SHIFT key and press any arrow key to move to the corresponding edge of the screen.

Forms gives you a working area 52 columns wide by 101 lines deep for creating your form. (The vertical area is broken into four parts — in other words, the program gives you four full screens on which to work.) Each character position on the screen is unique — when the cursor is where you want it, press the key for the appropriate graphic element (see Figure 1). The elements supported allow you to create a wide variety of lines and boxes for your form. As you add characters, the cursor position is not automatically changed; you'll need to move the cursor for each character you add. The actual cursor position, however, is always available at the bottom of the screen.

To add text at the current cursor position, press the CTRL key and enter the text you want. You can backspace over mistakes by using the left arrow — in the text mode, the left arrow does not update the graphics-screen cursor position. The text is represented on the screen by triangles (the CoCo 3 does not support a 52-column video mode). However, what you type appears at the bottom of the screen as you enter it. As you type, the cursor dashes are updated, making it easier to correctly position the text. When your text is complete, press CTRL again, or press ENTER. You are automatically returned to the graphics mode.

For printing purposes, *Forms* sets the printer linefeed to half its normal distance. For this reason, always skip a line between successive lines of text. However, graphics lines may be added between these lines with no problem.

To erase the graphic element or character at the current position, just press E. To minimize screen clutter, it is important that you never place a new character over an existing one — always erase the old character or element first.

As I mentioned earlier, *Forms* supports four screens for a total of 101 lines. When you have completed the top fourth of your form (Screen 1), press S to move to the next

See **Forms** on Page 8

Animation Tutorial

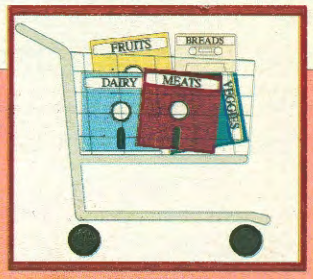
Key Frame Transformations

One of the earliest techniques used for computer animation is *key-frame interpolation*, a process for creating the intermediate visual frames between the critical positions (*key frames*) in an animated sequence. Key-frame interpolation was intended to replace the "in-betweening" ('tweening) used by countless animators for hand animation. As you might guess, 'tweening is a means of smoothing the movement of an animated figure from one key position to the next.

For many reasons, two-dimensional computer 'tweening was not very successful. The result is that such simplistic techniques gave computers a bad name in the character-animation industry. (Three-dimensional key-frame interpolation helped to restore the computer's reputation. A quick look at some of today's television commercials shows computer animation can be quite impressive.) Still, two-dimensional key-frame interpolation is a fascinating computer animation technique in its own right. While not particularly useful for traditional character animation, it works well for less well-defined objects and abstract shapes. It is particularly effective for transforming one graphics object into another (say, Africa into a Coke bottle).

See **Key Frames** on Page 12

CoCo Goes to Market
See Page 4



In this issue:

THE RAINBOW

Editor and Publisher
Lawrence C. Falk
Managing Editor Cray Augsburg
Associate Editor Sue Fomby
Submissions/Reviews Editor
Julie Hutchinson
Technical Editor Greg Law
Technical Assistant Ed Ellers
Contributing Editors Tony DiStefano, Martin Goodman, M.D., Eddie Kuns
Art Director Debbee Diamond
Designers Sharon Adams, Heidi Nelson
Consulting Editors Judi Hutchinson, Laurie D. Falk

Falsoft, Inc.

President Lawrence C. Falk

General Manager Peggy Lowry Daniels
Asst. General Mgr. for Finance
Donna Shuck
Admin. Asst. to the Publisher
Ellen Patterson
Editorial Director John Crawley
Director of Creative Services
O'Neil Arnold
Bookkeeper/Dealer Accounts
Beverly Quick
Asst. Gen. Manager For Administration
Tim Whelan
Corporate Business Technical Director
Calvin Shields
Customer Service Manager
Beverly Bearden
Chief of Printing Services Melba Smith
Business Assistant Shannon Yoffe
Chief of Building Security and Maintenance
Lawrence Johnson
Asst. General Manager for Advertising, Development and Production
Ira Barsky
Advertising Representatives Belinda Kirby, Kim Lewis
Advertising Assistant Carolyn Fenwick
(502)228-4492

For RAINBOW Advertising and Marketing Office Information, see Page 27.

THE RAINBOW is published every month of the year by FALSOFT, Inc., The Falsoft Building, 9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059, phone (502) 228-4492. **THE RAINBOW**, **RAINBOWfest** and **THE RAINBOW** and **RAINBOWfest** logotypes are registered trademarks of FALSOFT, Inc. ● Second class postage paid Prospect, KY and additional offices. USPS N. 705-050 (ISSN No. 0746-4797). POSTMASTER: Send address changes to **THE RAINBOW**, P.O. Box 385, Prospect, KY 40059. Authorized as second class postage paid from Hamilton, Ontario by Canada Post, Ottawa, Ontario, Canada. ● Entire contents copyright © by FALSOFT, Inc., 1992. **THE RAINBOW** is intended for the private use and pleasure of its subscribers and purchasers and reproduction by any means is prohibited. Use of information herein is for the single end use of purchasers and any other use is expressly prohibited. All programs herein are distributed in an "as is" basis, without warranty of any kind whatsoever. ● Tandy, Color BASIC, Extended Color BASIC and Program Pak are registered ®trademarks of the Tandy Corp. ● Subscriptions to **THE RAINBOW** are $31 per year in the United States. Canadian rates are U.S. $38. Surface mail to other countries is U.S. $68, air mail U.S. $103. All subscriptions begin with next available issue. ● Limited back issues are available. Please see notice for issues that are in print and their costs. Payment accepted by VISA, MasterCard, American Express, cash, check or money order in U.S. currency only. Full refund after mailing of one issue. A refund of 10/12ths the subscription amount after two issues are mailed. No refund after mailing of three or more magazines.

A Trip to Yesteryear

Our anniversary has come and gone, and this time of the year for THE RAINBOW always gets me thinking about the past. This is probably as good a time as any to talk about a couple of the things which have been sort of secrets regarding our little magazine. These are not big-time secrets — neither Oprah nor Sallie Jessie would care a whit about them — but every year at this time I tell myself I should mention them, then something more important comes up and they get shoved aside.

Our biggest secret involves our name. It was a subject of much debate in the earlier years when Jim Reed was the managing editor. He called the magazine RAINBOW, while I insisted the name was *THE* RAINBOW. Reed would write columns, letters and make announcements at RAINBOWfests about RAINBOW; and I would write my column, answer my letters and make my announcements about *THE* RAINBOW.

I asked Jim (who by the way is one of the founding members of the Louisville palindrome society) to come into my office one day, handed him the latest copy of the magazine, and asked him what its name was.

"RAINBOW," he replied.

"*THE* RAINBOW," I said.

"But, Lonnie, *I've* been calling it just RAINBOW for years."

"But, Jim, *I've* been calling it *THE* RAINBOW longer."

Anyone who can think up *YADDAY* (which Reed claims to have done) and similar palindromes doesn't give up on words easily.

"But just RAINBOW sounds better," he told me.

"But *THE* RAINBOW is its *name*."

"Well, I really think we should consider changing it, then. Especially since that is what I've been calling it."

"Except, Reed," I said, "all of our forms and all of our magazines and all of our bills are printed with a *the*. If we had to reprint them, it would cost a bunch of money."

"That's the trouble with you. Since you became a publisher, you worry about money instead of how things *sound*. Just RAINBOW sounds better."

"Well, Jim, I suppose we *could*. That is, if you don't mind waiting for your paycheck until we reprint all our materials."

The discussion seemed to end right there.

*****

Speaking of ends, look at the little thingie ("thingie" is a word coined by my daughter, Laurie) right above this paragraph. It is called a *star-dash*. It is five asterisks centered in a column and is a generally accepted term in the world of typesetting. But not here, it would seem.

Our first graphic artist, Sally Nichols, once came to me and asked how I wanted to set off several subjects in an article I wrote.

"Just star-dash them," I told her.

"Who-what?"

"Star-dash them."

And she walked away. Back an hour later, she had drawn a dash in the shape of an enlongated star.

"Is this what you want?"

*****

A better story about Sally is the time we got our first stat camera, a pretty heavy piece of equipment about 5 feet high and 3 feet deep with a sort of portable darkroom built in. The salesman who sold it to us had to have three men deliver it and said he would be by later in the day to teach us, particularly Sally, how it worked.

He showed up, talked to Sally and me for a minute or two, and then turned to her and said something like "Come on, Little Girl, and I'll show you how this works."

A full-fledged graduate of the University of Louisville's design program, Sally was not to happy to be a "Little Girl." But she and the salesman went off to the corner we had assigned to the camera.

When I came by, the salesman was "Little Girl"-ing Sally all over the place, and I could tell she was not pleased. An almost interminible hour passed and the salesman was finishing up when he said the last thing he needed to do was level the camera.

"OK, Little Girl," he told her, "you get down here and turn the little screw-in feet while I hold the camera up." Sally laid down on the floor, and he groaned and picked up the heavy camera. While he held the camera in the air, Sally said: "Oh, Wally, now where are those feet? This Little Girl is *so* confused."

Sally was never "Little Girl" again.

*****

It *is* true that long ago we ran a question-and-answer column by one of our technical people, Ed Ellers.

"What'll we call it?" Jim Reed asked me.

Ed's technical thoughts were sometimes somewhere in the stratosphere. On one occasion, I had hooked a new monitor to my CoCo and was having problems with RFI. I called Ed in to ask him why, only to be regaled by a lecture on everything from bandwidths to NTSB scan rates.

All I wanted was an answer. So, after about five minutes of this, I became impatient and finally said, "Earth to Ed, can you just *fix it*."

The column, thus, was named "Earth To Ed," somewhat over Ed's objection; but around Falsoft, the nickname "Earth" has stuck to this day.

I hope you've enjoyed my little trip down memory lane.

— Lonnie Falk

Letters to the RAINBOW

The Bottom Line

Editor:

Thank you for your continued support of our beloved CoCos. At first I was disappointed with your new format. After thinking it over, however, I realized THE RAINBOW could be printed on toilet paper and I would be happy. Just keep on publishing our only source of news.

Carla Sheridan
147 Lake Street
Bellingham, MA 02019

OS-9 and *Multi-Vue*

Editor:

I would like information about OS-9 Level II and *Multi-Vue* — our nearest Radio Shack doesn't have these products anymore. Also, where can I purchase these products?

Terence Holmes
211 Long Street
Leland, MS 38756

OS-9 is an operating system, a group of programs that controls how your computer operates. It offers more features and more control than is normally available through Disk BASIC. Multi-Vue *is a graphical user interface (GUI) intended to make OS-9 Level II a bit more user-friendly. Both products should be available through Tandy's Express Order system — you can call (800) 321-3133 for more information.*

Wants to Add EARS

Editor:

While reading through my back issues of THE RAINBOW, I saw numerous advertisements for a product called EARS. Is this product (or an equivalent) still available and, if so, where?

S. Remin
14 Wellington Road
Clayton, Victoria 3168
Australia

EARS, a hardware/software product designed to allow the CoCo to accept verbal input from a user, was originally marketed by Speech Systems. We know of no company currently marketing EARS. Perhaps another reader may be able to offer more assistance.

Music Program for the CoCo 3

Editor:

To all CoCo musicians and/or music programmers: help! For months I have been searching (so far in vain) for a CoCo 3 music program that supports onscreen music staves, chord input and note transposition as well as editing, playback and printing of music. It would be even better if it supported adding text to music (e.g., lyrics, guitar chords, etc.) and MIDI connection facilities. I have seen these features for PCs, and all the kids at our local Primary School have access to such features on their Apples. Why can't we get them on our illustrious CoCo? Any comments, advice or offers are most welcome. I think I've picked all the brains down here!

Keva Lloyd
8-12 Gallagher Street
Sea Lake 3533
Australia

We published a program by George Quellhorst in the April 1991 issue (Page 10) that supports onscreen staves. If using OS-9 Level II is not a problem, consider UltiMusE III *from Kala Software (3801 Brown Bark Drive, Greensboro, NC 27410) — it offers most of the features you mention and more. In the Disk BASIC market is* Lyra. *However, we are uncertain as to the current availability of this product.*

Make OS-9 Easy to Use

Editor:

I want to shout a super-hearty hallelujah in response to John Perry's comments in the April 1992 issue. I went to one of the Atlanta CoCofests and spent several hours talking to folks offering the new computers, begging them to make OS-9 easy to use. All I got were blank stares.

There is a severe dearth of CoCo hackers who remember when they *weren't*

omniscient. There is no doubt that OS-9 and OSK are very powerful systems, but they will always be hidden in a corner if they can't be used very easily be someone other than hackers.

Jim LaLone
9835 Standifer Gap Road
Ooltewah, TN 37363

User-Friendly Software

Editor:

Why are so many software producers letting programers cut them out of five to eight percent of their potential market? I don't know how many times I have read ads for programs that seem to be just what I want, but four little words tell me I can't use then: mouse or joystick required.

Some years ago I was involved in an automobile accident. I have recovered well, but my fine control of my hands and fingers is not all that great. Through 30 months of hard practice, I have gotten my typing speed back to about ⅔ of what it once was. The mouse and joystick, however, are still useless to me.

Have programmers forgotten about the arrow keys? I doubt it. In fact, I'll bet the vast majority give those arrows a real workout while they are writing their programs!

There are thousands of us out here (many in far worse condition than I). Can you who market these programs really afford to ignore this potential expansion of your customer base?

I have written a number of game and graphics programs for my grandsons — enough so that I can tell you it is not that hard to give users a choice of which control to use. Some of my programs have both a "text cursor" and a "mouse cursor." In these programs, when the keyboard is selected, pressing F1 puts the arrow keys into the Text mode and F2 puts them into the mouse mode; the CLEAR key is the "button."

And, please, don't forget diagonals. When both a vertical arrow and a horizontal arrow are pressed, the cursor should travel the diagonal path between the two.

Fred Rickards
P.O. Box 794
Trinity, NC 27370

Desperate for Help with *Hyper I/O*

Editor:

I have enjoyed THE RAINBOW for many years, and have found it to be a very helpful addition to my library of computer books and magazines. I want to thank all the people who have replied to my distress calls (through your great publication) for help and also THE RAINBOW staff, which has offered me help with software and hardware problems. It is rare to find such customer service these days, and I just thought you should know there are a lot of people out here who think you're doing a great job.

I would appreciate any help from anyone who uses (or has used in the past) a Burke & Burke hard-drive system with *Hyper I/O*. I picked up the system from a former CoCo owner. I am looking for any patches or programs that have been written to work with this hardware/software combination and would allow me to run several of my software packages with *Hyper I/O*.

Mychel Holtry
695 Park Avenue, Apt. 216
Idaho Falls, ID 83402
(208) 524-9027

Printing Pictures

Editor:

I hope you can help me. I have a 64K Color Computer 2 with a black-and-white TV, an FD-502 disk drive and a DMP-132. I can get the printer to print words on paper, but it won't print pictures, even though I can view the graphics onscreen. I have tried to get the people at Radio Shack to help, but they don't know how to do this either.

Daniel Miller
Route 1, Box 147
Church Hill, TN 37642

Printing PMODE graphics images with a printer requires a special program called a screen dump. See "Speedy PMODE Screen Dump" (THE RAINBOW, May 1992, Page 1) by Cray Augsburg for an in-depth look at printing these images.

Looking to Switch to OS-9

Editor:

We have a cassette-based CoCo 3 with 128K. We want to get a disk drive and use the OS-9 operating system, but we are very unsure as to where to go or what would be best for us.

When it comes to computers, we are all but lost. We've had ours for some time and also have *Telewriter-64* (with which we are not very happy). We use the CoCo for ministry records and mailing lists, auto-repair records and parts inventory, and myriad other personal and small-business uses. Can you give us some direction and sound advice?

Tiny and Wanda Brown
#13B Mystery Place
P.O. Box 2172
Alma, AR 72921

The OS-9 operating system, as we explained in our response to Terence Holmes above, is a group of related programs that manages the way the computer operates and the way we communicate with it. In the past, this operating system has been touted as not too user-friendly. The simple truth is, OS-9 is different than Disk BASIC and requires a little relearning. Since you have not yet experienced Disk BASIC, this relearning should not greatly affect you.

OS-9 has often been considered a hacker's operating system (as opposed to a user's system) since there are so many utilities and relatively few applications. But the necessary applications — word processors, filing programs and a spreadsheet — are available. In addition, a great deal of shareware and freeware is available through Delphi and other telecommunications services. Perhaps other readers will write you with their suggestions.

THE RAINBOW welcomes letters to the editor. Mail should be addressed to: Letters to Rainbow, The Falsoft Building, 9509 U.S. Hwy 42, P.O. Box 385, Prospect, KY 40059. Letters should include the writer's full name and address. Letters may be edited for clarity or to conserve space.

Letters to the editor may also be sent to us through our Delphi CoCo SIG. From the CoCo SIG> prompt, enter RAI to get to the Rainbow Magazine Services area of the SIG. At the RAINBOW> prompt, enter LET to reach the LETTERS> prompt, then select Letters for Publication. Be sure to include your complete name and address.

Flash UPDATE

Faster Throughput for the CoCo 3

Originally developed as a low-power (CMOS) alternative to the Motorola 6809, the Hitachi 6309 microprocessor has been around for some time. Until recently, however, few people knew of its hidden enhancements (undocumented features) and the benefit they might offer to users of the CoCo 3.

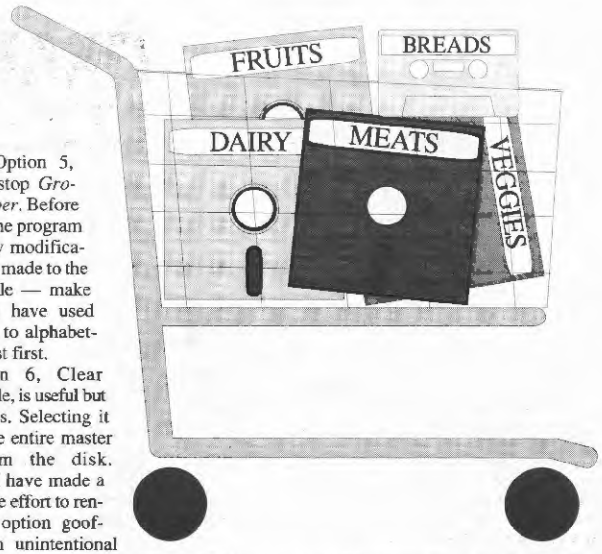
The 6309 is pin-for-pin compatible with the 6809. In addition to extra registers and support for high-speed block moves (four times as fast as the 6809), the 6309 sports two modes of operation. In the Emulation mode, the 6309 acts as if it were a 6809 but includes new instructions for access to the extra registers. In the Native mode, however, the 6309 executes instructions in up to 35 percent fewer instruction cycles.

Chris Burke (of Burke & Burke) learned of these hidden features earlier this year, and at the Chicago CoCofest (May 1992) introduced PowerBoost. This product includes a 63B09E and socket with which you replace the 68B09E in the CoCo 3, and software for Disk BASIC and OS-9 Level II that allows you to take advantage of the 6309's Emulation mode.

Early reports on PowerBoost indicate processor throughput increases of 10 to 50 percent, depending on the operation being performed — an average of 15 to 20 percent is expected. Because OS-9 is a modular operating system (in contrast to Disk BASIC) throughput increases are more visible with OS-9.

Feature Program

Go Grocery Shopping With the Coco



Beans, oranges, tofu, toilet tissue, pork chops . . . whew! This shopping list is horribly disorganized. If only there was a quick and easy way to alleviate some of the burden of weekly grocery shopping . . .

Your CoCo 3, along with a disk drive and printer, can do just this. *Grocery Helper* is a menu-driven program that lets you select from an alphabetical list the items you want to purchase and print them in an orderly form.

After you enter the program listing, save *Grocery Helper* to a fresh disk and run it. Make sure to keep the disk in the drive, though, so the CoCo can use it to store your data. The program stores your data in a file named `LIST.DAT`. If it can't find this file on the disk in Drive 0, the program assumes you haven't set up the master product list yet and takes you immediately to Option 1 (discussed below). However, if Grocery Helper does find a file with this name, a title screen appears and you see a menu of the programs six options.

Option 1, Add Items to Master File, allows you to add, delete, or change items in `LIST.DAT`. This master file holds up to 360 items of up to 18 characters each.

Grocery Helper supports nine product categories (see Figure 1). The category currently selected is shown on the third screen line. To move to the next category, hold the SHIFT key and press the up arrow. Similarly, use SHIFT-down arrow to move back one category.

The first item in the current category (or the first slot if there are no items in the category) is highlighted. The four arrow keys allow you to move the highlight bar over each listed item. To enter a new item or add to an existing one, position the bar and type the desired characters. In a slot that lists an item, *Grocery Helper* automatically positions the text cursor at the end of the existing characters. When you have finished entering an item, press ENTER (to move to the next slot) or any arrow key. Since the left arrow is used for cursor control, I wrote the program to iuterpret the CLEAR key as a backspace. To clear all the characters in the highlighted slot, press SHIFT-CLEAR.

While you are getting used to the program, it's not hard to forget one or more of the function keys. Press CTRL-H to call up a help window that outlines the most important functions. Press CTRL-X to return to the Main menu.

Option 2, Alphabetize Items, sorts the data in the master file in alphabetic order. *Always use this option immediately after you create or modify the master file.* Otherwise, the program will not correctly interpret the file, reudering the data useless.

Option 3, Select Items, is where you'll spend most of your time with *Grocery Helper*; use this option to select the items you want printed on your current shopping list. After you select this option, a list containing all the items in the file is displayed. The items are organized in alphabetic order, regardless of the category under which they fall. Use the up and down arrows to move the cursor through the list, and press ENTER to select an item. (Selected items are indicated by inverted text.) Pressing ENTER while the cursor is on a previously selected item cancels that selection.

To move to the top of the list, use SHIFT-up arrow. To go to the bottom, use SHIFT-down arrow. You can also press a single letter to move the cursor to the first item that begins with that letter in the list. As with Option 1, CTRL-H calls a help screen, and CTRL-X takes you back to the Main menu.

Option 4, Print List, is self-explanatory. To achieve a compact printout, I've inserted a control code to print the list in Tandy's microfont. If your printer (or eyes) don't support this small print, simply delete everything up to the `RESTORE` command in Line 4002 of the program, and change Line 4022 to `4022 GOTO 30`.

1. Bread and Cereal
2. Canned Goods
3. Dairy
4. Frozen Foods
5. Hygiene
6. Meats, Fish & Poultry
7. Miscellaneous
8. Paper/Detergent
9. Produce

Figure 1: Product Categories

Use Option 5, Exit, to stop *Grocery Helper*. Before ending, the program saves any modifications you made to the master file — make sure you have used Option 2 to alphabetize the list first.

Option 6, Clear Master File, is useful but dangerous. Selecting it erases the entire master file from the disk. Though I have made a reasonable effort to render this option goof-proof, an unintentional erasure can still result — please be careful.

One final piece of advice: When you are entering items (Option 1) with similar names, enter them "last name first," placing a comma between the words. For example, enter green beans as `BEANS, GREEN`. (The space is not required after the comma.) Then when the list is alphabetized, the item will appear in an ordered section of the master file — the various types of beans are listed together, making selection easier. When you print the list, *Grocery Helper* interprets the comma as a delimiter and rearranges the item name (e.g., to `GREEN BEANS`). When using this handy feature, be particular with your use of the comma.

David LeBlanc is a certified engineer who plans to attend the Technical University of Nova Scotia for a degree in electrical engineering. In addition to working with computers, David enjoys participating in various sports, and his hobbies include reading and listening to music.

