a 40-column PET, but will work on 80-column CBMs and 22-column VICs, with minor editing of the output statements. If you do use this program, I hope your students will enjoy it as much as these workshop participants said they did.

```
150 N=0:N$="NOBODY":T=0
160 DIM N$(100),T(100)
170 PRINT "{CLEAR}          {REV}TIME GUESSING P
    ROGRAM{03 DOWN}"
180 PRINT "DO YOU WANT INSTRUCTIONS('Y' OR 'N'
    )?";
190 GET Q$:IF Q$="" THEN 190
200 IF LEFT$(Q$,1)="N" THEN 330
210 IF LEFT$(Q$,1)<>"Y" THEN 170
220 PRINT "{HOME}{04 DOWN}THE OBJECT OF THE GA
    ME IS TO SEE WHO CAN"
230 PRINT "{REV}BEST{OFF} GUESS A 10 SECOND IN
    TERVAL WITHOUT"
240 PRINT "GOING OVER THE 10 SECOND LIMIT."
250 PRINT"{DOWN}YOU WILL BE ASKED YOUR NAME FI
    RST."
260 PRINT "THEN PRESS ANY KEY TO START THE TIM
    ER."
270 PRINT "{DOWN}WHEN YOU THINK 10 SECONDS IS
    UP PRESS A"
280 PRINT "KEY AND YOUR TIME WILL BE SHOWN ON
    THE"
290 PRINT "LEFT. THE BEST TIME OF THIS SESSION
     IS"
300 PRINT "SHOWN ON THE RIGHT."
310 PRINT "{DOWN}WHEN YOU FINISH READING THESE
               INSTRUCTIONS PRESS ANY KEY
    ."
320 GET Q$:IF Q$="" THEN 320
330 PRINT "{CLEAR}          {REV}TIME GUESSING P
    ROGRAM"
340 N=N+1
350 PRINT "{03 DOWN}TYPE PLAYER'S FIRST NAME A
    ND RETURN         "
360 INPUT "  ?{03 LEFT}";N$(N):IF N$(N)="?"THE
    NPRINT:PRINT"NAME PLEASE":GOTO360
370 N$(N)=LEFT$(N$(N),9)
380 PRINT "WHEN READY START THE TIMER BY PRESS
    ING  ANY KEY."
390 GET Q$:IF Q$="" THEN 390
400 TI$="000000"
410 PRINT "PRESS ANY KEY WHEN 10 SECONDS IS UP
    ."
420 GET Q$:IF Q$="" THEN 420
430 T(N)=INT(TI/60*100)/100
440 IF T(N)>10 THEN 460
450 IF T(N)>T THEN T=T(N):N$=N$(N)
460 PRINT "{CLEAR}          {REV}TIME GUESSING P
    ROGRAM"
470 PRINT "{DOWN}LAST PLAYER","BEST PLAYER"
480 PRINT "{02 DOWN}";N$(N),T(N),N$,T
490 PRINT "{DOWN}AGAIN('Y' OR 'N')?"
500 GET Q$:IF Q$="" THEN 500
510 IF LEFT$(Q$,1)="Y" THEN 340
520 IF LEFT$(Q$,1)<>"N" THEN 500
530 PRINT"{DOWN}HOPE THIS WAS FUN. THANKS FOR
    PLAYING.  HERE IS A LIST OF THE PLAYE
    RS ";
540 PRINT "AND THEIR SCORES.":PRINT
550 IF N=1 THEN PRINT N$(1),T(1):END
560 FOR I=1 TO N-1 STEP 2
570 PRINT N$(I),T(I),N$(I+1),T(I+1)
580 NEXT I
590 IF I=N THEN PRINT N$(N),T(N)
```

©

EDUCATIONAL SOFTWARE

- PROGRAMS FOR **PET** AND **VIC** COMPUTERS.
- OVER 200 NEW SELECTIONS FOR GRADES K-12
- PROGRAMMED WITH MOTIVATIONAL GRAPHICS AND SOUND
- MONEY BACK GUARANTEE

Order Your Free Catalog and . . .

Expect more from your PET® and VIC®

®PET and ®VIC are registered trademarks of Commodore Business Machines.

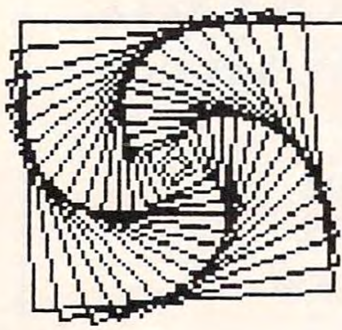
MICROGRAMS
INCORPORATED
P.O. BOX 2146, LOVES PARK, IL 61130
PHONE 815/965-2464

☐ Please send me a free K-12 PET catalog.
☐ Please send me a free K-12 VIC catalog.
☐ Please send me a sample program. Specify one. ☐ PET ☐ VIC
I have enclosed $3.00 for postage and handling.

NAME ______
ADDRESS ______
CITY ______
STATE ______ ZIP ______

www.commodore.ca

A Monthly Column



Friends Of The Turtle

David D. Thornburg
Associate Editor

LOGO Is Not Just Child's Play

Revolutionary periods are more than a time of change – they are often a time of great confusion as well. Those of us who are excited about the emergence of computer languages tailored to people's needs may be less sensitive than we should be to the way our message is being received.

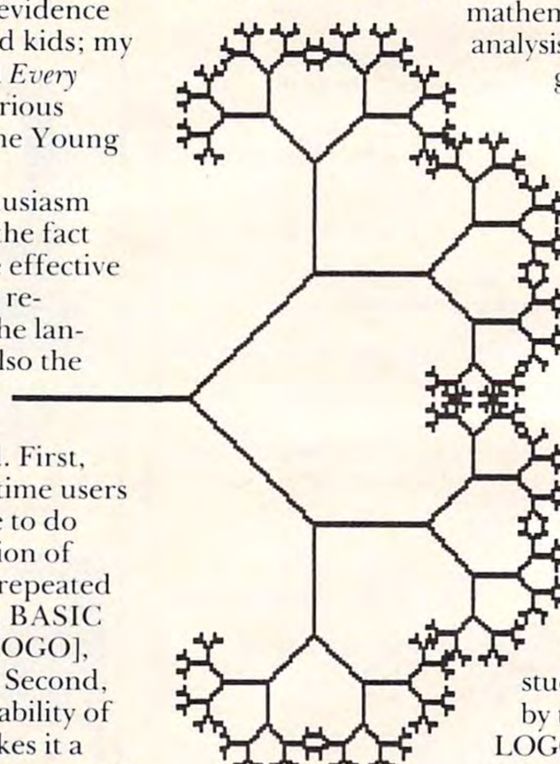
In the case of LOGO and Atari PILOT, this has had unfortunate consequences. Several readers have written to suggest that LOGO and Atari PILOT are "kid's" languages and are thus not worthy of serious attention. They cite as evidence Papert's *Mindstorms*, a book on LOGO and kids; my books, *Picture This!*, *Picture This Too!*, and *Every Kid's First Book of Robots and Computers;* various magazine articles; the very existence of the Young People's LOGO Association, etc.

Admittedly, much of the public enthusiasm for these languages has been devoted to the fact that, like English, PILOT and LOGO are effective communication tools for children. Let us remember, however, that while English is the language for "Baa, Baa, Black Sheep," it is also the language for James Joyce's *Ulysses* – the latter is definitely not for children.

The key to LOGO's power is twofold. First, much of it is very easy to learn, and first-time users find that within a short time they are able to do "interesting" things. (To me, the generation of logarithmic spirals is interesting, but the repeated printing of my name on the screen is not. BASIC has an easy time with the latter [as does LOGO], and has a horrible time with the former.) Second, LOGO is extensible by the user. This capability of LOGO, while of utility to youngsters, makes it a tremendous problem-solving language for users of any age. LOGO users readily develop skills in top-down programming and in the creation of building-block procedures that not only impart a logical order to programs, but also make them much easier to debug. When one adds to this such features as recursion, local variables, and list manipulation, it is obvious that LOGO is far more than just a kid's language. In fact, it is a far more useful language for many applications than many of the popular computer languages in use today.

Keep in mind that LOGO was a product of the artificial intelligence community. I assure you that something that is just a kid's language does not hold the interest of the MIT computer science department for over a decade.

Is turtle geometry easy to use? Of course it is. But, do LOGO's detractors know that finite differential geometry (turtle geometry's formal name) is a major tool for exploring some aspects of pure mathematics that have evaded analysis by traditional analytic geometry?

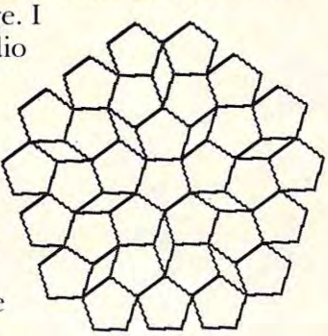


Those who think that LOGO is only for kids should read *Turtle Geometry* by Abelson and diSessa. If their treatment of relativity theory is too tame for your kids, try reading Buckminster Fuller's *Synergetics* (Fuller independently developed finite differential geometry and used it to make some very interesting discoveries). My Stanford graduate students get slowed down by that book, but perhaps LOGO's detractors will find it trivial reading. Fractal geometry – the subject of this column a few months back – lay virtually unexplored for more than 50 years because mathematicians lacked the tools to do the job.

This July it was my pleasure to give a lecture on the consequences of dimensionality on the conservation rules of geometry. Apple LOGO was my principal tool.

www.commodore.ca

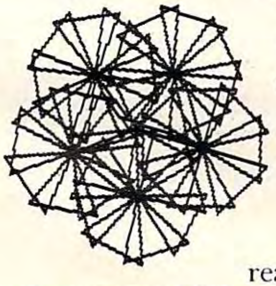
The reason I even care about this argument is that it has the promise of becoming a self-fulfilling prophecy. I have heard Apple dealers tell customers that LOGO is a kid's language. I have seen languages like Radio Shack Color LOGO that are excellent turtle graphics environments, but lack the list manipulation and other features that characterize a full LOGO implementation. In short, I have seen much confusion in the marketplace regarding LOGO and Atari PILOT.



So, please, know that languages like LOGO are marvelous tools for children – and that they are marvelous tools for almost everyone else as well. The power of a good tool is restricted only by the capabilities of its user. LOGO is a good tool.

Those Logophobes who feel like giving the language a second chance should read Harold Abelson's new book from Byte/McGraw Hill. The book is published in two editions, *Apple Logo* (for the LCSI LOGO sold by Apple), and *Logo for the Apple II* (for the MIT LOGO sold by Terrapin and Krell).

This book is excellent for all LOGO users simply because it is far more than a reference work. Abelson has managed to combine descriptions of LOGO primitives with projects that deepen the user's familiarity with the language. The first 60 pages are devoted to turtle graphics, and the remaining 150 concentrate on the other aspects of LOGO that make it a complete computer language. Thus, in addition to turtle graphics, readers become well versed in list manipulation, recursion, hierarchical structures, etc.



While it is fair to say that no prior experience in programming is required to read this book, those of you who are learning LOGO as a replacement of or supplement to another language will not find Abelson's book excessively wordy. The text follows several presentation styles: reference material, sample procedures, and projects for the user to solve on his or her own. Except for elementary grade school children, I can't think of any LOGO users who would not benefit from this book.

How To Grapple With A Turtle

In my last column I showed Apple LOGO users how to print screen images on the Silentype printer. The Silentype has many features (low cost, quiet operation, etc.), but it doesn't produce pictures with very high contrast. For high contrast one must consider using a dot matrix impact printer.

Because I need high quality screen images for various reasons, I invested in the Grappler printer interface card (from Orange Micro) for use with my Epson MX-100 printer. I have not done an exhaustive search of the printer interfaces for the Apple II, but I can't think of much I would want to do that can't be done with the Grappler.

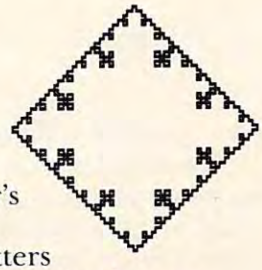
For example, this printer interface allows you to print a screen image at double size (rotated by 90 degrees) so it fits perfectly on an 8.5 by 11 sheet of paper. I enjoy the results of this print mode so much that I haven't explored any of the others.

To generate such prints for the Epson printer (there are Grapplers for other graphics printers as well), you should enter

```
SETSCRUNCH 0.84
```

before drawing any pictures. This compensates for the dot aspect ratio of the Epson printer. If you are using another printer (or another printing mode), you may have to experiment by drawing squares with various settings of SETSCRUNCH (or .ASPECT for those of you with MIT LOGO) until you get a picture that is perfectly square. The following procedure is all that is needed to generate a full-page image of your graphics screen. This procedure is written in Apple LOGO and assumes that the Grappler card is plugged in slot 1 of the Apple:

```
TO PRINTPICT
MAKE "CTRI CHAR 9
.PRINTER 1
PRINT WORD :CRI "GDR
PRINT CHAR 12
.PRINTER 0
END
```



This procedure gets the printer's attention with the character ctrl-I (CHAR 9), followed by letters that set the various options. G indicates that we want a graphics image, D means it should be double size, and R means it should be rotated by 90 degrees. If you want to use the enhanced print mode of the Epson printer, add an E to the list, and you will get a much denser print (with a longer print time, of course).

That's all there is to it! The accompanying figures are taken from my next book, tentatively titled *Discoveries of Beauty*. (This book should appear from Addison-Wesley about January 1983.) Most of the illustrations for this book were generated with the procedure shown above. As you can see, the Grappler lets your Epson printer do a fine job printing pictures generated with LOGO on your Apple computer. ©

www.commodore.ca

A Monthly Column

The World Inside The Computer

Fred D'Ignazio is a computer enthusiast and author of several books on computers for young people. He is presently working on two major projects: he is writing a series of books on how to create graphics-and-sound adventure games. He is also working on a computer mystery-and-adventure series for young people.
As the father of two young children, Fred has become concerned with introducing the computer to children as a wonderful tool rather than as a forbidding electronic device. His column appears monthly in **COMPUTE!**

This game will appeal to children of all ages. And it can teach both programming and subjects like English or history while it entertains.

The Story Game

Fred D'Ignazio, Associate Editor

Have you ever played MAD LIBS?®

MAD LIBS is an assortment of wacky party books designed to appeal to the six-year-old in all of us. Each book has a theme – monsters, movies, super heroes, current events, geography, mysteries, or whatever. A MAD LIBS book is a collection of stories, songs, and rhymes with key words left out. You select the words needed to complete the stories. But you must do it blind.

You play MAD LIBS by first picking a *reader*. The reader selects a MAD LIB from the book. Then he (or she) asks people for words to help fill the blanks in the MAD LIB. "Give me a plural noun," the reader might say. Or, "I want the name of a person in this room." The reader fills in the blanks, taking care to hide the story from everyone else in the room.

When all the blanks are filled, the reader stops asking questions and reads the completed story. Depending on people's moods and personalities, the completed MAD LIB might sound philosophical, ridiculous, funny, or shockingly raunchy. For example, an exchange between Mickey Mouse and Minnie Mouse might go as follows:

MINNIE: Mickey! Will you stop doing those __________ exercises.

MICKEY: Aw, shucks, Minnie. I have to keep my __________ in shape.

MINNIE: Well, watch out for the __________. Just look at the way your __________ hangs down over your __________.

To fill in the above blanks, the reader would ask people to volunteer an adjective, a plural noun, an adjective, and two singular nouns. Depending on people's answers, the above passage could end up anywhere from banal to cute, or from innocent to X-rated.

Computer MAD LIBS For Kids

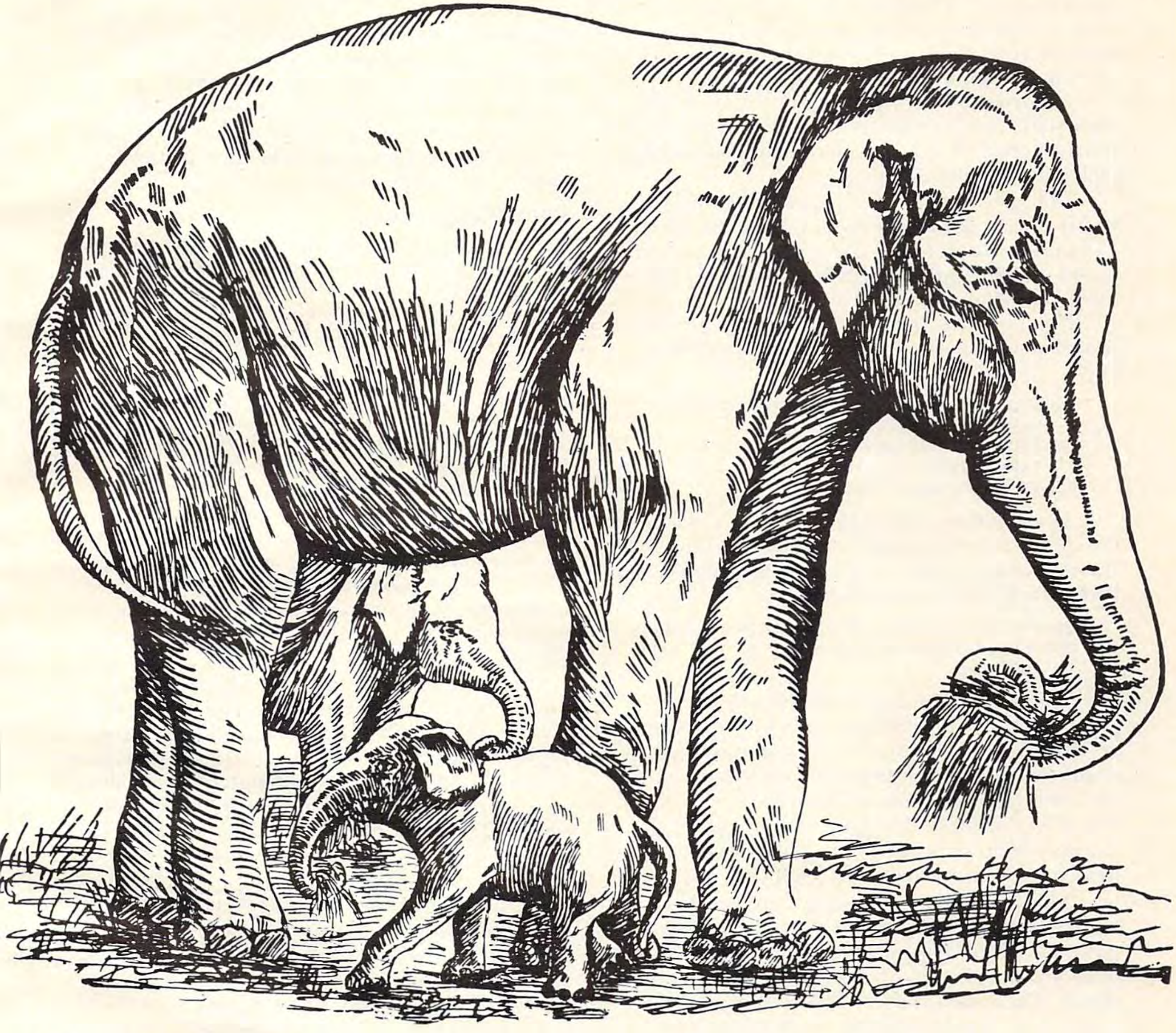
Japanese author Mitsumasa Anno has a book out called *Topsy-Turvies* (Weatherhill, 1970). Anno likes to play games with your vision and sense of perspective. The book is filled with colorful pictures of topsy-turvy buildings and people capable of walking up walls and strolling on ceilings. The book stretches the visual imagination in the same way as the bizarre paintings and drawings by M.C. Escher.

MAD LIBS games can do the same thing for kids' *verbal* and *literary* imaginations. And the "reader" who chooses the stories and asks for words can be a computer.

As a parent or teacher you can choose a MAD

www.commodore.ca

Micro-Ed is EXPANDING!



Send for free educational catalogs *

* Specify: Pet • Vic • TI • Apple • TRS-80 color computer
•The New Commodore 64

you can telephone us at
612-926-2292

Micro-Ed, Inc.
PO Box 24156 Minneapolis, MN 55424

www.commodore.ca

LIBS-type "skeleton" story from just about anywhere: a picture book, the Bible, a fairy tale, a comic book, a TV program, a song, a poem, or your own imagination. Or you can draw from yours or your child's personal experiences – sort of a fill-in-the-blank autobiography.

After you choose the source for a story, pick out a particularly vivid section of only about 25-300 words. You need to keep it short and fast-paced to maintain the child's interest.

Now go through the story and pick the key words you are going to leave out. Vary your choices. Try to take out different parts of speech: proper nouns (names), adjectives, adverbs, verbs, exclamations, plural nouns, and so on.

Next, type the story into your computer. The program will consist mostly of PRINT statements, like:

```
500 PRINT "FOUR SCORE AND SEVEN YEARS
    AGO OUR ";NOUN1$
510 PRINT "BROUGHT FORTH ON THIS
    CONTINENT"
520 PRINT "A NEW ";NOUN2$;"."
```

The variables NOUN1$ and NOUN2$ contain the child's answers to questions that the computer "reader" asked earlier. It asked the questions using PRINT and INPUT statements like:

```
200 PRINT "A PLURAL NOUN";INPUT NOUN1$
210 PRINT "A SINGULAR NOUN";INPUT
    NOUN2$
```

No matter what subject you choose, the computer stories are sure to fascinate children. They are likely to play the same story over and over, trying new words each time. And each time children try a new word, they immediately see its effect. The effect might be dramatic, zany, or silly. But it teaches children the different parts of speech and their roles in a sentence or in a story.

This also stretches children's imaginations and increases their confidence in using new words. After all, it's just a game, They can experiment with new words without being afraid of looking dumb. There won't be any all-knowing adults or smart aleck peers around to laugh at him if the words make the story crazy or absurd. Instead, it will be fun. And they can change the words on the program's next go around.

Dark Stories

We have a family tradition. Each night, my three-year-old and six-year-old take a bath and get into their pajamas. Then they tumble into bed, climb under the covers, and I turn out the lights. Then I tell them a "dark story." (It might be happy or sad, frightening or funny. It's a *dark* story because it can be told only in the dark.)

I make up a new dark story each night. A dark story is usually a heroic fantasy with lots of evil monsters, princes, princesses, spaceships, and adventures. The stories change, but two things remain the same. First, my daughter Catie is always the story's heroine, my son Eric is always the hero. Second, no matter where the stories end up – in a dismal dungeon or on a faraway planet – they always start someplace that is familiar to my children. That someplace might be their bedroom, their school, or their backyard.

Computer Fairy Tales

I tell a new dark story each night. I don't tell a new story just because I love to be creative. I do it because my memory is so bad. By the time bedtime arrives each night, even the previous night's dark story is usually nothing more than a faint smudge in my memory.

You and your family can create a new dark story each night, too, regardless of the state of your memories and imaginations. The storyteller can be your computer.

You can write programs that combine dark stories with our fill-in-the-blank program. What you get are fractured *fairy tales*. The kids can invent the new words to add to the fairy tales each night. If they are old enough, they can type them in themselves. And you can turn off the lights in the room where you keep the computer to make the fairy tales into true-blue dark stories.

At the end of this month's column I have a sample computer "story game" program for you to try. It takes up 4024 bytes and is written in Atari BASIC. It is a very simple, straightforward program that consists mostly of PRINT statements. It should be easy to modify to run on other popular computers. The only fanciness in the program is that it makes the stories appear in the enlarged Atari (graphics mode 2) character set.

Once you see how to create your own computer fairy tale, you can add to the program or change it completely. Right now, for example, the program asks only for nouns (proper names, places, things). You can add adjectives, verbs, nonsense words, etc. Also, the story is in a fairy tale format appropriate for short bedtime dark stories. But it needn't be. You can rewrite the story to be about anything. Whatever appeals to you and your kids.

And if you are a teacher, not a parent, you can use the story idea in your classroom. You can make up a story-writing assignment that combines programming, language arts, and history or social studies. The subject of the story is up to you.

The Story Game Unraveled

Lines 50-120: Program documentation (REM statements) and a data section. The child's answers are stored in variables ten characters long to

www.commodore.ca

Discover how easy it is for you to get useful results from your computer.

Dealer inquiries invited.

For PET/CBM
Understanding Your PET/CBM $16.95
Vol 1: Basic Programming
PET Graphics .... $ 6.95
For OSI C1P/C4P
Understanding Your C1P/C4P .... $ 9.95
A Workbook of BASIC Exercises
For VIC
Understanding Your VIC $13.95
Vol. 1: Basic Programming

Money Back Guarantee. VISA/MC accepted.
All prices include UPS or first class postage.

TIS INC
Total Information Services, Inc.
Box 921, Dept. C
Los Alamos, NM 87544

Scotch® Diskettes

Rely on Scotch® diskettes to keep your valuable data safe. Dependable Scotch diskettes are tested and guaranteed error-free. The low abrasivity saves your read/write heads. They're compatible with most diskette drives.

(800)235-4137

VISA

Dealer Inquiries Invited

PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. In Cal. call
(800) 592-5935 or
(805) 543-1037

SEI™ EDUCATIONAL SOFTWARE

For the Apple II*

Study Aids For College Board Exams (eg. PSAT, SAT or GRE)

Math Skills Pak - Algebra, Geometry & Trigonometry with Graphics
2 Diskettes - $50

Verbal Skills Pak - Vocabulary, Word Analogy, and Sentence Completion.
3 Diskettes - $60

Write For Complete Catalog
SEI
P.O. Box 7266-P
Hampton, VA 23666
(804) 826-3777

*Apple is a registered trademark of Apple Computer, Inc.

BATTERIES INCLUDED

village by the grange, 71 mccaul st. (f6) toronto m5t 2x1 telephone 596-1405

ARBITER 1.4 MULTI-USER DISK SYSTEM FOR COMMODORE 4.0 COMPUTERS

OVER THREE HUNDRED IN USE ACROSS ONTARIO

Since September 1981 **BATTERIES INCLUDED** has been installing the ARBITER system in classrooms of Commodore BASIC 4.0 computers. The computers are connected to CBM Disk Drives and printers. All users have access to all disk drives and printers plus a host of commands to make this system configuration really usable!

THE ARBITER 1.4 SYSTEM IS READY TO GO!

FEATURES

1) Easy installation.
2) Uses no RAM or Utility Sockets.
3) Up to 32 computers in one system.
4) System self initializes on power up.
5) Operation is completely transparent to the user.
6) Extended commands allow a friendly multi-user environment.
7) System design virtually eliminates interleaved printer output.

$150.00

SPECIAL COMMANDS

@S – Allows students to protect files with a five character password. A three character user ID is forced into the file name.

@L – Allows the students to load protected files if the password code is known.

LISTC – Used to produce program listings with a Commodore printer. Clumsy OPEN, CMD, LIST, PRINT#, CLOSE sequence not needed. It overcomes the listing problems found on other multi-user hardware systems.

LISTP – Used to get program listings on systems which have an ASCII printer. The cursor control characters are expanded and displayed in brackets. e.g. <home>

ALL FILE TYPES ARE SUPPORTED – During relative or sequential file access a delay has been built in so the computer will retain control of the system until the file is closed.

TEACHER UTILITY – A utility is supplied on disk to allow the teacher to produce a hardcopy listing and output from any of the protected or unprotected files selected. Once the files are chosen from the disk directory the teacher may do other tasks while the job is completed.

IF YOUR CLASSROOM WAS DESIGNED TO TEACH COMPUTER LITERACY OR STRUCTURED BASIC THEN THIS SYSTEM WAS DESIGNED FOR YOU.

Arbiter and Arbiter 1.4 are copyrights of Batteries Included.

www.commodore.ca

accommodate normal-sized words. The words could be longer, but you have to keep the size of your screen in mind, or you get word wraparound (the tail end of the word gets printed on the next line).

Lines 500-940: On line 535, the program clears the screen. On lines 540-560, the program prints the *game title*. There is a delay loop on 560 and in many other places throughout the story to slow the story down to the reading level of the child. You need to adjust these loops up or down to fit your kid's reading level.

On lines 800-940, the program asks the child for words to complete the story. My kids almost always put themselves in as the story's heroes. Their other answers are usually a surprise. Sometimes they are a shock! (Watching the child fill in story parts can be a real learning experience for the alert parent or teacher.)

Lines 950-1390: Lines 950 to 1008 print out the story title (taken from the child's answers). The title is partly centered and displayed in a special color (blue).

On lines 1009 to 1350 the computer tells the story. The lines are double-spaced and designed to accommodate the child's answers so they fit on the screen. The story stretches across several screens. Each screen is fairly full without being crowded with words. You can think of each screen as a "page" in a storybook.

Lines 1365 to 1390 print "THE END" (a key story ingredient) in a special color (blue, again).

Lines 1395-1480: Lines 1395 to 1430 enable the child to see the same story again (over and over and over!). Or the child can go back to the beginning of the program and invent a whole new story.

When the child is tired of making up stories, he reaches lines 1435 to 1480. The program says good-bye to the child, then clears the screen one last time and closes up shop. The POKE command on line 1445 makes the Atari screen cursor turn invisible for the computer's "good-bye" message. The POKE command on line 1475 makes the cursor reappear.

Next Month

Next month I'll show you how to teach the computer friend introduced in this column last month how to play games. The sample game will be the "Story Game" program you see below. You will be able to add up to 50 games to the friend's repertoire.

MAD LIBS® is a registered trademark of Price/Stern/Sloan Publishers, Inc. Price/Stern/Sloan is located at 410 North La Cienega Boulevard, Los Angeles, CA 90048.

Story Game Program

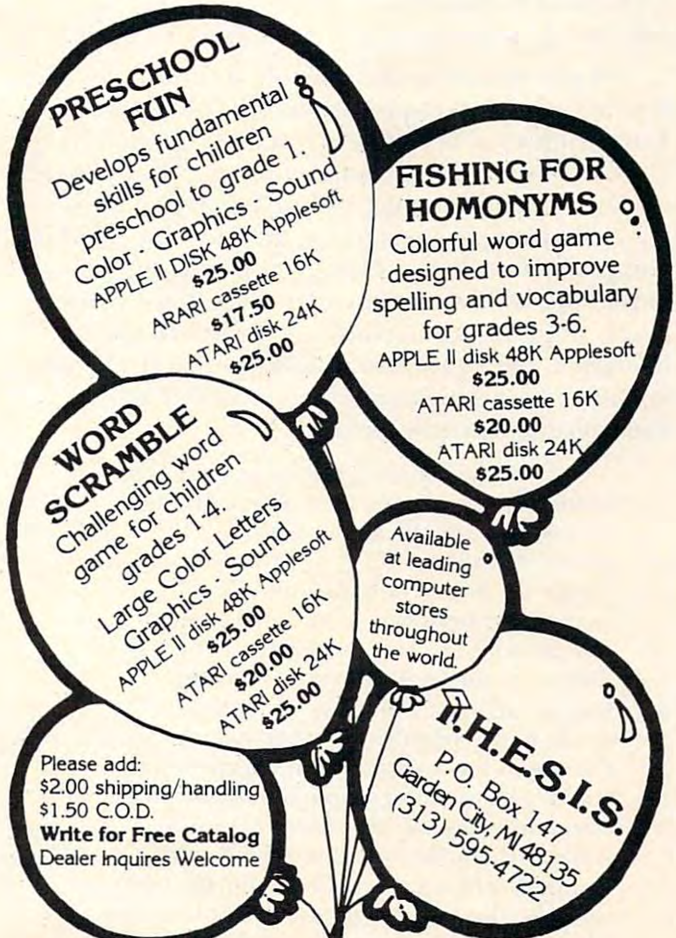
```
50 REM ***************
55 REM THE STORY GAME
60 REM ***************
65 REM ***
70 REM *** PROGRAM HELPS
75 REM *** CHILD AND PARENT
80 REM *** INVENT THEIR OWN
85 REM *** FAIRY TALE.
90 REM ***
95 REM *** DATA SECTION
96 REM ***
110 DIM N1$(10),N2$(10),N3$(10),N4$(1
   0),N5$(10),N6$(10)
120 DIM ANSWER$(1)
500 REM ***
510 REM *** PROGRAM INTRODUCES
520 REM *** ITSELF
530 REM ***
535 PRINT "{CLEAR}"
540 SETCOLOR 0,14,10
550 POSITION 8,4:PRINT "*** THE STORY
    GAME ***":PRINT :PRINT
560 FOR I=1 TO 500:NEXT I
800 REM ***
810 REM *** PROGRAM ASKS FOR
812 REM *** KEY INGREDIENTS
814 REM *** OF STORY
815 REM ***
840 POSITION 8,7:PRINT "NAME
    ";:INPUT N1$
870 POSITION 8,9:PRINT "ENCHANTED PLA
    CE";:INPUT N2$
890 POSITION 8,11:PRINT "VILLAIN";:IN
    PUT N3$
910 POSITION 8,13:PRINT "BAD PLACE";:
    INPUT N4$
930 POSITION 8,15:PRINT "MAGIC THING"
    ;:INPUT N5$
940 POSITION 8,17:PRINT "SILLY CREATU
    RE";:INPUT N6$
950 REM ***
960 REM *** PROGRAM TELLS STORY
970 REM ***
1000 GRAPHICS 2+16
1001 SETCOLOR 0,6,10
1003 POSITION 6,3:PRINT #6;N1$;" AND"
1004 POSITION 6,5:PRINT #6;"THE MAGIC
     "
1005 POSITION 6,7:PRINT #6;N5$
1008 FOR I=1 TO 1500:NEXT I
1009 GRAPHICS 2+16
1010 PRINT #6;" ONCE UPON A TIME,"
1015 PRINT #6;" "
1020 PRINT #6;" A BRAVE CHILD"
1025 PRINT #6;" "
1030 PRINT #6;" NAMED ";N1$
1035 PRINT #6;" "
1040 PRINT #6;" WENT EXPLORING"
1045 PRINT #6;" "
1050 PRINT #6;" IN AN ENCHANTED"
1052 PRINT #6;" "
1055 PRINT #6;" ";N2$;"."
1060 FOR I=1 TO 1500:NEXT I
1070 GRAPHICS 2+16
1080 PRINT #6;" IN THE ";N2$
1085 PRINT #6;" "
1090 PRINT #6;" LIVED A HUGE,"
1095 PRINT #6;" "
```

www.commodore.ca

```
1100 PRINT #6;" EVIL ";N3$
1105 PRINT #6;" "
1106 PRINT #6;" WHO LOVED"
1107 PRINT #6;" "
1110 PRINT #6;" TO EAT CHILDREN."
1115 FOR I=1 TO 1500:NEXT I
1116 GRAPHICS 2+16
1120 PRINT #6;" THE ";N3$
1125 PRINT #6;" "
1126 PRINT #6;" TRAPPED ";N1$
1127 PRINT #6;" "
1130 PRINT #6;" AND PUT"
1135 PRINT #6;" "
1140 PRINT #6;" ";N1$;" IN A DARK,"
1145 PRINT #6;" "
1150 PRINT #6;" STINKY ";N4$;"."
1155 FOR I=1 TO 1500:NEXT I
1156 GRAPHICS 2+16
1160 PRINT #6;" ";N1$;" SNEAKED"
1165 PRINT #6;" "
1170 PRINT #6;" OUT OF THE"
1175 PRINT #6;" "
1180 PRINT #6;" ";N4$;" AND"
1185 PRINT #6;" "
1190 PRINT #6;" GRABBED"
1195 PRINT #6;" "
1200 PRINT #6;" THE ";N3$;"'S"
1201 PRINT #6;" "
1202 PRINT #6;" MAGIC ";N5$;"."
1210 FOR I=1 TO 1500:NEXT I
1220 GRAPHICS 2+16
1230 PRINT #6;" ";N1$;" WAVED THE"
1235 PRINT #6;" "
1240 PRINT #6;" ";N5$;" AND"
1245 PRINT #6;" "
1246 PRINT #6;" TURNED THE"
1247 PRINT #6;" "
1250 PRINT #6;" ";N3$;" INTO "
1255 PRINT #6;" "
1260 PRINT #6;" A FAT, LAZY"
1265 PRINT #6;" "
1266 PRINT #6;" ";N6$;"."
1270 FOR I=1 TO 1500:NEXT I
1280 GRAPHICS 2+16
1290 PRINT #6;" ";N1$;" CHASED THE"
1295 PRINT #6;" "
1300 PRINT #6;" ";N6$;" THROUGH"
1301 PRINT #6;" "
1302 PRINT #6;" THE ";N2$;"."
1310 PRINT #6;" "
1311 PRINT #6;" THEN ";N1$;" RAN"
1315 PRINT #6;" "
1320 PRINT #6;" ALL THE WAY"
1325 PRINT #6;" "
1330 PRINT #6;" BACK HOME."
1350 FOR I=1 TO 1500:NEXT I
1360 GRAPHICS 2+16
1365 SETCOLOR 0,6,10
1370 POSITION 2,5:PRINT #6;"**  THE E
     ND  **"
1390 FOR I=1 TO 1500:NEXT I
1395 REM ***
1396 REM *** WOULD CHILD LIKE THE
1397 REM *** SAME STORY AGAIN?
1398 REM ***
1400 GRAPHICS 0
1402 PRINT "WOULD YOU LIKE TO SEE"
1403 PRINT "THE SAME STORY AGAIN";:IN
     PUT ANSWER$
1404 IF ANSWER$="Y" THEN 1000
1405 IF ANSWER$<>"N" THEN 1400
1406 REM ***
1407 REM *** WOULD CHILD LIKE TO
1408 REM *** INVENT A NEW STORY?
1409 REM ***
1410 GRAPHICS 0
1411 PRINT "WOULD YOU LIKE TO"
1415 PRINT "INVENT A NEW STORY";:INPU
     T ANSWER$
1420 IF ANSWER$="Y" THEN 535
1430 IF ANSWER$<>"N" THEN 1410
1435 GRAPHICS 0
1440 POSITION 6,8
1445 POKE 752,1
1450 PRINT "***  BYE!  BYE!  BYE!  **
     *"
1460 FOR I=1 TO 600:NEXT I
1470 GRAPHICS 0
1475 POKE 752,0
1480 END
```

©

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



www.commodore.ca

A Monthly Column
Learning With Computers

A good simulation is a game so real, so responsive, that it seems to imitate a real situation. Here are two reviews of excellent educational simulations and some ideas on how to create a simulation of your own.

Computer Simulations: Learning Through Exploration, Discovery, And Play

Glenn Kleiman
Teaching Tools: Microcomputer Services
Palo Alto, CA

I hear and I forget;
I see and I remember;
I do and I understand.

Many educators have extolled the virtues of learning through exploration, discovery, and play. These modes of learning are active – guided by the learner's own curiosity and interests. They provide opportunities to acquire new information, discover general principles, test ideas, and develop thinking and problem-solving skills. Active learning is both more enjoyable and more effective than learning that is imposed upon students. In *Democracy and Education,* the American philosopher John Dewey, one of the most influential advocates of active, experiential learning, wrote:

> The fundamental fallacy in methods of instruction ... consists in supposing that we can begin with ready-made subject matter of arithmetic, or geography, or whatever, irrespective of some direct personal experience.... The first stage of contact with any new material must inevitably be of the trial and error sort. An individual must actually try, in play or work, to do something with material ... and then note the interaction of his energy and that of the material employed. This is what happens when a child at first begins to build with blocks, and it is equally what happens when a scientific man in his laboratory begins to experiment with unfamiliar objects.... [Effective methods of education] give the pupils something to do, not something to learn; and the doing is of such a nature as to demand thinking, or the intentional noting of connections; learning naturally results.

Active learning has traditionally been limited by the number of objects, places, and experiences available for students to explore. But suppose we could make almost any type of experience available to students. They could learn about zero gravity environments by spending an hour in one. They could learn about city government by becoming the mayor and members of the city council. They could perform genetic engineering experiments with DNA, no matter how dangerous and unfeasible such experiments would be in actuality. They could experience being businessmen, air traffic controllers, architects, real estate brokers, generals, explorers, archeologists, or astronauts. Computer simulations make it possible for students to experience some aspects of all these roles and situations.

A computer simulation is a dynamic representation of a real object, situation, or environment. A representation reflects the main properties of an actual object or event as, for example, a map represents a city. A map, however, is static – it does not change in response to any type of actions. A simulation is called a *dynamic* representation because it responds and changes in a manner analogous to the real object, situation, or environment. Simulations can be actively explored, and students can

www.commodore.ca

WIN $5,000
Plus Royalties!*

For the best Talking Game

For the Atari® or Apple®II+

Using the VOICE BOX™ Speech Synthesizer

Now you can make your Atari® 400/800 or Apple® II games and other programs come alive with the VOICE BOX by the Alien Group — the first low-cost, smart speech synthesizer with unlimited vocabulary.

Add jokes to your programs. Insults. Compliments. Help messages. Stories. Alien voices. Animal roars. Have your computer talk to the fire department or police in emergencies. To kids. Or blind people. Teach touch typing with immediate spoken feedback. Or just about any other subject — the fun way. Or help a speech-impaired friend communicate . . . the possibilities are limitless.

The VOICE BOX plugs into your Atari's serial port. And talks directly through your TV set. Or into any Apple II slot. No power supply or special interfaces needed.

Just select from its simple screen menu. A dictionary with thousands of common words (on diskette or cassette) automatically translates your text into speech. It's that easy.

But don't let its friendliness fool you. The VOICE BOX has all 64 phonemes (basic sounds, like "ah") built in. So you can precisely create any word or sound you can imagine. And store it all on diskette or tape. Names or foreign language words, for example. Or wierd non-human languages.

Let me entertain you — The VOICE BOX is creative too. It will crack you and your friends up with non-stop random, grammatically correct sentences, using words you specify. It also has an amusing talking face with lip-sync animation — a real crowd-stopper. Best of all, you can call the VOICE BOX from any BASIC program and make your program really hum —literally!

Singing Apples? — Apple owners get all these capabilities too — as a plug-in card plus diskette. Or there's a deluxe version with the dictionary in ROM (no diskettes to bother with), speaker, and ability to "sing" (hey we're not making this up folks) in any key. (Both Apple versions require 32K or more. Applesoft and DOS 3.3).

Don't confuse the VOICE BOX with "dumb" speech synthesizers that can't learn new words. Or software-based ones with lower speech quality — and an annoying tendency to blank out the display when they talk. The VOICE BOX is a true breakthrough in speech synthesis. Small wonder thousands of Atari and Apple owners have already bought the VOICE BOX.

The VOICE BOX is available now at leading computer stores throughout the world. Or direct from the Alien Group, with 10-day money back guarantee if you're not completely satisfied.

VOICE BOX For Atari. $169.00
16K and 32K versions included
(Specify diskette or cassette).

VOICE BOX for Apple II.+ $139.00.
(Requires speaker.)

VOICE BOX for Apple II+. $215.00
(Includes dictionary in ROM and singing capability. Comes with speaker.)

Enclose check or money order, Visa or Mastercard #. Please include expiration date.

CHECK YOUR LOCAL COMPUTER STORE FOR TALKING "VOICE BOX" VERSIONS OF YOUR FAVORITE GAMES FROM LEADING GAME COMPANIES

*Win a $5,000 prize — plus royalties — for the best Atari 400/800 or Apple II+ game using the VOICE BOX. Deadline: May 30, 1983. Write for contest details.

Please mail to: The Alien Group, Department PT-2 , 27 W. 23 St., N.Y., N.Y. 10010
Or call in order to (212) 741-1770

Atari is a registered trademark of Atari Inc. Apple is a registered trademark of Apple Computer, Inc. VOICE BOX is trademark of the Alien Group.

learn from experiencing how the simulation responds to their actions.

Some simulations, such as the board game Monopoly, are familiar to almost everyone. Monopoly is a simulation of a real estate market. Each player assumes the role of an investor who buys, sells, and trades properties, trying to gain good locations to build houses and hotels, thereby amassing a fortune.

Children can learn a great deal from playing Monopoly. The game involves rents, taxes, utility bills, and banking, as well as financial successes and failures. Players must decide how to spend a limited amount of money wisely, and they learn about making investments for future returns. They practice and develop skills at negotiating, skills that can be developed only through experience. Money is constantly exchanged, so math skills are developed. I first learned about percentages as a result of landing on the dreaded income tax square and having to pay 10% of my assets. Some reading is required – I also learned the meaning of *assets*.

Simulations are never complete and precise representations. The aim of a simulation is to capture the main characteristics of what is being represented. The precision and completeness required depend upon the purposes for which the simulation will be used. For example, a fairly simple simulation of an airplane is sufficient for children, but a precise, detailed simulation is required to train pilots.

Computer simulations can be more complex, precise, and complete than any other type of simulations. Many things, such as zero-gravity environments and genetic engineering experiments, can be reasonably simulated only with a computer. Computer simulations can capture more aspects of reality and give people more flexibility in how they explore and experience the simulated environment.

A Roadtrip Simulation

A program called *Roadtrip* is a good example of a simulation which is both enjoyable and educational for children. Like many simulations, *Roadtrip* takes the form of a game. The aim of the game is to complete a 900-mile car trip from Dullsville to Greenstone Park. The player has a maximum of two days and $200 to spend. Along the way, he has to make many decisions like those in an actual trip.

The *Roadtrip* program is for Apple II computers and makes excellent use of high resolution, color graphics. The screen displays show a car dashboard and the views through the windshield and rear view mirror (see Figure 1). The dashboard has a speedometer, odometer, gas gauge, clock, alternator, and oil warning lights. The views

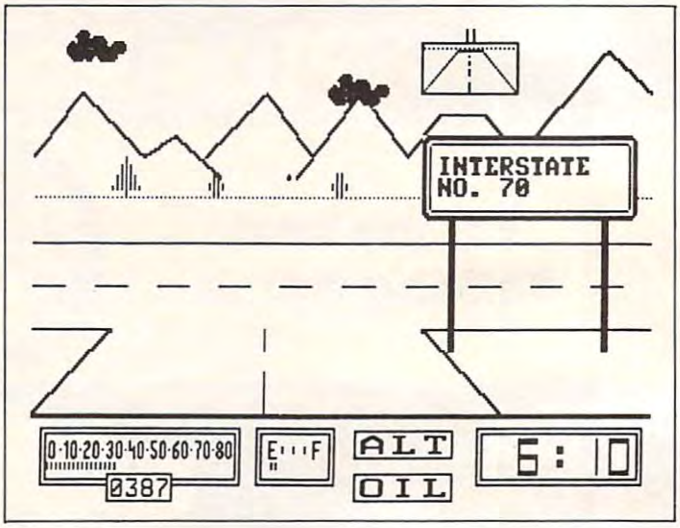


Figure 1.

through the windshield and rear view mirror change each time the car moves. The programmers of *Roadtrip* paid careful attention to every detail. For example, the sun comes up in the morning, and the stars appear at night.

While in the car, players have seven commands available. Pressing *F* moves the car forward; *R*, *L* and *T* turn the car to the right, left or completely around; *S* lets the player change the speed; *$* displays how much money is left; and *M* displays a map. The map is important: there are many crossroads and it is easy to get lost. Many a *Roadtrip* traveler has ended up in the swamp or Slipdisk City.

Along the way, players pass through towns where they can purchase gas, go to a restaurant, and check into a hotel. If you run out of gas, the game is lost. Failing to get sufficient food and rest increases the likelihood of an accident. If you get to a town late at night, the gas station and restaurant may be closed.

This simulation contains a number of events which may occur along the way. The computer's randomization capability is used, so it is impossible to predict if and when each event will occur. You may get a flat tire or have other car problems, have to wait for trains, pass hitchhikers, run into roads closed for construction, and so on. If you exceed the speed limit, you are likely to be stopped by a police car, be delayed, and have to pay a fine. Excessive speed, like lack of food and rest, makes accidents more likely.