CoCo 3

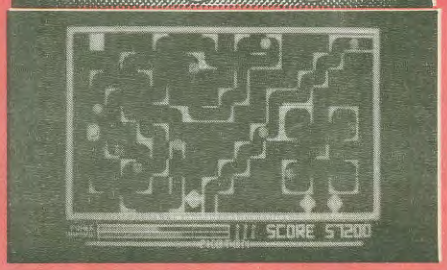
The Listing: `GROCERY`

```
0 DATA"Bread/cereal","Canned goo
ds","Dairy","Frozen foods","Hygi
ene","Meats, fish, and poultry",
"Miscellaneous","Paper/detergent
","Produce",9,2,7,4,3,6,1,8,5
1 WIDTH40:ONBRKGOTO8:PMODE0,1:PC
LEAR1:CLEAR8000:DIMF(361),I(40),
P(360),I$(360),P$(40):FORK=1TO9:
READG$(K):NEXT
3 CLS:LOCATE13,2:ATTR0,0,U:PRINT
"Grocery Helper";:ATTR0,0:LOCATE
13,5:PRINT"by Dave LeBlanc":LOCA
TE8,23:PRINT"(C) 1992 Rainbow Ma
gazine";
4 'GROCERY HELPER BY D. LEBLANC
5 'COPYRIGHT (C) 1992 FALSOFT
6 GOTO10
8 RUN
10 POKE65496,0:OPEN"D",#1,"LIST.
DAT",21:FIELD#1,18ASA$,3ASB$:POK
E65497,0
20 IFLOF(1)=0THENCLOSE:FORK=1TO7
50:NEXT:GOTO1000
22 LOCATE15,21:PRINT"Disk access
":POKE65496,0:FORK=1TO360:GET#1,
K:I$(K)=A$:P(K)=VAL(B$):NEXT:CLO
SE:POKE65497,0
24 LOCATE13,21:PRINT"Processing
data":FORK=1TO360
25 IFINSTR(I$(K),STRING$(18,32))
=1THENI$(K)="":GOTO28
26 IFASC(RIGHT$(I$(K),1))=32THEN
I$(K)=LEFT$(I$(K),LEN(I$(K))-1):
GOTO26
28 NEXT
30 ONBRKGOTO30:SOUND1,1:ATTR0,0:
CLS:LOCATE16,2:PRINT"MAIN MENU":
FORK=1TO8:LOCATE6,3+K:ATTR3,2:PR
INTSTRING$(28,32);:NEXT:LOCATE7,
5:PRINT"1. Add items to master f
ile";:LOCATE7,6:PRINT"2. Alphabe
tize items";:LOCATE7,7:PRINT"3.
Select items";:LOCATE7,8
31 PRINT"4. Print list";:LOCATE7
,9:PRINT"5. Exit";:LOCATE7,10:PR
INT"6. Clear master file";:LOCAT
E12,13:ATTR0,0:PRINT"Select opti
on > ";
32 EXEC44539:A$=INKEY$:IFVAL(A$)
<1ORVAL(A$)>6THENSOUND1,1:GOTO32
ELSEPRINTA$:EXEC43345:ONVAL(A$)G
OTO1000,2000,3000,4000,5000,6000
999 'MODIFY MASTER FILE MODULE
1000 ONBRKGOTO1000:C=0:CLS:ATTR0
,0:LOCATE10,0:PRINT"MODIFICATION
 MODULE":PRINT" CTRL-X to exit
       CTRL-H for help";:LOCATE1
,2:IFG=0THENG=1
1005 PRINT"Group #"+STR$(G)+": "
+G$(G):PRINT" "+STRING$(38,"-");
1010 FORK=1TO40:I(K)=0:NEXT:A=0:
FORK=(G-1)*40+1TO(G-1)*40+40:IFI
$(K)=""THENNEXTELSEA=A+1:I(A)=K:
NEXT
1011 IFC=0THENIFA=40THENC=1ELSEC
=A+1
1012 ATTR0,0:FORK=1TO40:IFK/2<>I
NT(K/2)THENB=3+((K+1)/2)ELSEB=3+
(K/2)
1014 IFK/2<>INT(K/2)THENLOCATE1,
8 ELSELOCATE21,8
1015 IFK=C THENATTR2,4
1016 PRINTI$(I(K))+STRING$(18-LE
N(I$(I(K))),32);
1017 IFK=C THENATTR0,0
1018 NEXT
1025 C$=I$(I(C))
1030 IFC/2<>INT(C/2)THENB=3+((C+
1)/2)ELSEB=3+(C/2)
1032 IFC/2<>INT(C/2)THENLOCATE1,
B ELSELOCATE21,8
1034 ATTR2,4:PRINTC$+STRING$(18-
LEN(C$),32);:ATTR1,0:LOCATE39,0
1036 GOTO1050
1039 'PRINT NORMAL ITEM [ENTRY:
C-POS
1040 IFC/2<>INT(C/2)THENLOCATE1,
B ELSELOCATE21,8
1042 ATTR0,0:PRINT C$+STRING$(18
-LEN(C$),32);:IFI(C)>0THEN1044EL
SEIFC$=""THENRETURNELSEFORK=(G-1
)*40+1TO(G-1)*40+40:IFI$(K)=""TH
ENI(C)=K:A=A+1:I$(K)=C$:RETURNEL
SENEXT:STOP
1044 I$(I(C))=C$:RETURN
1050 IFPEEK(341)=247THEN1100
1052 IFPEEK(342)=247THEN1150
1054 IFPEEK(343)=247ORPEEK(344)=
247THEN1200
105B A$=INKEY$:IFA$=""THEN1050
1059 IFA$=CHR$(189)ORA$="^"ORA$=
CHR$(8)ORA$=CHR$(9)ORA$=CHR$(10)
THEN1050
1060 IFA$="_"THEN1350
1062 IFA$="["THEN1360
1064 IFA$="X"ANDPEEK(342)=191GOS
UB1040:GOTO30
1066 IFA$="H"ANDPEEK(342)=191THE
N1380
1068 IFA$=CHR$(12)THEN1300
1070 IFA$="\"THEN1325
1076 IFA$=CHR$(13)THEN1310
1078 IFLEN(C$)=18THENSOUND1,1:GO
TO1050
1080 EXEC43345:C$=C$+A$:GOTO1030
1100 IFI(C)=0ORC$<>""THEN1110ELS
EIFC<3THENIFC+38>A THEN1050ELSEI
$(I(C))="":C=C+38:GOTO1010
1102 I$(I(C))="":C=C-2:GOTO1010
1110 IFC<3THENIFC+37>A THEN1050E
LSEGOSUB1040:C=C+38:GOTO1025
1112 GOSUB1040:C=C-2:GOTO1025
1150 IFI(C)=0ORC$<>""THEN1160ELS
EIFC>38THENI$(I(C))="":C=C-38:GO
TO1010
1152 IFC+2>A THEN1050ELSEI$(I(C)
)="":C=C+2:GOTO1010
1160 IFC>38THENGOSUB1040:C=C-38:
GOTO1025ELSEIFC+1>A THEN1050ELSE
```

SUNDOG SYSTEMS

PHOTON

NEW!



Energy is everything; your home world depends on it. However, someone or something is slowly siphoning it away. As your world's champion, you must climb into the experimental Power Tank to challenge this nemesis and his minions. Your key lies with the ability to teleport solid mass. Use this to manipulate and explore the endless stronghold of the enemy, and to exploit the free-floating DUPES (Dense Units of Photon Energy) to destroy the menacing Plasma Droids. Be cautious, though; those DUPES can be deadly, too! Photon, a fantastic new arcade game for your Coco3, contains spectacular 320x200 resolution, 16 color graphics, ultra-smooth 60 Hz animation, and loads of real-time music and sound effects. It will send your mind racing over endless possibilities, requiring quick decisions and reactions. Quite simply, Photon is incredibly addictive; it will deliver hours of excitement. Will you become your world's greatest hero, or just another energy slave? Req. 128K CoCo 3 and disk drive.

$34.95

GRAF 2.0 EXPRESS

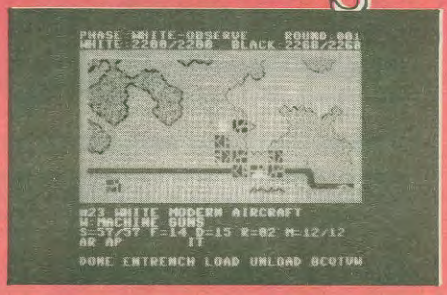
NEW!

GrafExpress 2.0 is a complete graphics and music programming environment. From the beginner to the accomplished professional, you can use GrafExpress to create lightning fast arcade games, graphic applications and utilities, and windowing multimedia demonstrations! The GrafExpress package includes two incredible systems. GrafExpress 16 works on all monitor types and offers support in 12 graphic resolutions (from 128x192 to 320x225). GrafExpress 256 offers 6 resolutions (from 128x192 to 160x225 on a composite monitor) in an astounding 256 colors! Ever see a CoCo do that before? Both systems include standard graphics commands (CIRCLE, FILL, etc.) that blow away the competition. For example, the BOX command peaks out at over 2 MegaPixels/second; that's 300 times faster than BASIC! 255 separate sprites of up to 100x100 pixels each are supported with window clipping and high-res pixel level collision checking. The 8-octave/4-voice music synthesizer has independent envelope, waveform, and volume controls, a 7+ KHz sampling rate, and much more. Other features include text/graphics mixing, different font sizes, fast window copying and scrolling, picture save/load, easy implementation from both BASIC and assembly language, multiple screen animation, and support for 128K/512K, double speed, and the high-res joystick interface. The package also contains support programs that are worth the purchase price of GrafExpress alone! These include an introductory demo, a picture editor, a waveform editor, and an art program that supports 256 colors! GrafExpress also comes with a 50 page manual that fully explains all of its incredible features. If you do any graphics programming or simply want to see what your little CoCo is capable of, GrafExpress is a must! Req. 128K CoCo 3 and disk drive.

$34.95

War Monger

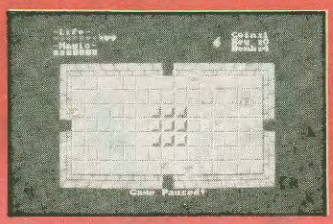
NEW!



The world is in unrest. Power-hungry villains and evil warlords are readying their forces. It falls to you to lead your people against these armies, and only your best strategic plans can save the day. Fight the good fight in any era or locale. Play a simple game of capture the flag armed with water balloons, or climb into the cockpit of a 100 foot high armored warrier. Explore the deepest dungeons, defend your galaxy, or create your own scenarios with this incredible war game construction set/simulator. Your imagination is your only limit. You will deploy your forces with total control over hostile terrain while you scroll a graphic bird's-eye window over an immense world. War Monger has terrific 320x200 resolution, 16 color graphics and includes a tile editor to create or edit your own. Play against the computer, battle with another player, or simply watch the computer plot against itself. The enemy is everywhere. Are you ready to take on the challenge as the War Monger? Req. 128K CoCo 3 and disk drive.

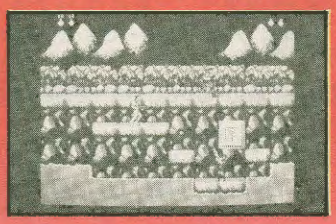
$29.95

The Quest for THELDA



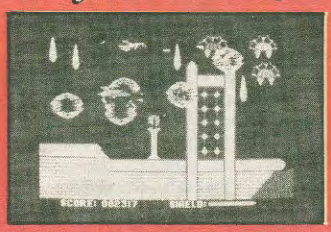
An immensely popular 128K CoCo 3 arcade/adventure. Over 500 screens of fast fantasy action and puzzle solving. Great graphics and sound effects. **$34.95.** Hint book only $4.95.

★ THE ★ CONTRAS



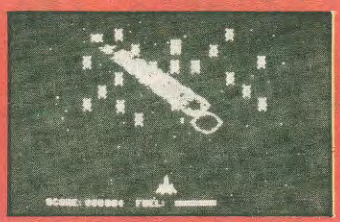
A 512K two player futuristic combat arcade game. Full screen 320x225 hardware scrolling and smooth animation. Back-ground music score and sound effects! 512K Coco3 only. **34.95.** Shipping soon!

Crystal City



This was THE game of '91! Ultra-fast space action with hardware scrolling on a 128K CoCo 3. Wild sound effects and over 30 MegaBytes of amazing graphics! **34.95.**

Zenix

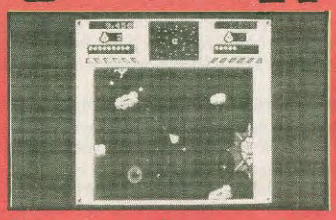


Lightning fast arcade game for the 128K CoCo 3. Terrific 320x225 graphics, back-ground music score and sound effects, and out-of-sight game play. **$29.95.**

ALSO AVAILABLE

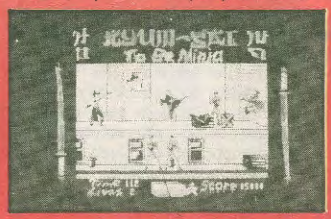
Warrior King	
CoCo 3	$29.95
In Quest of the Star Lord	
CoCo 3	$34.95
Hint Sheet	$ 3.95
Hall of the King 1, 2 or 3	
CoCo 1 - 3	$29.95 ea.
Hall of the King Trilogy	$74.95
White Fire of Eternity	
CoCo 1 - 3	$19.95
Dragon Blade	
CoCo 1 - 3	$19.95
Champion	
CoCo 1 - 3	$19.95
Paladin's Legacy	
CoCo 1 - 3	$24.95

Sinistaar



Everyone loves this 512K arcade game. 3 disks packed with spectacular graphics and eerie background digital sound effects. 512K CoCo 3 only. **$34.95.**

KYUM-SAI TO BE NINJA



The best selling 128K CoCo 3 martial arts arcade game. Now available in both RS-DOS and OS-9 versions. Play the incredible combat experience you've been missing under the operating system of your choice! **$29.95.**

SOUNDTRAX

A polyphonic digital sound sequencing system for your 128K/512K CoCo 3 with a user-friendly point-and-click graphic editor. Create music scores with your own sounds or from the many we provide. **$34.95.**
Sample instrument disks: 6 sides of sampled sounds/instruments. Only $12.95 each or $29.95 for all three.

Visa, Mastercard, Check, Money Order, and COD (USA only, please) accepted. All foreign orders must be sent in US currency Money Orders. Include $2.50 for shipping in USA and canada. $5.00 Foresign. $3.00 extra for COD orders. PA residents add 6% sales tax. Dealer inquiries welcome. Authors, we're looking for new software!

Sundog systems

SUNDOG SYSTEMS

P.O. Box 766 • Manassas, VA 22111

703/330-8989

```
GOSUB1040:C=C+2:GOTO1025
1200 IFI(C)=0ORC$<>""THEN1210ELS
EIFC/2<>INT(C/2)THENIFC=A THEN10
50ELSEI$(I(C))="":C=C+1:GOTO1010
1202 I$(I(C))="":C=C-1:GOTO1010
1210 IFC/2<>INT(C/2)THENIFC$=""T
HEN1050ELSEGOSUB1040:C=C+1:GOTO1
025
1212 GOSUB1040:C=C-1:GOTO1025
1300 IFC$=""THENSOUND1,1:GOTO105
0ELSEEXEC43345:C$=LEFT$(C$,LEN(C
$)-1):GOTO1030
1310 IFC$=""THENSOUNO1,1:GOTO105
0ELSEEXEC43345:GOSUB1040:C=C+1:I
FC>40THENC=1
1312 GOTO1025
1325 EXEC43345:IFI(C)=0THENC$=""
:GOTO1030ELSEI$(I(C))="":GOTO101
0
1350 GOSUB1040:G=G-1:IFG=0THENG=
9
1352 GOTO1000
1360 GOSUB1040:G=G+1:IFG>9THENG=
1
1362 GOTO1000
1380 GOSUB1040:LOCATE0,1:ATTR0,0
:PRINTSTRING$(40,32);:LOCATE10,1
:PRINT"Help screen engaged.";:FO
RK=1TO12:LOCATE7,6+K:ATTR3,2:PRI
NTSTRING$(26,32);:LOCATE34,7+K:A
TTR0,3
1381 PRINT" ";:ATTR0,0:PRINT" ";
:NEXT:LOCATE8,19:ATTR0,3:PRINTST
RING$(25,32);
1382 LOCATE8,8:ATTR2,4:PRINT"CTR
L";:ATTR3,2:PRINT" + ";:ATTR2,4:
PRINT"H";:ATTR3,2:PRINT" for HEL
P screen.";
1384 LOCATE8,9:ATTR2,4:PRINT"CTR
L";:ATTR3,2:PRINT" + ";:ATTR2,4:
PRINT"X";:ATTR3,2:PRINT" for MAI
N MENU.";
1386 LOCATE8,10:PRINT"ARROW KEYS
 move cursor.";:LOCATE8,11:ATTR2
,4:PRINT"CLEAR";:ATTR3,2:PRINT"
to backspace.";
1388 LOCATE8,12:ATTR2,4:PRINT"SH
IFT";:ATTR3,2:PRINT" + ";:ATTR2,
4:PRINT"CLEAR";:ATTR3,2:PRINT" t
o delete";:LOCATE9,13:PRINT"item
.";
1390 LOCATE8,14:ATTR2,4:PRINT"SH
IFT";:ATTR3,2:PRINT" + UP or DOW
N arrow";:LOCATE9,15:PRINT"moves
 to another group.";
1392 LOCATE8,17:PRINT"Hit any ke
y to resume...";
1394 LOCATE36,0:ATTR1,0:PRINT" "
;:EXEC44539:A$=INKEY$:ATTR0,0:LO
CATE1,1:PRINT"CTRL-X to exit
     CTRL-H for help";
1396 FORK=1TO15:LOCATE19,5+K:PRI
NT"  ";:NEXT:GOTO1010
1500 A$=INKEY$:IFA$=""THEN1500
1600 PRINTASC(A$)
1700 GOTO1500
1999 'ALPHABETIZE
2000 PRINT" Alphabetizing.":PRIN
T" Please wait a few moments..."
:LOCATE35,0:ATTR0,0,8:PRINT9;:AT
TR1,0:PRINTCHR$(8);:FORK=1TO360:
P(K)=K:NEXT:M=511
2010 FORK=1TO8:LOCATE35,0:ATTR0,
0,8:PRINT9-K;:ATTR1,0:PRINTCHR$(
8);:M=(M-1)/2
2020 IF360-M-1<0THEN2130ELSEM1=M
+1:I=M1
2030 I1=P(I):M2=I-M:J=1
2040 J1=M2-J+1:I2=P(J1)
2050 IFI$(I1))=I$(I2)THEN2100
2060 P(J1+M)=P(J1)
2070 IF J+M>M2 THEN2090
2080 J=J+M:GOTO2040
2090 P(J1)=I1:GOTO2110
2100 P(J1+M)=I1
2110 IF I+1>360THEN2130
2120 I=I+1:GOTO2030
2130 NEXTK
2140 GOTO30
3000 ONBRKGOTO3000:CLS:ATTR0,0:L
OCATE13,0:PRINT"SELECT MODULE":P
RINT" CTRL-X to exit          CTR
L-H for help":PRINT" "+STRING$(3
8,"-");
3006 FORK=1TO360:IFI$(P(K))=""TH
ENNEXT:SOUND1,1:GOTO30ELSEA=K
3010 ATTR0,0:B=A-1:FORK=1TO5
3012 LOCATE11,(6-K)*2+1:IFB=0ORI
$(P(B))=""THENPRINTSTRING$(18,32
);:NEXT:GOTO3020ELSEC$=I$(P(B))
3014 IFF(P(B))=1THENATTR2,4
3016 PRINTC$+STRING$(18-LEN(C$),
32);:IFF(P(B))=1THENATTR1,0:LOCA
TE39,0:ATTR0,0
3018 B=B-1:NEXT
3020 B=A:FORK=1TO6
3022 LOCATE11,(K*2)+11:IFB>360TH
ENPRINTSTRING$(18,32);:NEXT:GOTO
3030ELSEC$=I$(P(B))
3024 IFF(P(B))=1THENATTR2,4
3026 PRINTC$+STRING$(18-LEN(C$),
32);:IFF(P(B))=1THENATTR1,0:LOCA
TE39,0:ATTR0,0
3028 B=B+1:NEXT
3030 LOCATE8,13:PRINT"=>";:LOCAT
E30,13:PRINT"<-";:ATTR1,0:LOCATE
39,0
3040 IFPEEK(341)=247THEN3200
3042 IFPEEK(342)=247THEN3250
3043 K=PEEK(341)AND64:IFK>0THENW
$=""
3044 A$=INKEY$:IFA$=""THEN3040
3046 IFA$="@"ORA$=CHR$(189)ORA$=
"^"ORA$=CHR$(10)THEN3040
3048 IFA$=CHR$(13)THEN3300
3050 IFA$="_"THEN3006
3052 IFA$="["THENA=360:GOTO3010
3054 IFA$="X"ANDPEEK(342)=191THE
N30
3056 IFA$="H"ANDPEEK(342)=191THE
N3375
3058 IFPEEK(341)=191THENW$=W$+A$
 ELSEW$=A$
3059 IFLEN(W$)=1THENFORK=1TO360E
LSEFORK=A TO360
3060 IFINSTR(I$(P(K)),W$)<>1THEN
NEXT:SOUND1,1:W$="":GOTO3040ELSE
A=K:IFPEEK(341)<>191THENW$=""
3062 GOTO3010
3200 IFA-1=0ORI$(P(A-1))=""THEN3
040ELSEA=A-1:GOTO3010
3250 IFA+1>360THEN3040ELSEA=A+1:
GOTO3010
3300 IFF(P(A))=0THENF(P(A))=1ELS
EF(P(A))=0
3302 GOTO3010
3375 LOCATE0,1:ATTR0,0:PRINTSTRI
NG$(40,32);:LOCATE10,1:PRINT"Hel
p screen engaged.";:FORK=1TO17:L
OCATE7,4+K:ATTR3,2:PRINTSTRING$(
26,32);:LOCATE34,5+K:ATTR0,3
3376 PRINT" ";:ATTR0,0:PRINT" ";
:NEXT:LOCATE8,22:ATTR0,3:PRINTST
RING$(25,32);
3378 LOCATE8,6:ATTR2,4:PRINT"CTR
L";:ATTR3,2:PRINT" + ";:ATTR2,4:
PRINT"H";:ATTR3,2:PRINT" for HEL
P screen.";
3380 LOCATE8,7:ATTR2,4:PRINT"CTR
L";:ATTR3,2:PRINT" + ";:ATTR2,4:
PRINT"X";:ATTR3,2:PRINT" for MAI
N MENU.";
3382 LOCATE8,8:PRINT"UP arrow mo
ves cursor up";:LOCATE9,9:PRINT"
one item.";:LOCATE8,10:PRINT"DOW
N arrow moves cursor";:LOCATE9,1
1:PRINT"down one item.";
3384 LOCATE8,12:ATTR2,4:PRINT"SH
IFT";:ATTR3,2:PRINT" + UP arrow
moves";:LOCATE9,13:PRINT"cursor
 to top of items.";
3386 LOCATE8,14:ATTR2,4:PRINT"SH
IFT";:ATTR3,2:PRINT" + DOWN arro
w moves";:LOCATE9,15:PRINT"curso
r to end of items.";
3388 LOCATE8,16:ATTR2,4:PRINT"EN
TER";:ATTR3,2:PRINT" selects an
item.";:LOCATE8,17:ATTR2,4:PRINT
"A";:ATTR3,2:PRINT"-";:ATTR2,4:P
RINT"Z";:ATTR3,2
3390 PRINT" for letter search.";
:LOCATE8,18:ATTR2,4:PRINT"ALT";:
ATTR3,2:PRINT" for word search."
;
3392 LOCATE8,20:PRINT"Hit any ke
y to resume...";
3394 LOCATE36,0:ATTR1,0:PRINT" "
;:EXEC44539:A$=INKEY$:ATTR0,0:LO
CATE1,1:PRINT"CTRL-X to exit
     CTRL-H for help";
3396 FORK=1TO18:LOCATE7,4+K:PRIN
TSTRING$(27,32);:NEXT:GOTO3010
4000 PRINT" Make sure printer is
 on and positioned  to top of fo
rm and strike any key...":EXEC44
539
4002 POKE65496,0:PRINT#-2,CHR$(2
7)CHR$(77);:POKE65497,0:RESTORE:
FORK=1TO9:READA$:NEXT
4004 FORA=1TO3:FORB=1TO3:L(B)=0:
NEXT:FORB=1TO40:P$(B)="":NEXT:FO
RK=1TO3:READK(K):NEXT:FORK=1TO3:
FORL=(K(K)-1)*40+1TO(K(K)-1)*40+
40:IFF(L)=1THENL(K)=L(K)+1
4006 NEXTL,K:IFL(1)=>L(2)ANDL(1)
=>L(3)THENI=L(1):GOTO4012
4008 IFL(2)=>L(1)ANDL(2)=>L(3)TH
ENI=L(2):GOTO4012
4010 IFL(3)=>L(1)ANDL(3)=>L(2)TH
ENI=L(3)
4012 IFI=0THENNEXTA:GOTO30
4014 FORK=1TO3:B=0
4016 FORL=(K(K)-1)*40+1TO(K(K)-1
)*40+40:IFF(L)=0THENNEXTL:GOTO40
19ELSEB=B+1:C$=I$(L):Z=INSTR(C$,
","):IFZ=0THEN4018ELSEC$=MID$(I$
(L),Z+1)+" "+LEFT$(I$(L),Z-1)
4017 IFASC(LEFT$(C$,1))=32THENC$
=RIGHT$(C$,LEN(C$)-1):GOTO4017
4018 P$(B)=P$(B)+"___ "+C$+STRIN
G$(20-LEN(C$),32):NEXTL
4019 IFB<I THENFORL=B+1TO I:P$(L
)=P$(L)+STRING$(24,32):NEXT
4020 NEXTK:POKE65496,0:FORK=1TO
I:PRINT#-2,P$(K):NEXT:PRINT#-2,C
HR$(13):POKE65497,0:NEXTA
4022 POKE65496,0:PRINT#-2,CHR$(2
7)CHR$(19);:POKE65497,0:GOTO30
5000 PRINT" You sure? ";
5002 EXEC44539:A$=INKEY$:IFA$<>"
N"ANDA$<>"Y"THENSOUND1,1:GOTO500
2ELSEPRINTA$:EXEC43345:IFA$="N"T
HEN30
5004 PRINT" Saving memory to dis
k.":PRINT" One moment please..."
5006 OPEN"D",#1,"LIST.DAT",21:FI
ELD#1,18ASA$,3ASB$
5010 POKE65496,0:FORK=1TO360:LSE
TA$=I$(K):LSETB$=MID$(STR$(P(K))
,2):PUT#1,K:NEXT:CLS:END
6000 PRINT" You sure? ";
6002 EXEC44539:A$=INKEY$:IFA$<>"
N"ANDA$<>"Y"THENSOUND1,1:GOTO600
2ELSEPRINTA$:EXEC43345:IFA$="N"T
HEN30
6004 PRINT" NO RETURN beyond thi
s point!!":PRINT" Hit ENTER key
to clear disk and       restar
t. Hit ESC to abort..."
6006 EXEC44539:A$=INKEY$:IFA$<>C
HR$(13)THENSOUND1,1:GOTO6006ELSE
EXEC43345
6008 PRINT" Trashing file...":PO
KE65496,0:KILL"LIST.DAT":RUN
```

Product Review

KwikGen: Edit OS-9 Boot Files on the Fly

Have you ever tried to make a new boot disk with a single-drive system? Yeah, right! You'd probably rather have a tooth pulled with a pipe wrench. Or how many times have you wanted to make one small change to a boot file but found the only way to do it was to create a completely new boot disk from scratch?

Needless to say, creating or modifying a boot disk isn't the easiest job in the world. Of course, those who've used *EZGen* from Burke & Burke might say it isn't all that difficult. The only problem with *EZGen* is that inserting, moving and deleting modules can be slow, especially if you are using a floppy-only system. *KwikGen* from Gale Force Enterprises (licensed from Sardis Technologies) provides most if not all of the functionality of *EZGen* but works entirely in memory. This makes the process of adding, moving and deleting modules lightning fast. Included on the disk are versions for OS-9 Level I and II, the CoCo 1, 2 and 3, and terminals.

You begin by running *KwikGen* with an optional memory modifier. (The more memory you give to *KwikGen*, the larger the boot file it will let you edit.) A `modpatch` script is included in the manual for increasing the default to 40K; and up to 48K can be allocated on OS-9 Level II systems. With *KwikGen* running, you can either load an existing boot file from disk or use the boot file cnrrently in memory. Once the boot file is loaded, *KwikGen* vcrifies all modules contained in it. All modules with an invalid header parity or modnle CRC are purged from the buffer.

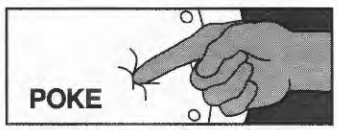
KwikGen allows you to delete, insert, move and even rename modules iu the buffer. If you rename a module, the new name *can be longer than the original name* — up to 26 characters in length. If the new name is longer than the original name, the new name is added to the end of the module.

Two of *KwikGen*'s handiest options allow you to "dump" a module from the buffer or patch it in memory. The module-dump listing is similar to that provided by the OS-9 `dump` command, with the contents shown in both hexadeciinal and ASCII formats. Patching a module works a little differently than when using `modpatch`, but the technique is similar. With *KwikGen* you enter the offset within the module to the byte you want to change. You are then shown the current byte at that offset and are prompted for the new byte. If yon want, you can even enter the data in ASCII format by preceeding the ASCII character with a single or double quote. If the ASCII value is preceeded by a double quote, the most-significant bit of the character is set.

KwikGen allows you to copy the OS-9 Kernel to Track 34. This is especially usc-ful for creating new boot disks or attempting to recover damaged boot disks. And if this isn't enough, the package includes extensive on-line help for all commands — and it's easily accessible.

KwikGen is an especially useful utility that greatly speeds the process of creating and altering boot disks. The only thing I would add to the package is the ability to work with non-boot files — perhaps an alternate write function that doesn't alter LSN 0. This could prove useful for creatiug customized shells and other files containing merged modules. *(Gale Force Enterprises, P.O. Box 66036, Station F, Vancouver, BC V5N 5L4, Canada; $19.95 U.S. plus $4 S/H.)*

— Greg Law



There are times when it could be important to disable the CoCo 3's keyboard so keystrokes have no effect. For instance, demonstration programs are often designed to run without intervention. Or perhaps you want to temporarily turn off the keyboard from within a program so no data is entered at a crucial time.