Roadtrip provides an opportunity for children to explore and learn about car travel. Map reading is critical to success, as is careful attention to the amount of money being spent, getting enough food and sleep, and obeying speed signs. This program lets children experience in play many of

www.commodore.ca

COW BAY COMPUTING

has a lot to offer you and your

PET/CBM

SOFTWARE FOR EDUCATION

o THE PET PROFESSOR $499.00

Whole Numbers Only	$235.00
Fractions Only	$175.00
Decimals Only	$175.00

A total arithmetic package with step-by-step instruction. 77 programs in addition, subtraction, multiplication and division on cassettes or diskettes. Ask for a sample.

o SINGLE CASSETTES $15.00
Math, English, Science.

WORKBOOKS FOR COMPUTER LITERACY

o Feed Me, I'm Your PET $4.95
Beginning lessons, exercises, worksheets, homework, quizzes.

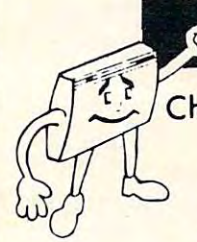
o Looking Good With Your PET $4.95
Intermediate. Same format as above.

o Teacher's PET $4.00
Lesson plans, answers keys for the workbooks.

Descriptive literature is available.

P.O.Box 515
Manhasset, N.Y. 11030

EDUCATIONAL SOFTWARE FOR YOUR VIC



Choose From Programs Such As:

★ GOTCHA MATH
★ ENGLISH INVADERS
★ MISSPELL

only **$9.95** Each

For FREE Program Catalog Call or Write:

COMM*DATA
COMPUTER HOUSE, INC.
P.O. BOX 325
MILFORD, MICHIGAN 48042
(313) 685-0113

We Handle Games For Your VIC, Too!
DRAGONS & TREASURE • VIC TREK
ASTRO COMMAND • AND MORE!

PET Educational Software Also Available

Announcing . . . THE WHOLE PET CATALOG

A two year compendium of the **Midnite Software Gazette** and other resources for users of Commodore, CBM, PET, and VIC computers.

*The **Whole PET Catalog** contains:*

- **Over 500 independent reviews of commercial products.**
- **Over 700 education programs reviewed & organized by course.**
- **Over 200 reviews of free games.**
- **Information on over 1,800 free programs.**
- **Information about dozens of PET and VIC user groups.**
- **Many pages of hints and helps for all Commodore users.**
- **"Commodore's Family Tree", by Jim Butterfield.**
- **Completely reorganized and greatly expanded edition.**
- **Typeset and printed full-size on bond paper.**
- **In all, 320 pages of useful information.**

If you've seen **Midnite** before, directly or reprinted in the **TorPET** newsletter, here it is, complete in one volume, completely reorganized for easy reference, and greatly expanded with new information from members of the Toronto PET Users' Group.

"I still use my copy of **The Best of the PET Gazette** regularly. It was a treasure trove of information, and a great bargain for $10 three years ago. I hope you'll feel the same way about my **Whole PET Catalog** someday. Considering that it's three times as long, completely organized by topic, printed on bond paper instead of newsprint, typeset instead of dot matrix printed, bound instead of stapled, and still only $10, in spite of inflation, I'm sure you will." *—Jim Strasma, Contributing Editor, **Micro***

Whole PET Catalog $8

252 Bethlehem Pike
Colmar, PA 18915
215-822-7727

A B Computers

WRITE FOR CATALOG
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular prices slightly higher. Prices subject to change.

www.commodore.ca

the situations people encounter while traveling.

Roadtrip was created as a class project by students of Jay Dean at the University of Minnesota. For Apple II computers with Applesoft, it is a public domain program, available (with two other class project simulations) for $10 from Softswap, San Mateo County Office of Education, 333 Main Street, Redwood City, CA 94063. For a catalog and order forms, send $1.00. *Roadtrip* uses a utility program called *Higher Text* to create type fonts. You do, however, have to obtain *Higher Text* and transfer the files to the *Roadtrip* disk. *Higher Text* is available for $40 from Synergistic Software, 5221 120th Avenue, S.E. Bellevue, WA 98006. (*Higher Text* is a very useful utility program if you do any programming yourself. It lets you create all kinds and sizes of type fonts on the Apple high resolution screen.)

A Logic Machine's Simulation

I have just received a review copy of an innovative new program which includes computerized simulations, tutorials, and demonstrations, combined into an educational game. The program, called *Rocky's Boots*, teaches about the building blocks of computer systems, such as AND, OR and NOT logic gates, flip-flops, clocks, delays, sensors and actuators. The operation of each device is explained and demonstrated. The simulation game has players use simple logical devices to build and test machines.

There are six levels in the program. Within each level is a set of rooms, each of which contains an explanation, demonstration, exercise, or other information. The player controls a cursor, moving it among the rooms by using either a joystick or the keyboard.

The first level of the program teaches how to move and pick up objects. The second level begins the lessons on building machines. The cursor contains electricity, so when it is placed on a socket of a device, electricity flows through the device. The flow of electricity is shown by the color orange.

Several *actuators* (devices which perform actions when connected to electricity) are introduced. These include a clacker which makes noise, a thruster which moves, and a boot which kicks. The player must discover what each one does by activating it with electricity from the cursor.

The concepts of *input* and *output* are then illustrated, and *sensors* are introduced. Each sensor detects a certain class of objects. One sensor detects green objects, another detects square objects, and so on. When a sensor detects an object, it sends electricity out. Players can, for example, connect a green sensor to a boot actuator, so that whenever a green object is detected, the sensor will send electricity to the boot actuator, which will then kick the green object. The program provides several practice rooms in which players can build and test all sorts of machines.

After some of the fundamentals are mastered, players are introduced to NOT, AND and OR logical gates. A NOT gate has one input line. When there is no electricity at the input line, the NOT gate sends electricity through its output line. When there is electricity at the input line, none is sent out. AND and OR gates each require two inputs. AND gates send electricity out only when there is electricity at both inputs. OR gates send electricity out when there is electricity at either one (or both) of the inputs.

Once these devices are understood, players are ready to try various games. Each game involves building a machine which will kick certain objects. For example, in one game the machine is to kick circles and squares, but not to kick triangles or crosses. Players receive points when their machine kicks an appropriate object and lose points if their machine kicks an inappropriate object. Once the machine is built, the player can throw a switch which causes various objects to flow into the room. The machine can be slowed or stopped at any point so its operation can be watched carefully.

Figure 2 shows a device which will kick squares and circles. The two boxes on the right are sensors. The top one detects circles, the bottom one detects squares. They are connected by wires to the input lines of an OR gate. The output line of the OR gate is connected to a boot actuator. When a square or circle comes into the room, the appropriate sensor will detect it and send electricity to the OR gate. Since electricity at either input causes electricity at the output of the OR gate, power is sent to the

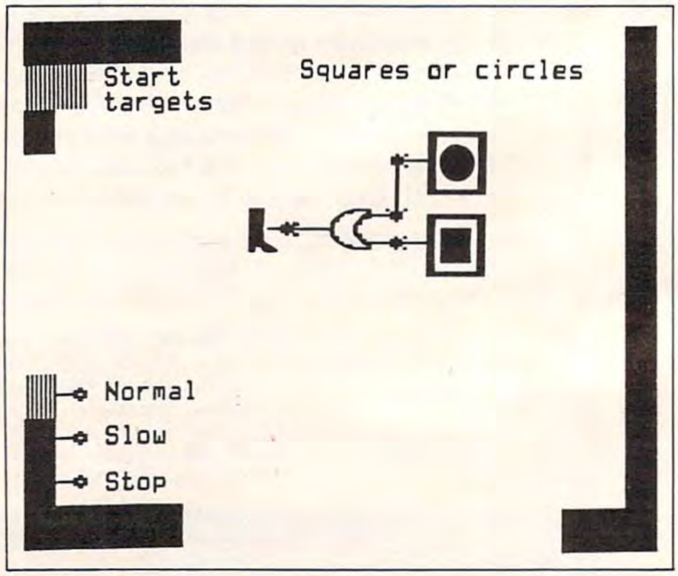


Figure 2.

www.commodore.ca

boot, which kicks the object.

The other levels of the program add more devices, such as flip-flops, clocks, and delays. It then presents a series of games requiring the building of progressively more complex machines. Players can explore all sorts of combinations of simple logical devices.

Students can learn a great deal by actively exploring and playing with *Rocky's Boots*. It presents a carefully structured environment so that new knowledge and understanding are built step-by-step. Children can explore this environment at their own speed and follow wherever their interests lead. *Rocky's Boots* is an exciting and innovative program, one which makes the often proclaimed educational potential of personal computers a reality.

Rocky's Boots is available for $75 from The Learning Company, 4370 Alpine Road, Portola Valley, CA 94025.

Creating A Simulation Program

Many other simulations are available. Some examples are: *Lemonade*, a simulation of a very small business; *Oregon Trail*, a simulation of traveling across the U.S. in the 1800s; *Three-Mile Island*, a simulation of controlling a nuclear reactor; *Windfall*, a simulation of the oil market; and *Air Traffic Controller*.

Since simulations are widely touted as having great educational potential, it may seem surprising that few good simulations are available. The reason is that good simulations are very difficult to create.

In order to create a simulation, you must first create a model of what is being simulated. You then translate the model into a computer program, designed for students to learn by exploring and playing. Often it is necessary to simplify the model, so students can manipulate certain factors and come to understand their effects.

Consider, for example, what we would have to do to create a simulation game which would allow students to role-play running a city government. Perhaps we could have one student take the part of the mayor and others the police chief, chairman of the board of education, city council members, and so on.

Students would have to control raising and spending money in their simulated city. We could arrange the program so students can set sales tax and property tax rates. But of course simply raising taxes does not always bring in more money. Raising sales taxes in a city often leads to more people shopping outside the city, so fewer tax dollars are collected. Raising property taxes may lead to businesses leaving the city, thereby lowering the tax base. The simulation program would have to include equations which reflect the effects of these factors. We might also include other ways cities raise money, such as from the state or federal government, tourism, or municipal bonds.

The students should be able to allocate how the available funds would be spent. In a realistic simulation, the city would not have sufficient income for everything the city council would like, so the various departments would have to compete for funding. The simulation should reflect some of the complexity of real cities. For example, departments which pay their employees well might function most efficiently, but overpaying some employees can cause problems. For example, raising police department salaries might result in reduced crime, but it might also cause the firemen to strike for equal wages.

Random events might be built into the simulation. A snowstorm, hurricane, or epidemic could cause difficulties which require funds to be reallocated. A gas shortage would put an excess strain on public transportation, while a visit by a foreign dignitary would require many policemen to work overtime, straining the department's budget.

We would also have to consider evaluating how well the students run the city. Do we require a balanced budget, good schools, public transportation, sanitation and parks, and minimal crime? Should we count whether more people and businesses move into the city than leave it? What about the amount of tourism?

I have described just a few of the initial considerations in creating this simulation. Many other factors could be included, and the program should reflect some of the complex interactions of a real city. Creating a good simulation program requires an expert's knowledge of what is being simulated, combined with skill in designing and implementing programs.

Computer simulations provide new ways of teaching and learning. They certainly have great potential in education. However, creating programs which fulfill this potential is a difficult task, one which requires a great deal of effort by talented and knowledgeable people. ©

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

www.commodore.ca

In Part I last month you were introduced to two Applesoft programs that create a powerful new language for the Apple. This language, called Turtle PILOT, combines PILOT, turtle graphics, and all of Applesoft's commands and functions. Turtle PILOT resembles Atari PILOT, allowing you to translate PILOT programs for the Atari to the Apple. The features of Turtle PILOT described last month were just the beginning. This month, along with the PILOT language for Atari, we will deal with the most interesting features of Turtle PILOT: the turtle graphics. Turtle graphics will add new commands to your computer which make high resolution graphics easier. At the end of this article you will find listings of three example programs in Turtle PILOT. These programs can be typed using the Editor and then translated to Applesoft or Atari BASIC with the Translator, as described in Part I.

Part 2

Turtle PILOT:

Including PILOT For Atari

Alan Poole
Loomis, CA

Introduction To Turtle Graphics

Turtle graphics gives you control over an imaginary and invisible turtle that lives on the screen. This turtle has a set of colored pens. When you instruct the turtle to move, it will leave a trail with its pens. All turtle commands must be preceded by a G: instruction. If there are no commands in the object of a G: instruction, page one of Apple high resolution graphics will be turned on without clearing. (This would result in Graphics 7 on the Atari.) Below is a description of each of the 11 turtle commands. Each description is followed by several examples using the command.

CLEAR

The CLEAR command sets Apple's high resolution graphics page one and clears the screen. (Atari goes to full-screen Graphics 7 and clears the screen.) This is identical to Applesoft's HGR command.

```
G:CLEAR
GY:CLEAR
```

TURN

The TURN command is followed by an expression. The value of the expression is the number of degrees added to the angle the turtle is presently headed. A positive number turns the turtle clockwise, and a negative number turns it counterclockwise.

```
G:TURN 90
G:TURN A
G(V<10):TURN N+INT(X*RND(1))
```

TURNTO

This command will change the angle of the turtle. The turtle's angle will be set equal to the value of the expression following the TURNTO command. Zero degrees is north, 90 degrees is east, 180 degrees is south, and 270 degrees is west.

```
G:TURNTO 20
GN:TURNTO ABS(K)+8
```

DRAW

The DRAW command will move the turtle at the angle it is presently heading, leaving a trail on the screen as it moves. The value of the expression following the DRAW command is the distance the turtle will move. If the turtle hits the edge of the screen, it will stay at the edge.

```
G:DRAW 25
G:DRAW D-6
GY(L<>1):DRAW RND(1)*50
```

GO

The GO command is similar to the DRAW command, except the turtle will not leave a trail.

```
G:GO 50
GY:GO S1+ABS(S2)
```

GOTO

This command moves the turtle to new coordinates on the screen without leaving a trail. The GOTO command is followed by the x and y coordinates separated by a comma. The coordinates the turtle uses are more like a normal graph than Applesoft's high resolution coordinates. The origin is in the middle of the screen instead of the upper left-hand corner. As you move up the graph, the y coordinate increases. As you move to the right, the x coordinate increases. The range of x coordinates is from -139 to 140, and the range of y coordinates is from -111 to 80.

```
G:GOTO 0,10
GN:GOTO -50,-5
GY(W>5):GOTO X,Y*N+2
```

PEN

The PEN command changes the color of the pen which the turtle uses to draw. It may be followed by the color names BLACK, GREEN, VIOLET, WHITE, BLACK2, RED, BLUE, or WHITE2. If the colors on your monitor are different, the color names can be changed in lines 5410-5445 of the Translator program. The PEN command may also be followed by UP or DOWN. Setting the pen to UP will cause all DRAW commands to move the

www.commodore.ca

turtle without leaving a trail. DOWN will return the pen to normal. A final option with the PEN command is to use the color ERASE. This will set the pen to the background color, which is normally black unless changed with a SCREEN command.

```
G:PEN RED
G:PEN ERASE
G(C=1):PEN UP
```

SCREEN

This command will clear the entire screen to the color that follows the SCREEN command. The color names are listed above under the PEN command.

```
G:SCREEN BLUE
GN:SCREEN WHITE2
```

FULL

Sets full screen graphics mode with no text at the bottom of the screen.

```
G:FULL
```

MIX

Sets the mixed text and graphics mode with four lines of text at the bottom of the screen.

```
G:MIX
```

QUIT

The QUIT command turns off the high resolution graphics and returns to the text mode. It is identical to Applesoft's TEXT command.

```
G:QUIT
```

Multiple Turtle Commands On A Line

The object of a G: instruction can hold up to six turtle commands. The commands are separated by semicolons. Below are some samples of multiple turtle commands on the same line.

```
G:DRAW 20; TURN 90
G:CLEAR; SCREEN BLUE; PEN RED; TURNTO 10
```

Turtle Loops

Suppose you want to draw a square. You could use the following series of commands.

```
G:DRAW 50
G:TURN 90
G:DRAW 50
G:TURN 90
G:DRAW 50
G:TURN 90
G:DRAW 50
G:TURN 90
```

This seems like a lot of work to draw a simple square. Wouldn't it be easier if there were a simple way to loop turtle commands? This is one of the features included in Turtle PILOT. You can place up to six turtle commands between parentheses and put an integer in front of them for the number of times to loop them. For example, the following instruction would draw the same square as the eight instructions above.

```
G:4(DRAW 50; TURN 90)
```

There are limitations with these loops. A loop cannot be placed inside another loop. Also, a command cannot be outside a loop on the same line. The following instructions would *not* be legal in Turtle PILOT.

```
G:4(DRAW 30; 3(TURN 120; DRAW 10))
G:PEN WHITE; 6(DRAW 75; TURN 60)
G:10(DRAW L; TURN 36); QUIT
```

Using High Resolution Page Two

If you write a long Turtle PILOT program that uses turtle graphics, you may find that there isn't enough memory. An extra 8K in the Apple's memory can be used by drawing on page two of high resolution graphics instead of page one. To do this, use a B:HGR2 instruction instead of G:CLEAR. All other turtle commands will work normally on page two.

PILOT Variables

Last month I mentioned that variables beginning with Q cannot be used in a Turtle PILOT program. This is not exactly true, but you must know how to use them correctly. Q variables are used in the translated program to execute some of the PILOT instructions. Below is a description of each of the variables and some possible uses for them in programs.

QM: This variable holds the number of the item that successfully Matched last. If more than one item Matched, QM will equal the number of the first item. This variable can be very useful, especially with programs containing a question with multiple choice answers. Program 3 with this article illustrates this.

QC: Conditioner flag, 0 = N, 1 = Y.

QR: Right margin, normally set to 40.

QI$: User's last response with an Accept instruction. The Match instruction normally uses the last response, but a string can be used instead by setting QI$ equal to the string immediately before the Match instruction. For instance, the following instructions would search for the word "TO" in R$.

```
C:QI$=R$
M:TO
```

QX,QY: Coordinates of the turtle.

QA: Angle of turtle.

QL: Length of line drawn by turtle.

QP: Pen position, 1 = UP, 0 = DOWN.

QB: Number of background color. Changes after a SCREEN command.

Q$(25): List of items to be Matched.

www.commodore.ca

Q(31): List of values for pitches of notes.
QT$: String to be Typed.
Q1, Q2, QT, QI, QK$: Temporary variables.

Next month will be the third and last article on Turtle PILOT. We'll translate an Atari PILOT program to Turtle PILOT, provide some documentation on the Editor and Translator programs, and include PILOT for Commodore machines. If you want more information on turtle graphics and PILOT, read the "Friends Of The Turtle" column in **COMPUTE!**.

Program 1.

```
1 *EXAMPLE 1
2 R:DRAWS INTERESTING PATTERNS
3 B:TEXT:HOME
4 T:TYPE AN ANGLE (BETWEEN 70 AND 150 IS BES
   T).
5 A:A
6 G:CLEAR;GOTO 0,-17; TURNTO 0; PEN WHITE; F
   ULL
7 C:L=1
8 *DRAW
9 G:DRAW L; TURN A
10 C:L=L+2
11 J(L<125):*DRAW
12 E:
```

Program 2.

```
1 *EXAMPLE 2
2 R:DRAWS STARS OF RANDOM SIZE
3 G:CLEAR; FULL; PEN WHITE
4 *START
5 C:X=RND(1)*220-139+30
6 C:Y=RND(1)*132-111+30
7 G:GOTO X,Y
8 U:*STAR
9 C:STARS=STARS+1
10 J(STARS<10):*START
11 E:
12 *STAR
13 C:SIZE=INT(RND(1)*25+5)
14 S:SIZE,75
15 G:5(DRAW SIZE; TURN 144)
16 E:
```

Program 3.

```
1 *EXAMPLE 3
2 R:EXAMPLE OF USING QM
3 T:WHAT IS THE NAME OF THE LARGEST OCEAN
4 T:          ATLANTIC OCEAN
5 T:          PACIFIC OCEAN
6 T:          INDIAN OCEAN
7 *ANSWER
8 A:
9 M:ATLANTIC,PACIFIC,INDIAN
10 J(QM=1):*ATLANTIC
11 J(QM=2):*PACIFIC
12 J(QM=3):*INDIAN
13 T:PLEASE TYPE ONE OF THE THREE ANSWERS.
14 J:*ANSWER
15 *ATLANTIC
16 T:NO, THE ATLANTIC OCEAN IS THE SECOND LAR
   GEST OCEAN. TRY AGAIN.
17 J:*ANSWER
18 *PACIFIC
19 T:YES, THE PACIFIC OCEAN IS THE LARGEST OC
   EAN.
20 E:
21 *INDIAN
22 T:NO, THE INDIAN OCEAN IS THE THIRD LARGES
   T OCEAN. TRY AGAIN.
23 J:*ANSWER
```

Turtle PILOT For The Atari

Charles Brannon
Editorial Assistant

Use Program 1, the Turtle PILOT Editor, to enter and edit PILOT programs. Program 2, the Translator, converts your PILOT program into a BASIC program that is ready to RUN. Program 1 requires 32K with a disk, or 24K with a cassette. Program 2 requires 40K with a disk, or 32K with a cassette. By adjusting the MAX variable (line 1410 in Program 1, line 240 in Program 2), you may be able to adapt PILOT to systems with less RAM.

Using Atari Turtle PILOT

Using the Turtle PILOT Editor is like typing in a BASIC program, but there are some important differences. The Editor has 13 commands to help you type in, edit, save, and load PILOT programs (see the Quick Reference Chart). Each command is acknowledged with an "OK" prompt. If you see the "READY" prompt, you've somehow returned to BASIC.

The ADD command is used to enter programs sequentially. Just type "ADD," and you will be prompted with a line number. You can then type in a PILOT line, which will be added to the end of your program. For example,

```
1>T:What is your name?
2>A:NAME$
3>
```

Press RETURN> alone on a line to exit the ADD command. While in the ADD mode, you can't cursor up to change previous lines, so be careful. You must use ADD to add lines to the bottom of your program, but you can use cursor-based editing to change any line already typed. Simple syntax

www.commodore.ca

checking is performed. The line must start with a valid PILOT command and must contain a colon.

List, Insert, Delete

LIST is used, as in BASIC, to display the program you are working on. Just type "LIST," and you will be asked for the starting and ending lines to list. "Default" answers are automatically provided, so if you want to list the whole program, just press RETURN twice. Otherwise, type over the default answers. While the program is listing, you can press "ESC" (for Escape) to abort the listing (used like the BREAK key, which is disabled in this program).

LMOD will find and list a specified module. For example, if you have a module named "*TURTLE," just type LMOD, and answer the prompt with "TURTLE" (the asterisk is supplied for you).

If you want to insert a line between two lines, enter "INS" (for INSert), and answer the prompt with the line number at which you wish to insert. The given line and all following lines will be "pushed down," and the given line will show as "BLANK." You can then LIST the program and cursor up to the blank line and make your addition.

To delete a line from a program, just type in its line number and press RETURN. The program will be automatically renumbered. To delete a range of lines, just type "DEL" and enter the start and end lines of the block of lines you want to delete. Use the NEW command to erase the entire program.

When using LOAD and SAVE, supply the complete filename (either C: or D:name), but don't use the optional three-character extender, as this is supplied automatically by the program. If you have a PILOT program in memory and type LOAD, the Editor will assume you want to append a program to the end of the one in memory. If you don't wish to do this, hit RETURN to exit the LOAD command, enter "NEW," and then type LOAD again.

Disk users will find the "DIR" command very helpful. It displays the directory of drive 1. Used in conjunction with PON, you can have a hardcopy listing of the directory. The PON command "turns on" the hardcopy option. After a PON command, all output will be sent to the printer (assuming you have one attached and turned on). You can use this feature to print listings of your PILOT programs. Use POFF to "de-select" the printer.

You can exit the Editor with either "BYE" or "RUN." The former will simply restore the break key, clear the screen, and return you to BASIC. "RUN" will run the Translator on disk-based systems. Make sure you save the program you're working on before you use "RUN" or "BYE."

Operation of the Translator is very simple. Just answer the filename prompt with the name of your PILOT program (you don't have to type D:). If you are using a cassette, position the tape to read the PILOT program, press PLAY, and answer with "C:". Press RETURN when you hear the beep. The Translator will then read in the PILOT

Quick Reference Chart

Editor Commands

ADD	Adds lines to program from keyboard.
LIST	Displays program.
LMOD	Lists module.
INS	Inserts line.
DEL	Deletes range of lines.
NEW	Erases program.
LOAD	Enters or appends program from tape or disk.
SAVE	Saves program to tape or disk.
DIR	Lists disk directory.
PON	Sets hardcopy feature.
POFF	Clears hardcopy feature.
RUN	RUNs Translator (disk only).
BYE	Exits to BASIC.

Turtle PILOT Commands:

T: Type line. Use $ and # to include variables.
A: Accept (ask for) input.
M: Match last accept with list of items separated by commas. Sets Y/N flag.
J:*LABEL Jump to indicated label.
U:*MODULE Use (call, GOSUB) indicated module (subroutine).
E:END Used to stop program or end a module.
C:Compute. Used to calculate variables. Similar to B:
R:Remark. Used to comment your program, like REM in BASIC.
S:Sound x,y x = tone, y = duration. Uses only voice 0. Use B: and SOUND for other effects.
B:line Compile a BASIC line.
* (No colon) Indicates a label.
G:Graphics (see below)

Graphics Subcommands

CLEAR	Clears screen; enters Graphics 7.
TURNTO A	Points turtle to angle A.
TURN A	Rotates turtle A degrees.
DRAW N	Moves turtle N units, leaving a trail.
GO N	Moves N units without leaving a trail.
PEN RED	
PEN GREEN	
PEN BLUE	
PEN ERASE	Selects drawing color.
PEN UP	
PEN DOWN	Permits or prevents drawing.
SCREEN ATARI	
SCREEN APPLE	Sets scale of drawing. (See text.)
GOTO X,Y	Go to absolute coordinate (-79,79; -47,47).
FULL	Removes text window.
MIX	Enables text window.
QUIT	Goes to Graphics 0.

www.commodore.ca

program and start to work on the translation. If you have a cassette, insert a blank tape, rewind it, and press RETURN when you hear two beeps.

When the Translator is finished, you can use ENTER to read the completed BASIC program into memory. (Use ENTER"C" for cassette or ENTER"D:name.ENT" for disk, where "name" is the name of the program.) The Translator automatically NEWs itself out on completion, so you don't have to worry about the program-merging effect of ENTER. You may want to change the "NEW" on line 390 to "END" while you are typing in and correcting the program. For safety's sake, SAVE a copy of the Translator before you RUN it.

The "BASIC" program which results from your efforts will run as is and will mimic the action of the PILOT program, albeit slower.

A Note On Graphics

This PILOT system was converted from the Apple version published last issue. Since Apple graphics differ from Atari graphics, a few things need to be mentioned. The Apple used high-resolution page one, which permits eight simultaneous colors, while the Atari version uses Graphics 7, a four-color mode with less resolution than the Apple screen. To allow Atari users to enter Apple programs without changes, the SCREEN command can be used in your PILOT program to select the "scale" of points plotted. SCREEN ATARI is the default mode, but if your program includes SCREEN APPLE, all coordinates are "scaled" from Apple coordinates (0-279) to Atari coordinates (0-159). All unimplemented Apple colors are plotted in COLOR 1.

Program 1.

```
100 REM TURTLE PILOT EDITOR
110 REM
120 GOSUB 1300:POKE 752,0
130 FOR W=15 TO 0 STEP -1:SOUND 0,10,
    10,W:NEXT W
140 SETCOLOR 4,6-5*PON,2+4*PON:? :? "
    OK."
150 INPUT #1;IN$:IF IN$="" THEN 150
160 IF IN$(1,1)=" " THEN IN$=IN$(2):G
    OTO 160
170 TRAP 190:V=VAL(IN$):IF V>0 AND LE
    N(IN$)>=LEN(STR$(V))+2 THEN IN$=I
    N$(LEN(STR$(V))+2):TRAP 40000:GOT
    O 580
180 IF V>0 AND V<=EL THEN A=V:B=V:GOT
    O 1030
190 IF IN$<>"DIR" THEN 230
200 TRAP 1190:CLOSE #2:OPEN #2,6,0,"D
    :*.*":TRAP 220
210 INPUT #2;IN$:? #6;IN$:GOTO 210
220 CLOSE #2:TRAP 40000:GOTO 130
230 IF IN$<>"PON" THEN 260
240 PON=0:TRAP 250:CLOSE #6:OPEN #6,8
    ,0,"P:":TRAP 40000:PON=1:GOTO 130
250 CLOSE #6:OPEN #6,8,0,"E:":SETCOLO
    R 2,6,2:? "Printer not ready.":?
    :GOTO 130
260 IF IN$="POFF" THEN CLOSE #6:OPEN
    #6,8,0,"E:":SETCOLOR 2,6,2:PON=0:
    GOTO 130
265 IF IN$="BYE" THEN GRAPHICS 0:POKE
     16,192:POKE 53774,192:END
270 IF IN$<>"RUN" THEN 320
280 ? "Press RETURN to RUN translator
    ,":? "Press ESC to abort...";
290 IF PEEK(764)=255 THEN 290
300 ? :IF PEEK(764)=12 THEN POKE 764,
    255:POKE 16,192:POKE 53774,192:RU
    N "D:PILOT.XLT":REM RUN TRANSLATO
    R
310 POKE 764,255:GOTO 130
320 IL=LEN(IN$)
330 F=0:FOR I=1 TO 8:IF IN$=CMD$(I*4-
    3,I*4-(IL<4)) THEN F=I:I=9
340 NEXT I:IF F THEN 370
350 GOSUB 390:REM ERROR SOUND
360 ? :? IN$;"? -- WHAT'S THAT?":GOTO
     140
370 ON F GOTO 420,610,750,880,1000,11
    00,1120,1210
380 STOP
390 FREQ=ASC(IN$)
400 FOR W=0 TO 15:SOUND 0,FREQ,12,W:N
    EXT W:FOR W=15 TO 0 STEP -0.2:SOU
    ND 0,FREQ,12,W:NEXT W:RETURN
410 REM ADD
420 ? :? EL+1;">";:INPUT #1;IN$
430 IF IN$="" THEN 140
440 ZL=EL:GOSUB 470
450 EL=EL+1:GOTO 420
460 REM * ENTER LINE *
470 IF IN$(1,1)=" " THEN IN$=IN$(2):G
    OTO 470
480 K=ASC(IN$):REM * SYNTAX CHECK *
490 F=0:FOR I=1 TO 12:IF K=ASC(PILOT$
    (I)) THEN F=I
500 NEXT I:IF F>0 AND F<12 THEN 530
510 IF F=12 THEN 560
520 GOSUB 390:? "{ESC}";CHR$(K);":  i
    s not a PILOT command":? :POP :GO
    TO 150
530 F=0:FOR I=1 TO LEN(IN$):IF IN$(I,
    I)=":" THEN F=1:I=LEN(IN$)
540 NEXT I:IF F THEN 560
550 GOSUB 390:? IN$;"? No colon":? :G
    OTO 150
560 LL(ZL)=LEN(IN$):L$(ZL*80+1,ZL*80+
    LL(ZL))=IN$
570 RETURN
580 IF V>EL THEN GOSUB 390:? "Use ADD
     to add to end.":GOTO 140
590 ZL=V-1:GOSUB 470:GOTO 150
600 REM LIST
610 IF EL=0 THEN GOSUB 390:? :? "No l
    ines to list!":GOTO 140
620 TRAP 620:? "Starting from line?1
    {2 LEFT}";:INPUT A:TRAP 40000
630 IF A<1 OR A>EL THEN 730
640 TRAP 640:? "To Line?";EL;:POKE 85
    ,9:INPUT B:TRAP 40000
650 IF B<1 OR B>EL THEN 730
```

www.commodore.ca

```
660 POKE 766,1:? :REM SHOW CTRLS
670 FOR I=A-1 TO B-1
680 IF LL(I)=0 THEN ? #6;I+1;" BLANK
    ":GOTO 720
690 T$="{3 SPACES}":T$(4-LEN(STR$(I+1
    )))=STR$(I+1):IF L$(I*80+1,I*80+1
    )<>"*" THEN T$(LEN(T$)+1)=" "
700 ? #6;T$;" ";L$(I*80+1,I*80+LL(I))
710 IF PEEK(764)=28 THEN I=EL:POKE 76
    4,255
720 NEXT I:POKE 766,0:GOTO 130
730 GOSUB 390:? "Can't do that.":GOTO
     150
740 REM * LMOD - LIST MODULE *
750 IF EL=0 THEN GOSUB 390:? :? "No l
    ines.":GOTO 140
760 ? "Module Name";:POKE 764,7:INPUT
     IN$
770 F=0:FOR I=0 TO EL-1
780 IF IN$=L$(I*80+1,I*80+LEN(IN$)) T
    HEN F=I+1:I=EL-1
790 NEXT I:IF F=0 THEN GOSUB 390:? "M
    odule ";IN$;" not found.":GOTO 14
    0
800 POKE 766,1
810 FOR I=F-1 TO EL-1
820 IF LL(I)=0 THEN ? #6;I+1;" * BLAN
    K *":GOTO 860
830 IF ASC(L$(I*80+1))=69 THEN POKE 7
    64,28:REM "PRESS" ESC KEY
840 ? #6;I+1;" ";L$(I*80+1,I*80+LL(I)
    )
850 IF PEEK(764)=28 THEN I=EL:POKE 76
    4,255
860 NEXT I:POKE 766,0:GOTO 130
870 REM INSERT
880 IF EL=0 THEN 750
890 ? "Insert at which line";:TRAP 89
    0:INPUT A:TRAP 40000
900 A=INT(A):IF A<1 OR A>EL THEN 730
910 EL=EL+1:A=A-1
920 FOR I=EL-2 TO A STEP -1
930 IF LL(I)=0 THEN LL(I+1)=LL(I):GOT
    O 960
940 T$=L$(I*80+1,I*80+LL(I))
950 L$((I+1)*80+1,(I+1)*80+LL(I))=T$:
    LL(I+1)=LL(I)
960 NEXT I
970 L$(A*80+1,A*80+9)=" BLANK ":LL(A)
    =7
980 GOTO 130
990 REM DELETE
1000 IF EL=0 THEN 750
1010 ? "Delete From";:TRAP 1020:INPUT
      A:TRAP 40000
1020 A=INT(A):IF A<1 OR A>EL THEN 730
1022 ? "From ";A;" to";:TRAP 1022:INP
     UT B:TRAP 40000
1023 B=INT(B):IF B<1 OR B<A OR B>EL T
     HEN 730
1030 OFF=(B-A)+1:EL=EL-OFF:A=A-1:B=B-
     1
1040 FOR I=A TO EL-1
1050 T$=L$((I+OFF)*80+1,(I+OFF)*80+LL
     (I+OFF))
1060 L$(I*80+1,I*80+LL(I+OFF))=T$:LL(
     I)=LL(I+OFF)
1070 NEXT I
1080 GOTO 130
1090 REM NEW
1100 EL=0:GOTO 140
1110 REM LOAD
1120 IF EL THEN ? "Append at line ";E
     L
1130 ? :? "Do not use extension{2 UP}
     "
1140 ? "Filename";:INPUT T$:? :FN$=T$
     :FN$(LEN(FN$)+1)=".PIL"
1150 IF T$="" THEN 130
1160 Z=EL:TRAP 1180:OPEN #2,4,0,FN$
1170 INPUT #2;T$:L$(Z*80+1,Z*80+LEN(T
     $))=T$:LL(Z)=LEN(T$):Z=Z+1:GOTO
     1170
1180 EL=Z:CLOSE #2:IF PEEK(195)=136 T
     HEN ? "Load OK":GOTO 130
1190 GOSUB 390:? "I/O Error #";PEEK(1
     95):CLOSE #2:GOTO 140
1200 REM SAVE
1210 IF EL=0 THEN GOSUB 390:? "Nothin
     g to save!":GOTO 140
1220 ? :? "Do not use extension{2 UP}
     "
1230 ? "Filename";:INPUT T$:? :FN$=T$
     :FN$(LEN(FN$)+1)=".PIL"
1240 TRAP 1270:OPEN #2,8,0,FN$
1250 FOR I=0 TO EL-1:PRINT #2;L$(I*80
     +1,I*80+LL(I)):NEXT I:TRAP 40000
1260 CLOSE #2:? "Save OK":GOTO 130
1270 CLOSE #2:GOTO 1190
1280 END
1290 REM INITIALIZATION
1300 GRAPHICS 0:OPEN #1,12,0,"E:"
1310 CLOSE #6:OPEN #6,8,0,"E:":SETCOL
     OR 2,6,2:POKE 752,1
1320 ? "{Q}{35 R}{E}"
1330 ? "{|}{35 SPACES}{|}"
1340 ? "{Z}{35 R}{C}"
1350 DIM T$(80),FN$(20)
1360 T$=" Turtle PILOT Editor "
1370 LM=INT(LEN(T$)/2)
1380 FOR I=1 TO LM
1390 POSITION 19-I,1:? "|";T$(LM-I+1,
     LM+I);" |":FOR W=1 TO 10:NEXT W
1400 NEXT I
1410 MAX=100:REM MAXIMUM # OF LINES
1420 DIM LL(MAX),L$(MAX*80):REM 80 CH
     ARACTER LINES
1430 L$=" ":L$(MAX*80)=" ":L$(2)=L$:R
     EM CLEAR L$ TO BLANKS
1450 DIM IN$(80):REM INPUT
1460 ? :? :?
1470 ? "{8 SPACES}Editor Commands:"
1480 DIM CMD$(40),PILOT$(12)
1490 CMD$="ADD LISTLMODINS DEL NEW LO
     ADSAVE"
1500 POKE 85,10:PILOT$="TAMJUECRSGB*"
1510 FOR I=1 TO 8:? CMD$(I*4-3,I*4);"
      ";:IF I=4 THEN ? :POKE 85,10
1520 NEXT I:?
1530 POKE 85,10:? "DIR  PON  POFF RUN
     "
1535 POKE 85,15:? "BYE"
1540 ? :? "Acceptable Turtle PILOT co
     mmands:":?
1550 FOR I=1 TO 11:? PILOT$(I,I);": "
     ;:NEXT I:? "*":?
1560 REM FOLLOWING LINE DISABLES BREA
     K KEY.  OPTIONAL FEATURE
1570 POKE 16,64:POKE 53774,64
1580 RETURN
```

www.commodore.ca

Program 2.

```
100 REM TURTLE PILOT TRANSLATOR
110 REM FILENAME "D:PILOT.XLT"
120 REM
130 GOSUB 2250
140 REM
150 REM *** OPEN ENTER FILE ***
160 N$(LEN(N$)-2)="ENT":LIST N$,20000
    ,32767
170 OPEN #2,9,0,N$:? #2;"5 GOSUB 2000
    0"
180 REM *** MAIN LOOP ***
190 REM
200 LINE=9:FOR NUM=0 TO NL-1
210 P$=PP$(NUM*80+1,NUM*80+PL(NUM))
220 ? "--":? NUM+1;" ";P$:?
230 LINE=LINE+1:LN$=STR$(LINE*10)
240 I=0:FOR L=1 TO 12:IF P$(1,1)=I$(L
    ,L) THEN I=L:L=12
250 NEXT L:IF I=0 THEN 2520
260 GOSUB 420:GOSUB 520:GOSUB 590
270 ON I GOSUB 660,860,940,1160,1280,
    1350,1440,1480,1510,1580,2170,221
    0
280 PRINT #2;LN$:? LN$
290 NEXT NUM:IF NSTRINGS=0 THEN 350
300 ZZ=4:LN=1:FOR I=0 TO NSTRINGS-1
310 ZZ=ZZ+1:IF ZZ=5 THEN ? :? #2:? #2
    ;LN*10;" DIM ";:? :? LN*10;" DIM
    ";:LN=LN+1:ZZ=0
320 ZZ$=SNAME$(I*10+1,I*10+NAMELEN(I)
    ):? #2;ZZ$;"(20)";:? ZZ$;"(20)";
330 IF ZZ<5 AND I<NSTRINGS-1 THEN ? #
    2;",";:? ",";
340 NEXT I
350 LINE=LINE+1:? #2:? :? :? LINE*10;
    "END"
360 ? #2;LINE*10;"END":? #2;"?";CHR$(
    34);"Your translated program is i
    n memory";CHR$(34)
370 CLOSE #1:CLOSE #2
380 ? :? "To load your translated pro
    gram into":? "memory, type ENTER"
    ;CHR$(34);N$
390 NEW :REM USE "END" HERE UNTIL YOU
    'RE SURE PROGRAM WORKS, AND A COP
    Y IS SAVED
400 REM
410 REM *** SPLIT PILOT LINE AT COLON
     ***
420 FOR L=1 TO LEN(P$):IF P$(L,L)=":"
     THEN T=L:L=80
430 NEXT L
440 IF P$(1,1)="*" THEN L$="*":R$=P$:
    RETURN
450 L$=P$(1,T-1):IF T=LEN(P$) THEN R$
    ="":RETURN
460 R$=P$(T+1)
470 T$=L$:GOSUB 2470:L$=T$
480 IF L$(1,1)="G" THEN T$=R$:GOSUB 2
    470:R$=T$
490 RETURN
500 REM *** FIND CONDITIONER
510 REM
520 C=0:IF LEN(L$)<2 THEN RETURN
530 IF L$(2,2)="Y" THEN LN$(LEN(LN$)+
    1)="IF QC=1 THEN ":C=1
540 IF L$(2,2)="N" THEN LN$(LEN(LN$)+
    1)="IF QC=0 THEN ":C=2
550 RETURN
560 REM
570 REM *** FIND EXPRESSION
580 REM
590 EX$="":IF L$(LEN(L$))<>")" THEN R
    ETURN
600 T=0:FOR L=1 TO LEN(L$)-1:IF L$(L,
    L)="(" THEN T=L:L=80
610 NEXT L:EX$=L$(T+1,LEN(L$)-1):LN$(
    LEN(LN$)+1)="IF":LN$(LEN(LN$)+1)=
    EX$:LN$(LEN(LN$)+1)=" THEN"
620 RETURN
630 REM
640 REM *** T: INSTRUCTION ***
650 REM
660 LL=LEN(LN$):LN$(LL+1)="QT$=":LN$(
    LL+5)=CHR$(34)
670 IF R$="" THEN LN$(LL+6)=CHR$(34):
    LN$(LL+7)=":GOS.20040":RETURN
680 FOR L=1 TO LEN(R$):T$=R$(L,L)
690 IF T$="$" THEN 730
700 IF T$="#" THEN 790
710 LN$(LEN(LN$)+1)=T$
720 NEXT L:LN$(LEN(LN$)+1)=CHR$(34):L
    N$(LEN(LN$)+1)=":GOS.20040":RETUR
    N
730 IF L>LEN(R$)-2 THEN 710
740 T=0:FOR L1=L+2 TO LEN(R$):IF R$(L
    1,L1)="$" THEN T=L1:L1=80
750 NEXT L1:IF T=0 THEN 710
760 LL=LEN(LN$):LN$(LL+1)=CHR$(34):LN
    $(LL+2)=":QT$(LEN(QT$)+1)="
770 ZZ$=R$(L+1,T):GOSUB 2540
780 LN$(LEN(LN$)+1)=R$(L+1,T):LN$(LEN
    (LN$)+1)=":QT$(LEN(QT$)+1)=":LN$(
    LEN(LN$)+1)=CHR$(34):L=T:GOTO 720
790 IF L>LEN(R$)-2 THEN 710
800 T=0:FOR L1=L+2 TO LEN(R$):IF R$(L
    1,L1)="#" THEN T=L1:L1=80
810 NEXT L1:IF T=0 THEN 710
820 LN=LEN(LN$):LN$(LN+1)=CHR$(34):LN
    $(LN+2)=":QT$(LEN(QT$)+1)=STR$("
830 LN$(LEN(LN$)+1)=R$(L+1,T-1):LN$(L
    EN(LN$)+1)="):QT$(LEN(QT$)+1)=":L
    N$(LEN(LN$)+1)=CHR$(34):L=T:GOTO
    720
840 REM
850 REM ** A: INSTRUCTION **
860 LN$(LEN(LN$)+1)="GOSUB 20130"
870 IF R$="" THEN RETURN
880 IF R$(LEN(R$))="$" THEN ZZ$=R$:GO
    SUB 2540
890 IF R$(LEN(R$))="$" THEN LL=LEN(LN
    $):LN$(LL+1)=":":LN$(LL+2)=R$:LN$
    (LEN(LN$)+1)="=QI$":RETURN
900 LN$(LEN(LN$)+1)=":":LN$(LEN(LN$)+
    1)=R$:LN$(LEN(LN$)+1)="=VAL(QI$)"
    :RETURN
910 REM
920 REM *** M: INSTRUCTION ***
930 REM
940 FOR L=0 TO 25:ML(L)=0:NEXT L:IF R
    $="" THEN 2520
950 T=0:FOR L=1 TO LEN(R$):IF R$(L,L)
    <>"," THEN ML(T)=ML(T)+1:M$(T*20+
    ML(T))=R$(L,L):GOTO 970
960 T=T+1
970 NEXT L
990 FOR L=1 TO T+1:LN$="":? #2;LINE*1
```