To disable the CoCo 3 keyboard, use `POKE &HFF01,0`. *Warning: Make sure you save any programs first!* Any keys pressed after this poke are ignored. To turn the keyboard on again, use `POKE &HFF01,4`.

OWL-WARE

Proven Technology

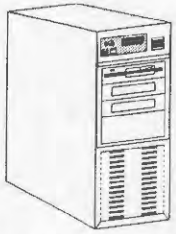
On the Razor's Edge of the Color Computer Frontier

In our 10th Year !

A DECADE OF SERVICE TO THE COMPUTER USER!

486SX-20 SYSTEMS - $1795.00!

Now You can enter the world of 486 computing at a reasonable cost!



the OWL SUPER ATOM - 486

High Powered Computing from a local, well established company.

- 33MHz / 50MHz i486 based Systems with Socket for Weitek CoProcessor
- System and Video BIOS in Cache
- Large Tower Case : (33MHz, FCC Class B) - (50MHz, FCC Class A)
- 230 Watt Power Supply & 8 Option Slots
- System Price Includes: 40MB HD, 4MB RAM, Std. Resolution Color VGA Monitor, High Resolution VGA Card, 2 High Density FD's, MS DOS 5.0

$1795 / $2095 / $2695 / $3595

486SX-20 ISA 486DX-33 ISA 486DX-50 ISA 486DX-50 EISA

- 105MB HD Upgrade Add $160.00
- Super VGA Upgrade Add $95.00

3- YEAR WARRANTY on all systems! Including One Full Year on Parts and Labor. Manufactures 3-Year Warranty on All Hard Drives

OWL SUPER ATOM - 386

- 25/40MHz 386DX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 4MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

$1565/$1645

25MHz 40MHz

OWL SUPER ATOM - SX

- 16/25MHz 386SX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 2MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

$1295/$1349

16MHz 25MHz

386-SX Notebook Computers

20MHz, 60MB HD, 1.4MB FD, 2MB RAM(exp. to 5MB), VGA 640X480 LCD w/32 shades of gray. Ports: 2 Ser, 1 Par, 1 VGA, DOS & Windows, 7.7LBS!

$1545

386-DX Notebook Computers

33MHz, 120MB HD, 1.44MB FD, 32KB CACHE, 4MB RAM(exp. to 16MB), Std. VGA LCD w/32 gray, Ext. Keypad inc., DOS & Windows, 7.7LBS!

$1995

OWL COMPUTER SERVICES

5950 Keystone Drive
Bath, PA (215)-837-1917

Kids & Us - RadioShack®	*St. Onge Systems*	*Computers & Games*
Pottstown Ave., RT. 663	*Wescosville*	*Muhlenberg Shopping Plaza*
Pennsburg (215)-679-3389	*Call for Appt. (215)-481-9775*	*Reading (215)-929-0540*

DISK DRIVES

Bonus! Special Bundled Software with any Disk Drive Purchase!

Floppy Drive Systems

The Highest Quality for Years of Service

Drive 0 Systems (Half Height, Double Sided,

SOLD OUT!
WE NEED CONTROLLERS! ~~$180.~~

IF YOU HAVE 502 CONTROLLERS, CALL US!

Drive 1 Systems (Half Height, Double Sided, Direct Drives) **$115.**

New 3.5", 720K Drives for OS-9 with case & Power Supply **$129. SALE!**

Drive 1 Systems have drive, case, power supply. (You may require optional cable and/or DOS chip to use)

Special for 0/1 Combos (0,1,2,3) $199.

(WITHOUT CONTROLLER)

HALF- HEIGHT DRIVE UPGRADES FOR RS HORIZONTAL CASES

Why only double the capacity of your system when you can triple in the same case? Kit includes: double-sided to fit your case, chip to run both sides of new drive, hardware, and detailed instructions. Easy! Takes only 5 minutes!

Model Only $119.

500 , 501, or 502

All drives are new and fully assembled. We ship only FULLY TESTED and CERTIFIED at these low prices. We use Fuji, YE Data, and other fine brands. No drives are used or surplus unless otherwise stated to you when you order. We appear to be the one of the few advertisers in Rainbow who can truly make this claim. We have 7 years experience in the CoCo disk drive market! We are able to provide support when you have a problem.

Drives 1 Year Warranty

OWL Phones

Order Numbers (only)
1-800-245-6228
1-215-682-6855
Fax: 1-215-837-1942
Technical Help
1-215-837-1917

OWL WARE Software Bundle

Disk Tutorial/Utilities/Games

DISK TUTOR Ver 1.1

Learn how to use your disk drive from this multi-lesson, machine language program. This tutor takes you through your lessons and corrects your mistakes for a quick, painless disk drive introduction. (This professionally written tutor is easily worth the bundle's total price.)

3 UTILITIES

A copy verify, copy, and DOS utility.

2 GAMES

We will select 2 games from our stock. These are sold for more than $20 each.

Do not mistake this software with cheap "Public Domain" software which others offer. All of this software is copyrighted and professional in quality. The tutor is unique with us and has helped thousands of new users learn their disk drive.

only $27.95
(or even better)
only $6.95 with
any Disk Drive Purchase!!

512K Upgrade

Again at a popular price. Fully assembled and tested before shipping. Easy to install. Uses fast 120 ns. chips.

SALE $79.

Now includes **memory test, Ram Disk Lighting, Printer Lighting, and Back-up Lighting.** All with an upgraded manual exclusive with OWL!

Our prices include a discount for cash but do not include shipping.

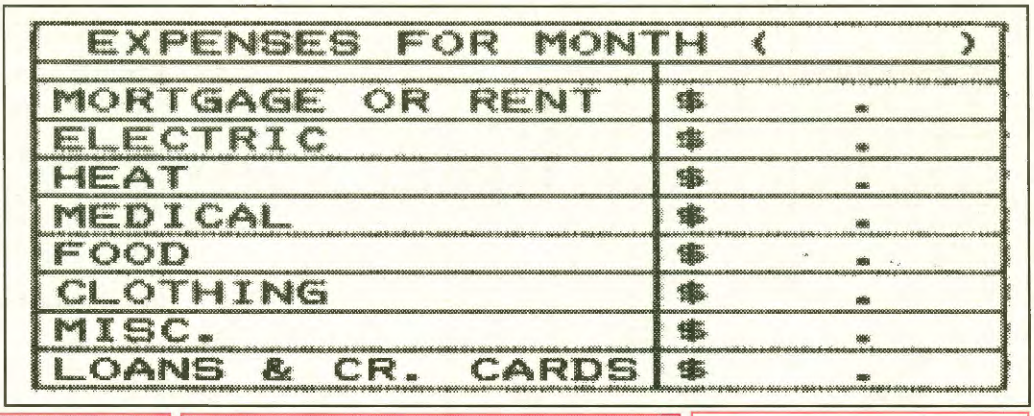
OWL-WARE has a liberal warranty policy. During the warranty period, all defective items will be repaired or replaced at our option at no cost to the buyer except for shipping costs. Call our tech number for return. Return of non-defective or unauthorized returns are subject to a service charge.

OWL-WARE
P.O. BOX 116
Mertztown, PA 19539

Forms *continued from Page 1*

screen. You'll then see a new screen, all of which is blank — except for the top line, which is a reprint of the last line on the previous screen. To get back to the original screen, press S three times — *Forms* cycles the screens from 1 through 4, then back again. The current screen is always indicated at the bottom. The ability to use four different screens is also handy for creating multiple forms on one sheet of paper.

All the key commands supported by *Forms* are shown in Figure 2, and we've covered the use of most of them. The Auto mode, however, also deserves some attention. It can be quite tiring to build a long line from individual graphic elements, and most forms include quite a few such lines. The Auto mode makes this job much easier. To use it, first position the cursor on a blank space, then press A followed by the right arrow. The program will automatically draw a horizontal line from the current position to the next non-space character or the next to last screen column, whichever comes first. The Auto mode works in a similar fashion for vertical lines — just press A followed by the down arrow.



Arrows	move around the screen
SHIFT+Arrows	move to edge of screen
E	erase character at current position
P	print current form (all four screens)
S	switch to next screen
CTRL	select text-entry mode
A	select auto-draw mode
R, Y, V, N	generate corner pieces
T, F, H, B	generate T pieces
G	generate cross piece
I	draw a vertical line
U	draw a horizontal line

Figure 2: Keyboard Controls

Element	Key	Value (decimal)	Program Lines
┌	R	240	210, 750
┬	T	243	220, 760
┐	Y	242	230, 770
├	F	244	240, 780
┼	G	250	250, 790
┤	H	249	260, 800
└	V	246	270, 810
┴	B	248	280, 820
┘	N	247	290, 830
─	U	241	300, 840
│	I	245	310, 850

Figure 3: Graphic-Element Codes

Forms is designed for use with a Radio Shack printer that supports the Tandy printer codes. The graphics elements and their corresponding CHR$ values are shown in Figure 3. By correlating these elements with the IBM Extended Character Set, it is possible to modify *Forms* for use with Epson/IBM-compatible printers. Line 910 sets the serial-port speed to 4800 bps. Change or omit this poke as necessary for your printer. *Forms* runs in the high-speed mode except when printing — pressing BREAK to exit the program also returns the CoCo to normal speed — so you do not need to cut the baud setting in half for printing purposes. The Tandy printer codes in lines 940 and 960 set the printer for half-forward and full-forward linefeed.

Feel free to experiment with *Forms*, and modify the program to meet your individual needs — I have not yet included a Save/Load feature, though it should be fairly easy to implement. The text and graphics elements are stored in simple string arrays.

Forms is handy program for just about anyone. I know it has helped me a great deal.

John Musumeci is a retired TV repairman whose sole hobby for the past eight years has been working with and programming the Color Computer. He may be contacted at 103-57 104 Street, Ozone Park, NY 11417, (718) 738-0212. Please include an SASE when requesting a reply.

CoCo 3

The Listing: FORMS

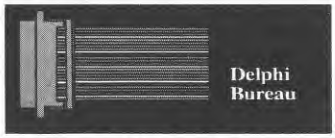
```
1 'FORMS
2 'BY JOHN MUSUMECI
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 CLEAR7500:WIDTH40:CLS5
20 PRINT"DO YOU WANT INSTRUCTION
S?      PRESS (Y)        IF NOT,
PRESS ANOTHER KEY"
30 ONBRKGOTO1110
40 I$=INKEY$:IF I$="Y" THEN 1200
  ELSE IF I$="" THEN 40 ELSE
50 CLS:PRINT"ADJUST PRINTER-PAPE
R PERFORATION     THEN          P
RESS <ANY KEY>"
60 DIMA$(101):TT=1
70 FORA=1 TO 101:A$(A)=STRING$(5
2,32):NEXTA
80 I$=INKEY$:IF I$=""THEN80
90 IF I$=CHR$(3)THEN1110
100 HSCREEN1:HCLS5:HCOLOR2:S=1:C
=1:D=26
110 GDTO 980
120 GOSUB1000:GOSUB1010:GOSUB102
0:GOSUB1030:GOSUB1040:GOSUB1060:
GOSUB1070:GOSUB1080:GOSUB1090
130 X=10:Y=10:Z=1:A=C
140 HDRAW"BM"+STR$(X)+",0;D5"
150 HDRAW"BM0,"+STR$(Y)+"R6"
160 GOTO450
170 I$=INKEY$:IF I$=""THEN170
180 IF I$=CHR$(3) THEN 1110
190 IF I$="A" OR I$="a" THEN GOS
UB1010:HPRINT(1,21),"AUTO":GOTO
1120
200 IF I$="P" OR I$="p" THEN HDR
AW"BM"+STR$(X)+",0;C5D5C2":HDRAW
"BM0,"+STR$(Y)+"C5R6C2":GOTO 910
210 IF I$="R" OR I$="r" THEN B=2
40:B$="ND2NR3":GOTO 430
220 IF I$="T" OR I$="t" THEN B=2
43:B$="ND2NL2NR3":GOTO 430
230 IF I$="Y" OR I$="y" THEN B=2
42:B$="ND2NL2":GOTO 430
240 IF I$="F" OR I$="f" THEN B=2
44:B$="ND2NU3NR3":GOTO 430
250 IF I$="G" OR I$="g" THEN B=2
50:B$="ND2NL2NU3NR3":GOTO 430
260 IF I$="H" OR I$="h" THEN B=2
49:B$="ND2NL2NU3":GOTO 430
270 IF I$="V" OR I$="v" THEN B=2
46:B$="NU3NR3":GOTO 430
280 IF I$="B" OR I$="b" THEN B=2
48:B$="NL2NU3NR3":GOTO 430
290 IF I$="N" OR I$="n" THEN B=2
47:B$="NL2NU3":GOTO 430
300 IF I$="U" OR I$="u" THEN B=2
41:B$="NL2NR3":GOTO 430
310 IF I$="I" OR I$="i" THEN B=2
45:B$="ND2NU3":GOTO 430
320 IF I$="S" THEN 620
330 IF I$="E" THEN B$="C5G2U5RD5
RU5RD5RU5RD5L3U2C2":B=32:GOTO 43
0
340 IF I$=CHR$(189) THEN U=0:GOT
O 470
350 IF I$=CHR$(9) THEN GOSUB600:
Z=Z+1:X=X+6:IF X>316 THEN X=316:
Z=52:GOTO 140 ELSE 140
360 IF I$=CHR$(93) THEN GOSUB600
:X=316:Z=52:GOTO 140
370 IF I$=CHR$(8) THEN GOSUB600:
Z=Z-1:X=X-6:IF X<10 THEN X=10:Z=
1:GOTO 140 ELSE 140
380 IF I$=CHR$(21) THEN GOSUB600
:X=10:Z=1:GOTO 140
390 IF I$=CHR$(94) THEN GOSUB610
:A=A-1:Y=Y-6:IF Y<10 THEN Y=10:A
=C:GOTO 140 ELSE 140
400 IF I$=CHR$(95) THEN GOSUB610
:A=C:Y=10:GOTO 140
410 IF I$=CHR$(10) THEN GOSUB610
:A=A+1:Y=Y+6:IF Y>160 THEN Y=160
:A=D:GOTO 140 ELSE 140
420 IF I$=CHR$(91) THEN GOSUB610
:A=D:Y=160:GOTO 140
430 HDRAW "BM"+STR$(X)+","+STR$(
Y)+B$
440 MID$(A$(A),Z,1)=CHR$(B)
450 GOSUB 1030:GOSUB 1040:HPRINT
(15,21),S:HPRINT(26,21),Z:HPRINT
(36,21),A
460 GOTO 170
470 GOSUB 1010:HPRINT(1,21),"TEX
T"
480 I$=INKEY$:IF I$="" THEN 480
490 IF I$=CHR$(189) OR I$=CHR$(1
3) THEN GOSUB1050:GOSUB 1010:HPR
INT(0,21),"GRAPHIC":GOTO 170
500 IF I$=CHR$(94) OR I$=CHR$(10
) THEN 480
510 IF I$=CHR$(9) THEN GOSUB600:
Z=Z+1:X=X+6:HDRAW"BM"+STR$(X)+",
0;D5":IF X>316 THEN X=316:Z=52:G
OTO 590 ELSE 590
520 IFI$=CHR$(8)ANDZ>1THENHDRAW"
BM"+STR$(X-6)+","+STR$(Y)+"C5G2U
5RD5RU5RD5RU5RD5L3U2C2":GOSUB600
:MID$(A$(A),Z-1,1)=CHR$(32):U=U-
1:Z=Z-1:X=X-6:HDRAW"BM"+STR$(X)+
",0;D5":GOSUB 1100:GOTO 590
530 IF I$=CHR$(8) THEN HDRAW"BM"
+STR$(X)+","+STR$(Y)+"C5G2U5RD5R
U5RD5RU5RD5L3U2C2":HDRAW"BM4,0;C
5D5C2":Z=Z-1:X=10:GOTO 600
540 MID$(A$(A),Z,1)=I$:GOSUB600:
HPRINT(U,23),I$
550 IF I$<>CHR$(32) THEN HDRAW"B
M"+STR$(X)+","+STR$(Y)+"L2E2F2L2
NU2"
560 X=X+6:Z=Z+1:IF X>316 THEN X=
316:Z=52
570 HDRAW"BM"+STR$(X)+",0;D5"
580 U=U+1:IF U>39 THEN U=39
590 GOSUB 1030:HPRINT(26,21),Z:G
OTO480
600 HDRAW"BM"+STR$(X)+",0;C5D5C2
":RETURN
610 HDRAW"8M0,"+STR$(Y)+"C5R6C2"
:RETURN
620 HCLS5
630 IF S=1 THEN C=26:D=51:S=2:GO
TO 670
640 IF S=2 THEN C=51:D=76:S=3:GO
TO 670
650 IF S=3 THEN C=76:D=101:S=4:G
OTO 670
660 IF S=4 THEN C=1:D=26:S=1:GOT
O 670
670 XX=10:YY=10
680 IFTT>3THEN690ELSEAA=C:GOTO70
0
690 FOR AA=C TO D
700 FORZZ=1TO52
710 II$=MID$(A$(AA),ZZ,1):N=ASC(
II$)
720 IF N=32 THEN870
730 IF N>32 AND N<128 THEN B$="L
2E2F2L2NU2":GOTO860
740 IF N<240 OR N>250 THEN870
750 IF N=240 THEN B$="ND2NR3":GO
TO860
760 IF N=243 THEN B$="ND2NL2NR3"
:GOTO860
770 IF N=242 THEN B$="ND2NL2":GO
TO860
780 IF N=244 THEN B$="ND2NU3NR3"
:GOTO860
790 IF N=250 THEN B$="ND2NL2NU3N
R3":GOTO860
800 IF N=249 THEN B$="ND2NL2NU3"
:GOTO860
810 IF N=246 THEN B$="NU3NR3":GO
TO860
820 IF N=248 THEN B$="NL2NU3NR3"
:GOTO860
830 IF N=247 THEN B$="NL2NU3":GO
TO860
840 IF N=241 THEN B$="NL2NR3":GO
TO860
850 IF N=245 THEN B$="ND2NU3"
860 HDRAW"BM"+STR$(XX)+","+STR$(
YY)+B$
870 XX=XX+6:NEXT ZZ:YY=YY+6:XX=1
0
880 IF TT<4 THEN 900
890 NEXT AA
900 TT=TT+1:GOTO120
910 POKE150,7:'***BAUD RATE***
920 POKE65496,0:'***SLOW-DOWN PO
KE***
930 FOR AA=1 TO 7:PRINT#-2:NEXT
AA
940 PRINT#-2,CHR$(27);CHR$(28):'
***HALF-FORWARD LINE FEED***
950 FOR AA=1 TO 101:PRINT#-2,TAB
(14);A$(AA):NEXT AA
960 PRINT#-2:PRINT#-2,CHR$(27);C
HR$(54)
970 FOR AA=1 TO 7:PRINT#-2:NEXT
AA
980 POKE65497,0:'***HIGH SPEED P
OKE***
```

```
990 GOTO 120
1000 HDRAW"BM0,164;R319DL319":RE
TURN
1010 HLINE(0,166)-(55,175),PRESE
T,BF:RETURN
1020 HLINE(126,166)-(136,177),PR
ESET,BF:RETURN
1030 HLINE(210,166)-(234,177),PR
ESET,BF:RETURN
1040 HLINE(290,166)-(319,177),PR
ESET,BF:RETURN
1050 HDRAW"BM0,184;C5R319DL319DR
319DL319DR319DL319DR319DL319C2":
RETURN
1060 HPRINT(0,21),"GRAPHIC":RETU
RN
1070 HPRINT(9,21),"SCREEN":RETUR
N
1080 HPRINT(20,21),"ACROSS":RETU
RN
1090 HPRINT(32,21),"DOWN":RETURN
1100 HDRAW"BM"+STR$(U*8)+",191;C
5U8RD8RU8RD8RU8RD8C2":RETURN
1110 POKE65496,0:WIDTH 32:END
1120 IF MID$(A$(A),Z,1)<>CHR$(32
) THEN GOSUB1010:GOSUB1060:GOTO1
70
1130 I$=INKEY$:IF I$="" THEN 113
0
1140 IF I$="A" OR I$="a" THEN GO
SUB 1010:GOSUB1060:GOTO170
1150 IF I$=CHR$(9) AND X<>316 TH
EN HDRAW"BM"+STR$(X)+","+STR$(Y)
+"NL2NR3BR6":MID$(A$(A),Z,1)=CHR
$(241):GOSUB600:Z=Z+1:X=X+6:HDRA
W"BM"+STR$(X)+",0;D5":GOTO 1180
ELSE IF I$=CHR$(9) THEN GOSUB 10
10:GOSUB1060:GOTO170
1160 IF I$=CHR$(10) THEN HDRAW"B
M"+STR$(X)+","+STR$(Y)+"ND2NU3BD
6":MID$(A$(A),Z,1)=CHR$(245):HDR
AW"BM0,"+STR$(Y)+"C5R6C2":A=A+1:
Y=Y+6:GOTO1190
1170 GOTO1130
1180 IF MID$(A$(A),Z,1)<>CHR$(32
) OR X=316 THEN GOSUB 1010:GOSUB
 1060:GOTO 170 ELSE 1150
1190 IF MID$(A$(A),Z,1)<>CHR$(32
) OR A=D THEN GOSUB 1010:GOSUB10
60:Y=10:A=C:GOTO150 ELSE1160
1200 CLS:LOCATE11,0:PRINT"GRAPHI
C MODE":PRINT
1210 PRINT"PRESS":LOCATE25,2:PRI
NT"FUNCTION"
1220 PRINT:PRINT"ARROWS":LOCATE1
5,4:PRINT"MOVE AROUND SCREEN"
1230 PRINT"SHIFT/ARROWS":LOCATE1
5,5:PRINT"MOVE TO EDGES"
1240 PRINT"E":LOCATE15,6:PRINT"T
O ERASE"
1250 PRINT"P":LOCATE15,7:PRINT"T
O PRINTER"
1260 PRINT"S":LOCATE15,8:PRINT"G
OTO NEXT SCREEN"
1270 PRINT"CTRL":LOCATE15,9:PRIN
T"ENTER TEXT MODE"
1280 PRINT"A":LOCATE15,10:PRINT"
ENTER AUTO MODE"
1290 PRINT:PRINT"R,Y,V,N":LOCATE
15,12:PRINT"DRAW CORNERS"
1300 PRINT"T,F,H,B":LOCATE15,13:
PRINT"DRAW T's"
1310 PRINT"G":LOCATE15,14:PRINT"
DRAW CROSS"
1320 PRINT"I":LOCATE15,15:PRINT"
DRAW VERT. LINE"
1330 PRINT"U":LOCATE15,16:PRINT"
DRAW HORIZ. LINE"
1340 PRINT:PRINT"DO NOT DRAW A G
RAPHIC OR CHARACTER OVER ANOTHER
 GRAPHIC OR CHARACTER WITHOUT
 ERASING FIRST."
1350 LOCATE12,22:PRINT"PRESS ANY
 KEY"
1360 I$=INKEY$:IF I$="" THEN 136
0
1370 HSCREEN1:HCLS5:HCOLOR2
1380 HPRINT(8,1),"DRAWING IN GRA
PHIC MODE"
1390 H=83:V=44:FOR X=1 TO 5:HCIR
CLE(H,V),10:H=H+40:NEXT X
1400 H=92:V=91:FOR X=1 TO 3:HCIR
CLE(H,V),10:H=H+40:NEXT X
1410 H=100:V=147:FOR X=1 TO 3:HC
IRCLE(H,V),10:H=H+40:NEXT X
1420 HPRINT(10,5),"R":HPRINT(15,
5),"T":HPRINT(20,5),"Y":HPRINT(2
5,5),"U":HPRINT(30,5),"I"
1430 HPRINT(11,11),"F":HPRINT(16
,11),"G":HPRINT(21,11),"H"
1440 HPRINT(12,18),"V":HPRINT(17
,18),"B":HPRINT(22,18),"N"
1450 HPRINT(12,21),"(KEY80ARD)":
HPRINT(9,23),"PRESS ANY KEY TO C
ONT."
1460 HPRINT(0,3),"draws ->":HDRA
W"BM80,30;U6R6BR37ND6NL3R3BR40NL
6LD6BU3BR35R6BR37NU3D3"
1470 HPRINT(0,9),"draws ->":HDRA
W"BM92,75;NU3ND3NR6BR40NL3NU3NR3
ND38R40NL3NU3ND3"
1480 HPRINT(0,16),"draws ->":HDR
AW"BM100,133;NU6R6BL3BR37NL3NR3N
U6BR37R6U6"
1490 I$=INKEY$:IF I$="" THEN 149
0
1500 WIDTH40
1510 LOCATE13,0:PRINT"TEXT MODE"
1520 PRINT:PRINT"ENTER TEXT AS Y
OU NEED TO, AT ANY TIME.":PRINT"
BACK-SPACE, TO CORRECT."
1530 PRINT"WHEN DONE, PRESS <ENT
ER> OR <CTRL>."
1540 PRINT:PRINT"DO NOT ENTER TE
XT DIRECTLY BELOW OTHER  TEXT WI
THOUT SKIPPING A LINE BECAUSE TH
EPRINTER IS IN HALF-FORWARD LINE
 FEED AND WILL PRINT OVER PART O
F CHARACTERS."
1550 PRINT"GRAPHICS CAN BE ENTER
ED BETWEEN TEXT    WITHOUT PROBL
EMS."
1560 LOCATE8,21:PRINT"PRESS ANY
KEY TO CONT."
1570 I$=INKEY$:IF I$="" THEN 157
0
1580 CLS:LOCATE13,0:PRINT"AUTO M
ODE"
1590 PRINT:PRINT"PRESS (A) TO RE
TURN, BEFORE USING, ELSE:"
1600 PRINT:PRINT"PRESSING RIGHT/
ARROW KEY WILL DRAW A    HORIZ.
LINE FROM THAT POINT TO NEXT TO
 LAST COLUMN."
1610 PRINT:PRINT"PRESSING DOWN/A
RROW KEY WILL DRAW A     VERT. L
INE FROM THAT POINT, DOWN THE
 SCREEN, STOPPING ROW BEFORE LAS
T."
1620 PRINT:PRINT"AUTO WILL NOT O
PERATE IF NOT STARTED    FROM A
BLANK SPACE. IT WILL STOP DRAWIN
G WHEN IT DOES NOT ENCOUNTER A B
LANK     SPACE."
1630 LOCATE12,21:PRINT"PRESS ANY
 KEY"
1640 I$=INKEY$:IF I$="" THEN 164
0
1650 GOTO 50
```

Thinking of buying a printer stand or monitor pedestal? Why not save a few bucks and make one instead? And when you look for materials, consider PVC pipe for the foundation.

PVC pipe is lightweight, easy to work with and relatively inexpensive at most chain hardware stores. It comes in long pieces (which you can easily cut with a hacksaw) and many joints (elbows, T fittings, etc.) are available. You'll also need a bit of cleaner and adhesive to connect the pieces.

To make a stand, construct two end frames using elbows, building each with four T fittings for connecting the frames together. Finally, connect the frames — it can be just about as wide as you need. To finish, you can put a piece of plywood across the top and paint the whole kit and kaboodle.