www.commodore.ca

```
   0+L-1;:? LINE*10+L-1;
1000 IF C=1 THEN ? #2;"IF QC=1 THEN "
     ;:? "IF QC=1 THEN ";
1010 IF C=2 THEN ? #2;"IF QC=0 THEN "
     ;:? "IF QC=0 THEN ";
1020 IF EX$<>"" THEN ? #2;"IF ";EX$;"
      THEN ";:? "IF ";EX$;" THEN ";
1030 LN$(LEN(LN$)+1)="Q$(":LN$(LEN(LN
     $)+1)=STR$((L-1)*20+1)
1040 LN$(LEN(LN$)+1)=")=":ZZ$=" ":ZZ$
     (2)=M$((L-1)*20+1,(L-1)*20+ML(L-
     1))
1050 IF ZZ$(LEN(ZZ$))="$" THEN ZZ$=ZZ
     $(2):GOSUB 2540:GOTO 1070
1060 ZZ$(1,1)=CHR$(34):ZZ$(LEN(ZZ$)+1
     )=CHR$(34)
1070 LN$(LEN(LN$)+1)=ZZ$:LN$(LEN(LN$)
     +1)=":"
1080 LN$(LEN(LN$)+1)="QL(":LN$(LEN(LN
     $)+1)=STR$(L-1):LN$(LEN(LN$)+1)=
     ")="
1090 IF ZZ$(LEN(ZZ$))=CHR$(34) THEN L
     N$(LEN(LN$)+1)=STR$(LEN(ZZ$)-2):
     GOTO 1110
1100 LN$(LEN(LN$)+1)="LEN(":LN$(LEN(L
     N$)+1)=ZZ$:LN$(LEN(LN$)+1)=")"
1110 ? #2;LN$;:? LN$;:IF L<T+1 THEN ?
      #2:?
1120 NEXT L:? #2;":GOSUB 20140":? ":G
     OSUB 20140":POP :GOTO 290
1130 REM
1140 REM *** J: INSTRUCTION ***
1150 REM
1160 IF R$="" THEN 1210
1170 IF R$(1,1)<>"*" THEN ZZ$=R$:R$="
     *":R$(2)=ZZ$
1180 T=0:FOR L=0 TO NL-1:IF PP$(L*80+
     1,L*80+PL(L))=R$ THEN T=L+1:L=25
     00
1190 NEXT L:IF T=0 THEN 2520
1200 LN$(LEN(LN$)+1)="GOTO":LN$(LEN(L
     N$)+1)=STR$(T*10+100):RETURN
1210 T=0:FOR L=NL TO 0 STEP -1:ZZ$=PP
     $((L-1)*80+1,(L-1)*80+1)
1220 IF ZZ$="A" THEN T=L:L=0
1230 NEXT L:IF T=0 THEN 2520
1240 GOTO 1200
1250 REM
1260 REM *** U: INSTRUCTION ***
1270 REM
1280 LN$(LEN(LN$)+1)="QU=QU+1":IF R$(
     1,1)<>"*" THEN ZZ$=R$:R$="*":R$(
     2)=ZZ$
1290 T=0:FOR L=0 TO NL-1:IF PP$(L*80+
     1,L*80+PL(L))=R$ THEN T=L:L=2500
1300 NEXT L:IF T=0 THEN 2520
1310 LN$(LEN(LN$)+1)=":GOS.":LN$(LEN(
     LN$)+1)=STR$(T*10+100):RETURN
1320 REM
1330 REM *** E: INSTRUCTION ***
1340 REM
1350 LN$(LEN(LN$)+1)="IF QU=0 THEN EN
     D"
1360 PRINT #2;LINE*10+5;:? LINE*10+5;
1370 IF C=1 THEN ? #2;"IF QC=1 THEN "
     ;:? "IF QC=1 THEN ";
1380 IF C=2 THEN ? #2;"IF QC=0 THEN "
     ;:? "IF QC=0 THEN ";
1390 IF EX$<>"" THEN ? #2;"IF ";EX$;"
      THEN ";:? "IF ";EX$;" THEN ";
1400 ? #2;"QU=QU-1:RET.":? "QU=QU-1:R
     ET.":RETURN
1410 REM
1420 REM *** C: INSTRUCTION ***
1430 REM
1440 LN$(LEN(LN$)+1)=R$:RETURN
1450 REM
1460 REM *** R: INSTRUCTION ***
1470 REM
1480 RETURN
1490 REM
1500 REM *** S: INSTRUCTION ***
1510 LN$(LEN(LN$)+1)="SO.0,"
1520 FOR L=1 TO LEN(R$):IF R$(L,L)=",
     " THEN T=L:L=LEN(R$)
1530 NEXT L:LN$(LEN(LN$)+1)=R$(1,T-1)
     :LN$(LEN(LN$)+1)=",10,8:FOR QW=1
      TO "
1540 LN$(LEN(LN$)+1)=R$(T+1):LN$(LEN(
     LN$)+1)=":NEXT QW:SO.0,0,0,0"
1550 RETURN
1560 REM *** G: INSTRUCTION ***
1570 REM
1580 IF R$="" THEN LN$(LEN(LN$)+1)="G
     R.7+32":RETURN
1590 F=0:IF ASC(R$)<48 OR ASC(R$)>57
     THEN 1620
1600 LN$(LEN(LN$)+1)="FOR Q1=1 TO ":L
     N$(LEN(LN$)+1)=STR$(VAL(R$))
1610 LN$(LEN(LN$)+1)=":":F=1:R$=R$(1,
     LEN(R$)-1):R$=R$(LEN(STR$(VAL(R$
     )))+2)
1620 REM FIND INDIVIDUAL COMMANDS
1630 FOR L=0 TO 6:GL(L)=0:NEXT L
1640 T=1:Z=1:FOR L=1 TO LEN(R$)
1650 IF R$(L,L)<>";" THEN GL$((T-1)*8
     0+Z)=R$(L,L):GL(T-1)=Z:Z=Z+1:GOT
     O 1670
1660 T=T+1:Z=1
1670 NEXT L:NN=T
1680 REM TRANSLATE EACH COMMAND
1690 FOR L=1 TO T
1700 GC=0:FOR L1=1 TO 11:IF G(L1-1)>G
     L(L-1) THEN 1740
1710 IF GL$((L-1)*80+1,(L-1)*80+G(L1-
     1))<>G$(L1*6-5,(L1-1)*6+G(L1-1))
      THEN 1740
1720 GC=L1:L1=11:IF GL(L-1)=G(GC-1)+1
      THEN GL(L-1)=0:GOTO 1740
1730 GL$(L*80-79,L*80)=GL$(L*80-79+G(
     GC-1)+1):GL(L-1)=GL(L-1)-G(GC-1)
     -1
1740 NEXT L1
1750 IF GC=0 THEN 2520
1755 ZZ$="":IF GL(L-1) THEN ZZ$=GL$(L
     *80-79,(L-1)*80+GL(L-1))
1760 ON GC GOSUB 1810,1830,1850,1870,
     1890,1980,2010,2060,2080,2100,21
     20
1770 IF L<NN THEN LN$(LEN(LN$)+1)=":"
1780 NEXT L:IF F=1 THEN LN$(LEN(LN$)+
     1)=":NEXT Q1"
1790 RETURN
1800 REM CLEAR COMMAND
1810 LN$(LEN(LN$)+1)="GR.7":RETURN
1820 REM TURNTO COMMAND
1830 LN$(LEN(LN$)+1)="QA=90-":LN$(LEN
     (LN$)+1)=ZZ$:RETURN
1840 REM TURN COMMAND
1850 LN$(LEN(LN$)+1)="QT=":LN$(LEN(LN
```

www.commodore.ca

```
$)+1)=ZZ$:LN$(LEN(LN$)+1)=":GOS.
     20190":RETURN
1860 REM DRAW COMMAND
1870 LN$(LEN(LN$)+1)="QL=":LN$(LEN(LN
     $)+1)=ZZ$:LN$(LEN(LN$)+1)=":GOS.
     20220":RETURN
1880 REM PEN COMMAND
1890 IF ZZ$="UP" THEN LN$(LEN(LN$)+1)
     ="QP=1":RETURN
1900 IF ZZ$="DOWN" THEN LN$(LEN(LN$)+
     1)="QP=0":RETURN
1910 TT=1:LN$(LEN(LN$)+1)="COLOR "
1920 IF ZZ$="ERASE" THEN TT=0
1930 IF ZZ$="RED" THEN TT=1
1940 IF ZZ$="GREEN" THEN TT=2
1950 IF ZZ$="BLUE" THEN TT=3
1960 LN$(LEN(LN$)+1)=STR$(TT):RETURN
1970 REM SCREEN COMMAND
1980 TT=0:IF ZZ$="APPLE" THEN TT=1
1985 LN$(LEN(LN$)+1)="QSCR=":LN$(LEN(
     LN$)+1)=STR$(TT)
1990 RETURN
2000 REM GOTO COMMAND
2010 FOR L1=1 TO GL(L-1)
2020 IF ZZ$(L1,L1)="," THEN T=L1:L1=2
     55
2030 NEXT L1:LN$(LEN(LN$)+1)="QX=":LN
     $(LEN(LN$)+1)=ZZ$(1,T-1)
2040 LN$(LEN(LN$)+1)=":QY=":LN$(LEN(L
     N$)+1)=ZZ$(T+1):RETURN
2050 REM FULL COMMAND
2060 LN$(LEN(LN$)+1)="GR.7+16+32":RET
     URN
2070 REM MIX COMMAND
2080 LN$(LEN(LN$)+1)="GR.7+32":RETURN

2090 REM QUIT COMMAND
2100 LN$(LEN(LN$)+1)="GR.0":RETURN
2110 REM GO COMMAND
2120 LN$(LEN(LN$)+1)="QP=1:QL=":LN$(L
     EN(LN$)+1)=ZZ$
2130 LN$(LEN(LN$)+1)=":GOS.20220:QP=0
     ":RETURN
2140 REM
2150 REM *** B: COMMAND ***
2160 REM
2170 LN$(LEN(LN$)+1)=R$:RETURN
2180 REM
2190 REM *** LABEL
2200 REM
2210 LN$(LEN(LN$)+1)="REM ":LN$(LEN(L
     N$)+1)=R$:RETURN
2220 REM
2230 REM *** INITIALIZE
2240 REM
2250 GRAPHICS 0:DETRAP=40000
2260 ? "{Q}{6 W}{23 R}{6 W}{E}"
2270 ? "{A}{5 S}{D}TURTLE PILOT 
     TRANSLATOR{A}{5 S}{D}"
2280 ? "{Z}{6 X}{23 R}{6 X}{C}"
2290 ? :?
2300 DIM N$(14),T$(80),LN$(255),P$(80
     ),SNAME$(10*50),NAMELEN(50):REM
     UP TO 50 STRINGS IN PROGRAM
2310 ? "What is the name of the PILOT
     ":? "program? (Do not use extens
     ion)"
2320 INPUT N$:IF N$="" THEN ? "{UP}
     {DEL LINE}";:GOTO 2320
2330 IF LEN(N$)>1 THEN IF N$(2,2)<>":
     " THEN T$="D:":T$(3)=N$:N$=T$
2340 TRAP 2530:N$(LEN(N$)+1)=".PIL"
2350 OPEN #1,4,0,N$:TRAP DETRAP
2360 DIM I$(12):I$="TAMJUECRSGB*"
2370 DIM G$(6*11),G(11),ZZ$(80),GL$(1
     0*80),GL(10)
2380 FOR I=0 TO 10:READ ZZ$:G$(I*6+1,
     I*6+LEN(ZZ$))=ZZ$:G(I)=LEN(ZZ$)-
     1:NEXT I
2390 DATA CLEAR,TURNTO,TURN,DRAW,PEN,
     SCREEN,GOTO,FULL,MIX,QUIT,GO
2400 DIM R$(80),EX$(20),L$(80),M$(25*
     20),ML(25)
2410 ? "Now reading PILOT program..."
     :?
2420 MAX=100:DIM PP$(MAX*80),PL(MAX)
2430 TRAP 2450:INPUT #1;P$:TRAP DETRA
     P
2440 PP$(NL*80+1)=P$:? P$:PL(NL)=LEN(
     P$):NL=NL+1:GOTO 2430
2450 IF PEEK(195)<>136 THEN 2530
2460 ? :CLOSE #1:RETURN
2470 IF T$="" THEN RETURN
2480 ZZ$="":FOR L=1 TO LEN(T$)
2490 IF T$(L,L)<>" " THEN ZZ$(LEN(ZZ$
     )+1)=T$(L,L)
2500 NEXT L:T$=ZZ$:RETURN
2510 RETURN
2520 CLOSE #1:CLOSE #2:? "ERROR IN PI
     LOT LINE #";NL;CHR$(253):END
2530 CLOSE #1:PRINT "UNABLE TO LOAD "
     ;N$;CHR$(253):END
2540 REM SEARCH "STRING NAME TABLE" F
     OR ZZ$, ADD IT IF NOT PRESENT
2550 IF NSTRINGS=0 THEN 2600
2560 ZZ=0:FOR I=0 TO NSTRINGS-1
2570 IF ZZ$=SNAME$(I*10+1,I*10+NAMELE
     N(I)) THEN ZZ=I+1:I=NSTRINGS
2580 NEXT I:IF ZZ THEN RETURN
2600 SNAME$(NSTRINGS*10+1,NSTRINGS*10
     +LEN(ZZ$))=ZZ$:NAMELEN(NSTRINGS)
     =LEN(ZZ$)
2610 NSTRINGS=NSTRINGS+1
2620 RETURN
2630 REM THE FOLLOWING LINES ARE NOT
     PART OF THE TRANSLATOR, BUT ARE
2640 REM INCLUDED IN EVERY TRANSLATED
      PROGRAM
20000 DIM Q$(25*20),QL(25),QS(31),QI$
      (80),QT$(80):OPEN #4,12,0,"E:"
20010 COLOR 1:QX=0:QY=0:QC=-1:QR=40:Q
      A=90:QQ=3.1415927/180
20020 FOR Q1=0 TO 25:QL(Q1)=0:NEXT Q1
20030 RETURN
20040 IF QT$="" THEN PRINT :RETURN
20045 QTAB=85+572*(PEEK(87)<>0)
20050 QT=0:IF QT$(LEN(QT$))="&" THEN
      QT$=QT$(1,LEN(QT$)-1):QT=1
20060 FOR Q1=1 TO LEN(QT$):IF QT$(Q1,
      Q1)=" " AND PEEK(QTAB)>QR-9 THE
      N GOSUB 20090
20070 PRINT QT$(Q1,Q1);:NEXT Q1:IF QT
      =0 THEN PRINT
20080 RETURN
20090 QF=0:FOR Q2=Q1+1 TO Q1+QR-PEEK(
      QTAB)-1:IF Q2>=LEN(QT$) THEN Q2
      =1000:QF=1:GOTO 20110
20100 IF QT$(Q2,Q2)=" " THEN Q2=1000:
      QF=1
20110 NEXT Q2:IF QF=0 THEN PRINT :Q1=
```

```
      Q1+1
20120 RETURN
20130 INPUT #4;QI$:RETURN
20140 QM=0:QC=0:FOR Q1=1 TO 25:IF LEN
      (QI$)<QL(Q1-1) OR QL(Q1-1)=0 TH
      EN 20180
20150 FOR Q2=1 TO LEN(QI$)-QL(Q1-1)
20160 IF Q$((Q1-1)*20+1,(Q1-1)*20+QL(
      Q1-1))=QI$(Q2,Q2+QL(Q1-1)-1) TH
      EN QC=1:QM=Q1:Q1=25:Q2=300
20170 NEXT Q2
20180 NEXT Q1:FOR Q1=0 TO 25:QL(Q1)=0
      :NEXT Q1:RETURN
20190 QA=QA-QT:IF QA>360 THEN QA=QA-3
      60
20200 IF QA<0 THEN QA=QA+360
20210 RETURN
20220 IF PEEK(87)<>7 THEN GRAPHICS 7
20221 QS=1:IF QSCR THEN QS=0.576:REM
      SCALE FOR APPLE
20225 IF QP=1 THEN 20250
20230 TRAP 20250
20240 PLOT QX*QS+79,48-QY*QS:TRAP 400
      00
20250 QX=QX+QL*COS(QA*QQ):QY=QY+QL*SI
      N(QA*QQ)
20260 IF QP=1 THEN RETURN
20270 TRAP 20290
20280 DRAWTO QX*QS+79,48-QY*QS:TRAP 4
      0000
20290 RETURN
```

©

Use the card in the back of this magazine to order your **COMPUTE! Books**.

OSI C1P

34 programs on tape for $29.95 *less than $1 each!* Includes many arcade style games. Program summaries available for $1.00. Add $1.50 Post & Hand. for tape.

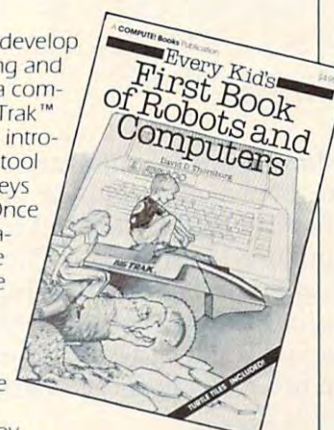
VICTORY SOFTWARE, INC.®

2027-A S.J. RUSSELL CIRCLE, ELKINS PARK, PA 19117
(215) 576-5625

An Intriguing New Release from **COMPUTE! Books:**

Every Kid's First Book Of Robots And Computers

By David Thornburg



From the author's preface:

"This book allows children to develop skills in computer programming and geometry through the use of a commonly available toy – the Big Trak™ robot vehicle. Programming is introduced as the communication tool through which the child conveys instructions to the machine. Once the machine's language limitations are understood, it can be made to follow any procedure which has been entered by the user.

"Our use of turtle commands as the programming language mirrors the process-based descriptions commonly used by children. For example, a child is likely to describe a nearby location, such as a friend's house, by a procedure (Go two blocks, turn right, go another block, turn left,...). Because turtle geometry has been incorporated as the graphics environment in several computer languages available for the popular desk-top computers, these programming ideas can continue to be used as the child learns to operate other computers."

In *Every Kid's First Book Of Robots And Computers,* author David Thornburg conveys a uniquely exciting learning experience for children, parents, and teachers. The book uses Big Trak, PILOT/LOGO type languages, and Turtle Tiles™ to explore the concepts and techniques of robot/computer programming. Turtle Tiles, included with every book, are designed to provide hands-on programming experience to children without access to a Big Trak or a personal computer. Additionally, the Tiles can be used in conjunction with either of these items to share and reinforce the exercises in the book.

Ask for
Every Kid's First Book Of Robots And Computers
at your computer retailer, local bookstore,
or order directly from:

COMPUTE! Books
P.O. Box 5406
Greensboro, NC
27403

For Fastest Service,
Call Toll Free
800-334-0868
In NC 919-275-9809

$4.95 plus $1.00 shipping and handling.
ISBN 0-942386-05-1. Perfect bound, 96 pages plus Turtle Tiles™. Fully illustrated.

Dealer and educator quantity discounts are available.

Big Trak is a trademark of the Milton Bradley Company.
Turtle Tiles are a trademark of David D. Thornburg and Innovision, Inc.

www.commodore.ca

This VIC game demonstrates how the motor-impaired can communicate in several ways with a computer – using only the button on a joystick. There are also suggestions on adapting the game for the blind, to other computers, and for use with other kinds of input devices.

A Bi-monthly Column

Micros With The Handicapped

Susan Semancik and C. Marshall Curtis
The Delmarva Computer Club
Wallops Island, VA

Many kinds of computer entertainment require keyboard interaction or other motor coordination that can be difficult and even impossible for motor-impaired individuals. This month we'll digress from our series on developing a daily communications program to explore how a game program can be modified to accept alternative input devices in order to allow the motor-impaired to interact with it.

In this "Color Master" game, the computer randomly fills a four-block pattern, choosing from seven colors, with repeats possible. The user tries to duplicate the hidden color pattern within ten guesses. The user's guesses are usually given by typing a letter or number for each color in the guess. To make this game more accessible to the motor-impaired, a menu of choices the user can make will be presented on the computer's screen, with a moving vertical arrow indicating the current menu choice which will be made if an input device is activated.

Figure 1 shows a typical layout for this game on the VIC computer's screen. The blocks for the user's ten possible guesses appear in the middle of the screen and are initially set to all white. A horizontal arrow will point to the current guess on which the user is working. The menu of choices appears at the bottom of the screen and includes blocks in each of the seven possible colors, movement left or right within the four blocks of the current guess, speeding up (+) or slowing down (-) of the menu's arrow, and requesting the computer to score the user's current guess. An advantage of a menu-driven game is that no written directions are needed to see what options are available during the game, since the choices are always visible in the menu.

The computer will score a guess in two ways, which are described at the top of the screen. Under the P-score, the computer will tell how many of the colors in the user's guess are correct colors in the right positions. The C-score will tell how many other colors are correct, but in the wrong positions. When the two scores add up to four, the user will have identified all of the colors in the hidden pattern. If the P-score is exactly four, then their positions are correct as well, and the game ends. Each time a P-score occurs, a whistle sound will be heard as an audible reward.

When the game is over, at the top of the screen will appear a score, which is inversely related to the number of guesses used to successfully duplicate the hidden pattern. The score ranges from a low score of zero, if not guessed within ten tries, to a top score of ten, if guessed in one try. The user's best score is also recorded at the top of the screen.

When the program is run, a horizontal arrow will point to the first row of white blocks which the user will be filling with his/her first guess. The first block of this row will be flashing to indicate that the user's response will be with respect to this block. A moving vertical arrow will point in turn to each of the possible responses the user can make from the menu. Program 1 assumes a joystick is attached to the VIC, and that the user will push the joystick button to indicate a response.

The program can be changed to permit the use of other means of input that may be more suitable to the needs of a particular handicap. Even sounds could be used in place of the colors so that a blind person could also participate, though fewer choices and a review option might be needed in this case as well. Lines 1040, 1080, and 241 need to be changed so that any activity on the joystick will indicate a user's response. (However, joystick movement to the right will *not* be picked up by this routine. Change the =158 in lines 1040 and 1080 to <>190, and change line 241 to =190 instead of <>158.) To change it so that any activity on a device attached to the user port will indicate a user's response, change the 37139 in line 9 to 37138. Also change the =158 in lines 1040 and 1080 to <>255, and the <>158 in line 241 becomes =255. Table 1 contains a description of the program's variables so that the program's logic will be easier to follow.

Try converting some of your favorite games to a menu-driven approach for alternative input. In future columns, the rest or our series on developing a communications program will provide additional techniques in this area.

The Delmarva Computer Club
P.O. Box 36
Wallops Island, VA 23337

www.commodore.ca

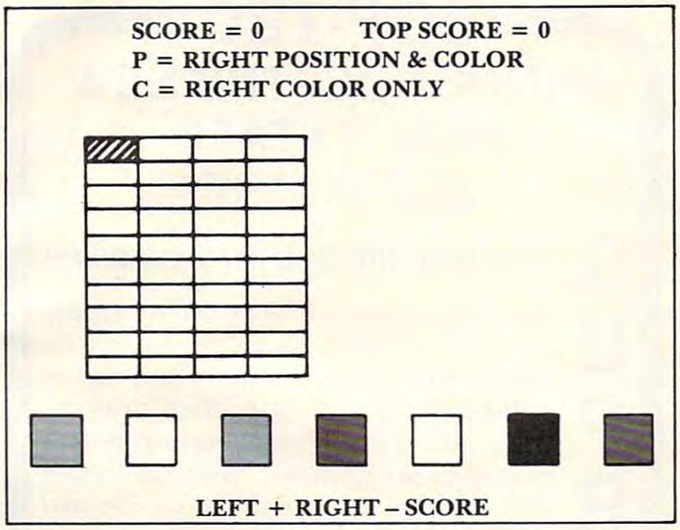


Figure 1.

Table 1.

Line 4 DE controls the time delay for the arrow moving through the menu. Note: After a selection is made, the arrow pauses again at that selection, thereby allowing immediate multiple selections of a menu item.

Line 5 C() contains the color codes for the menu blocks.

Lines 9, 10 V1, V2, and PL control spacing and positioning within the menu for the vertical arrow.
37139 sets the data direction register for the joystick.
36879 sets the screen to a cyan border with a black background.
36878 sets the volume for sound.

Line 20 Draws ten rows, each with four white blocks.

Line 22 H() contains the computer's hidden color pattern.

Lines 24,26 Print the menu.

Line 30 At the start of the game, a horizontal arrow points at the first row of white blocks. L contains the screen line for the current guess row of blocks.

Line 40 P indicates which block in the current guess row will flash. G() contains the user's guess, which is preset to all white.

Line 50 B contains the color map location for the screen location of the flashing block; S is the screen location of the flashing block.

Line 75 Looks for a SCORE response.

Line 80 On a LEFT response, moves the flashing to the next block on the left in the current guess row, as long as it isn't already the leftmost block.

Line 90 On a RIGHT response, moves the flashing to the next block on the right in the current guess row, as long as it isn't already the rightmost one.

Lines 100-110 Set the flashing block in the current guess row to the selected color.

Line 120 Indicates the next block to the right will be the next block to flash, if not already at the rightmost block in the current guess row.

Line 140 K() is a copy of the hidden pattern and will be used for scoring.

Line 150-170 Y counts the P-score, which is the number of matches in both position and color.

Lines 171-175 Whistle sound for each P-score.

Lines 180-190 B counts the C-score, which is the number of matches only in color.

Line 195 Checks for end of game by a correct guess.

Line 200 Checks for end of game by running out of guesses.

Line 210 Points to the next guess row.

Lines 220-230 Reveal the hidden pattern.

Lines 222-229 Update the score and top score.

Lines 240-242 Wait for the user to respond before starting a new game.

Line 1000 V contains the current menu item number. VL contains the screen location of the current menu item.

Line 1020 Positions the vertical menu arrow.

Line 1030-1060 Flash the current block in the guess row and delay the arrow at the current menu item.

Line 1040 Looks for a user's response on the joystick button.

Line 1080 Waits for the user to release the joystick button in order to eliminate a "keyboard-bounce" type problem.

Lines 1089-1110 Code menu item's function with respect to the arrow's position.