OS-9 SIG Database Primary Keywords

Last month I promised to describe what kinds of files are stored in each database in the OS-9 SIG, now that the database reorganization is completed. Most of the database names are self-descriptive, but there are always some files that defy classification. In figures 1 through 4, I've listed the primary keywords for several of the databases. The remaining databases — Standards, Games & Graphics, Music & Sound, Programmers Den, OSK Applications, OSK Telecom, OSK System Modules, and Tutorials & Education — don't have primary keywords set up yet. I will report the rest as they are finalized.

Announcements
Archives
Humor
News
Reviews
Update

Figure 1: Primary Keywords for General Information

BBS
Doors
Patches
Protocols
Terminal Programs
UUCP

Figure 3: Primary Keywords for Telecom (6809)

Disk I/O
Parallel Communications
Serial Communications
Screen Drivers
Speciality (other)

Figure 4: Primary Keywords for System Modules (6809)

The primary keywords should give you a general flavor of what kinds of files belong in each database topic. I'll add some prose to describe the contents of each topic:

New Uploads: *Do not upload files to this topic!* This database is a temporary staging area where all new database groups appear for about a month (a little bit longer if I'm unusually busy at work, a little bit less when I catch up at the end of the month, but always long enough so that people who check in at least once a month have to check only one database). Submit your group to the database you want it moved to after it has spent a month in New Uploads.

General Information: This is the database to check for product information, as product announcements and reviews belong here. General Information is also the "everything else" database, containing random news, humor, politics, and other stuff that doesn't quite fit in any other database.

Applications (6809) and OSK Applications: These are the "meat-and-potatos" database topics for 6809- and 68000-based OS-9 users, respectively. This is where you will find file archivers, calculators, calendar programs, disk utilities, analysis tools, editors, spreadsheets, and other applications and utilities. Note that some applications and utilities are located in other database topics. A graphics digitizer should be placed in Games & Graphics; terminal programs belong in the Telecom databases. Patches for any program that belongs in this topic should also be placed here. For example, a patch for *DynaCalc*, which is an application, belongs here. A group containing executables for both OS-9/6809 and OS-9/68000 should be placed in the Applications (6809) database, at least for now. The same rule applies to the following two topic pairs.

Archivers
Database
Patches
Productivity
Text Processing
Utilities

Figure 2: Primary Keywords for Applications (6809)

Telecom (6809) and OSK Telecom: These topics are self-descriptive. Any file, programs, or data related to telecommunicating belongs here. This includes Bulletin Board Systems (BBSs), file-transfer protocols such as Kermit, terminal programs, and any utilities that are telecommunications oriented.

System Modules (6809) and OSK System Modules: Patches and updates to the operating system reside here. A patch for `AciaPak` belongs here — even though it is used for telecommunications — because `AciaPak` is a system module. Custom drivers and enhancements to OS-9 are stored here as well.

Games & Graphics: Graphics files such as VEF and GIF images belong here, as do the programs that allow you to view them. You must have the right to upload any picture files, however. Do not upload a digitized news photo or a picture from a magazine or cartoon: *these pictures are copyrighted.* Games also belong in this database. If you upload a picture file, please put the type of picture file at the end of the group name. Here is an example:

```
MARINE CORPS EMBLEM (VEF)
    DATA  JAN-92  BRWOOLSTRUM
```

The above example broaches another topic: meaningful group names. Brian Woolstrum could have named his group `USMC.VEF`. The name he gave above is much more meaningful; it describes exactly what is drawn. If you upload a data file for a game, such as Rick Adams' *OAI*, then follow the group name with `(OAI)`, as above.

You may notice that there is only one Games & Graphics database topic. Uploads for both OS-9/6809 and OS-9/68000 belong here. If your program or data is useful only on a 68000-based system or on any single computer, mention this in the group description and add an appropriate keyword. If your program runs on any 68000-based computer, then the keyword `OSK` is sufficient. The same rules apply to all of the following topics.

Music & Sound: Any program or data that creates, plays, digitizes or reproduces sound or music belongs here, just as the topic name suggests. In this topic, you find scores of *UltiMusE* files and many digitized sounds, as well as the programs to play them. Any MIDI-related programs belong here, such as MIDI patch editors for synthesizers. Also, any documentation related to music, sound, MIDI, or programs belonging here should be placed in this database topic. The same rule applying to Games & Graphics applies here: *Do not upload copyrighted digitized sounds.*

Programmers Den: This database topic is where budding programmers can find new libraries, programming tools such as `make` and `lex`, disassemblers, program skeletons, compilers and interpreters, programming demos designed to give programming tips, and documentation to any of the above. Basically, anything designed to make a programmer's life easier should be uploaded here.

Tutorials & Education: Beginners and people trying to do something for the first time (such as install *Multi-Vue* or program in C) should look here to find help. You can find articles describing disk fragmentation (and how to avoid it), an introduction to OS-9, help configuring your floppy drives, a tutorial explaining how to upload to the databases, a new OS-9 `help` utility and many more useful groups.

Standards: This is the emptiest database in the OS-9 SIG. If you are uploading information about an existing or proposed standard, this is where it belongs. Thus, information about the Ymodem protocol should be uploaded here, but information about a program implementing Ymodem belongs in one of the Telecom topics.

You will notice files in the databases that do not follow the rules as stated above. Deciding which topic a group belongs to can be confusing at times. As Greg Law and I find files that belong in a different topic, we will move them. If you have trouble deciding which topic to upload your group to, you may want to see what is already in the database topics you are trying to decide between. Greg Law uploaded `ALPHA DIRECTORY`, which contains a full directory of each database topic with descriptions of each group in each topic. Be warned that this file is about 340K after it is decompressed!

March 1992 Uploads

In the General Information database, **James Jones** (JEJONES) and **Marty Goodman** (MARTYGOODMAN) uploaded some very exciting information about the Hitachi 6309 chip — a drop in replacement for the 6809. The 6309 has some hidden features that could prove very useful. **Rick Adams** (RICKADAMS) released some utilities that manipulate a disk's granule allocation table. While these utilities can be very useful, use them with extreme caution! As with any disk editor, you can easily corrupt your disk.

Charles West (SANDRIDER) released the latest version of Ron Bihler's RiBBS, Version 2.02S. RiBBS is a Fidonet-compatible CoCo BBS. In the System Modules (6809) topic, **Eugene Anderson** (01GEN40) uploaded a patch to `Init` that allows you to place the `CC3Go` module in your `CMDS` directory rather than in your `OS9Boot` file; you recover about a page (256 bytes) of OS-9 system space by doing this, because you make your `OS9Boot` file smaller. **Matthew Thompson** (MATHOMPSON) announced the new SCSI hard disk drivers that he is working on. These drivers support 512-byte sectors.

Richard Kottke (RICHKOTTKE) uploaded a public domain Motorola floating-point math subroutine library module for OS-9/6809. Richard also submitted a cross assembler written in BASIC09 for the Intel 8051 microcontroller. In the OSK Applications topic, **Mike Haaland** (MIKEHAALAND) released `fstat`, similiar to the *Multi-Vue* version of `fstat`, that shows a file's file-descriptor information. **Bryan Clingman** (BRYANC) uploaded *MicroEmacs 3.11C* — the latest version available. **John Donaldson** (VAXELF) submitted a description of the OS-9/68000 `F$Rename` Set Status call that was released onto USENet by Microware.

In the Standards topic, **Ed Gresick** (EDELMAR) submitted a proposed printer standard for OS-9/68000 — PrintCap, similiar to TermCap.

The CoCo SIG databases were unusually slow during March. Marty Goodman published the 6309 secrets article in the Source for 6809 Assemblers topic. This is the same article he posted to the OS-9 SIG General Information database. In the Utilities & Applications topic, **Denver Page** (DENPAG) released an updated version of `DISKUTILITIES`. This program allows you to copy, move, delete, and rename files, or just erase an entire disk. You can also use the program to scramble a directory, making the disk unusable until you unscramble the directory. **Richard McNabb** (RICKMAC) uploaded `PRINTOUT` — a program that dumps 32-, 40- or 80-column text screens to your printer.

Eddie Kuns is pursuing a doctorate in physics at Rutgers University. He lives in Aurora, Illinois, and works as a programmer and researcher at Fermilab. Eddie is the OS9 Online database mamnager; his username is EDDIEKUNS

OS-9 SIG

General Information

```
INFO ON UPGRADE2.5 PATCH KIT
DONALDLF              Leslie Donaldson
6309 UNMASKED!
MARTYGOODMAN          Marty Goodman
HITACHI 6309 CPU INFO
JEJONES               James Jones
```

Applications (6809)

```
FILE ALLOCATION TABLE UTILITIES
RICKADAMS             Rick Adams
INVENTORY PROGRAM - DEMO
MOHRT                 Tim Mohr
TICKLE: REMINDER SYSTEM
RICKGRAY              Rick Gray
CAL2TEXT: DYNACALC FILE FIXER
JIMHRUBIK             Jim Hrubik
ADDLF - ADD LINE FEED UTILITY
JMLSOFT               Jim McDowell
CALC: SIMPLE CALCULATOR
RICKULAND             Rick Ulland
TELEPHONE LOG, ETC.
JIMHRUBIK             Jim Hrubik
SEE: TEXT FILE VIEWER
JIMBM                 Jim Manning
GROW: GARDENING PROGRAM
GOOCH1                Phillip Vouers
NEW GCAL FOR MULTIVUE
DKINDBERG             Darren Kindberg
```

```
COMPRESS 4.3:COMPRESSION UTILITY
PETRAS1               Michael Petracci
DYNACALC BUSINESS TEMPLATES
JIMHRUBIK             Jim Hrubik
TIMECARD: PUNCH THAT CLOCK!
JIMHRUBIK             Jim Hrubik
GOALCHART: STUDENT GOALS
JIMHRUBIK             Jim Hrubik
MERGE MODULES INTO 8K BLOCKS
RAINMAKER             Edward J. Niklas
UNLZH E5: DECOMP LHA/LHARC FILES
COGITATR              Norman Rheaume
GSORT: MULTIVUE DIRECTORY SORT
WOAY                  Jim Martin
ALARM: ALARM CLOCK COMMAND
RICHKOTTKE            Richard Kottke
```

Telecom (6809)

```
RIBBS V2.02S SHAREWARE BBS
SANDRIDER             Charles West
TODAY: UPDATED HISTORY FILES
BSCHWING              Baron Schwing
```

System Modules (6809)

```
INITPCH: FOR CC3GO IN CMDS DIR
01GEN40               Eugene Anderson
80 COLUMN /TERM WINDOW
EARTHER               Shawn Driscoll
NEW WINDOW DEVICES
EARTHER               Shawn Driscoll
512-BYTE COCO SCSI ANNOUNCEMENT
```

MATHOMPSON Matthew Thompson

Games & Graphics
MM/1 RAYTRACED GRAPHICS
MIKEHAALAND Mike Haaland

Programmers Den
FLOATING POINT MATH MODULE
RICHKOTTKE Richard Kottke
8051 ASSEMBLER
RICHKOTTKE Richard Kottke

OSK Applications
GIFSHOW 2.0 ED. 6 FOR THE MM/1
MIKEHAALAND Mike Haaland
LHARC V1.03
MIKEHAALAND Mike Haaland
FSTAT: FILE STAT. UTILITY
MIKEHAALAND Mike Haaland
MICROEMACS 3.11C EXECUTABLE
BRYANC Bryan Clingman
MICROEMACS 3.11C
BRYANC Bryan Clingman
1991 US TAX TEMPLATE (FORM 1040)
KSCALES Ken Scales
RENAME SETSTAT FOR OSK 2.4
VAXELF John Donaldson
GCC MEMORY BUG PATCH
VAXELF John Donaldson
PEARLS V1.02
PAGAN Stephen Carville

OSK System Modules
XWINDOWS (X11R4) DEMO KIT
THEFERRET Philip Brown

Standards
PRINTER CONTROL
EDELMAR Ed Gresick

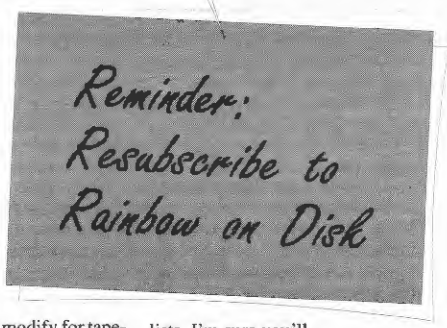
CoCo SIG

Source for 6809 Assemblers
6309 REVEALED!
MARTYGOODMAN Marty Goodman

Utilities & Applications
DISK UTILITIES
DENPAG Denver Page
TEXT SCREEN PRINTOUT
RICKMAC Richard McNabb

Feature Program

CoCo Makes a Quick Note Taker



Ed is a "quickie" 32-column screen editor — it allows you to type text on a 32-by-16 screen and save that text to disk. The program is great for "jotting down" quick notes. (Since the program uses the 32-column screen, it is also ideally suited for use with the TP-10 printer; I use PRT from the February 1990 issue of THE RAINBOW.)

The idea behind *Ed* is simple: The Color Computer (any model) stores its 32-column screen text in memory locations 1024 through 1535 ($400 through $5FF). Line 70 saves these locations in binary format. Once you've saved the text, all you need to do is load the file as a machine-language program and the text reappears onscreen — the text is placed directly into screen memory.

Ed was written for the CoCo 3 and is designed for use with a disk drive. But the program is easy to modify for tape-based CoCos — simply change SAVEM to CSAVEM and LOADM to CLOADM. CoCo 1 and 2 users can make use of the general idea presented, but you must delete Line 10 and insert a check for a save-and-quit key other than BREAK. You can use pretty much any key — just make sure you won't need it for the text you want to enter.

I hope you enjoy this simple screen editor. I find it easier and less bothersome at times than loading a word processor. Since it loads in a flash and is so easy to use, *Ed* is ideal for writing quick notes and shopping lists. I'm sure you'll find other uses for it as well.

Trevor Boehm is a tenth grade student whose greatest passion is challenging computers with new programs. He has participated in several science fairs and has received numerous awards for his work. He can be contacted at 77 Inwood Cres., Winnipeg, MB R2Y 1A2, Canada. Please include an SASE when requesting a reply.

CoCo 3

The Listing: ED

```
1 'ED 1.0  LO-RES SCREEN EDITOR
2 'BY TREVOR BOEHM
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 PALETTE13,0:PALETTE12,63:ONBR
KGOTO70
20 CLS:PRINT"ED 1.0":PRINT"LORES
 TEXT SCREEN EDITOR":PRINT"BY TR
EVOR BOEHM":PRINT"<C> 1991 BY FA
LSOFT, INC.":PRINT"ALL RIGHTS RE
SERVED."
30 LINEINPUT"FILENAME>>";F$
40 PRINT"PRESS THE <BREAK> KEY T
O SAVE":FORX=1TO1000:NEXT
50 CLS
60 PRINTINKEY$::GOTO60
70 SAVEMF$,1024,1535,1024
80 PRINT"USE THE COMMAND:":PRINT
:PRINT"LOADM"+CHR$(34)+F$+CHR$(3
4):PRINT:PRINT"TO SEE THE FILE"
90 EXEC44539:RUN
```

Key Frames *continued from Page 1*

Going Ahead With the Computer

Computer animation is simply a process by which a set of graphics lines and/or points are made to move from one location to another. In computerized key-frame animation, the animator defines the points in two critical frames and the computer is used to compute the point locations between pairs of points in the starting and ending frames.

In a conventional animated story there are many sets of key frames, requiring hours of tedious hand drawings. Given the resources, the challenge when using a computer is to create interesting animation in spite of the computer's artifacts resulting from linear interpolation. Both approaches require several 'tweens for smooth animation, but a hand-drawn line is always smoother than a curve as seen on a computer monitor.

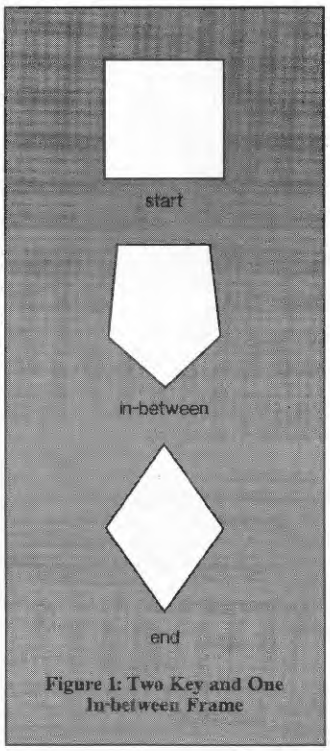


Figure 1: Two Key and One In-between Frame

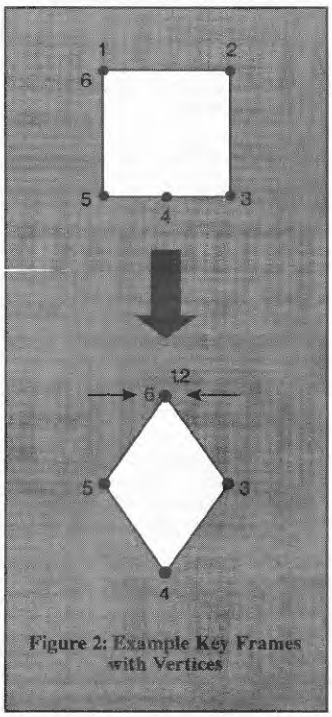


Figure 2: Example Key Frames with Vertices

Let's look at an application of key-frame interpolation. We'll use the technique to transform a simple square into a diamond shape. Figure 1 shows the two key frames along with one in-between frame. One "rule" of key-frame animation is that there must be the same number of points in the starting and ending frames. The secret is to choose the positions of these vertices carefully. The illusion shown in Figure 1 is deceptive. The eye sees only the four vertices in the starting and ending frames. However, the object in each frame really contains six points.

Figure 2 shows two techniques for hiding the extra vertices necessary for this animation example. Point 4 does not show in the bottom line of the square object (starting frame) because it is in line (colinear) with points 3 and 5. In the end frame, showing the diamond, points 1 and 2 have converged into the same spot, so they appear to be one point in the last frame.

Also illustrated by figures 1 and 2 is the importance of placing the points and shapes in such a way that the changes between shapes flow in an interesting pattern. This takes time to design, and it is a necessary part of the art of computer animation that is not handled the same way for traditional animation.

Were we actually animating the transformation from a square to a diamond, we would use more than one in-between frame. The number of frames used affects how slowly and smoothly the change appears to take place.

Linear vs. Non-linear Movement

To better understand the application of key-frame interpolation, let's look at how we can smoothly move an object from one place to another. A single point will do nicely and suffers no loss of generality. To move the point from a starting position to an end position, we might divide the distance between the two positions into four equal parts. This means the point will appear in five different positions, each at a different time (see Figure 3).

Dividing the distance from the starting to end positions into equal parts results in smooth even movement. Not only that but it is easy to program. However, the results of this approach are not always desirable. Smooth movement is the reason "flying" logos on television are not very exciting. Indeed, cartoons often rely on irregular speeds for humorous effect.

The human visual system detects fine differences in acceleration and deceleration, and the brain uses these distinctions as clues for understanding what we see. Very regular movement is more mechanical than natural, so simple computer animation is often more successful with subjects that don't need to move naturally or follow the laws of physics (e.g., bouncing balls). Simple non-linear timing differences (in this case, deceleration) can be achieved with our moving-point example by halving the distance moved at each frame (see Figure 4). From a programming viewpoint, one way this can be accomplished is by using a two-dimensional array.

Of course most animated objects are not limited to either horizontal or vertical movement. (It would be quite boring if they were.) To achieve diagonal movement we can divide both the x (horizontal) and y (vertical) distances into the same number of even divisions (see Figure 5). Notice, however, the divisions don't have to be the same size — the requirement is that the *number* of divisions be equal.

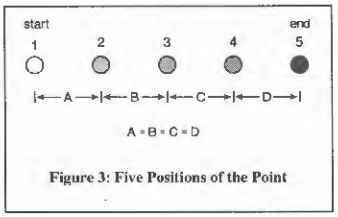


Figure 3: Five Positions of the Point

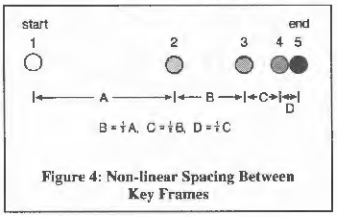


Figure 4: Non-linear Spacing Between Key Frames

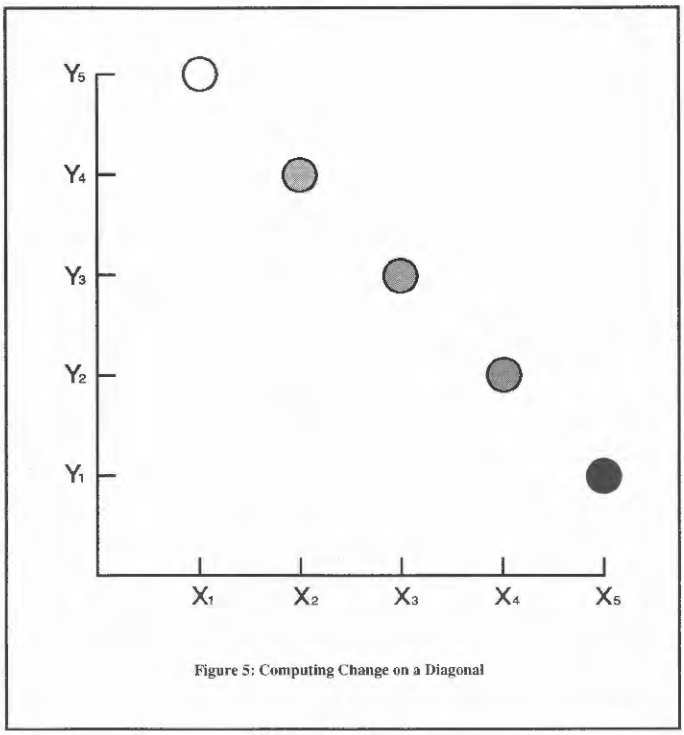


Figure 5: Computing Change on a Diagonal

Keep in mind the foregoing discussion applies to time as well as distance. For example, we can make the moving point appear to slow down by *decreasing* the distance between successive frames and/or by *increasing* the amount of time it takes to reach each frame in the series. It takes a little experience to play the trade-offs and obtain smooth results; and the best way to get experience is to experiment.

```
iniz w4
merge /dd/sys/stdfonts >/w4
display 1b 20 08 00 00 28 18 07 0a 0a >/w4
shell i=/w4&
```

Figure 6: Procedure to Create Window 4

Variations and Other Considerations

In addition to using non-linear distances/timing, consider moving the entire object across the screen *while* it transforms. Especially effective on the CoCo 3 might be to change palettes between frames. Finally, having several shapes appear to change and recombine is more work, but it looks very impressive.

To a certain extent, it is possible to animate stick figures. Sooner or later, though, you will discover why linear interpolation did not solve the 'tweening problem for character animation — body parts are of a fixed length and move in an arc, not in a line. You can hide some of the problem by making more key frames and computing only a few 'tweens. But this does not cut down very much on the programming work required to achieve smooth animation.

Program Operation

Listings 1 and 2 show two programs that illustrate the concepts introduced in this article. The program in Listing 1, KEYF.BAS, is a BASIC program that runs on any CoCo with at least 16K and Extended BASIC. Enter the program and save it to tape or disk. If you don't have a CoCo 3, however, remove lines 10 and 20 before running the program. Further, if your CoCo 1 or 2 does not work well with the high-speed poke, remove Line 30 as well. Note that Line 240 creates a delay to reduce screen flicker — if you run the program without the high-speed poke, you should remove this line, too.

When you run KEYF.BAS, you are prompted for the number of divisions. Enter a number from between 0 and 40; smaller numbers decrease the number of frames, increasing the speed with which the end frame is reached. After you enter a valid value, you'll see five shape sets:

- a moving dot
- a triangle folding over itself
- a square transforming into a diamond
- a D changing to an S
- a leg kicking upward

The data for the shape sets is found in lines 660 through 860. Each set consists of three parts:

- a single number indicating the number of points in the shape

- *x,y* pairs for the starting shape
- *x,y* pairs for the ending shape

Use this format to add your own shapes at the beginning of the shape data. Then change Line 370 to limit the number of shapes the computer draws before it starts over.

The program in Listing 2 is written in C and works with OS-9 Level II. This program is similar to that in Listing 1 except that it performs only the square-to-diamond transformation.

All the necessary graphics calls are built into `keyf.c` using `#define` and `printf` statements. For this reason you don't need to have or use the `cgfx.l` library to compile the program. However, `keyf` is designed to be run on a Type 8 graphics window. Before running the compiled program, use `build` to create the OS-9 procedure (script) file shown in Figure 6 and run this procedure to open Window 4 (use another window number if you are already using Window 4). Use the CLEAR key to select the new window, then execute `keyf`.

Summary

This article has introduced several aspects of computer animation and focused on the key-frame interpolation technique. Granted, the examples we have looked at are very simple in nature. However, the concept remains the same regardless of the number of points used.

Dawn Smith has been programming personal computers since 1977. She began using the Color Computer because of the relatively inexpensive X-Pad. Dawn completed a masters program with emphasis on graphics, CAD and computer imaging. Her hobbies include archaeology, geology and dance. She may be contacted at 4 Eagle Street, Apt. B, Rochester, NY 14608. Please include an SASE when requesting a reply.

16K Extended

Listing 1: KEYF

```
1 'KEYF
2 'BY DAWN A. SMITH
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW
10 POKE 65497,0: REM HI-SPEED CO
CO III
20 WIDTH 32:TP=3:GOTO 280:
 REM COCO III
30 POKE 65495,0:TP=2:GOTO 280:
     REM HI-SPEED COCO 2
40 TP=1:GOTO 280
50 REM ------ KEY-FRAME ANIMATIO
N ------
60 REM  BY DAWN A. SMITH
70 REM  FILE NAME = KEYF/BAS
80 REM ------------------------
--------
90 REM ----- DRAW SHAPES -----
100 FOR C=0 TO DV
110 REM ----UNDRAW LAST SHAPE --
---
120 IF C=0 GOTO 180
130 FOR A=1 TO N-1
140 DX = CX(A)*C:DY=CY(A)*C
150 FX=CX(A+1)*C: FY=CY(A+1)*C
160 LINE (SX(A)-CX(A)+DX,SY(A)-C
Y(A)+DY) -(SX(A+1)-CX(A+1)+FX,SY
(A+1)-CY(A+1)+FY),PRESET
170 NEXT A
180 REM ---- DRAW NEW SHAPE -----
190 FOR A=1 TO N-1
200 DX = CX(A)*C:DY=CY(A)*C
210 FX=CX(A+1)*C: FY=CY(A+1)*C
220 LINE (SX(A)+DX,SY(A)+DY) - (
SX(A+1)+FX,SY(A+1)+FY),PSET
230 NEXT A
240 FOR Z=1 TO 30:NEXT Z
250 NEXT C
260 RETURN
270 REM
280 REM ----- STORE POINT -----
290 DIM SX(30), SY(30):   REM --
- START SHAPE X,Y
300 DIM EX(30), EY(30):   REM --
- END SHAPE X,Y
310 DIM CX(30), CY(30):   REM --
- CHANGE IN X,Y
320 REM
330 REM ----- BEGIN PROGRAM ----
-
340 INPUT "NUMBER OF DIVISIONS "
; DV: DV=DV+1
350 IF DV<1 GOTO 450
360 REM ----- LOOP THROUGH THE 5
 SHAPES -----
370 FOR CT=1 TO 5
380 READ N:               REM
NUMBER OF POINT IN SHAPES
390 GOSUB 490:            REM
LOAD SHAPES
400 PMODE 4,1:SCREEN1,1:PCLS
410 GOSUB 90              REM
DRAW SHAPES
420 NEXT CT
430 RESTORE
440 GOTO 330              REM
RE-RUN PROGRAM
450 REM ---- END OF PROGRAM ----
460 IF TP=3 THEN POKE 65496,0
470 END
480 REM
490 REM ----- LOAD START SHAPE
-----
500 FOR A = 1 TO N
510 READ SX(A)
520 READ SY(A)
530 NEXT A
540 REM ----- LOAD END SHAPE ---
---
550 FOR A = 1 TO N
560 READ EX(A)
570 READ EY(A)
580 REM ----- COMPUTE SIZE OF DI
VISIONS -----
590 CX(A) = ((EX(A)-SX(A))/DV)
600 CY(A) = ((EY(A)-SY(A))/DV)
610 NEXT A
620 RETURN
630 REM
640 REM ----- SHAPE DATA -----
650 REM
660 REM ---- NUMBER OF POINTS
670 DATA 2
680 REM ---- STARTING SHAPE X,Y,
...
690 DATA 20,20,  20,20
700 REM ---- ENDING SHAPE X,Y .
. .
710 DATA 120,20, 120,20
720 DATA 4
730 DATA 20,30, 110,50, 100,20,
20,30
740 DATA 20,60, 100,30, 20,100,
20,60
750 DATA 6
760 DATA 20,40, 60,40, 60,80, 40
,80, 20,80, 20,40
770 DATA 40,40, 40,40, 60,60, 40
,80, 20,60, 40,40
780 DATA 8
790 DATA 20,20, 20,30, 20,40, 30
,40, 38,38, 38,22, 30,20, 20,20
800 DATA 20,36, 27,40, 38,38, 38
,32, 30,30, 24,26, 28,20, 38,22
810 DATA 3
820 DATA 20,20, 20,40, 22,40
830 DATA 20,20, 40,20, 42,20
840 DATA 9
850 DATA 40,20, 44,22, 45,25, 44
,27, 40,30, 37,28, 35,25, 37,22,
 40,20
860 DATA 40,55, 45,56, 47,58, 45
,59, 40,60, 35,59, 32,58, 35,56,
 40,55
```

Listing 2: KEYF.c

```
/* ------- KEYF.c  -------------
*/
/*    Key Frame Animation
*/
/*      by Dawn A. Smith
*/
/* ------------------------------
*/
#include <stdio.h>

#define CLR    12 /* c  clear
screen */
#define GRAF   27 /* 1b */
#define FORGND 50 /* 32 */
#define BCKGND 51 /* 33 */
#define BORDER 52 /* 34 */
#define SET    64 /* 40 position
graphics cursor */
```

```
#define LN_MV  70 /* 46 draw line and move gr.crsr*/

#define BLACK   2
#define RED     4
#define YELLOW  5
#define MAGENTA 6
#define CYAN    7

int    sx[30], sy[30]; /* start shape */
int    ex[30], ey[30]; /* end shape */
double cx[30], cy[30]; /* change */

main ()
{  /* ----- begin program ----- */
    int dv;     /* divisions */
    int n;      /* number of points */

    gr_setup();
    do
    {
         /* get the number of divisions */
         printf("number of divisions? ");
         scanf("%d",&dv);
         if (dv < 0) break;
         dv++;
         n=shapesetup();
         make_incr(dv,n);
         drawfrm(n,dv);
     }
     while (dv > -1);
} /* ----- end program -----*/

/*
 * ----- Load the data -----
 */
int shapesetup()
{
int n = 6;
  /* ----- start shape ---- */
  sx[1]=sx[6]-20;
  sy[1]=sy[6]-40;
  sx[2]=60; sy[2]=40;
  sx[3]=60; sy[3]=80;
  sx[4]=40; sy[4]=80;
  sx[5]=20; sy[5]=80;
  /* ----- end shape -----*/
  ex[1]=ex[6]-40;
  ey[1]=ey[6]-40;
  ex[2]=40; ey[2]=40;
  ex[3]=60; ey[3]=60;
  ex[4]=40; ey[4]=80;
  ex[5]=20; ey[5]=60;
  return (n);
}
make_incr(idiv,n)
int n;
int idiv;
{ /*-- compute the size of the divisions --*/
    int a;
    double div;
    div = (double)idiv;
    n++;
    for (a=1; a<n; a++)
    {
      cx[a]=((double)(ex[a]-sx[a]))/div;
      cy[a]=((double)(ey[a]-sy[a]))/div;
    }
}
/*
 * ---- screen setup ----
 */
gr_setup()
{
    printf("%c%c%c",GRAF,BCKGND,MAGENTA);
    printf("%c%c%c",GRAF,BORDER,MAGENTA);
    printf("%c%c%c",GRAF,FORGND,BLACK);
    printf ("%c",CLR);
}
/*
 * ----- draw shape -----
 */
drawfrm(n,dv)
int dv;     /* number of positions*/
int n;      /* number of points */
{
int a, m, x, y;
  for (m=0;m<dv+1;m++)
  {
    printf ("%c",CLR);
    x = sx[1] + (int)(cx[1]*m);
    y = sy[1] + (int)(cy[1]*m);
    printf ("%c%c%c%c%c%c",GRAF,SET,0,x,0,y);
    for (a=2;a<n+1;a++)
    {
      x = sx[a] + (int)(cx[a]*m);
      y = sy[a] + (int)(cy[a]*m);
      printf ("%c%c%c%c%c%c",GRAF,LN_MV,0,x,0,y);
    }
  }
} /* end drawfrm */
```

Feature Program

ARTIFACTING MEETS THE COCO 3

When the Color Computer 3 was first introdnced, many people believed the age of artifacting colors for graphics had come to an end. After all, the CoCo 3 supports 16 colors (selectable from a palette of 64) on a 320-by-192-pixel graphics screen. This goes way beyond the CoCo 1 and 2 with their two-color limit. Still, I say we can use artifacting to get even more.

In terms of working with the Color Computer, artifacting colors (creating the false impression of colors) relies on an inherent characteristic of color composite monitors and televisions. The PMODE4 graphics mode supports only two colors, and most programs use black and white. However, if you draw a single vertical line on the PMODE4 screen, it appears either blue or red (depending on the internal timing of the CoCo). If you erase that line and draw a similar vertical line, but one space to the right or left of the first, it appears in the other color — the false "primary" colors alternate vertically. If you put two vertical lines side-by-side, you get a line that appears in the selected foreground color (black or white). By setting different pixels in a defined grid, this "defect" can be used to create the appearance of many different hues. Now imagine the possibilities with 16 selectable colors rather than two.

Color 256 is a simple BASIC program that combines color artifacting with the CoCo 3's 16-color graphics screen (HSCREEN2) to effectively produce 256 different colors on one screen. The program contains two sections, the first of which sets the palettes and draws the individual lines used for artifacting. *Color 256* works much like artifacting on the CoCo 1 and 2 where alternating, adjacent vertical lines are used to create the effect of more colors. It takes some time for the program to draw the lines, so be patient. The high-speed poke is used (Line 20) to speed up the process.

The second section of *Color 256* rapidly cycles the screen through the entire palette of 64 colors. This is accomplished using very simple palette switching. If you press BREAK before the program reaches this color cycling, you'll end up with an odd palette setting and the computer will still be in the high-speed mode (normal speed is enabled in Line 220). Make sure you slow it down before performing any tape or disk I/O. The best way to set things right is with a full reset of the Color Computer (CTRL-ALT-Reset).

It is important to note that artifacting colors with the CoCo 3's HSCREENs is generally much more effective with a color composite monitor or television. RGB monitors more accurately display the correct information, and it is easy to distinguish between the vertical lines that make up the artifacted colors.

Feel free to study the techniques used by *Color 256* and introduce color artifacting into your BASIC programming efforts. Experienced programmers might consider using the interrupts to enable the full 64-color palette onscreen at one time (see "Color Chart for the CoCo 3" on Page 20 of the January 1987 issue of THE RAINBOW). Then it should possible to produce 4096 colors at the same time . . .

———---

Adam Breindel plans to attend the University of Chicago in the fall of 1992, where he will study economics. Currently, he can be contacted at 328 Abbey Lane, Lansdale, PA 19446. Please include an SASE when requesting a reply.

CoCo 3

The Listing: COLOR256

```
1 'COLOR 256
2 'BY ADAM BREINDEL
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 ON ERR GOTO 220
20 POKE 65497,0
70 '******************
80 '*WHEN THIS SECTION OF
    *PROGRAM IS DONE,THE DISPLAY
    *SEQUENCER SECTION RUNS.
90 '*THIS CAN BE ACCESSED AT
    *LINE 220 IF PROGRAM IS
100 '*STOPPED AFTER INITIAL
     *SCREEN IS DRAWN
110 '*THIS PROGRAM USES 320*192
120 '*16-COLOR GRAPHICS AND THE
     *2MHZ POKE AND RUNS ON A
     *128K COCO 3
130 '******************
140 HSCREEN 2
150 FOR Y=0 TO 15:PALETTE Y,48+Y
:NEXT
160 FOR Z=0 TO 15:FOR Y=0 TO 15
170 GOSUB 190
180 NEXT Y,Z
190 FOR A=Z*16 TO Z*16+15:IF A/2
=INT(A/2) THEN HDRAW"C"+STR$(Y)
ELSE HDRAW "C"+STR$(Z)
200 HLINE (A,12*Y)-(A,12*Y+11),P
SET
210 NEXT:RETURN
220 POKE65496,0
230 ON BRK GOTO 360
240 '******************
250 '**COLOR-256
260 '*COLOR SEQUENCER
270 POKE&HE6E4,&HE6
280 HSCREEN 2
290 POKE&HE6E4,&HE7
300 FOR X=0 TO 63
310 FOR Y=X TO X+15:IF Y>63 THEN
 330
320 PALETTE Y-X,Y
330 NEXT Y
340 NEXT X
350 GOTO 300
360 PALETTE 13,63:PALETTE 12,0
```

Product Review

VED/68000 for High-Performance Editing

One of the most important tools for any computer system is a powerful but easy-to-use text editor. *VED* (short for *V*isual *ED*itor) is just such an offering from Bob van der Poel Software. *VED*'s origins lie with the CoCo and OS-9, and the program is now offered for OS-9/68000-based machines — the MM/1, TomCat and System IV. The software is supplied on a 3½-inch, high-density disk that contains five directories holding the program modules, documentation files, environment files, help files and source files.

Although the on-line documentation files fully describe the functions of the files on the distribution disk, setting up *VED* is quite easy. The *VED* executable module must be copied from the CMDS directory of the disk to the CMDS directory of your hard drive or any floppy disk. In addition, a help file and an environment file must be copied into the SYS directory. Supplied are a few different versions of the help file, including one with documentation for all the standard C library routines; this is extremely helpful for programmers. The environment file describes to *VED* the specific computer you are using; environment files are supplied for the MM/1, the System IV, VT100 terminals and the Color Computer.

Also included on the distribution disk is a program named VSPLIT, which allows you to break extremely large text files into smaller files that *VED* can more easily accept. Still, I tested *VED* with some large, unsplit text files (about 250K in length). Even with this large amount of text, copying a large block of text is almost instantaneous. The source code for VSPLIT is also supplied and can be found in the SRC directory of the distribution disk.

VED's basic editing screen does not include any status bars or symbols, so you can use the entire screen for viewing and editing a file. Any carriage returns in the file are represented onscreen by the tilde (~) character, making it easy to see where paragraphs actually end. A special end-of-buffer character is visible at all times immediately to the right of the last character in the file.

VED features many movement, insertion, and deletion functions, each of which is mapped to a particular control-key combination. These key combinations can be modified through the environment file, and some commonly used functions are mapped to arrow and movement keys on the IBM-style keyboard most OS-9/68000 systems use. Two of the more unusual features are a Jump function (which lets you move to a position in the file by line number, percentage position, or test label) and Case Toggle (which cycles a word between all upper- or lowercase characters, and normal capitalization. In addition, *VED* sports an Undo function that operates on the line currently being edited as well as for word, line and block deletions.

The Search and Replace functions offer the usual search (in both directions) as well as Find Next and Find Last. You can use the wildcard character (?) when replacing text — a feature many programs do not offer. Block-editing commands are provided, giving you the ability to cut, copy and paste text. You can also save a block to disk, sort the lines within a block, print a block to the printer or a disk, and display word- and line-count information about a block.

VED supports a full complement of macro capabilities; up to 26 user-defined macros may be defined and saved to disk at any time while you are editing a file. In addition, there are eight predefined macros, some of which allow you to list the current input and output files or extensions, automatically generate increasing numbers for auto-numbering applications. Two user-definable Time macros give you the ability to easily insert the current time/date string in the format you choose.

Printing is supported by *VED*, and the output can be sent to either a printer or a disk file. Options such as margin settings, new page, effect sequences (such as underline on/off) and headers may be defined using "dot" commands in the document. For more complex formatting needs, Bob van der Poel Software also offers *VPrint*, a separate product that can be used in conjunction with *VED*.

There are a few miscellaneous and very useful functions in *VED*. An OS-9 shell can be called at any time through a simple command sequence. Memory and file information can be displayed at any time, and commands can be easily repeated a number of times. Cursor blinking can be turned on and off, and the auto-numbering mode can be engaged at will to insert line numbers after each carriage return.

VED's on-line help is completely menu-driven and generally easy to use. The manual also describes the format of the help and environment files for those users who may be interested in modifying them for their own use. Getting help for any command requires only that you remember that ESCAPE-H is used to bring up the Help display.

VED appears to be a well-designed product that is surprisingly intuitive, setting it apart from other line editors, which often send users running for a sledge hammer. Movement commands are very easy to remember, especially since most are mapped to the and movement keys on the keyboard. Many of the commands are grouped into two-letter sequences. For example, to use one of the Options commands, press ESCAPE-O followed by the letter specifying the subcommand you want to use. If you hesitate after initiating the first command, *VED* automatically lists the available letters at the top of the screen.

Bob van der Poel's attention to speed is very noticeable from the performance of *VED*. For example, *VED* checks for any keystrokes entered while the screen is being updated and does not redraw the complete screen if the next update will fill the screen with new information. For this reason, using OS-9's key-repeat feature with Page Up and Page Down is extremely fast.

If you are looking for a quality editor for your OS-9/68000 system, *VED* is a sure winner. The price is reasonable, and *VED* may be the most often used piece of software on your system, especially for writers or programmers. With the addition of *VPrint*, you can count on the most advanced text tools for the OS-9/68000 system to deliver the performance you need. *(Bob van der Poel Software, P.O. Box 57, Wynndel, BC V0B 2N0, Canada, 604-866-5772; or P.O. Box 355, Porthill, ID 83853-0355; $39.95 plus $3 S/H.)*

— Jordan Tsvetkoff

Intercom

CoCo Clubs

CALIFORNIA

☎ StG Net West, Alan Sheltra, P.O. Box 38713, Hollywood, 90038, (818) 761-4135, BBS (818) 761-4721

COLORADO

☎ Colorado Springs Color Computer Club, Bud Ward, 1118 Claiborne Road, Colorado Springs, 80906-5513, (719) 392-8268

CONNECTICUT

☎ Connecticut CoConnt Connection, Charles Joseph Scanlon, 2 Eagle Lane, Simsbury, 06070, (203) 657-8373

FLORIDA

☎ The Color Computer 3 Users Group, Tom Batchelder, 6042 Syrcle Ave., Milton, 32570, (904) 623-4405

GEORGIA

☎ Atlanta Computer Society, Inc., Alan R. Dages, 4290 Bells Ferry Road Suite 10639, Kennesaw, 30144, (404) 469-5111 voice, (404) 636-2991 modem

IDAHO

☎ Snake River Color Computer Club, Emil Franklin, 1750 Carmel Drive, Idaho Falls, 83403, (208) 522-0220

ILLINOIS

☎ Cook County Color Computer Club, Howard Luckey, 10 McCarthy Rd., Park Forest, 60466-2122, (708) 747-0117

☎ Motorola Micro Computer Club, Steve Adler, 1301 East Algonquin Rd., Shaumburg, 60196, (708) 576-3044

IOWA

☎ Metro Area Color Computer Club, Joe Cavallaro, 2425 Ave A, Co. Bluffs, 51501, (712) 322-2438

☎ Mid Iowa & Country CoCo, Terry Simons, 1328 48th Street, Des Moines, 50311, (515) 279-2576

KENTUCKY

☎ Hardin County Color Computer Club, Paul Urbahns, 2887 Republic Ave., Radcliff, 40160, (502) 351-4757

LOUISIANA

☎ The CoCo SIG, Christopher Mayeux, 20 Gibbs Drive, Chalmette, 70043, (504) 277-6880 voice, (504) 277-5135 modem

MARYLAND

☎ Arkade, John M. Beck, 3513 Terrace Drive #D, Suitland, 20746, (301) 423-8418

MASSACHUSETTS

☎ NorthEast CoCo Club, Jose Joubert, 440 North Ave., Bldg. 9 #210, Haverhill, 01830, (508) 521-0164

MICHIGAN

☎ Color Computer Owners Group, Bernard A. Patton, 388 Emmons Blvd., Wyandote, 48192, (313) 283-2474

☎ Greater Lansing Color Computer Users Group, E. Dale Knepper, P.O. Box 14114, Lansing, 48901, (517) 626-6917

MISSISSIPPI

☎ Mississippi OS-9 User Group, Boisy G. Pitre, Southern Station, Box 8455, Hattiesburg, 39406-8455, (601) 266-2807

MISSOURI

☎ CoCoNuts User Group, Clyde Lloyd, 2116 N. Columbia, Springfield, 65803, (417) 866-8738

☎ KC CoCo, Gay Crawford, P.O. Box 520084, Independence, 64052, (913) 764-9413

NEBRASKA

☎ Bruce Gerst c/o Metro Area CoCo Club, P.O. Box 3422, Omaha, 68103

NORTH CAROLINA

☎ Raleigh CoCo Club, P.O. Box 10632, Raleigh, 27605, (919) 878-3865

☎ The Tandy Color Computer Users of Charlotte, Eric Stringer, 1022 Noles Dr., Mt. Holly, 28120

OHIO

☎ The Greater Toledo Color Computer Club, Bill Espen, 1319 North St., Bowling Green, 43402, (419) 471-9444

☎ Tri-County Computer Users Group, Ron Potter, 10914 Oliver Road, Cleveland, 44111, (216) 476-2687

PENNSYLVANIA

☎ Cumberland Valley Users Group, Thomas Martin, 9085 Newburg Road, Newburg, 17240, (717) 423-5525

RHODE ISLAND

☎ New England "CoCoNuts" Color Computer Club, Arthur J. Mendonca, P.O. Box 28106 North Station, Providence, 02908, (401) 272-5096 (Sig3)

SOUTH CAROLINA

☎ Spartanburg CoCo Club, Jesse W. Parris, 152 Bon Air Ave., Spartanburg, 29303, (803) 573-9881

SOUTH DAKOTA

☎ Empire Area Color Computer Users Group of South Dakota, Carl Holt, P.O. Box 395, Brandon, 57005, (605) 582-3862

TEXAS

☎ The Codis CoCo Symphony, William C. Garretson, 2902 Harvard St., Irving, 75062, (214) 570-0823

UTAH

☎ Salt City CoCo Club, L. Todd Knudsen, 6357 S. Lotus Way, West Jordan, 84084, (801) 968-8668

WASHINGTON

☎ Bellingham OS-9 Users Group, Rodger Alexander, 3404 Illinois Lane, Bellingham, 98226, (206) 734-5806

☎ Port O' CoCo, Donald Zimmerman, 3046 Banner Rd. SE, Port Orchard, 98366-8810, (206) 871-6535

AUSTRALIA

☎ Australian National OS-9 Users Group, Gordon Bentzen, C/- 8 Odin Street, Sunnybank, Queensland, 4109, (07) 344-3881

☎ Brisbane Southwest Colour Computer Users Group, Bob Devries, 21 Virgo St., Inala, Queensland, 4077, (07) 372-7816

CANADA

☎ Club d'Oridinateur Couleur du Quebec Inc., 8000 Metropolitain est, Anjou, Quebec, H1K 1A1, (514) 354-4941

GERMANY

☎ OS-9 Users Group in Europe, Burghard Kinzel, Leipziger Ring 22A, 5042 ERFTSTADT, +49-2235-41069, (OS-9/6809)

THE NETHERLANDS

☎ European OS-9 User Group, Peter Tutelaers, Strijperstraat 50A, 5595 GD Leende, s88405777@hsepml.hse.nl, +31-4906-1971, (OSK)

PUERTO RICO

☎ Puerto Rico Color Computer Club, Luis R. Martinez, P.O. Box 2072, Guaynabo, 00657-7004, (809) 799-8217 or (809) 728-2314

Bulletin Board Systems

State/City	BBS Name	Access Number	Parameters (Speed-Parity-Word Bits-Stop Bits)	SysOp
Arkansas				
Sheridan	The Grant County BBS	(501) 942-4047	300/1200/2400-N-8-1	Eddie Gilmore
California				
Hollywood	Zog's Cavern BBS	(213) 461-7948	300/1200/2400-N-8-1	Alan Sheltra
Connecticut				
Manchester	Silk City BBS	(203) 649-9057	300/1200/2400-N-8-1	Darren Kindberg
Waterbury	Applause BBS	(203) 754-9598	300/1200/2400-N-8-1	Carmen Izzi, Jr.
Hawaii				
Ft. Shafter	CoCo'Nuts BBS Service	(808) 845-7054	300/1200/2400-N-8-1	Tommie Taylor
Idaho				
Idaho Falls	Snake River Computer Club BBS[1]	(208) 523-3796	300/1200-N-8-1	Jon Gould
Illinois				
Carpentersville	The Pinball Haven BBS	(708) 428-8445	300/1200/2400-N-8-1	Jeffrey R. Chapin
Elmhurst	Glenside's Cup of CoCo BBS	(708) 428-0436	300/1200/2400-N-8-1	Tony Podraza
Kentucky				
Elkhorn City	Cross-N-Crown BBS	(606) 754-9420	300/2400-N-8-1	Tim McIntosh
Michigan				
Manistee	Crystal Palace	(616) 723-0146	1200/2400-N-8-1	Nelson Howard
Mississippi				
Hattiesburg	The OS-9 Zone[2]	(601) 266-2807	300/1200/2400-N-8-1	Boisy G. Pitre
New York				
Wappingers Falls	The Dutchess CoCo	(914) 838-1261	300/1200/2400-N-8-1	Chris Serino
North Carolina				
Wilmington	Bill's Board	(919) 395-4366	300/1200/2400-N-8-1	Bill Medcalf
North Dakota				
Minot AFB	The 9-Line BBS	(701) 727-6826	300/1200-N-8-1	David Hensley
Ohio				
Columbus	Springwood BBS	(614) 228-7371	300/1200/2400-N-8-1	Edward Langenback
Pennsylvania				
Conshohocken	Charlie's Help Line	(215) 825-3226	300/1200-N-8-1 or N-7-1	Charles DiMartino
Rhode Island				
Central Falls	The Weather Connection II BBS	(401) 728-8709	300/1200/2400-N-8-1	Eric Chew
Virginia				
Fall Mills	Clem's Corner BBS[3]	(703) 322-4053	300/1200-N-8-1	Richard Douglas Bailey
Washington				
Fircrest	OS-9 Tacoma	(206) 566-8857	300/1200/2400-N-8-1	Chris Johnson
Wisconsin				
Marinette	Phoenix Interstate Data Systems[4]	(715) 732-1036	300/1200/2400/9600-N-8-1	Joe Boburka
Canada				
Twillingate, NF,	ColorNET BBS	(709) 884-2176	300-N-8-1	Jason Woodford
Windsor, Ontario	Color Connection	(519) 948-1879	300/1200-N-8-1	Cory Richert

Notes:
[1]Snake River Computer Club BBS supports all types of computers.
[2]The OS-9 Zone is up from 10 p.m. to 6 a.m. seven days a week.
[3]Clem's Corner BBS is up from 6 p.m. to 11 p.m. seven days a week.
[4]Phoenix Interstate Data Systems has a .75/hr charge for premium services, paid in advance.

Submitting Material To Rainbow

Contributions to THE RAINBOW are welcome from everyone. We like to run a variety of programs that are useful, helpful and fun for other CoCo owners.

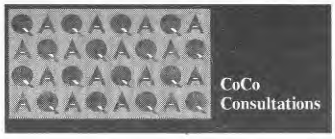
WHAT TO WRITE: We are interested in what you want to tell our readers. We accept for consideration anything that is well-written and has a practical application for the Tandy Color Computer. If it interests you, it will probably interest lots of others. However, we vastly prefer articles with accompanying programs that can be entered and run. The more unique the idea, the more the appeal. We have a continuing need for short articles with short listings. These are especially appealing to our many beginners.

FORMAT: Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs and debug our typing errors. All programs should be supported by some editorial commentary explaining how the program works. We also prefer that editorial copy be included in ASCII format on the tape or disk, using any of the word processors currently available for the Color Computer. Also, please include a double-spaced printout of your editorial material and program listing. Do not send text in all capital letters; use upper- and lowercase.

COMPENSATION: We do pay for submissions, based on a number of criteria. Those wishing remuneration should *so state* when making submissions.

For the benefit of those wanting more detailed information on making submissions, please send a self-addressed, stamped envelope (SASE) to: Submission Guidelines, THE RAINBOW, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit material currently submitted to another publication.



MIDI Hints

Those who have downloaded Lyra *and* UltiMusE *files from Delphi might be interested in the following recommendations for MIDI keyboards that can be used to play these files. Although* Lyra *can play its files via the speaker in your monitor or TV, the result often sounds quite poor because the files were composed for a MIDI keyboard. The following is a list of MIDI keyboards in the price range of $200 to $1000: Casio models CPS720, CT636, CT656, CT670, and CT680; Kawai FS690, K1II, K4, M8000, and PH50; Yamaha PSR300, PSR400, PSR500, PSS795, and YPR20.*

Do not buy the older Yamaha models PSR-48 or PSR-38 because they lack the capability to play enough notes at once. My own favorites in this group are the Casio CT680 and the Yamaha PSS795. I will at times chain these two keyboards by running a cable from the MIDI Thru connector of one into the MIDI In connector of the other so they both are playing at the direction of my CoCo. In this way the weaknesses of one are balanced by the strengths of the other. The built-in speakers in such keyboards are generally of poor quality. You should send the output to a stereo amplifier and decent high-fidelity speakers. I personally find UltiMusE 3 *superior to* Lyra. UltiMusE 3 *requires OS-9 and a 512K CoCo 3. Call Kala Software and/or Rulaford Research for more details on these two programs.*

Danny Faye (DFYE)
Independence, Missouri

Disk Drives and the CoCo

Can you refresh my memory about which models of Color Computer floppy-drive systems were made by Radio Shack? I'm specifically interested in knowing what models of Radio Shack disk controllers work with a CoCo 3 without a Multi-Pak. Also, please tell us how to identify these controllers?