Color Master

```
4 DE=17
5 X=RND(-TI):C(2)=5:C(3)=28:C(4)=159:C(5)=15
    6:C(6)=30:C(7)=31:C(8)=158
9 POKE37139,0:V1=8100:V2=8166:VL=V1:PL=3
10 POKE36879,11:POKE36878,15:PRINT"{CLEAR}{GR
    GRN}SCORE=";D;"{YEL}TOP{RIGHT}SCORE="
    ;E;
15 PRINT"{CYN}P{WHT}=RIGHT POSITION &       CO
    LOR"
16 PRINT"{BLU}C{WHT}=RIGHT COLOR ONLY"
18 PRINT"{REV}{CYN}                       ";
20 PRINT"{WHT}";:FORL=1TO10:PRINT"{RIGHT}";:F
    ORC=1TO4:PRINT"{REV}L:";:NEXTC:PRINT:
    NEXTL:PRINT"{03 DOWN}  ";
22 FORI=1TO4:H(I)=INT(7*RND(1)+2):NEXTI
24 FORI=2TO8:PRINT"{REV}";CHR$(C(I));" ";"{02
     RIGHT}";:NEXTI
26 PRINT:PRINT:PRINT" LEFT {GRN}+ {CYN}RIGHT ~
    {GRN}- {PUR}SCORE{WHT}"
30 PRINT"{HOME}{05 DOWN}{10 RIGHT}_ ";:L=5
40 P=1:FORI=1TO4:G(I)=2:NEXTI
50 B=38400+L*22+1
```

www.commodore.ca

```
53 T=128
55 C=B+(P-1)*2:T=-T:S=C-30720
58 GOSUB1000:V=V-1
75 IFASC(A$)=13THEN140
80 IFA$="{LEFT}"ANDP>1THENPOKES,204:POKES+1,2
   50:S=S-2:P=P-1:GOTO50
90 IFA$="{RIGHT}"ANDP<4THENPOKES,204:POKES+1,
   250:S=S+2:P=P+1:GOTO50
95 IFA$<"2"ORA$>"8"THEN55
100 A=ASC(A$)-48
105 POKES,204:POKES+1,250
110 G(P)=A:POKEC,A-1:POKEC+1,A-1
120 P=P+1:IFP>4THENP=P-1
130 GOTO53
140 FORI=1TO4:K(I)=H(I):NEXTI:POKES,204:POKES+
    1,250
150 PRINT"{CYN}P{WHT}=";:Y=0
160 FORI=1TO4:IFG(I)=K(I)THENY=Y+1:K(I)=0:G(I)
    =9
170 NEXTI:PRINTY;"{BLU}C{WHT}=";:B=0
171 IFY=0THEN180
172 FORJJ=1TOY
173 POKE36878,15:FORLL=148TO220STEP2:POKE36876
    ,LL:NEXTLL
174 FORLL=128TO200STEP2:POKE36876,LL:NEXTLL:FO
    RLL=200TO128STEP-2:POKE36876,LL:NEXTL
    L
175 POKE36876,0:POKE36876,0:FORLL=1TO50:NEXTLL
    :NEXTJJ
180 FORI=1TO4:FORJ=1TO4:IFG(I)=K(J)THENB=B+1:G
    (I)=9:K(J)=0
190 NEXTJ,I:PRINTB;
195 IFY=4THENFORI=14TOL+1STEP-1:PRINT:NEXTI:PR
    INT"CORRECT. THE ANSWER IS";:GOTO222
200 L=L+1:IFL=15THEN220
210 PRINT"{10 RIGHT}_ ";:GOTO40
220 PRINT"THE ANSWER IS"
222 D=15-L
224 F=(D=10):POKE7686,32:IFFTHENPOKE7686,49:PO
    KE7687,48:GOTO227
225 POKE7687,48+D
227 IFD<=ETHEN230
228 E=D:G=(E=10):POKE7699,32:IFGTHENPOKE7699,4
    9:POKE7700,48:GOTO230
229 POKE7700,48+E
230 PRINT" {REV}";:FORI=1TO4:PRINTCHR$(C(H(I))
    );"L:";:NEXTI:PRINT"{WHT}"
240 PRINT"RESPOND TO CONTINUE.";
241 IFPEEK(37137)<>158THEN241
242 GOTO5
1000 V=V+1:IFV=8THENVL=V2:PL=4
1010 IFV>12THENVL=V1:V=1:PL=3
1020 POKEVL,30:POKEVL+30720,1
1030 J=1
1032 K=PEEK(S):M=PEEK(S+1)
1034 POKES,K+T:POKES+1,M+T
1040 IFPEEK(37137)=158THENPOKEVL+30720,0:GOTO10
     80
1042 T=-T
1050 J=J+1
1060 IFJ<DETHEN1032
1070 POKEVL+30720,0:VL=VL+PL:GOTO1000
1080 IFPEEK(37137)=158THEN1080
1089 IFV<8THENA$=MID$(STR$(V+1),2):RETURN
1090 IFV=8THENA$="{LEFT}":RETURN
1095 IFV=9THENDE=DE-.25*DE:A$="A":RETURN
1100 IFV=10THENA$="{RIGHT}":RETURN
1105 IFV=11THENDE=DE+.25*DE:A$="A":RETURN
1110 IFV=12THENA$=CHR$(13):RETURN
1120 END
```

©

SOF-TEC
ANNOUNCES IT'S NEW REAL ESTATE SOFTWARE

SUPR-AN

The ultimate in User Flexibility!

Control in sensitivity analysis, Financing Terms (Any Combination), Investor Orientation. Use your own identifiers for income and expense categories, property titles, addresses, etc. Measure before and after-tax benefits, both for Future and Present Value Estimates. For any property.

Required PET/CBM 32K (specify Disk or Cassette, with or without printer)

***NOW:* $79.95** (Add 4% in Georgia)

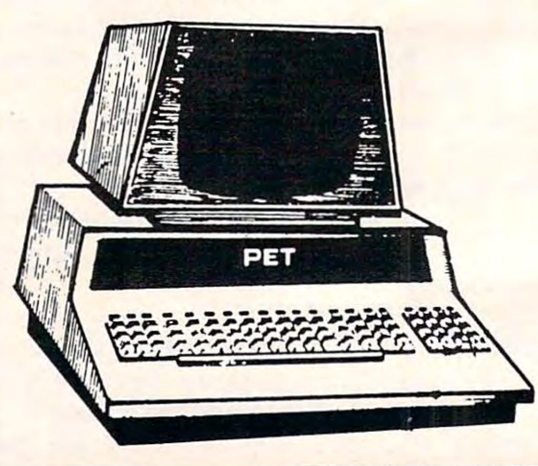
ORDER NOW

OTHER SOFTWARE AVAILABLE:

Real Estate:	Subdivision Analysis	- $79.95
	FHA 221-d(4) Analysis	- $19.95
	ELLWOOD Tables	- $14.95
Other:	Retail/Banking Market Share Analysis	- $149.95
	Merchandise Line Purchasing Power Analysis for Geographical Area	- (Call)
	Demographical Data any Area	- (Call)

ORDER NOW!!

SOF-TEC • 1043 First Ave., Columbus, Ga 31901
(404) 327-1221



www.commodore.ca

Review:

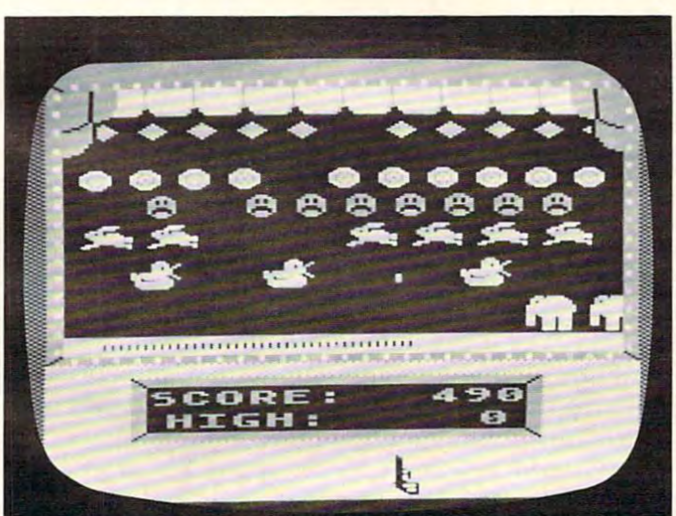
Four Atari Games

Charles Brannon
Editorial Assistant

As software developers discover and exploit more and more of the Atari's features, the games become more colorful, dazzling, and exciting.

DataSoft's four new games, *Canyon Climber*, *Pacific Coast Highway*, *Clowns And Balloons*, and *Shooting Arcade*, are of this type. All these games show off the graphics and animation capabilities of the Atari. New graphics techniques are used to allow fine scrolling of multicolor playfield objects at varying speeds (a feat normally impossible, but it looks like it's done here with DLI's and the four-color character modes 4 and 5).

Shooting Arcade is a most attractive game, with a display just like the carnival game. Bouncing, quacking, twisting, and flashing targets invite you to shoot, but you had better be accurate – you have a limited supply of bullets. Run out of ammunition and the game is over. If you clear the screen, you



Shooting Arcade

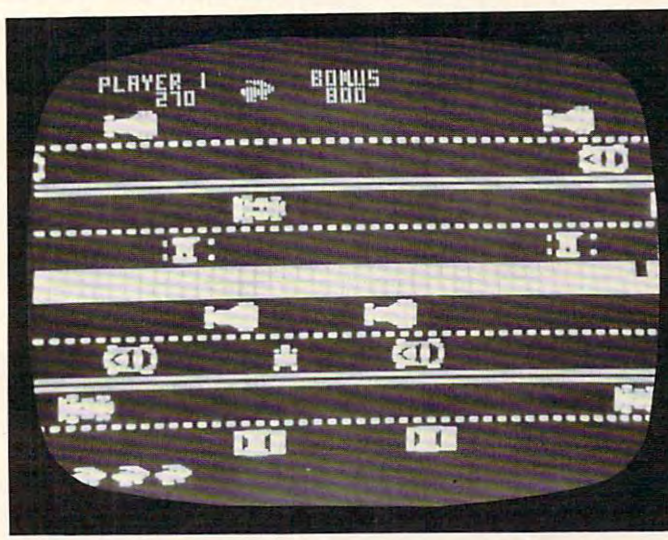
can shoot a cagey bear for bonus points, and play again against a faster set of targets.

There is a row of faces that alternate between happy and sad. Hit a sad face and you get another bunny to shoot. The music, color, and smoothness are sure to make the game popular.

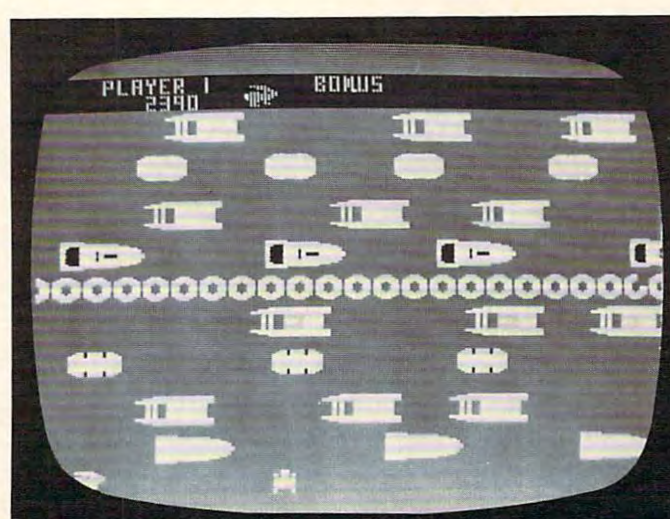
Rush Hour At 1.79 MHz

DataSoft is sure to score a hit with its *Pacific Coast Highway*. As either a turtle or a rabbit, you must try to cross a busy California freeway as you try to make it to the beach. Once there, you must hop (yes, the turtle can hop!) from surfboard to surfboard in search of the ultimate goal, bonus points.

The game is divided into two screens, a highway and a water scene. Each successful crossing makes the game more difficult. In the two-player game, the classic contest of turtle vs. rabbit is re-enacted.



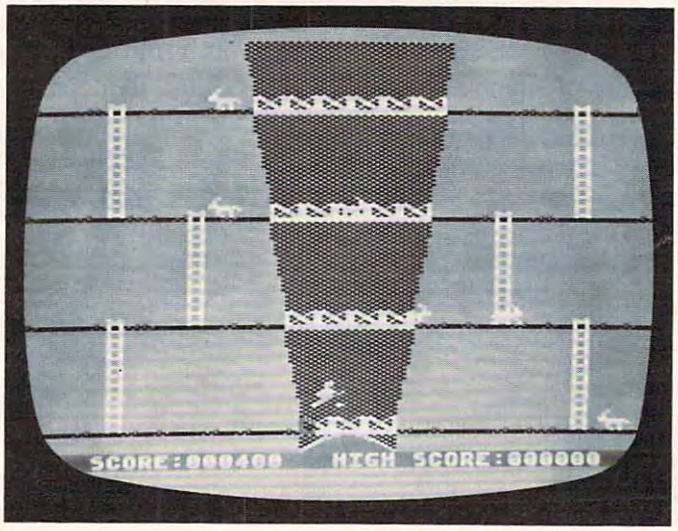
Pacific Coast Highway



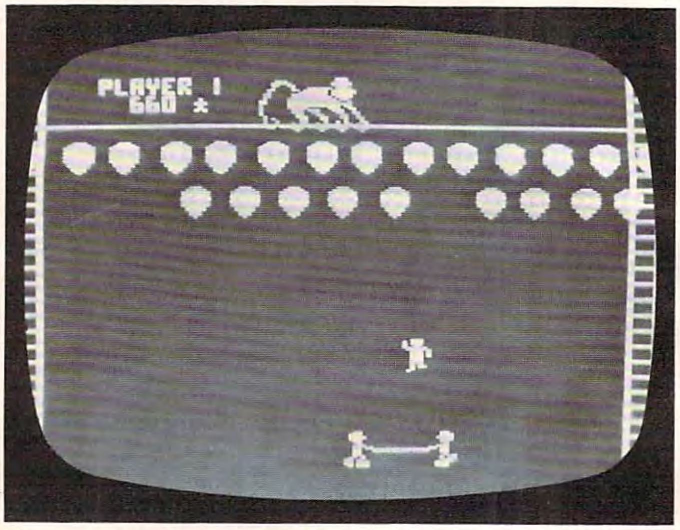
Pacific Coast Highway

A frustrating aspect of the game is that if one player gets hit (or takes a plunge), both players have to start over.

www.commodore.ca



Canyon Climber



Clowns and Balloons

The animation is in "strips" of screen (a by-product of the graphics technique), but it suits this game very well. If you didn't know better, you'd think the Atari could control dozens of multicolor "sprites," as cars whiz and hum, and boats drift lazily along. The police car even flashes its lights!

Canyon Climber is similar to *Donkey Kong,* in the way that *Pac-Man* is like *Head-On,* or *Galaxians* is like *Space Invaders. Canyon Climber* is a "theme" game, where a little man you control with a joystick can run, jump, climb, wield a shield, blow up bridges, leap over obstacles in a single bound, or even fly! It's not an adventure game. These possibilities are just integrated into the game, as hitting barrels with a hammer is part of *Donkey Kong.*

Canyon Climber is really several games in one. The first level involves setting charges on various bridges, and then detonating them. You must evade malevolent goats that are determined to butt you to the bottom of the Grand Canyon. Assuming success, you advance to the second screen, where you dodge Indian arrows (or grab a shield that temporarily deflects them) as you wend your way to the top.

If you make it this far (don't count on it), you soar into the air as you progress to the third screen. Here you leap from rock to rock as you attempt to scale the canyon walls. Meanwhile, pesky (and surprisingly intelligent) birds overhead rain down rocks upon your weary head. If you can withstand this final barrage, you find yourself standing triumphant at the top of the Grand Canyon. The attainment of your goal is rewarded by a charging goat who knocks you all the way back to the first screen.

To fit a large-scale screen onto a standard TV set is quite a challenge. DataSoft uses small playing characters, but detailed settings. The graphics are less elaborate (and the colors are mostly Arizona dun and orange) than the other DataSoft games reviewed, but the overall animation and execution are perhaps the best of the four.

Shenanigans At The Circus

A seemingly simple game, *Clowns and Balloons* involves maneuvering a clown-driven trampoline across the bottom of the screen with either a joystick or paddle controller. A third clown climbs a ladder and leaps out to seeming doom. Ah, but that's your job, to save the clown, and what's more, bounce him to the top of the big top! Rolling along across the top three rows of the screen is an array of colorful circus balloons.

What this boils down to is an unusual janitorial duty. You try to clear the screen of balloons. Clear out a row at a time to reap bonus points. Meanwhile, a mischievous monkey keeps blowing up more balloons. More balloons will appear if you clear an upper row before a lower one, but the monkey does not stir from his high-wire perch, unless you clear the screen when he tips his hat at you.

The animation remains fairly simple, though smooth. The sound and music are some of the best I've heard. Despite the simple theme, *Clowns and Balloons* turned out to be great fun, and inspired hours of frenzied joystick twisting. Perhaps its appeal can be compared to that of *Breakout,* a similar game where you bounce a ball to clear out a brick wall. *Breakout* is one of the most popular games in arcade history (that's B.P. – Before *Pac-Man*).

With the release of these games, DataSoft has issued an implicit challenge to game producers: use the Atari's features to the utmost.

DataSoft Inc.
19159 Business Center Dr.
Northridge, CA 91324
Available on cassette or disk.
$29.95 each.

©

www.commodore.ca

ATARI 400 48K MEMORY EXPANSION KIT

ORDER FACTORY DIRECT **124.95** SUGGESTED RETAIL $139.95

WHY BUY OUR NEW, STATE-OF-THE-ART 48K MEMORY EXPANSION KIT FOR YOUR ATARI 400? JUST ASK A FELLOW ATARIAN...

"Thank you for the shipment of the 48K memory expansion kit for my Atari 400. I found your kit very well documented, easy to assemble, and very well designed. I was previously using a 32K board which left a series of vertical lines on the left half of my TV. My hat is off to you for providing an affordable, quality product to make my computer even more enjoyable than before."

Gary Nance
Spokane, Washington

THANKS GARY, WE COULDN'T HAVE SAID IT BETTER OURSELVES!!!

DEALERS:
IN THE EAST CONTACT: JERSEY SYSTEMS
(800)526-3647
IN N.J. – (201)287-9462

ATARI IS A REGISTERED TM OF ATARI INC.

ADD $2.00 FOR POSTAGE AND HANDLING.
SEND CHECK OR MONEY ORDER TO:

DYNAMIC TECHNOLOGIES
P.O. Box 351
ALLEN, TEXAS 75002

TEXAS RESIDENTS ADD 5% SALES TAX

(214) 542-6012

ORDER 7 DAYS A WEEK 24 HOURS A DAY

YOUR PROGRAM IS WORTH DOLLARS

COMPUTERforce will

1) Run and Evaluate your program as to:

(i) Technical competence

(ii) Suggestions for technical improvements in screen presentation, output formats, disk and cassette operating systems, or revision for other hardware etc.

(iii) Potential for marketability and profits

(iv) Ideas to expand your program to service a different potential market

(v) Ideas to streamline your efforts toward a more specific market

2) Catalogue marketable programs

3) Enter into Royalty agreements with you, and on your behalf sell and promote your programs to the correct mass markets

COMPUTERforce will not

1) Copy your program

2) Modify your program

3) Catalogue, promote, or sell your program without your participation and consent

4) Keep your program. We will return your program by mail if you place a self addressed stamped label on your shipping container.

YOU MUST

1) Advise specifically your design hardware

2) Send cheque for $35.00 payable to:

COMPUTERforce
Box 175
Sault Ste. Marie, Michigan
49783

OR

Box 699
Sault Ste. Marie, Ontario
Canada P6A-5N2

3) Allow four weeks for evaluation and mail.

You will receive a written report documenting the areas noted, and if we deem your program to be valuable, you will be immediately contacted by our firm.

www.commodore.ca

Review:

High Orbit For Apple

Erann Gat
Oak Ridge, TN

I opened the package with anticipation. *High Orbit* seemed pretty ordinary for a computer game: a disk, some P.R. from Gebelli Software, the company that sells *High Orbit*, and a sheet of rather cryptic instructions.

I booted the disk in the usual way, and *High Orbit* immediately became very unordinary. My mouth fell open as I listened to the fastest disk boot I had ever heard. I later timed the furious "clickclickclick" of the head stepper motor: it was reading seven tracks per second! Apple DOS generally reads a track and a half per second.

The program then went into a nice demo mode which included some animated three-dimensional graphics, but nothing to give a clue as to what the game was all about. I tried for five minutes to start the game. I tried every key, but nothing worked. Oh well, when all else fails, read the directions. Aha! Control-R starts the game.

High Orbit starts with three dots that zoom onto the screen from the depths of space, which is gratifyingly free of stars. The object of the game is to "construct a space station" by moving a little fuzz ball (which represents a piece of the station) onto each of the dots using a tractor beam. To make it a bit more challenging, the dots spin around each other in a circle, and you can use the tractor beam for only a limited amount of time before it has to recharge. On top of that, there are the ubiquitous enemy spaceships that zip onto the screen and destroy your fuzz balls, so you have to start all over again. (You can destroy enemy spaceships, *if* you are fast enough, and that is a big "if.")

When (and if) you manage to maneuver a fuzz ball onto each of the dots at the same time, the space station is suddenly transformed into an abstract, three-dimensional shape which undergoes some breathtaking gyrations, splits in two, and starts spinning again.

The next phase is to "energize" the space station by moving yet another fuzz ball into the center (and I do mean the *exact* center) of the station and zapping it with your laser. Enemy spaceships will again try to destroy your supply of fuzz balls before you can get one fuzz ball into the center and destroy it.

If you are successful, the station stops spinning, becomes rainbow colored, and turns itself inside out, depositing the "crew" in deep space. The crew of the space station is just three little humanoid figures which pop onto the screen and do not move. The space station drops back into the depths of space, giving the impression that the crew is being launched into high orbit (hence the name of the program).

The last and final phase consists mainly of watching a shuttle pick up the crew. According to the instructions, you have to move the crew in front of the shuttle with your tractor beam, but I never had to. The shuttle seems to know where to go, and it will even destroy enemy spaceships that stray too close.

So how do you lose? Enemy spaceships cannot destroy you; in fact, you cannot be destroyed at all. Aye, but here's the rub: the space station must be constructed and energized before time expires. You get about two minutes to finish. If you do not, the game stops, and "mission incomplete" flashes on the screen.

If you do manage to complete a station within the time limit, you get a new station to build, but this one has four points instead of three. This goes on until you complete a six-point station. Then you go back to three points, but enemy spaceships get more aggressive. Every time you complete a station, a little colored square appears in a long hollow bar at the bottom of the screen. The bar is very long; I managed to fill up only about one-fifth of it with colored squares. You can always restart the game at the point where you last ran out of time.

High Orbit is a unique and challenging game. The graphics are well done and use the Apple's color capabilities to their fullest. It is a joy to play, provided you use a joystick. Paddles can be frustrating, and keyboard control was a frightening experience. (One nice feature of the keyboard control, though, is that you can redefine which key controls which function.) There seem to be enough levels of difficulty to keep even the best player occupied for a long, long time (although I was not able to get past the first few levels!).

All in all, *High Orbit* is an excellent game for all ages. It is challenging but not frustrating, simple but not boring. It requires a 48K Apple II with a disk drive. A joystick is not necessary, but it is *very* desirable.

High Orbit
Gebelli Software
1787 Tribute Road
Suite G
Sacramento, CA 95815
Requires 48K, disk
$29.95

©

www.commodore.ca

Review:

Raster Blaster

G. L. Kopp, Indianapolis, IN

After Atari introduced video games to America, old-fashioned, flipper-smashing, steel-ball-rebounding, mechanical pinball faced a notable decline in popularity until manufacturers moved into the computer age and introduced some incredibly sophisticated pinball. Now the game is back, this time in video format.

Raster Blaster, first produced for the Apple, is now available in an Atari version. The game boasts the standard fare of point-counting obstacles in its display: channels along the top which are lighted when the ball passes through them; four round bumper posts; targets in the center and on one side; a spinner and "ball saving shields" at the bottom which are always functioning during "easy" play, but must be turned on by hitting targets in the "hard" version (the only difference between the two). In addition, "Raster Blaster claws" can be enabled, which catch up to three of five balls allotted during play (a new ball replaces the one caught each time) and then releases them for multiple-ball play once all three claws have been activated.

Although the game is a masterfully written program, it is not without a glitch (I hesitate to call it a "bug.") Most of the time, the player will be able to give the ball the old one-two flip – slightly deflecting it off the tip of one flipper and catching it a split second later with the other. Often, however, if one flipper is up (they stay in that position until the fire buttons are released) and the ball passes just beyond it toward the bottom of the screen, the second flipper will not function, even though the ball is in its range. There are other occasions when the player must endure the non-functioning flipper phenomenon as well, though infrequently.