Dave Myers (DAVEMYERS)
Ypsilanti, Michigan

A Radio Shack made, as best I can recall, five revisions of its disk controller. The very first drive system (I mean disk controller with drives, case and power supply) was Catalog No. 26-3022. This disk controller is unique in that all of its chips were socketted. It used a 40-pin WD1793 Floppy Disk Controller (FDC) chip and required both +5 and +12 volts. This controller cannot be used with a CoCo 2 without a Multi-Pak Interface or a Slot Pak because the CoCo 2 does not supply 12 volts. This controller cannot be reliably used with a CoCo 3 at all (even with a Multi-Pak or Slot Pak) because its data separator was a tad on the sloppy side, and so most versions of the 26-3022 controller won't work properly with a CoCo 3 when the CoCo 3 is running at high speed. I have received occasional reports from users who succeeded in using this ancient controller with a CoCo 3, but you cannot count on such success (even with a Multi-Pak) and I strongly recommend you don't even bother to try it. This disk controller was usually packaged with a gray-case full-height TEC drive. The drive itself was a single-sided, 35-track unit that was incapable of stepping faster than 20ms per track. This full-height TEC drive also had a notoriously poor cam-drive head-step mechanism and tended to go out of allignment quite easily.

The second CoCo disk system Tandy offered (it first appeared in the 1984 catalog) was Catalog No. 26-3029. The disk controller of this system used a 40-pin FDC chip, which was a clone of the WD1793 chip made by either Fujitsu (MB8877A) or Mitsubishi (M5W1793-02P). These two chips represented an improvement over the original Western Digital 1793 in that they did not require a source of +12 volts but ran happily off a single +5-volt supply. In the 1984 catalog this system was referred to as "Color 2 Disk #0 Kit." The FDC chip was socketted, as was the 8-pin data-separator chip (FDC 9216), but other chips were soldered to the circuit board. This was — in the opinion of many assembly-language hackers who wrote copy-protection systems or programs to break or clone copy-protected disks — the best, the most reliable and the most stable of CoCo disk controllers ever made, though the three units that came after this one were almost as good. This controller and all that followed work quite happily with all models of Color Computer, including the CoCo 3.

The 26-3029 system was usually packaged in a white case with a full-height, 40-track, single-sided drive capable of stepping at 6ms. The drive was usually one whose mechanism was made by Tandon (for its Tandon TM 100-1 drives), but whose logic board was manufactured by Texas Peripherals Inc. for Tandy. This was a most rugged and reliable drive, one of the best of the full-height, single-sided drives ever made.

Around 1985 Tandy began to offer a completely redesigned Color Computer disk system. This used a disk controller that still was full-size like its two predecessors, but which employed a single 28-pin WD1773 FDC chip that did the work of the 40-pin 1793 *and* the 8-pin 9216 data-separator chip. I'm not too sure about the details of this intermediate offering, but it may have been called the FD-500 drive system. The controller was accompanied by a white case that sat horizontally on the desk with a half-height, 40-track, single-sided drive capable of stepping at 6ms. The case had space for a second half-height drive.

By 1986 Tandy was offering in its catalog the FD-501 (Catalog No. 26-3131) drive system. This was fundamentally similar to the FD-500, but was offered with a redesigned disk controller that was physically smaller (shorter) than the older disk controllers and came in a correspondingly smaller plastic case. Like the FD-500, it used a 28-pin WD1773 FDC chip and worked happily with all models of Color Computer. The half-height drive that usually was supplied with this system was a TEC 501 drive, an especially rugged and reliable single-sided, 40-track drive that was capable of stepping at 6ms.

The last drive system offered by Tandy (which appeared first in the 1988 Radio Shack catalog) was the FD-502 system, Catalog No. 26-3133. This featured a circuit board with a design relatively similar to that of the FD-500 and FD-501 but physically smaller. The FD-502 was unique among all systems offered by Tandy in that it was supplied with a half-height, 40-track, double-sided drive that was capable of stepping at 6ms. This drive was, I believe, made by Tandon and enjoyed a reasonably good track record in the field. This system works happily with all models of the CoCo.

The bottom line is that all Tandy disk controllers except the ancient 26-3022 work with all models of the Color Computer, and the controllers from these systems can be used with double-sided drives of your choice.

Single- vs. Double-Sided

I have on occasion formatted the back side of a single-sided disk, and it appears to format and work just fine. Am I letting myself in for any problems when I do this?

Edward Stroh
Thornton, Illinois

A In the early days of floppy disks, there may have been some significance to labeling a disk single-sided versus double-sided. In those days the technology for laying down the media on the disk was likely poorer than it has been for the last half decade or so, and so boxes of disks that were not certified and tested as working on both sides might actually contain a small but significant percentage of disks that actually had bad media on the other side. This almost certainly is no longer true. I suspect almost all disks that have been sold as single-sided in the last five years are perfectly good on both sides. Indeed, it is hard to find any disks labeled single-sided these days. What makes all of this a moot point is that certified 5¼-inch, double-sided disks are commonly available for under 25 cents each . . . often for as little as 10 cents each. At this price, it hardly pays to bother with ancient, single-sided disks.

Viewing GIF Files in Disk BASIC

Is there a simple way to view GIF files on a CoCo 3 (without having to first convert them) using a program running under Disk BASIC?

John Burke (JBURKE)
Fremont, California

According to Brian Flahive (BFLAHIVE), there is. *The Projector*, a program available in the Graphics database on Delphi's CoCo SIG, gives users the ability to directly view GIF files.

Disk-Drive Terminators

I've heard about the need to remove terminator resistors from disk drives when adding a second drive, and the requirement that there must be one and only one terminator resistor in a given drive system. Can you explain to me just what a terminator resistor is, what it does, why there must be only one, and what it looks like?

Daniel Holley (MRINTENSITY)
Frankfort, Kentucky

A Electrically, a terminator resistor is a package that contaios several (usually seven or eight) individual resistors. These packages often look like DIP ICs with 14 or 16 pins. These types of terminator resistors have each internal resistor hooked across each facing pair of pins. Such terminators are usually composed of 150-ohm resistors and indicated as such by a designatiou "151" printed on them. The "151" translates to "15 X 10 to the 1st power." However, some disk drives can take other forms of terminator resistors. Some older Tandy drives used terminator resistors in the form of single in-line pin packages. Such packages appear as a small rectangular blob of epoxy resin with five, six, seveu or eight pins in a row coming out of the blob. These are typically internally wired so that one lead of all the internal

resistors goes to one (marked) pin, and the other leads go individually to the other pins. Usually the terminator resistor is socketted. With such drives, be sure there is one and only one terminator resistor in the drive system.

On a very few 5¼-inch disk drives (such as the Tandy FD-502) and on all newer 3½-inch third-height drives, the terminator resistors are 1000 ohms and are permanantly soldered on the drive logic board. If you are adding another drive with a 1000-ohm resistor pack, just add it and don't worry about the terminators. If you want to add a drive that takes a 150-ohm terminator, try adding the drive with a terminator resistor installed. Better yet, find a 300-ohm terminator (instead of the usual 150-ohm terminator) that fits the socket on the drive, and use that if you want to use such a drive with a soldered 1000-ohm terminator.

The reason terminator resistors are required is that many of the control lines coming out of a disk controller originate in open-collector driver chips (usually 7416 or 7406 chips in a CoCo controller, or a 7438 on many IBM-PC type controllers). Open collector gates have the ability to pull a signal down to Ground, but lack the ability to push a signal High on their own. They must be used with pull-up resistors if they are to function properly. These pull-up resistors in the case of the system are located on the drive itself and called the terminator resistor. The reason for this arrangement is that open-collector gates with pull-ups at the other end of the controlling line make for especially reliable, noise-free communications along the cable between the controller and the drive.

If there is too small a terminator resistor (as is the case if there are several 150-ohm terminators in the system, effectively putting those resistors in parallel with each other), the open collector-gate is pulled too strongly toward +5 volts (Logic 1), and finds itself unable to pull the signal to Ground (Low) when told to do so. This is the reason too many terminator resistors in a drive system result in total failure or in unreliable operation.

In more modern drive arrangements, such as the newer 3½-inch third-height drives, it was learned that a 1000-ohm terminator is adequate for pulling up the open-collector gates, and yet even if four such terminators are paralleled, the effective terminator resistance on each line is still only 250 ohms — still more than the 150-ohms used in older systems. I hope this gives you some more insight on what is going on with these mysterious terminator resistors.

Tying up the Keyboard

A Several folks have asked me recently how much time the CoCo spends scanning the keyboard under BASIC. My own guess was that the figure is around five percent, but I asked Art Flexser (ARTFLEXSER) to suggest a straight-forward way to empirically measure this? Here is his answer:

Q *If you enter* POKE &HADEB,&H39 *on a CoCo 3, you will knock out the check for* BREAK *and* SHIFT-@ *that is done between every* BASIC *statement when a* BASIC *program is run and replace the start of that subroutine with an* RTS. *Using this you should be able to construct a simple test to see how much time a* BASIC *program takes to run with and without that part of the keyboard scan knocked out. I agree with your guess of five percent of the total running time, but I await your report on the results of the experiment.*

A I've not had time to do this experiment. Interested readers are invited to try it in various forms and write THE RAINBOW with what they find.

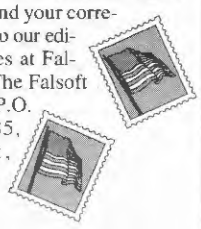
Martin H. Goodman, M.D., a physician trained in anesthesiology, is a longtime electronics tinkerer and outspoken commentator — sort of the Howard Cosell of the CoCo world. On Delphi, Marty is the SIGop of THE RAINBOW*'s CoCo SIG. His non-computer passions include running, mountaineering and outdoor photography. Marty lives in San Pablo, California.*

About Your Subscription

Your copy of THE RAINBOW is sent second class mail. You must notify us of a new address when you move. Notification should reach us no later than the 15th of the month prior to the month in which you change your address. Sorry, we cannot be responsible for sending another copy when you fail to notify us.

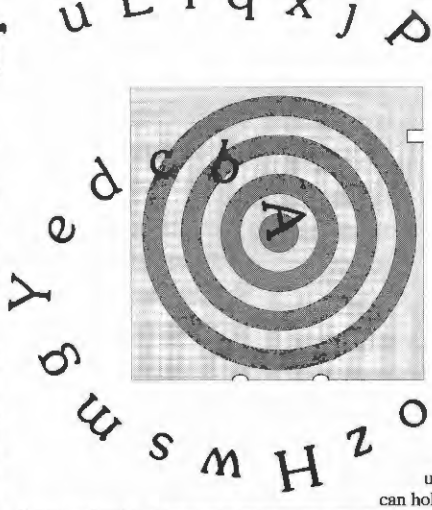
Your mailing label also shows an account number and the subscription expiration date. Please indicate this account number when renewing or corresponding with us. It will help us help you better and faster.

For Canadian and other non-U.S. subscribers, there may be a mailing address shown that is different from our editorial office address. Send your correspondence to our editorial offices at Falsoft, Inc., The Falsoft Building, P.O. Box 385, Prospect, KY 40059.



Feature Program

Get Your Disks in Order



Few would disagree that having your disk files in alphabetical order makes it much easier to find a specific file quickly. I wrote *Disk Sorter* for just this purpose — it reads a disk's directory, sorts the information contained there, then writes it back to the disk.

Disk Sorter is a CoCo 3 program designed to work with standard 35-track disks. However, it is easy to modify for use with earlier versions of the Color Computer. (I'll show you how in just a minute.) First, enter the program as shown in the listing. Be especially careful and check for possible syntax errors since a stray character could cause the program to crash a disk during the sorting process. When you are sure the program is "clean," save it to disk.

To sort a disk, run *Disk Sorter*; the program prompts you to enter a drive number. Enter the number (from 0 to 3) of the drive that holds the disk you want the program to sort. Drive 0 is assumed if you simply press ENTER at this prompt. Depending on the number of files on the disk, it may take a little while for *Disk Sorter* to do its thing. After the sort is complete, the program executes a `DIR` command to show you a listing of the sorted directory.

Those of you with CoCo 1's and 2's cannot use *Disk Sorter* as printed because it coutains pokes and statements applicable only to the CoCo 3. To modify the program for use with earlier CoCos, you must delete or edit a few program lines. First delete Line 160, which is used to set the screen width. Then delete Line 210, which disables the BREAK key on the CoCo 3. Also delete Line 220, the error trap, and lines 710 through 740. Delete Line 360, which enables the high-speed mode, or edit it for the CoCo 1 and 2 by changing `&HFFD9` to `&HFFD7`. Similarly; delete Line 560 (disables the high-speed mode) or change `&HFFD8` to `&HFFD6` in that line.

Whenever you are using or modifying *Disk Sorter*, make sure the computer is not in the high-speed mode. Otherwise an I/O error may result, or the CoCo may trash the disk. *Disk Sorter* enables the high-speed mode only to increase the speed of the sort routine (a simple bubble sort). It is also important that you never press BREAK or the Reset button while the program is running to avoid the possibility of trashing the disk. (CoCo 3 users need not worry about pressing BREAK since that key is trapped.)

CoCo Disk Structure

A standard Color Computer disk contains 35 tracks (numbered 0 through 34) for data storage. These tracks are arranged as concentric circles on the disk, with Track 0 being the outermost. Each track is divided into 18 sectors (numbered 1 through 18). One sector on a CoCo disk contains 338 bytes of which 256 are used for data storage (the remaining bytes are used for system controls).

Track 17 is a special track that holds the directory for the disk. The actual directory entries are stored in sectors 3 through 11, and the file-allocation table (FAT) is located in Sector 2. To increase performance, sectors on a disk are grouped in *granules* — one granule is comprised of nine contiguous sectors — and there are two granules per track, 68 user granules per disk. A granule is the smallest unit of disk storage available with Disk BASIC. In other words, even a one-line program that is only 40 bytes in length is allocated a whole granule (2304 bytes). The FAT is a map of the disk granules occupied by each file stored on the disk.

Each directory entry on Track 17 requires 32 bytes, so eight directory entries (8 entries x 32 bytes equals 256 bytes) fit into one of the nine sectors (3 through 11) set aside for this purpose. Again, since a granule is the smallest supported unit of disk storage, a CoCo disk can hold a maximum of 68 files.

`D`	selected disk drive
`DB$`	disk buffer
`DE$`	array of directory entries
`I`	loop counter/directory-entry index
`J`	directory-entry offset in disk buffer
`N`	number of deleted and existiug directory entries
`P`	sort pass counter
`S`	sector
`T$`	temporary variable used for sort
`X$`	temporary variable

Figure 1: Variable Dictionary

The first eight bytes of a directory entry hold the filename proper, and the extension is stored in the next three bytes. If the first byte of a specific directory entry (filename) is $00, the file orginally pointed to by that directory entry has been deleted and the entry is available. If the first byte is $FF (decimal 255), this and all subsequent entries have never been used and are free.

Disk Sorter helps speed the system a little by pushing all deleted (previously used) directory entries to the front of the directory when it sorts. When you use `DIR` to get a listing, you won't see these empty entries. However, since they are in front, Disk BASIC won't require as much time to find a free entry when storing a file as it does when the free entries are spaced throughout the directory.

Disk Sorter is a great tool for sorting the directory information on your CoCo disks. I hope you enjoy using the program and that it relieves some of the headaches of trying to find files in the disk haystack. If you have any questions or comments about *Disk Sorter*, please feel free to contact me. Make sure to send an SASE (with Canadian postage) if you need a reply.

Geoff Friesen has a bachelor of science degree in computer science and mathematics. He is the author of several published articles about computers. He may be contacted at General Delivery, Dauphin, MB R7N 2T3, Canada, (204) 638-7302. Please include an SASE when requesting a reply.

CoCo 3/32K Mod.

The Listing: `DISKSORT`

```
1 'DISK SORTER
2 'BY GEOFF FRIESEN
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
100 'DSORT (DISK SORT)
110 '
120 'INITIALIZE
130 '
140 CLEAR 3000
150 DIM DB$(2),DE$(72)
160 WIDTH 32
170 PRINT "DSORT (DISK SORT)"
180 PRINT
190 INPUT "DRIVE (0-3)"; D
200 IF D<0 OR D>3 THEN 190
210 POKE &HE42B,&H21
220 ON ERR GOTO 710
230 '
240 'INPUT DIRECTORY
250 '
260 I=0
270 FOR S=3 TO 11
280 DSKI$ D,17,S,DB$(0),DB$(1)
290 FOR J=1 TO 97 STEP 32
300 DE$(I)=MID$(DB$(0),J,32)
310 DE$(I+4)=MID$(DB$(1),J,32)
320 I=I+1
330 NEXT J
340 I=I+4
350 NEXT S
360 POKE &HFFD9,0
370 '
380 'DETERMINE NUMBER OF ENTRIES
390 '
400 N=0: I=0
410 X$=LEFT$(DE$(I),1)
420 IF X$=CHR$(255) THEN 450
430 N=N+1: I=I+1
440 IF I<72 THEN 410
450 IF N<2 THEN 700
460 '
470 'SORT DIRECTORY
480 '
490 FOR P=0 TO N-2
500 FOR I=0 TO N-P-1
510 IF DE$(I)<=DE$(I+1) THEN 550
520 T$=DE$(I)
530 DE$(I)=DE$(I+1)
540 DE$(I+1)=T$
550 NEXT I,P
560 POKE &HFFD8,0
570 '
580 'OUTPUT DIRECTORY
590 '
600 I=0
610 FOR S=3 TO 11
620 FOR J=1 TO 97 STEP 32
630 MID$(DB$(0),J,32)=DE$(I)
640 MID$(DB$(1),J,32)=DE$(I+4)
650 I=I+1
660 NEXT J
670 I=I+4
680 DSKO$ D,17,S,DB$(0),DB$(1)
690 NEXT S
700 DIR D
710 POKE &HE42B,&H27
720 IF ERNO=-1 THEN END
730 PRINT "ERROR"; ERNO; "@";
740 PRINT ERLIN
```

Switching Slots

Here is the answer you requested for Kent Holcomb in the October 1991 issue.

First, it should be mentioned that OS-9 Level II (at least on my disks) has `t3.dd` *and* `m1.dd` *incorrectly set up on the Config disk. The manual and help messages on the disk say that* `/t3` *is for the RS-232 Pak in Slot 2 of the Multi-Pak Interface and that* `/m1` *is for the Direct Connect Modem Pak in Slot 1. If you inspect the actual descriptors, you will see that they are backwards:* `/t3` *is set to work with* `modpak` *instead of* `aciapak` *while* `/m1` *is set to work with* `aciapak` *instead of* `modpak`. *Worse still,* `aciapak` *does not get its slot information from the descriptors but (as you said) has the slot hard coded. There is, however, a simple answer to Kent's problem using the following* `modpatch` *script.*

```
l aciapak
* replace 03 with x3 where x=slo
t (0-1, 1-2, 2-3, 3-4)
* this patch set for slot 2
c 68 03 13
v
```

At this point, you should use `cobbler` *to create a new boot disk so that the change is made permanent.*

Robert Gault
Grosse Pointe Woods, Michigan

Multiple Windows Onscreen

The cover of the October 1991 issue of THE RAINBOW looked really sharp! What are the parameters you used for setting up the three windows as shown on the monitor in the bottom-left corner of the cover?

Ernest Bazzinotti, Jr.
Dorchester, Massachussetts

To create the windows as shown on the cover, use `build` or an editor to create a shell script with the commands

```
wcreate -z
/w1 -s=2 00 00 39 24 02 03 02
/w2 40 00 40 12 00 04
/w3 40 13 40 11 00 01
```

and run the script. (Note that you must use a shell script because `wcreate` switches windows after it creates the first window. If you don't, OS-9 will create the windows on separate screens.) Once the windows have been created, enter

```
shell i=/w1&
shell i=/w2&
shell i=/w3&
```

to start shells on the windows. You can then use CLEAR to move to each window and start your applications. (Bear in mind you'll need 512K to get this fancy with windows.)

If you so desire, you can change the color of each window by modifying the last three numbers (02 03 02) for window `/w1`, and the last two numbers (00 04 and 00 01) for windows `/w2` and `/w3`. The parameters represent, in order from left to right:

starting x (horiz.) position
starting y (vert.) position
x size in characters
y size in rows
foreground color
background color
border color

The border color is specified for the first window only — all other windows on the same screen use the border color of the first window.

You may notice that we used rather odd dimensions for the windows. For example, the first window is 39 columns wide (x size=39) and the other two windows start at column 40. Also the second window is 40 columns by 12 rows, while the third window is 40 columns by 11 rows starting at row 13. This gives a 1-character margin between each window. If you prefer to have 40-column by 24-row and 40-column by 12-row windows without the margin, change the script as follows:

```
wcreate -z
/w1 -s=2 00 00 40 24 02 03 02
/w2 40 00 40 12 00 04
/w3 40 12 40 12 00 01
```

Alternatively, you may want to totally change the locations and sizes of the windows or add more windows to better suit your needs.

OS-9 vs. OS-9

The first time I saw the OS9: prompt, I was trying to dump a DL Logo *picture file to the printer and discovered I needed more "tools." I'm still trying to get the picture printed, but in the meantime I have a few questions coming from an absolute beginner who taught himself BASIC and is trying to start all over with OS-9 — which, so far, is a lot like Latin I studied in school: totally great stuff you never use anywhere.*

A while ago a Logo product was produced for the CoCo called DL Logo, *which ran under OS-9 Level I. This, like a lot of Level I programs, won't boot on the CoCo 3. But some Level I Version 2 programs, such as* DeskMate, *do boot and run on the CoCo 3.*

Conceptually, I understand OS-9 as a vanilla operating system with all these possibilities out there somewhere. So I have OS-9 Level I (I always manage to buy something just about the time it is worthless), and I have OS-9 Level II. What I don't understand is the difference between OS-9 Level I files and OS-9 Level II files. Or is there a difference? If OS-9 is just an operating system, shouldn't it be possible to move DL Logo *to an OS-9 Level II system?*

I don't mind venturing off into the future. I might even buy one of the new Tomcats. It would just be nice if, as technology jumps another parsec into the future, someone would leave behind a little packet of clues for those of us who follow at a distance but who must move through the same hoops and learning curves. This constant change and assumption that customers are continually thrilled about new stuff and are always ready to plunk down money puts off many people who assume that if they don't have the latest in point-and-click MacBurgers, they can't compete or compute. I work with these types every day. They laugh at my CoCo and then run off in search of cheap memory chips. It's a strange world.

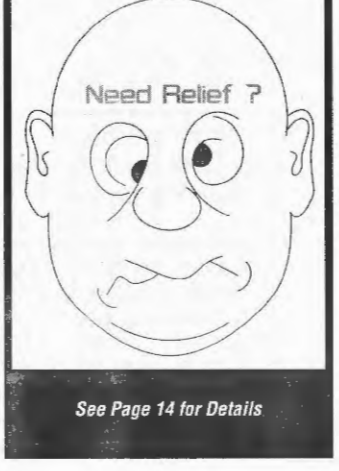
Michael Franich
Milton, Washington

OS-9 Level I versions 01.00.00 and 01.01.00 are virtually identical in all respects, and neither version will boot on the CoCo 3. This is because they both use the memory area between $FE00 and $FEFF. On the CoCo 3, this memory area is used by the GIME chip to store interrupt vectors and other goodies. OS-9 Level I Version 02.00.00 was written specifically in preparation for OS-9 Level II and the CoCo 3. This version is different in that the screen driver was split into three modules: `C032.io` handles the I/O for the CoCo's 32-column screen, `C080.io` handles the I/O for the PBJ WordPak-RS 80-column video board, and the module that handles I/O for the keyboard. As you can probably tell, this was done in preparation for Level II's windows.

OS-9 Level II splits the video handlers into three classifications: `VDGInt` handles OS-9 Level I compatible video, `GrfInt` is the standard windows driver, and `WindInt` replaces `GrfInt` to add pull-down menu support for *Multi-Vue*.

If you boot the system using `Term_VDG` (the 32-column green screen), most OS-9 Level I software will run with OS-9 Level II. For example, *DL Logo* should work fine when run from the VDG screen. To do so, boot OS-9 Level II and insert the *DL Logo* disk into the drive. Use the `chd` and `chx` commands to change the current directories to the *DL Logo* disk. (For example, `chd /d0` and `chx /d0/cmds`.) Try running the program to see if it works.

As you are alluding, OS-9 is just an operating system. Software written for OS-9 Level I should run with OS-9 Level II without any problems. Keep in mind that if a particular program uses any *tricks*, it may not work properly with OS-9 Level II. An example of this is *Profile* distributed by Computerware. *Profile* would have worked fine with OS-9 Level II, but it attempts to link to the `CCGo` module to search for the string `TANDY` to make sure the program is running on a CoCo. Of course, since the CoCo 3 uses the `CC3Go` module, *Profile* aborts with an error.

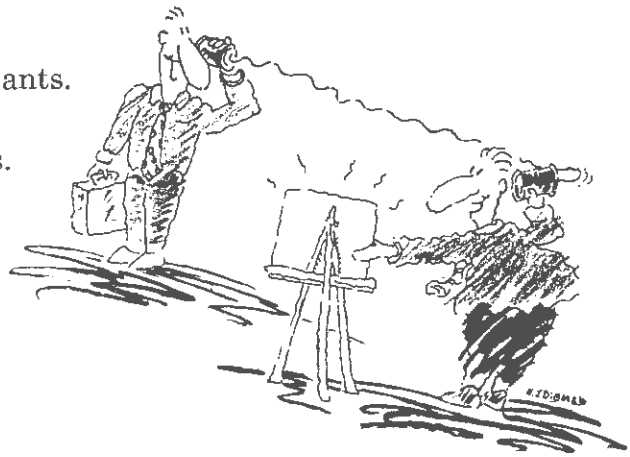


See Page 14 for Details

DELPHI – The $1 per hour online solution!

DELPHI's 20/20 Advantage Plan sets the standard for online value: 20 hours for only $20, for all the services you want!

- Thousands of files to download.
- Chat lines with hundreds of participants.
- Worldwide e-mail.
- Hobby and computer support groups.
- Multi-player games.
- Local access numbers in over 600 cities and towns.



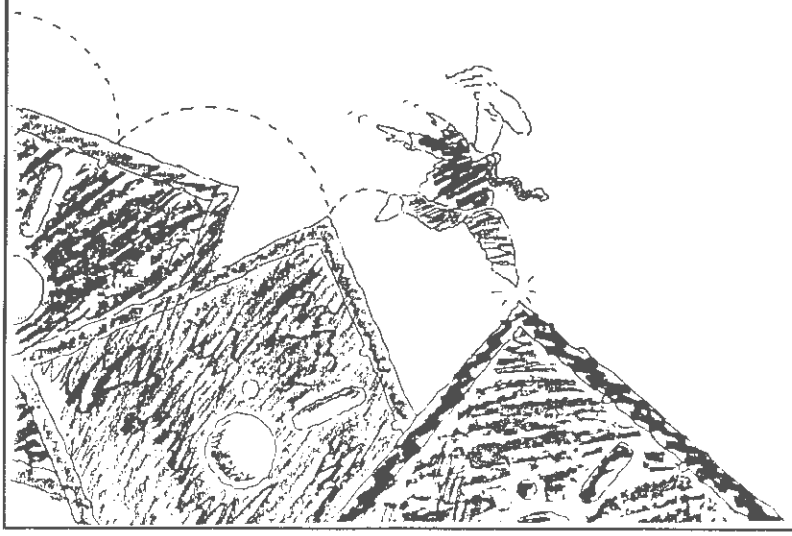
Trial Offer: 5 hours for $5!

Try DELPHI at $1 per hour. Join today and get 5 hours of evening and weekend access for only $5. If you're not satisfied, simply cancel your account before the end of the calendar month with no further obligation. Keep your account active and you'll automatically be enrolled in the 20/20 plan for the next month.

1. Via modem, dial 1-800-365-4636.
2. When connected, press RETURN once or twice.
3. At *Password*, enter **RB55**

Questions? Call 1-800-695-4005.

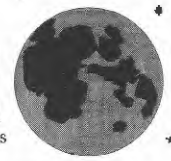
Rates apply for evening and weekend access from within the mainland US. There is a one-time enrollment fee of $19 when you join the 20/20 Advantage Plan. Further details are provided during the online registration.



DELPHI is a service of
General Videotex Corporation
1030 Massachusetts Avenue
Cambridge, MA 02138-5302

800-695-4005 • 617-491-3393

Animation Creation Through Machine Language



Since the first time I saw an 8mm cartoon, I've thought animation was something special; but it isn't restricted to those people pushing paper and pencil for Walt Disney. No, graphics animation is a natural use for the Color Computer — especially when combined with the speed of machine language. In this article, we'll look at how machine language can help the CoCo strut its stuff.

Traditional character animation involves rapidly "flipping" through a "deck" of pictures in which each successive image is slightly different from the last. Needless to say, the number of images required for a smooth animation sequence can be quite high. This is also true when using a computer for animating objects. Through *Animator* I'll show you how to use the CoCo to draw and save eight graphics pages, each containing twelve 64-by-64-pixel frames for animation, producing an animated sequence with a total of 96 frames.

To rapidly show 96 successive frames, we need to keep them in the computer's memory while flipping through them. After all, loading each page from disk when it is needed would use up valuable time and cause the sequence to appear jumpy. A standard PMODE4 graphics page occupies 6144 bytes, and there are eight pages, so we need to find 48K of free memory. We'll simply have the computer store them starting at a very low location in memory and going all the way up to $F800. Since memory above $8000 contains the CoCo's ROM routines, we'll poke a value into Location $FFDF so we can use the corresponding high-RAM area.

The workhorse of *Animator* is shown in Listing 1. The short assembly-language routine between lines 210 and 340 is used to place each graphics page in an appropriate memory area as it is loaded. (This routine works in conjunction with the BASIC program in Listing 3, which loads the actual graphics images from disk.) A standard PMODE4 graphics page starts at Address $0E00 and ends at $25FF. The assembly routine stores the first page at Location $3800 and the second page $2600 bytes above this. Each successive graphics page is stored $2600 bytes above the last as it is loaded by the BASIC program; memory storage ends at Address $F800. Lines 240 and 330 disable and enable the interrupts, while lines 250 and 320 set the high- and low-RAM flags as needed.

When all eight pictures have been loaded and stored, the assembly-language routine starting at Line 360 can be used to show the frames. As I mentioned earlier, each frame is 64 pixels high and 64 pixels wide. However, I want to double this size before displaying the frames, so each bit must be repeated in a line and each line shown twice. The FRAME macro takes care of this; it starts with the first frame in memory (Line 390) and uses SHOWIT (Line 670) to display a 64-by-64 area as a 128-by-128 pixel image. Let's look at how SHOWIT works.

Memory Location $1208 represents the top-left corner of a centered 128-by-128-pixel screen area. The graphics area we are using is eight bytes (64 bits) long and 64 bits wide, and the values for the image in this area are stored in ACROSS and DOWN. Register U is always used to hold the location of the current frame in memory. Store the first byte of the frame in Register B and use shifts to check each bit. An arithmetic shift left (ASLB) will remove the left-most bit and save it in the "carry register" (the carry bit in the condition-codes register). If that bit is 0, the register is clear and you can branch (BCC) to the next bit check.

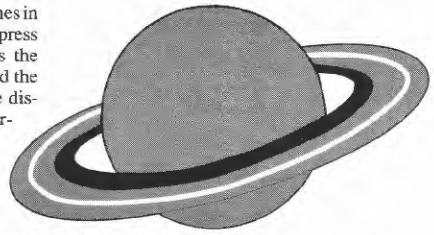
If the carry bit is High (a logic one), however, you must set two bits in Register A (remember, we're doubling the frame size). The first time through, we are dealing with Bit 7, so we'll need to set bits 7 and 6 of Register A. Do this either by ORing the contents with 192 (128+64) or ADDing 192 to Register A. The same procedure is carried out for Bit 6; if it is High, set the next two bits (5 and 4) in Register A using 48 (32+16). If Bit 5 of the graphics byte is set, set bits 3 and 2 in Register A by increasing it by 12 (8+4). Finally, if Bit 4 is High, set bits 1 and 0 of Register A by increasing it by 3 (2+1). Notice at this point we're only halfway through the number, and Register A is full; save Register A as N1, clear Register A, and repeat the process for bits 3 through 0 of the graphics byte. When you are finished, save Register A as N2.

When all of the first graphics byte has been checked, you have two numbers (N1 and N2) that "double" this byte. The program loads Register D with these values and stores them (as two bytes) 32 bytes away (one line) from the current screen location, and then again at the current screen location. The horizontal screen location is then increased by two (,X++) and the program is ready for the next byte.

When the first row is complete, the program jumps 24 bytes in the array (LEAU 24,U) to the start of Row 2 of that frame. It also jumps half a line plus another entire line on the screen (LEAX 48,X). Now it's just a matter of repeating everything until we've gone down all 64 rows of the frame. I've included a delay in Line 1100 that you can adjust in the BASIC program (Listing 3); a higher value increases the delay between frames. Once the first frame is shown, the FRAME macro increases the current graphics location by eight bytes (64 bits — the width of a frame) to get the next frame. When all four frames in the first row have been shown, the macro is recalled using the start of the next four frames. Altogether, the macro is called 24 times, displaying four frames each time. This certainly cuts down on the length of the source code, saving a lot of writing. Enter the source code in Listing 1 and check it for errors with A/NO/NS/WE. When it's error-free, save the source code using W ANIM.ASM and assemble it using A ANIM.BIN /NS/WE.

Now that we have a machine-language program to display successive frames of graphics, we need to find some graphics images to animate. Listing 2 shows a BASIC program that draws a series of images that simulate three planets revolving around a sun (or, perhaps three moons revolving around a planet). When you run this program, it will draw and save to disk eight full PMODE4 graphics pages.

Once the eight images are saved on disk, you can run the program shown in Listing 3, ANIMSHOW. This short BASIC driver loads the machine-language program (if necessary). It then loads the eight separate graphics pages and displays all 96 frames in succession. ANIMDRAW loops until you press the BREAK key. Line 250 contains the name of the ANIM file to be shown and the speed at which the animation will be displayed. You can experiment with different speeds by poking values into $37C4; After stopping the program using BREAK, you can resume the animation with the new speed by entering GOTO 220.



You can use these routines to animate just about any sequence by altering ANIMDRAW (Listing 2). Follow the same general format shown in the listing. The heart of the program is in lines 80 through 140 and Line 290. Once the twelve frames for one PMODE4 screen are drawn, the full image must be saved. Be sure the titles in Line 270 are different from any that you've already saved unless you want to erase the originals. Line 230 was necessary in my planetary sequence since everything revolves around the center of each frame. Depending on how you're drawing your animation, you may want to start HH and VV at (0,0) or at (0,63). Increase HH by 64 for each frame, and increase VV by 64 for each row, looping back to their original values at the end of each 12 frames. [*Editor's Note: Included on the August RAINBOW ON TAPE/DISK is different set of eight files that, when run through Animator, depict a 3D cube rotating in space. To view the sequence, remove the* REM *marker from Line 260 of* ANIMSHOW *and place a* REM *marker at the beginning of Line 250.*]

Another alteration you may want to make is to convert *Animator* to generate color animations, but you'll probably have to double the frame width resulting in the ability to use only half as many frames. CoCo 3 users may be able to use a lot more memory and increase the number of frames. A quick animation sequence like my drawing in Listing 2 could be saved directly into RAM instead of saving it to disk and reloading it. If you're zooming in on a Mandelbrot Set, however, image drawing takes so long that you'll need to save each page as you go and reload it later.

That's all for this month. If you have any suggestions or ideas for future articles, or questions about any of my articles, please let me know.

Bill Nee bucked the snowbird trend by retiring to Wisconsin from a banking career in Florida. The success of his 13-part series, "Machine Language Made BASIC" (July 1988 to July 1989), prompted him to continue writing articles about Color Computer machine-language programming. You may contact Bill at Route 2, Box 216C, Mason, WI 54856-9302, (715) 746-2952. Please include an SASE when requesting a reply.

64K Disk

Listing 1: ANIM

```
00100 FRAME     MACRO
00110           LDU     #$\0
00120           LBSR    SHOWIT
00130           LDU     #$\0+8
00140           LBSR    SHOWIT
00150           LDU     #$\0+$10
00160           LBSR    SHOWIT
00170           LDU     #$\0+$18
00180           LBSR    SHOWIT
00190           ENDM
00200
00210           ORG     $3500
00220 LOAD      LDX     #$E00
START OF GRAPHICS
00230           LDU     #$3800
START OF STORAGE (CHANGES)
00240           ORCC    #$50    DIS-
ABLE INTERRUPTS
00250           CLR     $FFDF
HIGH-RAM FLAG
00260 L1        LDD     ,X++    GET
FIRST 2 GRAPHICS BYTES
00270           STD     ,U++    AND
SAVE THEM
00280           CMPU    #$F800  END
OF STORAGE AREA
00290           BHS     L1A     OUT
OF ROOM
00300           CMPX    #$2600  END
OF GRAPHICS PAGE
00310           BLO     L1      GET
NEXT 2 BYTES
00320 L1A       CLR     $FFDE   LOW-
RAM FLAG
00330           ANDCC   #$AF    EN-
ABLE INTERRUPTS
00340           RTS
00350
00360           ORG     $3520
00370 SHOW      ORCC    #$50
00380           CLR     $FFDF
00390 PAGE1A    FRAME   3800
FIRST FOUR FRAMES
00400 PAGE1B    FRAME   4000    NEXT
```

CBASIC Editor/Compiler

The ULTIMATE Color Computer Basic Compiler!!!

If you want to write fast machine language programs but you don't want to spend the next few years trying to write them in Assembly Language , then CBASIC is the answer!!!

CBASIC is the only fully integrated Basic Compiler and program editing system available for the Color Computer. It will allow you to take full advantage of all the capabilities available in your color computer without having to spend years trying to learn assembly language programming. CBASIC allows you to create, edit and convert programs from a language you are already familiar with Extended Disk Color Basic, into fast efficient machine language programs easily and quickly.

"The most complete Editor/Compiler I have seen for the CoCo... "-The RAINBOW March 1986

CBASIC is a powerful tool for the Beginner as well as the Advanced Basic or M.L. programmer. CBASIC features well over 150 compiled Basic Commands and Functions that fully support Disk Sequential and Direct access files, Tape, Printer and Screen I/O. CBASIC supports ALL the High and Low Resolution Graphics, Sound, Play and String Operations available in Enhanced Color Basic, including Graphics H/GET, H/PUT, PLAY and H/DRAW, all with 99% syntax compatibility.

Specify Coco 1, 2 or 3 Disk $149.00

DataPack III Plus V1.1

AUTOPILOT and AUTO-LOG Processors
X-MODEM DIRECT DISK FILE TRANSFER
VT-100 & VT-52 TERMINAL EMULATION

- No lost data even at 2400 Baud on the Serial I/O port.
- 8 Selectable Display Formats, 32/40/64/80 columns
- ASCII & BINARY disk file transfer via XMODEM.
- Directly record receive data (Data Logging).
- VT-100 emulation for VAX, UNIX and other systems.
- VT-100/52 cursor keys, position, PF & Alt. Kbd. keys.
- Programmable Word Length, Parity, Stop Bits
- Complete Full and Half Duplex operation,
- Send full 128 character set from Keyboard .
- Complete Editor, Insert, Delete, Change or Add .
- 9 Variable length, Programmable Macro Key buffers.
- Programmable Printer rates from 110 to 9600 Baud.
- Send Files from the Buffer, Macro Key Buffers or Disk.
- Display on Screen or Print the contents of the Buffer.
- Freeze Display & Review information On line .
- Built in Command Menu (Help) Display.

Supports: Modem-Pak & Deluxe Pak or Serial Port.

Specify Coco 1, 2, 3 Disk $49.95

EDT/ASM III

EDT/ASM III is a Disk based co-resident Text Editor & Assembler. It is designed to take advantage of the new features of the COCO 3. It has 8 Display formats from 32/40/64/80 columns. The disk also contains a free standing ML Debug Monitor.

EDT/ASM III has the most powerful, easy to use Text Editor available in and Editor/Assembler package for the Color Computer.

* Local and Global string search and/or replace.
* Full Screen line editing.
* Load and Save standard ASCII formatted files.
* Block Moce & Copy, Insert, Delete, Overtype.
* Create and Edit files larger than memory.

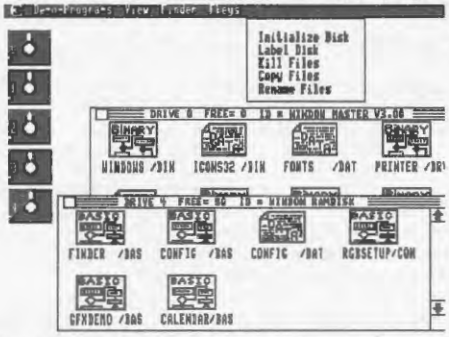
The Assembler features include:

* Conditional If/Then/Else assembly.
* Disk Library files up to 9 levels deep.
* Supports standard Motorola directives.
* Allows multiple values in FCB/FDB directives.
* Allows assembly from Buffer, Disk or both.

Specify Coco 1, 2 or 3 Disk $59.95

CER COMP

Window Master V3.0



The Ultimate User Friendly Point & Click operating System for your Coco 3. Simple enough even for children to use, just point and click to run programs, select files, do disk or file maintenance or almost any task you currently do by typing commands. You also get things like a print spooler, Programmable Function Keys, a Buffered Keyboard, Ramdisk, Serial I/O port and Deluxe Pak support along with Windows, Icons, Buttons, Pull Down Menus, Edit Fields and Mouse functions all in one program. It has multiple fonts in 54 possible sizes and styles, Enhanced Basic Editing and much much more. It add over 50 commands and Functions to Basic to fully support the Point & Click System without OS9.

$49.95

"..it offers so many features that it is probably underpriced. I recommend this software to all CoCo3 owners." -The Rainbow Feburary 1989

It is completely compatible with existing Basic programs and takes absolutely no memory away from basic. It requires 1 Disk Drive, R.S. hires interface & Joystick or Mouse.

Includes 128 & 512K Versions Only $69.95

LOOK!

"The Source"

The SOURCE will allow you to easily and quickly Disassemble machine language programs directly from disk and generate beautiful, Assembler Source Code. And " The Source" has all the features and functions you are looking for in a Disassembler .

* Automatic Label generation.
* Allows specifying FCB, FCC and FDB areas.
* Disassembles programs directly from Disk.
* Automatically locates addresses.
* Output listing to the Printer, Screen or both.
* Generates Assembler source directly to disk.
* Built in Hex/ASCII dump/display.
* 8 Selectable Display formats 32/40/60/80.
* Built in Disk Directory and Kill file commands.
* Menu display with single key commands.
* Written in Ultra fast machine language.

Specify Coco 1, 2 or 3 Disk $49.95

Window - Ware

Window Writer- A point & click Word Processor, powerful formatting capability, works wi...th any printer. On screen Italic, bold etc. WYSIWYG..... ..... $59.95

"Window Writer is a powerful word processor that is fun to use, very user friendly..... The Enviornment compares favorably to that of Microsoft Windows " - RAINBOW Dec. 1989

Window Writer/W- for non W/M owners.....$79.95
Window Basic Compiler-similar to CBASIC...$99.00
Window Edt/Asm- A full featured Assembler...$49.95
Font/Icon Editors- Basic & M.L. versions..... $19.95
Advanced Programmers Guide- $24.95
The Memory game- Concentration game $19.95
Desk Accessory Pak-7 resident programs $39.95

Deluxe Terminal

A Completely New and Easy to use Terminal Program designed specifically for the Coco III.

With advanced features you would expect to find only in a Hi-Priced MS-Dos program. It has a 26 Entry Phone Directory with complete Configuration information for Communications and Automatic logon. Supports the Serial I/O Port up to 2400 baud, Deluxe & Modem Pak and the Disto Serial I/O board up to 9600 baud. It has a Full Screen Text Editor, X/Y Modem File transfer support, Split Screen Conference Mode, Macro Keys. Full Disk Support including Multifile Copy, Kill, Rename, Arc/Un-Arc and Disk Initialize and it is Completely Compatible with ADOS.

Requires 128K, Disk & 80 Col. Display $44.95

CoCo III Tool Kit

Disk Commands
Backup, Initialize, Directory, Verify, Compare, Search, Edit, Erase, Speed Test, Step Rate Test, Gran Table Analysis & Repair

File Commands
Arcive, Copy, Kill, Rename, Erase, View, Edit, Print, Compare, Salvage, Search, Verify, Test Arcive, Un-Arcive, Xmodem Send/Rec.

The Disk Utility Program that you always dreamed of is now a Reality.

Coco Tools is a comprehensive disk utility Program providing the most complete set of functions available for the standard R.S. DOS disk system. Comparable in scope and functionality to that of the famous utility available for MS-DOS computers "PC-TOOLS"!

Coco Tools is also the most Comprehensive Disk Repair program available, it Automatically diagnosis and repairs file allocation errors, locates corrupt directory information and cross linked files. It provides fast and easy recovery of deleted files without the drudgery normally associated with killed file recovery.

Requires 128K, Disk & 80 col. display $49.95

This Months Specials

Savings Up To 35%

EDT/ASM III $49.95
Window Master.....$49.95
Window Master & Window Writer.....$79.95
Deluxe Terminal.....$34.95

VISA, MASTERCARD or COD

Call Toll Free 1-800-383-8529

(Monday thru Saturday, 8am to 5pm PST).

To order by mail, send check or money order for the amount of the program plus $4.00 for shipping to:

Cer-Comp Ltd.
5566 Ricochet Avenue
Las Vegas, NV 89110
702-452-0632

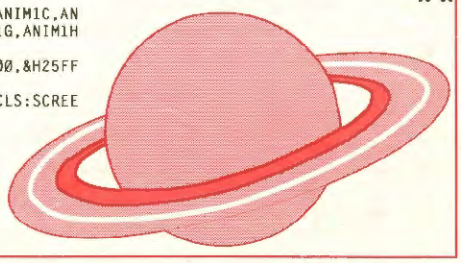
```
FOUR FRAMES
00410 PAGE1C  FRAME   4800
00420 PAGE2A  FRAME   5000
00430 PAGE2B  FRAME   5800
00440 PAGE2C  FRAME   6000
00450 PAGE3A  FRAME   6800
00460 PAGE3B  FRAME   7000
00470 PAGE3C  FRAME   7800
00480 PAGE4A  FRAME   8000
00490 PAGE4B  FRAME   8800
00500 PAGE4C  FRAME   9000
00510 PAGE5A  FRAME   9800
00520 PAGE5B  FRAME   A000
00530 PAGE5C  FRAME   A800
00540 PAGE6A  FRAME   B000
00550 PAGE6B  FRAME   B800
00560 PAGE6C  FRAME   C000
00570 PAGE7A  FRAME   C800
00580 PAGE7B  FRAME   D000
00590 PAGE7C  FRAME   D800
00600 PAGE8A  FRAME   E000
00610 PAGE8B  FRAME   E800
00620 PAGE8C  FRAME   F000
00630         CLR     $FFDE
00640         ANDCC   #$AF
00650         RTS
00660
00670 SHOWIT  LDX     #$1208
CENTER THE DISPLAY
00680         LDA     #64     #
ROWS DOWN
00690         STA     DOWN
00700 L2      LDB     #8      #
BYTES ACROSS
00710         STB     ACROSS
00720 L3      CLRA
00730         LDB     ,U+     GET
GRAPHICS BYTE
00740 R1      ASLB
CHECK BIT 7
00750         BCC     R2
BRANCH IF ZERO
00760         ORA     #192    SET
BITS 7,6
00770 R2      ASLB
CHECK BIT 6
00780         BCC     R3
BRANCH IF ZERO
00790         ORA     #48     SET
BITS 5,4
00800 R3      ASLB
00810         BCC     R4
00820         ORA     #12
00830 R4      ASLB
00840         BCC     R5
00850         ORA     #3
00860 R5      STA     N1      SAVE
FIRST HALF
00870         CLRA
READY FOR BITS 3-0
00880         ASLB
CHECK BIT 3
00890         BCC     R6
BRANCH IF ZERO
00900         ORA     #192    SET
BITS 7,6
00910 R6      ASLB
00920         BCC     R7
00930         ORA     #48
00940 R7      ASLB
00950         BCC     R8
00960         ORA     #12
00970 R8      ASLB
00980         BCC     CONT
00990         ORA     #3
01000 CONT    STA     N2      SAVE
SECOND HALF
01010         LDD     N1      PICK
UP DOUBLED VALUE
01020         STD     32,X    AND
SHOW
01030         STD     ,X++    IT
TWICE
01040         DEC     ACROSS
01050         BNE     L3
01060         LEAU    24,U
START OF NEXT ROW
01070         LEAX    48,X    SKIP
1 1/2 LINES
01080         DEC     DOWN
01090         BNE     L2
01100 DELAY   LDY     #$2000
ADJUSTABLE DELAY
01110 D1      LEAY    -1,Y
01120         BNE     D1
01130         RTS
01140 ACROSS  RMB     1
01150 DOWN    RMB     1
01160 N1      RMB     1
01170 N2      RMB     1
01180         END     LOAD
```

Listing 2: ANIMDRAW

```
1 'ANIMATION DRAWER
2 'BY WILLIAM P. NEE
3 'COPYRIGHT (C) 1992
4 'FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 FOR N=1 TO 8:READ TI$(N):NEXT
20 PMODE4,1:COLOR0,5:PCLS:SCREEN
1,1
30 R1=12:R2=14:R3=16:PI=4*ATN(1)
40 HH=32:VV=32
50 FOR N=0 TO 95
60 A=N*3.75:A=A*ATN(1)/45
70 A1=3*A:A2=2*A:A3=A
80 IF N=12 THEN 280
90 IF N=24 THEN 280
100 IF N=36 THEN 280
110 IF N=48 THEN 280
120 IF N=60 THEN 280
130 IF N=72 THEN 280
140 IF N=84 THEN 280
150 X1=INT(HH+R1*COS(A1+PI/6))
160 Y1=INT(VV-R1*SIN(A1))
170 X2=INT(HH+R2*COS(A2+PI/6))
180 Y2=INT(VV-R2*SIN(A2))
190 X3=INT(HH+R3*COS(A3+PI/6))
200 Y3=INT(VV-R3*SIN(A3))
210 CIRCLE(HH,VV),8
220 PSET(X1,Y1):PSET(X2,Y2):PSET
(X3,Y3)
230 HH=HH+64:IF HH>224 THEN HH=3
2:VV=VV+64:IF VV>160 THEN VV=32
240 NEXT N:VERIFYON
250 SAVEM TI$(8),&HE00,&H25FF,&H
E00
260 GOTO 260
270 DATA ANIM1A,ANIM1B,ANIM1C,AN
IM1D,ANIM1E,ANIM1F,ANIM1G,ANIM1H
280 VERIFYON
290 SAVEM TI$(N/12),&HE00,&H25FF
,&HE00
300 PMODE4,1:COLOR0,5:PCLS:SCREE
N1,1:GOTO150
```

Listing 3: ANIMSHOW

```
1 'ANIMATION SHOW
2 'BY WILLIAM P. NEE
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 CLEAR 200,&H3500-1
20 IF PEEK(&H3500)<>142 THEN LOA
DM"ANIM":POKE&HFF40,0
30 PMODE4,1:COLOR0,5:PCLS:SCREEN
1,1
40 READ FI$,SP:POKE &H37C4,SP
50 POKE &H3504,&H38
60 LOADM FI$+"A":EXEC &H3500
70 POKE &H3504,&H50
80 LOADM FI$+"B":EXEC &H3500
90 POKE &H3504,&H68
100 LOADM FI$+"C":EXEC &H3500
110 POKE &H3504,&H80
120 LOADM FI$+"D":EXEC &H3500
130 POKE &H3504,&H98
140 LOADM FI$+"E":EXEC &H3500
150 POKE &H3504,&HB0
160 LOADM FI$+"F":EXEC &H3500
170 POKE &H3504,&HC8
180 LOADM FI$+"G":EXEC &H3500
190 POKE &H3504,&HE0
200 LOADM FI$+"H":EXEC &H3500
210 POKE &HFF40,0
220 PMODE4,1:COLOR0,5:PCLS:SCREE
N1,1
230 EXEC &H3520
240 GOTO 230
250 DATA ANIM1,&H10
260 'DATA ANIM2,&H20
```



Feature Program

CoCo Finds the Right Location

Centering a phrase or title onscreen while writing programs can be a real chore. It isn't all that difficult but involves enough trial and error that it becomes a laborious task. That's why I wrote *Print At*.

Print At is a simple BASIC program that helps you place a short line of text just about anywhere on the screen. After running the program, you are prompted to enter the phrase; type up to 17 characters (to allow longer lines, change IFL>17 in Line 90 to a greater value) and press ENTER. Then use the arrow keys to move the text line around the screen. As you move the text, the actual PRINT@ location is displayed at the bottom of the screen. You can move the phrase anywhere from the top line down to two rows from the bottom.

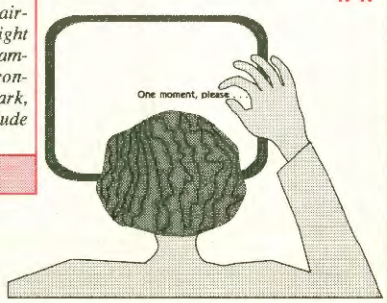
When the text is positioned to your satisfaction and you have noted the proper location, press BREAK. Then you can use the location with the PRINT@ statement in your BASIC program. It's amazing how such a simple program can be so handy.