Another adjustment a pinball wizard must make is in holding the globe on the flipper to apply more than blind luck to direct it where he wants it. In mechanical pinball, the ball slides along the flipper on release until it reaches the *kill* point you know so well from playing a machine until your fingerprints are gone. Sorry, not so in the video version. Once in motion, always in motion, is the computer game style. The ball bounces lightly on the flippers, which puts the player into partnership with luck.

On the other hand, there are occasions in which the ball can be observed to pass *through* a flipper which is actually in the way of the other flipper's return shot. This rarity does make you smile and glance about to see if anyone noticed that a law of physics was broken in your favor. This same break has been observed to go the other way, however, allowing the ball to pass right through the bottom side channel railing and slip out-of-play behind the flippers.

In spite of its shortcomings, though, *Raster Blaster* is addictive, which speaks well of any arcade game. About the only feature true pinball fanatics will find missing is being able to flip the ball so hard it slaps the inside of the TV screen. Requirements for play are a disk drive, 24K of memory, *two* joysticks (accommodating one to four players), and a good deal of patience.

Raster Blaster, *$30*
BudgeCo, 428 Pala Avenue
Piedmont, CA 94611

©

SAVE YOUR QUARTERS AND LEARN TO PLAY LIKE A PRO WITH

The Survival Kit for Apple Computer Games

by Raymond Spangenburg and Diane Moser

Learn how to play more than two dozen of the most popular computer games like a pro. And, the better you get, the more challenges there are in store for you. **THE SURVIVAL KIT** covers games in four categories—Adventure, Arcade, Fantasy, and Strategy. For each game you'll find detailed, fun-to-read descriptions, winning strategies, tips from experts—players, programmers, and game designers—and the authors' personal quality rating. All the games run on the Apple home computer, and hardware requirements tell you at a glance on what other machines you can play each game.* Illustrated. Paperbound. Available December 15, 1982.

*Many games run on the Atari 400 or 800, the TRS-80, and the IBM PC.

Plus...to help you keep up to date in this fast-paced world, you'll want the 1983 COMPUTER ERA CALENDAR.

This 9" x 12" full-color calendar has plenty of room for your own important input while relating the history of computing through facts and photographs.

WADSWORTH ELECTRONIC PUBLISHING COMPANY
10 Davis Drive • Belmont • CA 94002

ALL ORDERS MUST BE PREPAID AND SHOULD BE SENT TO: WADSWORTH ELECTRONIC PUBLISHING CO., 10 DAVIS DRIVE, BELMONT, CA 94002. (Postage and handling will be paid by publisher.)

Please send me:

___ copies of THE SURVIVAL KIT FOR APPLE COMPUTER GAMES @ $9.95 each (I understand that the book will not be available until 12/15/82) $______

___ copies of the 1983 COMPUTER ERA CALENDAR @ $6.95 each $______

Residents of CA and KY please add appropriate sales tax $______

☐ Enclosed is my check **Total** $______

☐ Please charge my ☐ Visa ☐ Mastercard

Account # ______ Exp. Date ______

Name ______

Address ______

City ______ State ______ Zip ______

Signature ______

www.commodore.ca

Review:

Four New Cartridges For VIC-20

Harvey B. Herman
Associate Editor

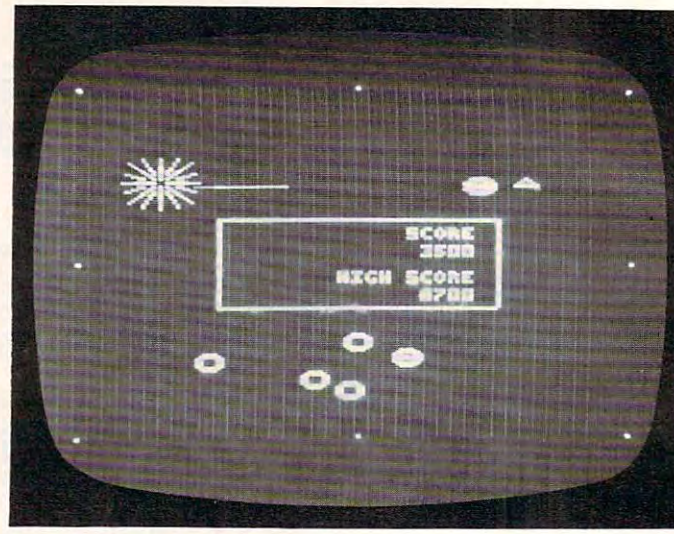
It seemed like Christmas in July when I received these cartridges to review. They turned out to be excellent examples of the full capabilities of the VIC. Some may have received more play than others, but all were challenging and fun to use. I think most adults will be very pleased with their purchase, and the kids who are arcade game freaks will especially like two of them (*Omega Race* and *Gorf*). *Sargon II Chess* is more a thinking person's game, and *Visible Solar System* is an interesting educational tool.

I am truly amazed at the ingenuity of machine language programmers. In the video games particularly, they use all the VIC's features. That is, color, sound, and graphics are integrated into a tour de force of gamesmanship – my word for their art. The programmers seem to be improving with each new release, so I hesitate to give numerical ratings as I am not sure what the future will hold.

Omega Race

An alien race, the Omegans, run a contest to improve their fighting skills. You are an Omegan fighter, and the command ship which you control must evade and destroy three types of enemy ships and two kinds of mines. Points are awarded for each ship and mine destroyed. The contest is played between two concentric rectangles. The central rectangle is impenetrable and displays the current score, the previous high, and the number of your remaining ships (turns). Force field boundaries, which resemble rubber bands, keep the players inside the outer rectangle.

The game is very fast-moving and difficult to master. Enemy droid ships start out with limited fire power, but improve with time. They occasionally become enemy command ships which are more deadly and become even more so when they



Omega Race

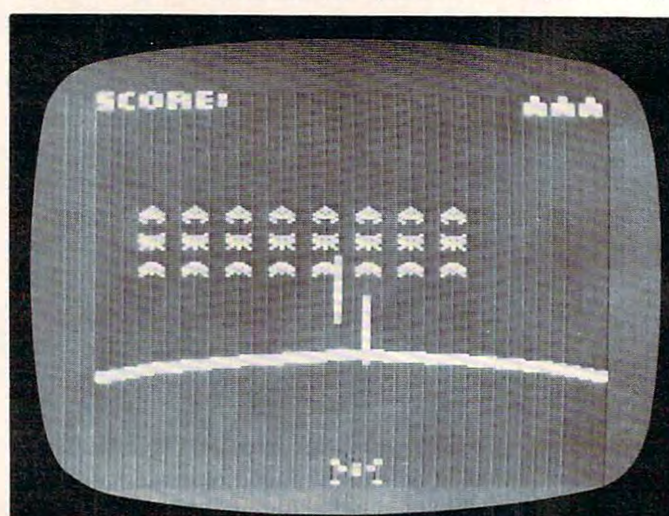
metamorphize into widely spinning death ships. Watch out for newly laid mines!

At the start of the game, you are given a choice of screen and character colors, three or five ships (turns), and either joystick or paddle control of your ship. I found paddle control easier to use, and my tiny testers agreed. I never scored very high, but one of the testers easily scored over 135,000 points starting with five men. Many of the testers preferred this game over any of the others reviewed here. A real winner for the VIC.

Gòrf

The Gorfian Empire poses a major threat to the Earth. Narrative translation: Get them before they get you, or you "bite the dust."

Gorf really is four games in one. You are in control of a fighter under attack. The Gorfian attacks come in waves. Wave 1 is called "Astro Battles" and is reminiscent of *Space Invaders*. Three rows of Droids, controlled by a Gorf, keep coming

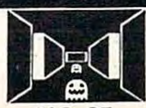


Gorf: Defeat the Invaders

www.commodore.ca

nüfekop

PROUDLY PRESENTS FOR THE VIC-20



3-D MAN

$12.95

THE EXACT MAZE FROM PROBABLY THE MOST POPULAR ARCADE GAME EVER WITH PERSPECTIVE ALTERED FROM OVERHEAD TO EYE LEVEL. THE DOTS. THE MONSTERS. THE POWER DOTS. THE SIDE EXITS. NEW ON SCREEN RADAR. THE GAME IS AMAZING.

3-D MAN REQUIRES AT LEAST 3K MEMORY EXPANDER BUT WILL RUN WITH ANY MEMORY ADD ONS (8K, 16K, 24K, ETC.) THAT WE HAVE COME ACROSS EXCEPT THE 3K SUPER EXPANDER.

By Ken Grant $16.95

FIRST THE BAD NEWS . . . THIS GAME IS LITERALLY FULL OF BUGS. THE GOOD NEWS? WE GUARANTEE HOURS OF EXCITING ENTERTAINMENT TRYING TO REMOVE THEM. SOME BUGS YOU ARE LIKELY TO COME UP AGAINST ARE SPIDERS, SNAILS, FLEAS, AND CENTIPEDES IN THIS RAPID FIRE, 100% MACHINE LANGUAGE, EXCEPTIONAL QUALITY GAME.

EXTERMINATOR RUNS IN STANDARD 5K VIC!

DEFENDER ON TRI

$12.95

AS PILOT OF THE EXPERIMENTAL DEFENDER STYLE SHIP "SKYES LIMITED" YOU ARE THE ONLY HOPE FOR ADVANCE PARTY OF SCIENTISTS TRAPPED IN ANCIENT ALIEN SPHERE WHICH SUDDENLY (HEAT FROM COLLISION COURSE WITH SUN PRESUMABLY - G.E.) CAME TO LIFE. 4 SCREENS WORTH OF UNIQUE DEFENSES, ON-OFF SHIELDS, FUEL DEPOSITS, ALIEN TREASURES, RUNNING TIMER, ENERGY, SCORE, AND VERY NICE GRAPHICS DISPLAY MAKE THIS ONE THAT DOES NOT QUICKLY WAX OLD.

DEFENDER ON TRI REQUIRES AT LEAST 3K MEMORY EXPANDER, BUT WILL RUN WITH ANY MEMORY ADD ON (8K, 16K, 24K, ETC.) WE HAVE COME ACROSS EXCEPT THE 3K SUPER EXPANDER.

ANTIMATTER SPLATTER!

$12.95

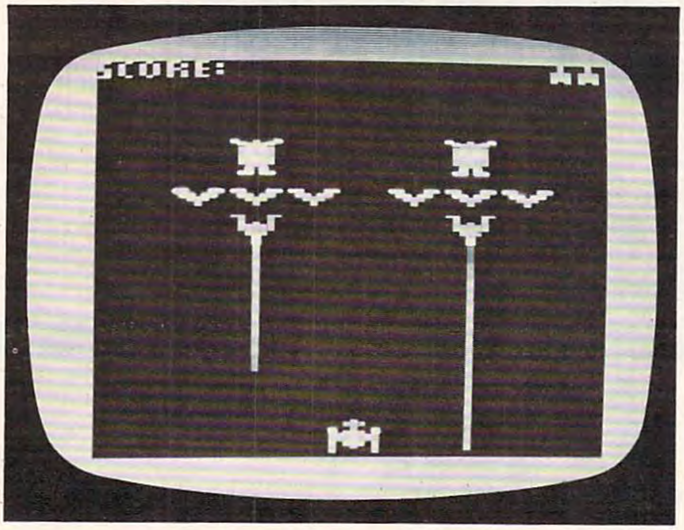
A MORE DASTARDLY ALIEN COULD SCARCE BE FOUND THAN ONE WHO WOULD WIPE OUT AN ENTIRE CIVILIZATION BY DROPPING ANTI-MATTER ANTI-CANNISTERS, RIGHT? IF YOUR OPINION OF THIS ALIEN TROUBLE MAKER IS THE SAME AS OURS PROBABLY YOUR FIRST THOUGHT WAS, GET SOME MATTER! WE SAY CALM DOWN! ALL IS NOT LOST. A MOBILE RAPID SPLATTER CANNON CAPABLE OF BOTH BREAKING THROUGH HIS STANDARD ALIEN MOVING FORCE FIELDS AND LAYING WASTE TO THE EVER INCREASING NUMBER OF ANTI-CANNISTERS IS EVEN NOW HOVERING ABOVE US. IF ONLY OUR CANNONEER HADN'T CALLED IN SICK . . . SAY, WHAT ARE YOU DOING TODAY?

ANTI-MATTER SPLATTER IS 100% MACHINE LANGUAGE AND RUNS IN STANDARD 5K VIC.

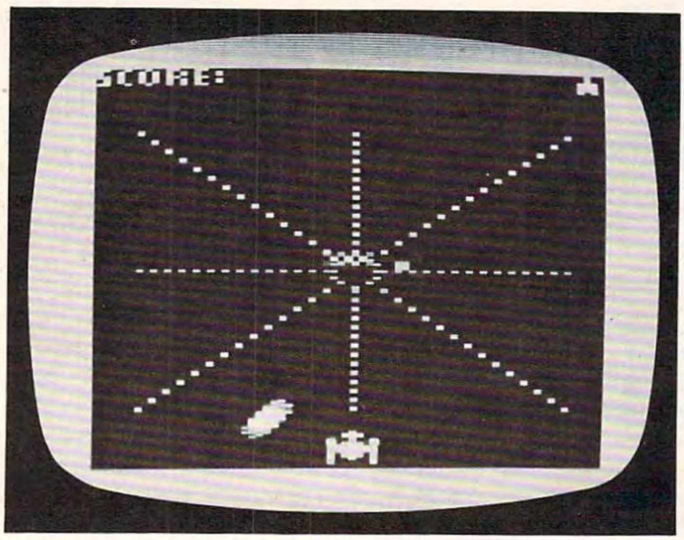
WRITE **nüfekop** P.O. BOX 156 • SHADY COVE, OREGON 97539 OR CALL 503-878-2113

MASTERCARD – VISA – C.O.D. ***OUR CATALOG FREE!***

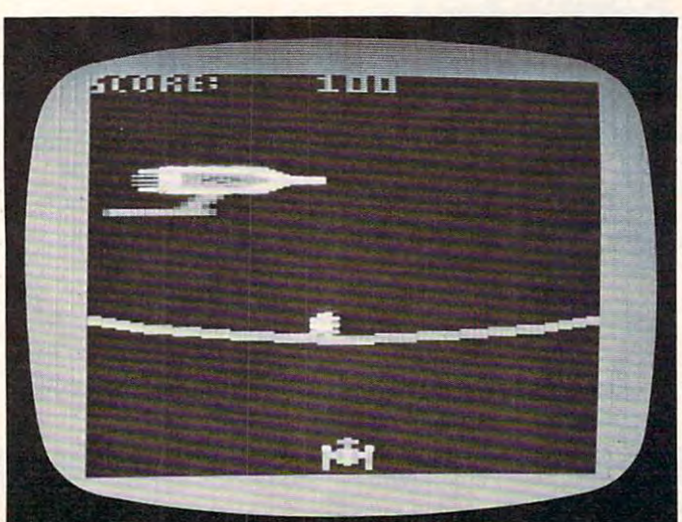
www.commodore.ca



Gorf: Vanquish the Laser Armada



Gorf: Evade and Eliminate in 3-D



Gorf: The Final Challenge – The Mother Ship

at you. You have some protection by a force field, but that doesn't last long. Kill or be killed is your motto throughout all the waves. Wave two, assuming you survive wave one, is called "Laser Attack." Two groups – a Gorf, three attack ships and a laser ship – have a serious grudge against you. Keep out of the way of the laser power ray, or you'll be sorry. The third wave, called "Space Warp," is the most difficult. The attacks come in a spiral formation and could make you dizzy if you watch too long. Avoid the smart torpedoes which seem to zero in on you. The flagship wave is the only remaining hurdle. You must destroy the flagship's power reactor while dodging fireballs and chips from the damaged vessel. The reward for completion of all four missions is a promotion and the right to oppose an even more powerful Gorfian force.

I find this game fascinating to watch while an "expert" plays. You are in a different world. I can't seem to get past the first or second waves, but experts can evade the enemy for mission after mission. How do they do it? I have no explanation except to say that I have the same feeling of awe when I watch professional sports on television. Why can't I do that – it looks so easy?

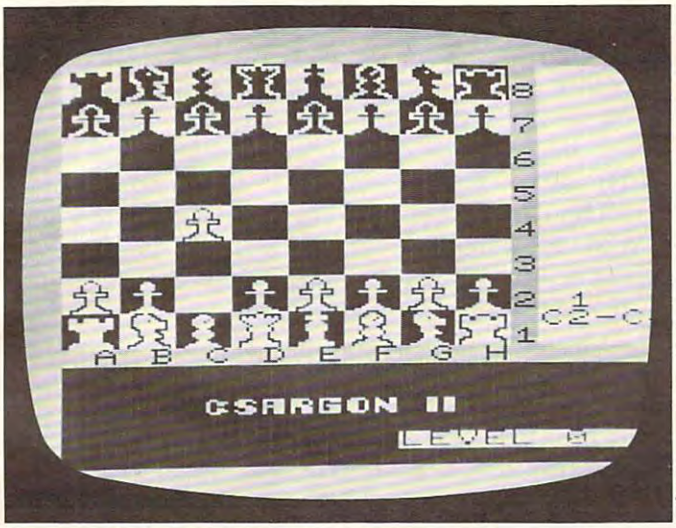
Sargon II Chess

Chess is considered a more "serious" game. Parents who would object to their child playing a video game would probably not object to chess. Chess certainly requires more thinking than most games, although the video games do have their own best strategies. Many people feel that the thinking associated with chess is good for us, and could carry over into other areas. Whatever the case, *Sargon II* is both fun and thought-provoking at the same time, and cannot help but make more people interested in one of the world's oldest games.

Sargon II probably has the best reputation of all the microcomputer chess programs. You are playing against the computer. At the start, you are given a choice of white or black pieces and the level of play. Beginners should choose level 1 or 0. Twenty seconds is given as the average response time for level 1. If you are a masochist, you can choose level 6, where the average response time is four hours! Of course, the play is much better at the highest levels as the computer is "thinking" further ahead.

The chess board is displayed with stylized pieces, which take a little getting used to at first. After awhile there is no problem. The last five moves are shown in a box next to the board. The notation used to show moves is algebraic (E2-E4) as opposed to descriptive (P-K4). However, the capture of a piece and castling is shown by X and O-.

www.commodore.ca



Sargon II Chess

This program has several nice touches:

1. Moves can be made either by typing the from-to locations with the keyboard or with a joystick. In the latter case, the cursor is placed over the piece, which is then "picked up" and "set down" at the new location.

2. It is possible to correct errors or try a chess problem with the set-up mode. I tried a few simple chess problems, with which *Sargon II* had no trouble. This mode could be used to correct the only deviation from normal chess rules that I could find – pawn promotion was always to a queen.

3. At the higher levels, *Sargon II* will, at your request, tell you what it thinks is your best move. You are free to accept or reject this suggestion. It is usually a good suggestion!

I would guess that this is the most sophisticated program available for the VIC.

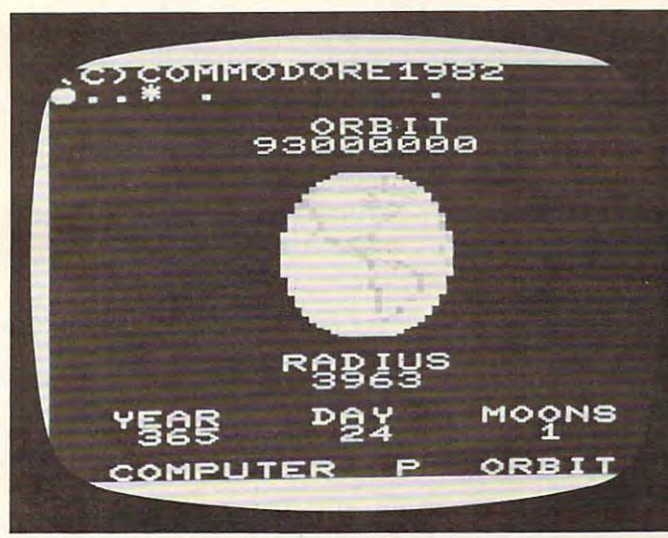
Visible Solar System

This program can hardly be called a game – it begins with a computer simulation of the solar system. You are in command of a spaceship which is on a tour of a scaled model of the planets in orbit around the sun. Additional features include a section displaying descriptive models of each of four planets and an "astrocalc," which gives detailed facts about the first six planets. Do not expect any unfriendly extra-terrestrials to appear when you use this program. If they do, turn off the VIC.

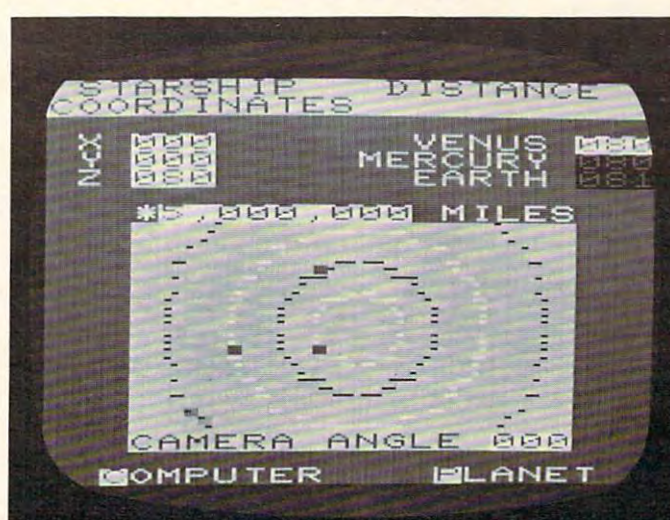
I would expect this simulation to get more classroom than home use. Flying through space shows in realtime perspective what can only be read about in textbooks. You have complete control of the position of the spaceship in three dimensions. The screen shows what one would see from an operator-controlled television camera. It can be positioned to view at any angle in an arc of 180 degrees relative to the ship. It does take some practice to get a feel for the program, and the preliminary text that comes with the cartridge is quite helpful. However, I believe an experienced teacher would be even more so.

In spite of my minor objections, *Visible Solar System* is the kind of program I would like to see more of. We have plenty of good games – I like to play them myself – but what we really need is more programs which can be used in teaching. They probably are more difficult to write effectively, and they may not sell as well, but they have a unique value.

Omega Race, Gorf, Sargon II Chess,
Visible Solar System
Commodore International
487 Devon Park Drive
Wayne, PA 19087
$39.95 Each Cartridge



Visible Solar System: 3-D Planetary Display



Visible Solar System: A Star's Eye View

© www.commodore.ca

The Code Works is no longer publishing its cassette magazine for PET, CURSOR, *but the back issues reviewed below are still available. ($5.95 each or $4.95 for orders of 12 or more.)*

Review:

CURSOR: Issues 23 Through 28

Marlene R. Pratto
Greensboro, NC

If your school is fortunate enough to own and use PET/CBM microcomputers, you can build your software library with programs from *CURSOR*. *CURSOR*, a cassette magazine, is published by Ron Jeffries of Code Works in Goleta, California.

Programs on *CURSOR* cost less than one dollar each, but are worth much more. I have classified *CURSOR* programs from issues 23-28 for children from kindergarten to eighth grade. In addition to five general classifications, I have added a sixth classification, TL, for Tools. Tools are those *CURSOR* programs which enable a user to program more effectively and with less effort (for example, *X-REF*) or to do other work more efficiently than without a program (*Repair*).

These tool programs may be used in a school setting. *X-REF* provides a cross-reference of variables used in a program, and *RE-NUM* renumbers the lines in a BASIC program.

One kind of "tool" program can also be used in schools to aid teachers, media specialists, and administrators in their work. In addition, students may use these tools to learn more about current and potential uses of computers. These programs enable the students to "do work" rather than to gain skills to be used later or to learn a body of knowledge. The children will learn skills and gain knowledge, but in a context different from the drill and simulation programs frequently used with children.

The *Repair* program mentioned previously can be used as a library checkout system. The program forms a file of items to be repaired. A record for a customer consists of a tag, name, amount, and location. A school media center could use the program to keep track of the books checked out to various rooms or units within the school. The tag would be the call number of the book; the name, the title of the book; the amount, the due date, such as 12.3 for December 3; and the location, the room number or unit. The program could also be used for checking out books to individuals, depending on the call number to identify the book and using the name as the name of the borrower.