John Musumeci is a retired TV repairman whose sole hobby for the past eight years has been working with and programming the Color Computer. He may be contacted at 103-57 104 Street, Ozone Park, NY 11417, (718) 738-0212. Please include an SASE when requesting a reply.

16K Extended

The Listing: PRINTAT

```
1 'PRINT AT
2 'BY JOHN MUSUMECI
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 CLS:PRINT@107,"**PRINT@**"
20 FORX=1TO1000:NEXTX
30 PRINT@270,"BY":FORX=1TO600:NE
XTX
40 PRINT@329,"JOHN MUSUMECI":FOR
X=1TO2000:NEXTX
50 CLS:C=0:R=0:AT=0
60 SOUND225,1:PRINT"ENTER PHRASE
 (17 CHAR. MAX.)"
70 INPUTA$:L=LEN(A$)
80 CLS:PRINT@AT,A$
90 IFL>17THENPRINT:PRINT"PHRASE
T-O-O L-AR-GE":SOUND75,10:FORX=1
TO600:NEXTX:GOTO50
100 FORX=1TO600:NEXTX:PRINT@106,
"USE ARROWS":SOUND225,1
110 B$=INKEY$:IF B$=""THEN110
120 IF ASC(B$)=9 THEN C=C+1:B=C+
L:IF B>31 THEN C=32-L
130 IF ASC(B$)=8 THEN C=C-1:IF C
<0 THEN C=0
140 IF ASC(B$)=94 THEN R=R-1:IF
R<0 THEN R=0
150 IF ASC(B$)=10 THEN R=R+1:IF
R>13 THEN R=13
160 CLS
170 AT=32*R+C:PRINT@AT,A$
180 PRINT@481,"PRINT@";AT;
190 GOTO110
```



Feature Program

Protect Your Parcels With Care Labels

MAGNETIC MEDIA: DO NOT X-RAY

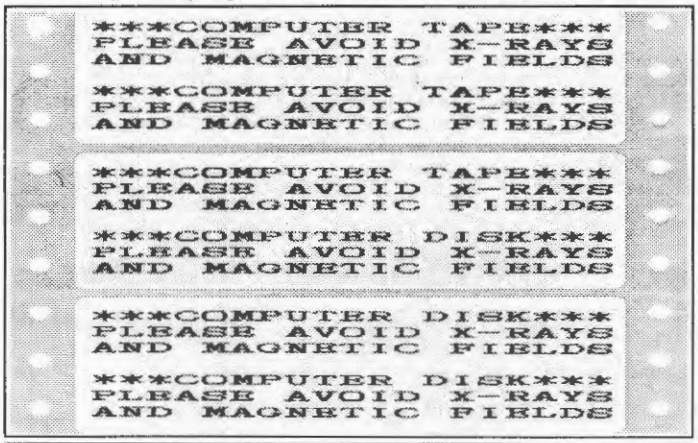
When you send a tape or disk through the mail, good packing techniques are essential. It can also help to let postal employees know that what you are sending is somewhat delicate. You can do this by putting warning labels on your packages.

Post Care is designed for use with a CoCo and a DMP-130 printer, and prints warning labels for you. The program supports one-up labels that are 4 inches wide by 1½ inches tall and prints two copies of the warning on each label. Enter POSTCARE from the listing and save it to tape or disk. Line 10 sets the computer's sending speed to 2400 bps. Alter this poke to match your setup. The control codes used are supported by most Tandy printers, though you may need to change them if your printer doesn't support the same features as the DMP-130 (or if your printer is not a Tandy). The control codes are commented in the listing.

When you run the program, you are prompted for whether you want a label for a disk (press D) or a tape (press T). To end the program, press E. The label is then printed and the prompt returns, allowing you to print a bunch of labels at one time.

Only once did I forget to attach a label to a tape I sent from Australia to the U.S. That package went through an electronic sorting machine in San Francisco and was thoroughly destroyed (I don't know how the machine fared). Since labels aren't the end-all for protection, also exercise care when packing your tapes or disks — place stiff cardboard on both sides of a disk, and pack tapes in sponge rubber or corrugated cardboard. [*Editor's Note: We've noticed some people also wrap aluminum foil around disks and tapes before packing them. Based on the number of disks and tapes we receive daily, our experience shows this has little effect on whether or not the materials arrive intact. Consider saving your pennies, folks.*]

Keiran Kenny's interests lie mainly with the Color Computer's graphics and math capabilities. But in his own words, "I like to try everything." He may be contacted at 11/5 Milson Road, Cremorne, NSW, Australia 2090.



16K ECB

The Listing: POSTCARE

```
1 'POST OFFICE TAKE CARE!
2 'BY KEIRAN KENNY
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 POKE150,18 '2400 baud
20 PRINT#-2,CHR$(27);CHR$(18);CH
R$(27);CHR$(14);CHR$(27);CHR$(31
); 'DMP-130 NLQ, Elongated,
    Bold
30 CLS:PRINT@224,"(D)ISK OR (T)A
PE MAILER OR (E)ND"
40 K$=INKEY$:IFK$<>"D"ANDK$<>"T"
ANDK$<>"E"THEN40
50 IFK$="E"THEN120
60 IFK$="T"THEN90
70 PRINT#-2:PRINT#-2,"***COMPUTE
R DISK***":PRINT#-2,"PLEASE AVOI
D X-RAYS":PRINT#-2,"AND MAGNETIC
 FIELDS"
80 GOTO100
90 PRINT#-2:PRINT#-2,"***COMPUTE
R TAPE***":PRINT#-2,"PLEASE AVOI
D X-RAYS":PRINT#-2,"AND MAGNETIC
 FIELDS"
100 NR=NR+1:IFNR/2=INT(NR/2)THEN
PRINT#-2
110 GOTO30
120 PRINT#-2,CHR$(27);CHR$(15);C
HR$(27);CHR$(32); 'End Elongated
, Bold
```

Received and Certified

The following products have recently been received by THE RAINBOW, examined by our staff and issued the Rainbow Seal of Certification, your assurance that we have seen the product and have ascertained that it is what it purports to be.

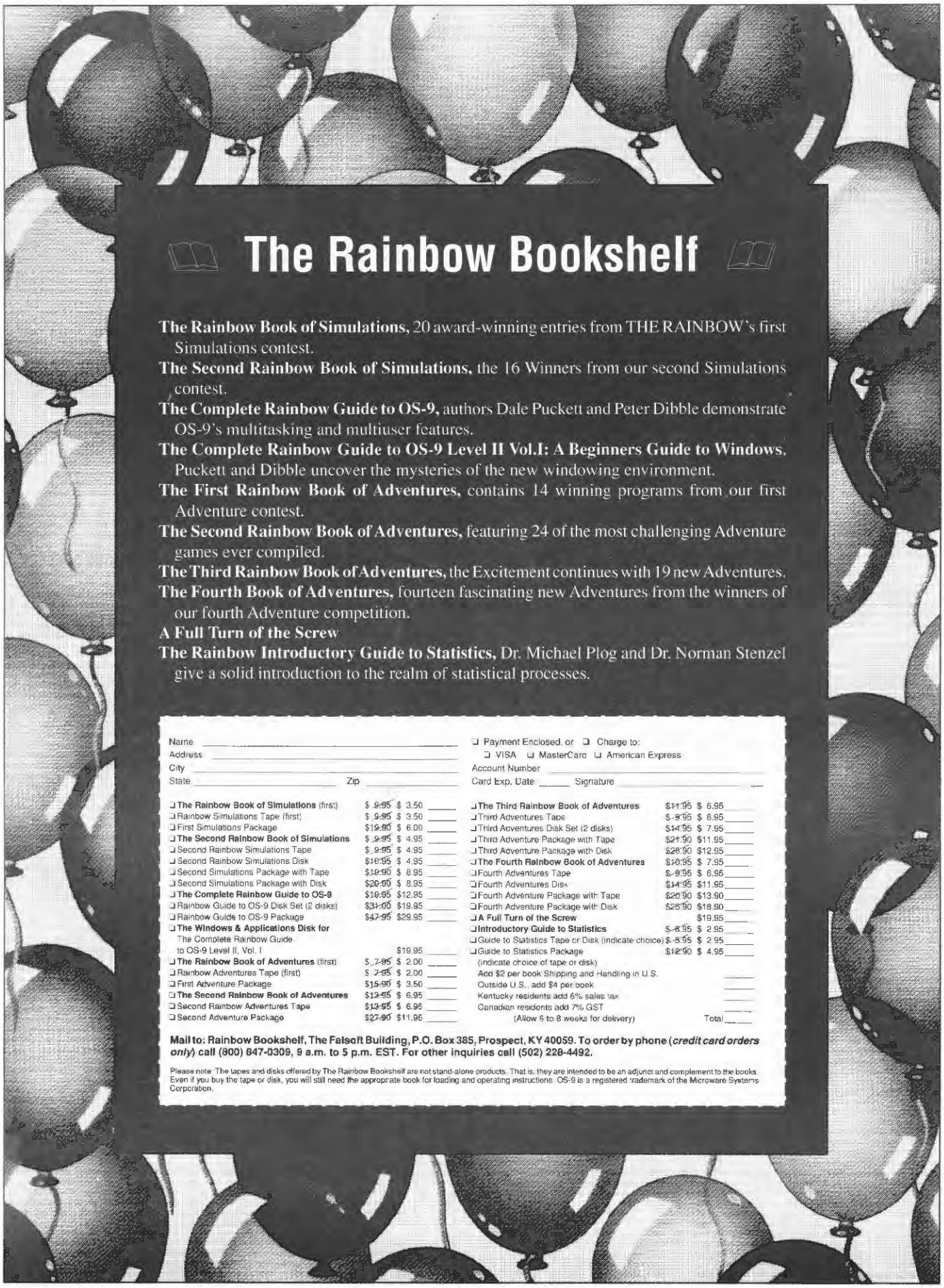
CCTools, a shareware windowing environment for the CoCo 3 and OS-9 Level II. This environment integrates File, Disk and Utility Management with application launching capabilities. Requires a 512K CoCo 3, OS-9 Level II, *Shell+* and at least 1 Meg of free disk storage. *Micro 80 Users Group, 598 Riverton Ave., Winnipeg, MB R2L 0P1, Canada; $15 U.S. or $18 Cdn. registration fee, money orders only.*

CF83-5: Assembler Extension Word Set, an extended set of mnemonics for the CF83 assembler. Requires CF83 Forth and the Standard Required Words Set. *BDS Software, P.O. Box 485, Glenview, IL 60025-0485, (708) 998-1656; $22 with printed manual, $15 with manual on disk; Canadian orders add $3, all other foreign orders add $10; all funds U.S.*

CF83-8: Uncontrolled Reference Words Plus, a portion of the Uncontrolled Reference Words Set from the Forth-83 Standard. Also includes a complete Case Structure package, a complete String Operations package, and a Printer Control package. Requires CF83 Forth and the Standard Required Words Set. *BDS Software, P.O. Box 485, Glenview, IL 60025-0485, (708) 998-1656; $20 with manual, $10 with manual on disk; Canadian orders add $3, all other foreign orders add $10; all funds U.S.*

Icon-Basic09 V 1.9a, the latest version of *Icon-Basic09*, a graphic user interface for BASIC09 under OS-9 Level II. (See Received & Certified listing in the June 1992 issue, Page 22.) *HAWKSoft, P.O. Box 7112, Elgin, IL 60121-7112, (708) 742-3084; $20.*

The Rainbow Seal of Certification is open to all manufacturers of products applicable to the Tandy Color Computer, regardless of whether or not those companies advertise in THE RAINBOW. By awarding the Seal, we certify the product exists — we have a sample copy and have examined it. However, this does not constitute any guarantee of satisfaction. As soon as possible, these products will be forwarded to reviewers for evaluation.



The Rainbow Bookshelf

The Rainbow Book of Simulations, 20 award-winning entries from THE RAINBOW's first Simulations contest.

The Second Rainbow Book of Simulations, the 16 Winners from our second Simulations contest.

The Complete Rainbow Guide to OS-9, authors Dale Puckett and Peter Dibble demonstrate OS-9's multitasking and multiuser features.

The Complete Rainbow Guide to OS-9 Level II Vol.I: A Beginners Guide to Windows. Puckett and Dibble uncover the mysteries of the new windowing environment.

The First Rainbow Book of Adventures, contains 14 winning programs from our first Adventure contest.

The Second Rainbow Book of Adventures, featuring 24 of the most challenging Adventure games ever compiled.

The Third Rainbow Book of Adventures, the Excitement continues with 19 new Adventures.

The Fourth Book of Adventures, fourteen fascinating new Adventures from the winners of our fourth Adventure competition.

A Full Turn of the Screw

The Rainbow Introductory Guide to Statistics, Dr. Michael Plog and Dr. Norman Stenzel give a solid introduction to the realm of statistical processes.

Name _____

Address _____

City _____

State _____ Zip _____

❑ Payment Enclosed, or ❑ Charge to:
❑ VISA ❑ MasterCard ❑ American Express

Account Number _____

Card Exp. Date _____ Signature _____

❑ **The Rainbow Book of Simulations** (first)	~~$ 9.95~~	$ 3.50	_____
❑ Rainbow Simulations Tape (first)	~~$ 9.95~~	$ 3.50	_____
❑ First Simulations Package	~~$19.90~~	$ 6.00	_____
❑ **The Second Rainbow Book of Simulations**	~~$ 9.95~~	$ 4.95	_____
❑ Second Rainbow Simulations Tape	~~$ 9.95~~	$ 4.95	_____
❑ Second Rainbow Simulations Disk	~~$10.95~~	$ 4.95	_____
❑ Second Simulations Package with Tape	~~$19.90~~	$ 8.95	_____
❑ Second Simulations Package with Disk	~~$20.90~~	$ 8.95	_____
❑ **The Complete Rainbow Guide to OS-9**	~~$19.95~~	$12.95	_____
❑ Rainbow Guide to OS-9 Disk Set (2 disks)	~~$31.00~~	$19.95	_____
❑ Rainbow Guide to OS-9 Package	~~$47.95~~	$29.95	_____
❑ **The Windows & Applications Disk for** The Complete Rainbow Guide to OS-9 Level II, Vol. I		$19.95	_____
❑ **The Rainbow Book of Adventures** (first)	~~$ 7.95~~	$ 2.00	_____
❑ Rainbow Adventures Tape (first)	~~$ 7.95~~	$ 2.00	_____
❑ First Adventure Package	~~$15.90~~	$ 3.50	_____
❑ **The Second Rainbow Book of Adventures**	~~$13.95~~	$ 6.95	_____
❑ Second Rainbow Adventures Tape	~~$13.95~~	$ 6.95	_____
❑ Second Adventure Package	~~$27.90~~	$11.95	_____

❑ **The Third Rainbow Book of Adventures**	~~$11.95~~	$ 6.95	_____
❑ Third Adventures Tape	~~$ 9.95~~	$ 6.95	_____
❑ Third Adventures Disk Set (2 disks)	~~$14.95~~	$ 7.95	_____
❑ Third Adventure Package with Tape	~~$21.90~~	$11.95	_____
❑ Third Adventure Package with Disk	~~$28.90~~	$12.95	_____
❑ **The Fourth Rainbow Book of Adventures**	~~$10.95~~	$ 7.95	_____
❑ Fourth Adventures Tape	~~$ 9.95~~	$ 6.95	_____
❑ Fourth Adventures Disk	~~$14.95~~	$11.95	_____
❑ Fourth Adventure Package with Tape	~~$20.90~~	$13.90	_____
❑ Fourth Adventure Package with Disk	~~$25.90~~	$18.90	_____
❑ **A Full Turn of the Screw**		$19.95	_____
❑ **Introductory Guide to Statistics**	~~$ 8.95~~	$ 2.95	_____
❑ Guide to Statistics Tape or Disk (indicate choice)	~~$ 8.95~~	$ 2.95	_____
❑ Guide to Statistics Package (indicate choice of tape or disk)	~~$12.90~~	$ 4.95	_____
Add $2 per book Shipping and Handling in U.S.			_____
Outside U.S., add $4 per book			_____
Kentucky residents add 6% sales tax			_____
Canadian residents add 7% GST			_____
(Allow 6 to 8 weeks for delivery)		Total	_____

Mail to: Rainbow Bookshelf, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. To order by phone (*credit card orders only*) call (800) 847-0309, 9 a.m. to 5 p.m. EST. For other inquiries call (502) 228-4492.

Please note: The tapes and disks offered by The Rainbow Bookshelf are not stand-alone products. That is, they are intended to be an adjunct and complement to the books. Even if you buy the tape or disk, you will still need the appropriate book for loading and operating instructions. OS-9 is a registered trademark of the Microware Systems Corporation.

PHOTON

The critics will be raving about this strategy game! Based on an original concept by author Jeff Steidel, *Photon* is an addictive time-muncher in the spirit of *Lemmings*™ and *Tetris*™. Match wits with Ludevide, the evil power droid, as you reason your way through over 60 devious levels. The numerous original music scores, digitized speech and sound effects, and pleasing animation and graphics enrich *Photon* to make it an unparalleled gaming experience. Soon to be released on a variety of computer platforms, the CoCo Community is lucky enough to be given first glance at this fantastic game! Req. 128k CoCo-3, disk drive, and joystick.

$34.95

50% OFF!

Sundog Systems is *blowing out* selected back-stock software, and you can now take advantage of these bargains to complete your Sundog collection! For a limited time, you can buy some of the best CoCo games at *50% off retail price!!* Half off factory-new game software... you'll never find a better deal!

Paladin's Legacy	reg. $24.95	*now $12.45!*
Hall of the King 1, 2, or 3	reg. $29.95	*now $14.95!*
Kung-Fu Dude	reg. $24.95	*now $12.45!*
White Fire of Eternity	reg. $19.95	*now $9.95!*
Dragon Blade	reg. $19.95	*now $9.95!*
Champion	reg. $19.95	*now $9.95!*

Sundog Systems

PO Box 766 Manassas, VA 22111
(703) 330-8989

Visa, MC, check, MO, and COD (US only, please) accepted. All foreign orders must be sent in US currency MO's. Include $2.50 for shipping in USA & Canada, $5 foreign. $3 extra for COD orders. PA residents add 6% tax.

Advertisers Index

Call:
Kim Lewis
Advertising Representative

Call:
Belinda Kirby
Advertising Representative

The Falsoft Building
9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059
(502) 228-4492 • FAX (502) 228-5121

We appreciate your mentioning THE RAINBOW *when you contact these advertisers.*

Announcing

Icon Basic09

The next programming language for OS-9!

Icon Basic09 is a graphical user interface (GUI) to Basic09, which will make programming easier than ever! **Icon Basic09** takes an innovative approach by using graphic representations, or icons, to represent statements and keywords for writing Basic09 programs and procedures. Instead of constantly typing while writing a program, the user can simply **point & click** to choose the desired statement!

Icon Basic09 can also be very useful in studying procedures and programs written by others to learn how they operate. The package contains a full set of icons...or, you may edit or create icons using the included icon editor. Icon Basic09 requires a CoCo-3 with at least 256k, mouse or joystick, and OS-9 lv 2.

$20

Dual hi-res joystick adapter (RS/Colorware)	**$40**
Hi & Lo-res joystick adapter	**$27**
HAWKsoft keyboard extension cable	**$25**
Domination ("Risk"-like wargame!)	**$18**
MyDOS full-featured DOS extension	**$15**

Hawksoft

PO Box 7112
Elgin, IL 60121-7112
(708) 742-3084 eves & ends

US and CDN S&H always included. Terms: MO, check, or COD.

Burke & Burke

P.O. Box 733 Maple Valley, WA 98038
U.S. ORDER DESK: (800) 237-2409
INT'L & TECHNICAL: (206) 432-1814

See you at CoCo Fest '92!

RUN OS9 LEVEL 2 VISIBLY FASTER - GET A $29.95 POWERBOOST FROM BURKE & BURKE!

You can see the difference in OS9 Level 2 when you install the PowerBoost kit from Burke & Burke. It includes the amazing Hitachi HD63B09E microprocessor, a 100% compatible high-performance replacement for your CoCo 3' MC68B09E. Using the HD63B09E's added registers and high-speed instruction set, block moves and other functions are up to 4x as fast! The PowerBoost kit includes Burke & Burke's PowerBoost software, which modifies your OS9 Level 2 operating system for faster multi-tasking, graphics, and disk I/O when using the HD63B09E. SOLDERING IS REQUIRED TO INSTALL THE HD63B09E.

PowerBoost (HD63B09E w/ Software for OS9) *$29.95*

The 6309 Book by Chris Burke. This book describes the instruction set of the HD63B09E microprocessor, and gives assembly language examples that help you detect the high-performance processor and take advantage of its capabilities in your programs. Includes disk with OS9 Assembler / Disassembler / Debugger patches for HD63B09E. REQUIRED 'ASM' ASSEMBLER, 'DEBUG' DEBUGGER, AND HD63B09E MICROPROCESSOR NOT INCLUDED.

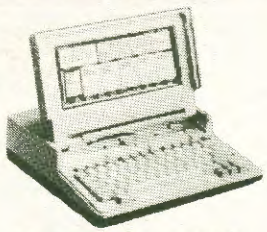
The 6309 Book (Includes disk with development tools for OS9) *$24.95*

WORLD CLASS CHESS -- Use Cyrus Chess w/ Level 2*	*$29.95*
FILE SYSTEM REPACK 1.1 -- Faster disk defragmenter	*$29.95*
*R. S. B. * -- Disk BASIC for Level 2 (BASIC ROM required).*	*$39.95*
EZGEN 1.09 -- Handy & powerful OS9 bootfile editor	*$19.95*
ZCLOCK - Continuous time / date display on Level 2 screen	*$9.95*
COCO XT -- Use PC hard drive w/ CoCo ! OS9 S/W incl.	*$69.95*
(Add $30 for Real-Time Clock. Add $20 for auto-boot ROM)	

WA RESIDENTS ADD 8.2% SALES TAX. MasterCard & VISA accepted. U.S. COD's add $3.75. Min. U.S. shipping $4.00. Min. to Canada $5.00. Please allow 2 weeks for delivery. Overnight or 2nd- day available for in-stock items. Software upgrades $5.00 each w/receipt, including U.S. shipping.
Call or write for a free catalog of more exciting Color Computer products!

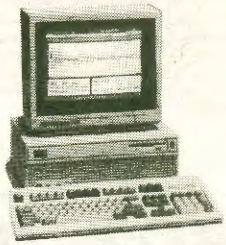
From Computer Plus to YOU . . .

PLUS *after* PLUS *after* PLUS

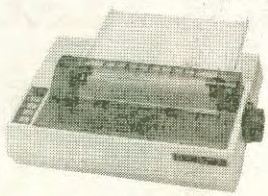


Tandy 1110 HD $929*
Tandy 1800 HD $929*
Tandy 3820 $2499

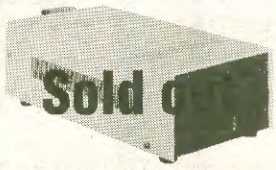
Tandy 2500 SX/25 HD $1099.00
Tandy 4825 SX/25 HD $1739



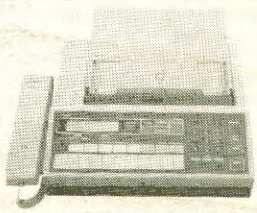
Tandy 4850 EP $2399



DMP-136 $199*



Color Computer Disk Drive
Drive 0 $239 Drive 1 $149



TandyFax 1500 $539*

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Tandy 1000 RLX HD with VGM-220	839.00
Tandy 1110 HD 1 Drive 640K	929.00 *
Tandy 1800 HD 1 Drive 1 Meg RAM	929.00 *
Tandy 1000 RL HD with CM11	569.00 *
Tandy 1100 FD 1 Drive 640K	469.00 *

PRINTERS	
Tandy DMP-136 213 CPS	199.00 *
Tandy DMP-302 270 CPS	469.00
Tandy DMP-202 180 CPS	299.00
Tandy DMP-442 300 CPS	539.00
Tandy LP-950 Laser Printer	1299.00
Tandy DMP-240 192 CPS 8 color	399.00
Panasonic KXP 1180 192 CPS	189.00 *
Panasonic KXP 1123 24 Wirehead	259.00 *
Panasonic KXP 1124 i300 CPS	329.00
Okidata 320 300 CPS	369.00
Okidata 380 180 CPS 24 Wire HD	239.00
OKI Laser 400 4PPM	679.00 *

MODEMS	
Tandy DCM-6	52.00
Tandy DCM-7	85.00
Cardinal 1200 Baud External	99.00
Cardinal 2400 Baud External	129.00

COLOR COMPUTER MISC.	
Tandy Drive Controller	89.00 *
Extended Basic Rom Kit (28 pin)	19.95
64K Ram Upgrade Kit (2 or 8 chip)	39.00
Tandy Deluxe Keyboard Kit	24.95
HI-RES Joystick Interface	8.95
Color Computer Deluxe Mouse	44.00
Multi Pak Pal Chip for COCO 3	14.95
COCO 3 Service Manual	29.95
Serial to Parallel Converter	59.95
Tandy Deluxe Joystick	19.95
Magnavox 8135 RGB Monitor	299.00
Magnavox Green or Amber Monitor	99.00
CoCo 3 Gime Chip	29.95
Tandy Pistol Grip Joystick	26.95
PBJ OK COCO 3 Upgrade Board	29.95
PBJ 512K COCO 3 Upgrade	89.00
Tandy OK COCO 3 Upgrade Board	39.95
Tandy 512K COCO 3 Upgrade	99.00 *

COLOR COMPUTER SOFTWARE	TAPE	DISK
The Wild West (COCO 3)		25.95
Worlds of Flight	34.95	34.95
Mustang P-51 Flight Simul.	34.95	34.95
Flight 16 Flight Simul.	34.95	34.95
Tandy Educatioal Software		2.00
Spinnaker Software		2.00
Max 10 by Colorware		79.95
AutoTerm by PXE Computing	29.95	39.95
TW-80 by Spectrum (COCO 3)		39.95
TeleWriter 64	49.95	59.95
TeleWriter 128		79.95
Elite Word 80		79.95
Elite Calc 3.0		69.95
CoCo 3 512K Super Ram Disk		19.95
Home Publishing by Tandy (CoCo 3)		35.95
Sub Battle Sim. by Epyx (CoCo 3)		26.95
Thexder by Sierra (CoCo 3)		22.45
Kings Quest III by Sierra (CoCo 3)		31.45
Flight Sim. II by SubLogic (CoCo 3)		31.45
QS-9 Level II by Tandy		71.95
QS-9 Development System		89.95
Multi-View by Tandy		44.95
VIP Writer (disk only)		69.95
VIP Integrated Library (disk)		149.95

Prices are subject to change without notice. Please call for shipping charges. Prices in our retail store may be higher. Send for complete catalog.

***Sale prices through 8-10-92**

CALL TOLL FREE
1-800-343-8124

- LOWEST POSSIBLE PRICES
- BEST POSSIBLE WARRANTY
- KNOWLEDGEABLE SALES STAFF
- TIMELY DELIVERY
- SHOPPING CONVENIENCE

VISA AMERICAN EXPRESS MasterCard

computer plus

P. O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (508) 486 3193