The program is flexible, menu driven, and easy to use, and could be used by children in the media center. What a nice way to introduce children to the variety of uses of a computer.

The *Mail* program from *CURSOR* 25 is quite valuable. *Mail* creates and maintains a file of names for generating mailing labels. Letting children think of other ways to use the *Mail* program will help them learn new uses for other computer programs. One great advantage of *CURSOR* programs is that they are not protected; we can list them, change them, or make them into new programs when we or the children want to or are able to.

Several of these programs are usable by two persons at the same time. Among these is "*Mwhiz!*". A mathematical statement is printed on the screen. Each person tries to determine if the statement is true or false and then to press the appropriate button before the other person. This certainly makes learning mathematics enjoyable.

Maxit is a clever and challenging game which can be played against the PET to learn some skills and strategy (hence the DT classification below) before playing with another person. Other two-person programs are *Ambush!* and *Tank!*.

One program, *Safe!*, can be played by sight-handicapped children. The program simulates cracking a safe – the child listens to the clicks as he turns the knob by pressing the number pad. The game does use graphics, but the player does not have to look at what is on the screen to "crack the safe."

Enigma should send many children on a search for information about coding and decoding in World War II. Using and learning about *Enigma* is fascinating.

Strictly speaking, *Printsit* requires a printer so that children may print the pictures they have made on the screen. However, the program can be enjoyed whether or not the pictures can be saved.

Some of the *CURSOR* games can be played at several levels. Younger children can start playing at the lowest level, and as they build up playing skills they can move to higher levels. Older children can start at higher levels. Frequently, the highest level in a game is a real challenge to even the best computer users. Multiple levels allow many children in a school to use the same programs. This provides for discussion among the different age levels and for a pleasant learning atmosphere.

www.commodore.ca

One of the most congenial programs from these *CURSOR* issues is *Miser*, an adventure game. *Miser* was played continuously for two months at Erwin Open School, where it was the topic of both intense and casual conversations. Children exchanged information about what was hidden where. They used a thesaurus to look up alternative words when they could not make the computer take action. Some people think that personal computers will lead to fewer human conversations, but this program resulted in much conversation and cooperation.

Classification Of CURSOR Programs Issues 23 To 28

LEVEL	PROGRAM	ISSUE	CLASSIFICATION					
K-2	LAWN!	26	HE			PS	FF	
	PRINTSIT	24	HE			PS	FF	
	RESCUE!	25	HE			PS		
3-4	All of the above							
	BLASTO!	28				PS	FF	
	DEFEND!	24	HE			PS		
	EMAZE!	27	HE			PS		
	FLAGS!	28		DT			FF	
	MISER	27				PS		
	MWHIZ!	23		DT		PS		
	RACER!	24	HE			PS		
	SAFE!	26	HE			PS	FF	
	TANK!	26				PS		
	VOZ	28			LS			
5-6	All of the above							
	AMBUSH!	23			LS	PS		
	ENIGMA	23		DT	LS	PS		
	MAXIT	25		DT	LS	PS		
	PROCHAR	27				PS		TL
	RE-NUM	24				PS		TL
	RECIPE	23				PS		TL
	SKEET!	28	HE			PS	FF	
7-8	All of the above							
	ATTACK!	27				PS		
	DUEL!	27			LS	PS		
	G-WORD	24			LS	PS		
	MAIL	25						TL
	ORRERY	23		DT				
	RAM	26			LS			
	REPAIR	25				PS		TL
	STOP!	28			LS	PS		
	X-REF	25		DT		PS		TL
Teachers and aids								
	TEST	26						TL

Codes For Classifications

- **HE** hand and eye coordination
- **LS** logical skills
- **FF** fun and familiarity with the PET
- **DT** drill and tutor
- **PS** problem solving
- **TL** tool

Perhaps adventure style games have benefits beyond the social involvement and program solving. Because *Miser* and other adventure games have a restricted set of words that they understand, the player may know what to do, but not how to make the computer do it. This is similar to learning a programming language. The potential programmer may know what he/she wants the computer to do, but he/she must learn the words of the programming language used. Each computer language is a small subset of the language that humans know.

One of the programs, *Test*, will help teachers in grading. This program is nicely designed and even has its own example data to demonstrate what it does.

Our school here continues to find *CURSOR* an excellent resource for its PET microcomputers. The children have maintained their interest in computing over the past two and one-half years, and the newer children are quick to make friends with the PETs.

CURSOR
The Code Works
Box 550
Goleta, CA 93116

©

...PET/CBM/VIC? SEE SKYLES...

PET owners everywhere sing

Thanks for the Memories

to good old Bob Skyles

. . . they should . . . because Bob Skyles is the only complete source for memory boards for *any* PET ever sold. Old Bob won't forget you.

And the Skyles memory systems have the highest quality control of any computer product ever. Over 100 million bits of Skyles memory boards are already in the field. First quality static and dynamic RAMS, solid soldered on first quality glass epoxy. That is why they are **guaranteed**—*in spite of the new lower prices*—**for a full two years.**

The boards, inside the PET/CBM, install in minutes without special tools or equipment . . . just a screwdriver.

Because of our new dynamic memory design, and to celebrate old Bob's 30^{th}_{16} birthday, here are the smashing new prices:

8K Memory System	*orig. $250.00*	***now $200.00***	*Save $ 50.00*
16K Memory System	*orig. $450.00*	***now $300.00***	*Save $150.00*
24K Memory System	*orig. $650.00*	***now $400.00***	*Save $250.00*

. . . *For any PET ever made.* When ordering, just describe your PET by model number and indicate the amount and type (or brand) of memory currently in the unit.

Shipping and Handling. . . . . (USA/Canada) **$3.50** *(Europe/Asia)* **$15.00**

California residents must add 6%/6½% sales tax, as required.

Visa/Mastercard orders: call tollfree **(800) 227-9998** (except California). California orders: please call **(415) 965-1735.**

Skyles Electric Works
231E South Whisman Road
Mountain View, California 94041
(415) 965-1735

...PET/CBM/VIC? SEE SKYLES...

www.commodore.ca

Review:

Meteorites And Red Alert For Sinclair/Timex

Tom R. Halfhill
Features Editor

As popular as the Sinclair ZX-81 computer has been in the U.S. (reportedly 300,000 sold), it has been an even bigger hit in the country of its birth, Great Britain. There, the ZX-81 (and its predecessor, the ZX-80) is the microcomputer most often found in thousands of households and hundreds of schools. For one thing, it has been available there longer than in the U.S.

It's no surprise, then, that some of the best software written for the Sinclair has come out of Britain. The large number of Sinclair users there has created both the market and the labor pool for quality software development.

A New York firm, Softsync, Inc., recently arranged to import some of that British software. Softsync's first two releases are arcade-style space games. They are being sold in Britain by a company called Quicksilva under the names *Asteroids* and *Scramble*. However, Softsync is repackaging and selling the games here as *Meteorites* and *Red Alert*.

The games are as playable, and the action as fast, as games available for machines costing many times the Sinclair's $99.95 price.

Both are one-player games compatible with the new Timex TS-1000, a version of the ZX-81 that Sinclair has licensed the watch company to market in the U.S. Both games come on cassette tape and require the 16K RAM memory expansion module.

Meteorites

Meteorites (neé *Asteroids*) is patterned after the popular coin-op arcade game. Basically, you have to defend your spaceship against oncoming hailstorms of space rocks. The game starts with your ship centered on the screen while meteorites drift randomly by. To aim, you press the "6" key to rotate the ship counterclockwise, and the "7" key to rotate clockwise. Pressing the "0" key fires a stream of torpedoes. Hitting the "9" key fires the engines and moves the ship in whatever direction it is pointed.

Because of the Sinclair's low graphics resolution, it was not possible to represent the ship with a graphics shape. Instead, the game uses a numeric character from "1" to "8" to represent the ship and its orientation. That is, the character "1" means the ship is pointed "north" (the 12 o'clock position); a "2" means the ship is pointed northeast; a "3", east; and so on. Although this might sound awkward, I had no trouble adapting to the system.

The meteorites start off as graphics shapes, and split into five "0" characters when hit by your torps. These smaller pieces are then blasted out of existence by further hits. Screen wraparound is supported, which means objects can leave one side of the screen and emerge on the other.

Although the graphics effects in *Meteorites* are sparse (there are no fancy explosions), it is a tribute to the 3K machine language program that so many objects can be moving on the screen at once without noticeably slowing down the action.

According to the instructions, a bonus spaceship is awarded at 10,000 points, although my coordination deficiencies foiled persistent attempts to verify this feature.

Interestingly, the game's skill level can be varied by POKEing numbers into certain memory locations. For example, you can increase the number of points at which bonus ships are awarded, or vary the number of ships you start off with, or change the firing pitch of torps. These features should keep the game challenging for advanced players.

Red Alert

Red Alert resembles the popular arcade game *Defender*. A random landscape of mountains and valleys scrolls horizontally at the bottom of the screen, lending illusory motion to your spacecraft. This ship can be moved up or down with the cursor-up and cursor-down keys (the "7" and "6" keys, respectively). The control is surprisingly responsive, and it takes some flying practice to keep from clipping tall peaks or bumping into the top screen border – especially since either mishap blows the ship to pieces.

There's not much opportunity to dally around practicing, though. For one thing, flocks of alien ships keep zooming across the screen head-on into your path. The aliens try to destroy you in three ways: by firing missiles, by kamikaze charges, and

www.commodore.ca

by tricking you into dodging so fast that you hit either a mountain or the top of the screen. The last tactic is often the most annoying; just when you're congratulating yourself for evading the latest wave, you suddenly notice that your ship is rocketing headlong into a cliff. Unfortunately, there's no ejection seat.

You can do more than just dodge around, of course. Pressing another key fires your own missiles at the aliens, and they're even worse at evasive actions than you are. Knocking off a couple of them at the outset makes it easier to dodge the rest.

Red Alert would be good enough if this were all you had to worry about, but the programmer tossed in two more wrinkles. First, there are alien bases on the ground which present additional scoring opportunities. Daring (and presumably skillful) pilots can skim the surface, blasting the alien bases with missiles. This is a risky maneuver, however, since the slightest descent (accidental or provoked by the aliens above) reduces your ship to flying fragments.

Anyway, some of the bases are in valleys, protected from your missiles by flanking slopes. This forces you to resort to another method – your bombs. Pressing the "9" key drops one or a few bombs at once, destroying anything on contact. Like flying the ship, however, it takes some practice to learn how to properly "lead" the targets.

As a final twist, there are also some alien ships based on the surface. If you don't destroy them before they take off, they launch into a kamikaze path straight for your spacecraft.

The most amazing thing about *Red Alert* – and *Meteorites* – is that so much complex action happens simultaneously at relatively high speeds. Many Sinclair owners have been frustrated because so much software for their machines is written in BASIC. Both of these games are written in machine language, and it shows. In both cases the programmers have made the most of the Sinclair's capabilities. Of the two, *Red Alert* has the better graphics and arcade-style play, but both are top-shelf games that Sinclair users deserve.

Meteorites
Red Alert
Softsync, Inc.
P.O. Box 480
Murray Hill Station
New York, NY 10156
$14.95 each
plus $1.50 shipping/handling

©

In addition to Jim's review here, see Marlene Pratto's review of CURSOR, issues 23 through 28, on p. 136, along with complete ordering information.

Book Review:

PET Fun And Games

Jim Butterfield
Associate Editor

CURSOR magazine has been notable for several reasons. It's not a paper publication; the magazine is issued on cassette tape containing a "cover program" and five other PET/CBM programs. The programs are entertaining and of very high quality. And the price is surprisingly low.

CURSOR's programs haven't been exclusively games; a number of serious applications and utilities have been included over the years. But it's the games we remember best, and many of the *CURSOR* games have been memorable.

While *CURSOR* magazine ceased production with issue 30, back issues can still be obtained, and *CURSOR* may be making the transition from magazine to software house. Your dealer may stock the back issues or you may write *CURSOR* magazine at the address given in the book's Introduction.

Tape To Paper

If you can get the programs on tape, why bother with the book? After all, you can list the programs yourself. Well, the book is quite inexpensive. It is a collection of many "favorite" programs all in one place. And the program listings are useful for study. It's handy to have the game and its instructions in an easy-to-find location.

Since the book is limited to BASIC programs, some of *CURSOR*'s excellent machine language programs are not included. No worry: there's quite enough good material here to keep the reader busy and entertained. I must confess that I miss some of the classics that are too big to fit into the book: Ken Morley's "Phuzzy" and "Wuzzy" stories, for example.

Users who have both tape and book versions of a program may notice slight differences. These are usually small, cosmetic, and of no great importance.

The Games

If the book were called "PET Exercises, Simulations, and Challenges," it might enjoy more appeal in the

www.commodore.ca

educational community. The word "games" seems to be taboo in some quarters. Yet games are what they are, and they're great fun.

The back cover of the book claims that 30 games and puzzles are included. I count 31. Thirteen of them are written by Glen Fisher, one of the book's editors. The remaining 18 are by various contributors. I wish that the authors' names had been included in the table of contents. There's no easy way to find a given author's programs. I would also have liked to see a cross-reference to the particular *CURSOR* issue which carried the game.

The games are divided into six sections: Action Games, Puzzles, Games of Risk, Games of Strategy, Games of Chance, and Games for Fun. These sections are somewhat arbitrary. Many games could be listed in any of several divisions.

Some are old standbys. *Reversi* and *Master Mind*, for example, are well known in many versions: ancient, computer, and commercial. Others are new, witty, and well suited to computers. *RAT-RUN* and *FIRE*, for example, are nicely animated and play well.

There are many styles of games. Some are action, some thoughtful. Some have the computer as an active player; in others, the computer just enforces the rules. Some are involved with handling words and numbers, others with graphic objects. You'll get a good cross section with this book.

PET Fun and Games: Selected CURSOR Programs.
by Ron Jeffries and Glen Fisher
Osborne/McGraw-Hill, 171 pages
$11.95

©

VIXEN:™ The magazine for *VIC-20 users ©1982
*VIC-20 is a trademark of Commodore Business Machines, Inc.

Vixen will provide you, each month, with 5 or more ready-to-run programs on cassette - exciting, imaginative game, educational, and utility programs for all VIC memory configurations. Novice and pro alike, VIXEN is the magazine for you. Be a sly little fox for only $43 a year - Subscribe to:

VIXEN™

P. O. Box 507, Deer Park, Texas 77536
P.S. Foxhounds; look for Foxtales on side 2.
Vixen is a division of : Foxfire Systems, Inc.
3811 Newton, Pasadena, Texas

Dealer Inquiries Invited
Texas residents add 5% Sales Tax
Canada and overseas $53
Orders pre-paid,
U.S. Dollars only

THE COMMUNICATOR

For 4.0 Commodore Computer

THE HARDWARE

- Select speeds up to 4800 baud
- Conforms to RS232C standard
- Uses no existing CBM connectors
- Easily installed

THE SOFTWARE

- Emulates ADDS Regent 100, ADM 31, TeleVideo 950 ¹
- Uploads and downloads data files
- Load and run CBM programs from host
- Resides in ROM; no loading from disk
- Uses only 512 bytes of RAM
- Runs coresident with BASIC programs
- Text editor to work with Communicator and host computer

$200 for THE COMMUNICATOR
$40 for the text editor

$625 for THE COMMUNICATOR and U.D.S. 1200 baud modem
$770 for THE COMMUNICATOR and new Hayes 300/1200 baud modem

We have full line of U.D.S. and Hayes modems. Write for our discount prices.

AMPLIFY, INC.
2325 Macbride
Iowa City, Iowa 52240
319/-4775

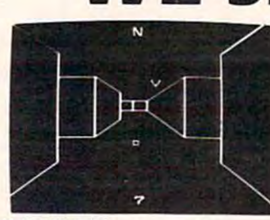
¹trademarks Adds Regent, Inc., Lear Siegler, Inc., Televideo Systems, Inc.

VIC-20® COMMODORE

VIC-20® COMMODORE

VICTORY SOFTWARE, INC.®

WE SELL FUN!™



TREASURES OF THE BAT CAVE $14.95
Battle the vampire bats as you search their cave for gold bullion. Fast, real time action will keep you playing for hours. Of course, you are in a different cave every time you play.

COSMIC DEBRIS $14.95
This highly addictive arcade type game will keep you battling the aliens for days.

GRAVE ROBBERS $13.95
Introducing the first GRAPHIC ADVENTURE ever available on the VIC-20! With realistic audio-visual effects, you explore an old deserted graveyard and actually see the perils that lie beyond.

NIGHT RIDER $12.95
High speed night time driving simulator.

STREET SWEEPERS $14.95
Gobble up all of the dots in the maze before the ensuing nemesis gets you. The maze is different every time, and if you succeed in getting all the dots, you get progressively harder mazes to complete as your skills increase. Does this sound like Pac Man? It isn't! Highly recommended and extremely addicting.

Send for free catalog. All programs fit in the standard VIC memory, and can be controlled from the keyboard. All programs on cassette tape.

Ordering—Please add $1.50 postage and handling per order. PA residents please add 6% sales tax.

Mastercard or VISA Charges—Please send your account number and card expiration date.

VICTORY SOFTWARE INC.
2027-A S. J. Russell Circle, Elkins Park, PA 19117
(215) 576-5625

VIC-20® COMMODORE

VIC-20® COMMODORE

www.commodore.ca

"Pixelator" is an easier way to design custom characters for the VIC. Three accompanying programs let you save and load the character data from cassette and convert it into DATA statements – ready to use in a program. If you don't want to type everything in, the author has offered to make tape copies (see information at the end of the article).

PIXELATOR

James Calloway
Morrisville, NC

The first time you design your own characters on the Commodore VIC-20, the process can be downright thrilling. Marking off graph paper in eight by eight squares and drawing in a figure. Converting each line into a number as if the dark squares were binary one and the light squares were binary zero. Storing the numbers in memory.

Then you POKE the magic address, 36869. The screen fills with gobbledygook. But wait! Isn't that a space ship there where the "A" of READY is supposed to be? And that three-legged alien must be the "D."

Once the thrill wears off, the work can turn to drudgery. Converting your design into numbers is bad enough, but the job of typing all those numbers into DATA statements is not only boring but also subject to typographical error. A slip of the finger and your beautiful rocket cruiser looks as if it had been shot full of laser holes.

Designing Characters With Pixelator

A program called "Pixelator" restores some of the thrill of designing screen characters. Pixelator gives you four large eight by eight work areas on the screen for creating, editing, and comparing characters. Pixelator then stores those characters in RAM. On standard VICs with 3.5K memory, Pixelator will store up to 64 characters. With additional memory, the program will store up to 128 characters; it also can retrieve from memory any character you have already stored. You can even copy from the VIC's own ROM character set and change those characters to suit your needs.

Like most small computers, the VIC stores mosaics or maps of its characters in ROM (addresses 32768 to 36863). Unlike some other computers, whose characters may be five pixels wide by seven pixels tall, the VIC's characters are eight by eight. (A pixel is simply the smallest portion of the video image that a particular computer can control.) That makes the VIC's characters look a bit squat, but it's a tidy use of memory. Eight bytes are needed to describe a single character, with each byte corresponding to a horizontal line of the character. The vertical information comes from breaking the bytes into binary ones and zeros, corresponding to dark and light areas.

Just Enough Memory

By POKEing different numbers into address 36869, you can change where the Video Interface Chip looks for its character maps. You do this automatically when you change the keyboard from graphics to text mode. Graphics is a value of 240 at 36869, and text is 242. The value in between, 241, represents reversed graphics characters, but using the reversed characters doesn't normally change the value at 36869.

A value of 252 moves the map location to 4096, the start of standard 3.5K memory. Above 252 the corresponding address increases by increments of 1024, up to 7168 for a value of 255. Because of the length of the Pixelator program, it uses the highest value. (For a fuller explanation of what happens at address 36869, consult Jim Butterfield's "Browsing the VIC Chip" in the April 1982 issue of **COMPUTE!**.)

The Pixelator program, once it is up and running, consumes almost 3K of memory. On VICs that haven't been expanded, that leaves just room enough to store 64 characters. That limit coincides with the fact that the second half of the map memory starting at 7168 corresponds to screen memory in most machines. We'll discuss a way of getting around this 64-character limit later.

Of course, with expanded memory, all you have to do is select a memory location that doesn't interfere with screen memory. Sometimes the problem is solved automatically because the screen memory moves (as do the screen color addresses). The three variables in line 20 allow you to change the program to compensate. XX is map memory and should always be a multiple of 1024. SC is screen memory. CL is color memory.

When you run Pixelator, you first are offered a choice of creating a new character or retrieving an old one from memory. The choices are color-coded green and cyan, respectively. If you select "new character" by pressing the programmable key F1, the border changes from white to green, and you are asked to select one of the four work frames by keying F1, F3, F5, or F7. Next you are asked to select the character at the address where you intend to design a new shape.

Four Options Following Design

Once you've selected a character, you'll see a half-height dot screen figure pop up in the top left corner of the frame. That's your cursor, and you

www.commodore.ca

can move it anywhere within the frame by using the cursor controls. To design a character, use the space bar. SHIFT/SPACE leaves a trail of red spaces in its wake. Without shifting, the SPACE bar returns the spaces to white. You can clear a cluttered frame simply by holding the SPACE bar down until all the red is gone.

After you have worked on the character to your satisfaction, you have four options. F1 stores your creation in the appropriate eight bytes of memory and then returns you to the opening format. F3 aborts the frame, returning you to the opening format without storing the character. F5 renames the character, enabling you to reassign it to a memory location different from the one for which it originally was named. This is of more use when retrieving characters from memory than when creating new ones, but it works in both modes. F7 allows you to work on a series of characters without having to go through the "select frame – select character" process every time. The command stores the current character, jumps to the next frame, and increments the character name. You can keep doing this until you have stored the question mark, at which point you are returned to the opening format.

If at the opening format you opt to retrieve a character from memory, the border changes to cyan, and you are given five choices. F1 retrieves from RAM; i.e., it accesses either characters you have already stored or whatever garbage happens to be in memory at the time. F2 accesses the VIC's ROM characters from the graphics mode, and F4 calls up the reverse of those characters. F6 and F8 are for text mode, the latter key again applying to reversed characters. You can freely mix characters from all modes and modify them to suit your needs. (If you need a full alphabet to go along with your custom characters, there is a short cut, provided you store your characters at 7168. After POKEing 255 into 36869, you can use RVS ON to get any normal character from "@" to "?". RVS OFF gives you your custom characters. This works only at 255.)

From there you are asked to select frame and select character again, but if you call up a graphics character (or, in text mode, an uppercase character) from ROM, you will be asked to rename it to something with a screen value less than 64. You now have the same options as before: to store, to abort, to rename, or to store and increment. If you have renamed a character, both the original character and its new name will be incremented.

Saving Your Custom Characters

More than likely, you will want to use Pixelator to create characters for use in some other program, such as a video game. Three shorter programs allow you to save the information the Pixelator has created. To save the characters directly on cassette as a data file, interrupt the Pixelator with the STOP key and type NEW to get rid of the program. Then load "Pixaver" into the VIC. Pixaver allows you to save a block of characters of any size, up to 64, on tape as a single data file. The first number in the file represents the screen value of the first character; the second number is the last character. This allows you to record as many different blocks as you like. Each file will contain the information necessary to store the data in the right place. Also, for convenience, each file will be tagged with the name of its first character. Now you can turn your VIC off.

The "Pixeloader" program will read the data off the cassette and enter it back into memory. Notice line 10, which sets the value of XX, the start of map memory. By changing that value, you can load character data into many different memory locations, thus bypassing the 64-character limit. Be sure that XX is a multiple of 1024, or else the characters won't properly correspond to the keyboard.

A third accessory program, called "Pixdata," will convert a block of RAM character memory into DATA statements, one for each character. The line numbers of the DATA statements will correspond to the screen value of the characters, plus 5000. DATA statements are highly inefficient, memory-wise, for storing that information, but they are much more convenient than cassette data files because they can be included within a program, which saves you the trouble of loading the characters separately.

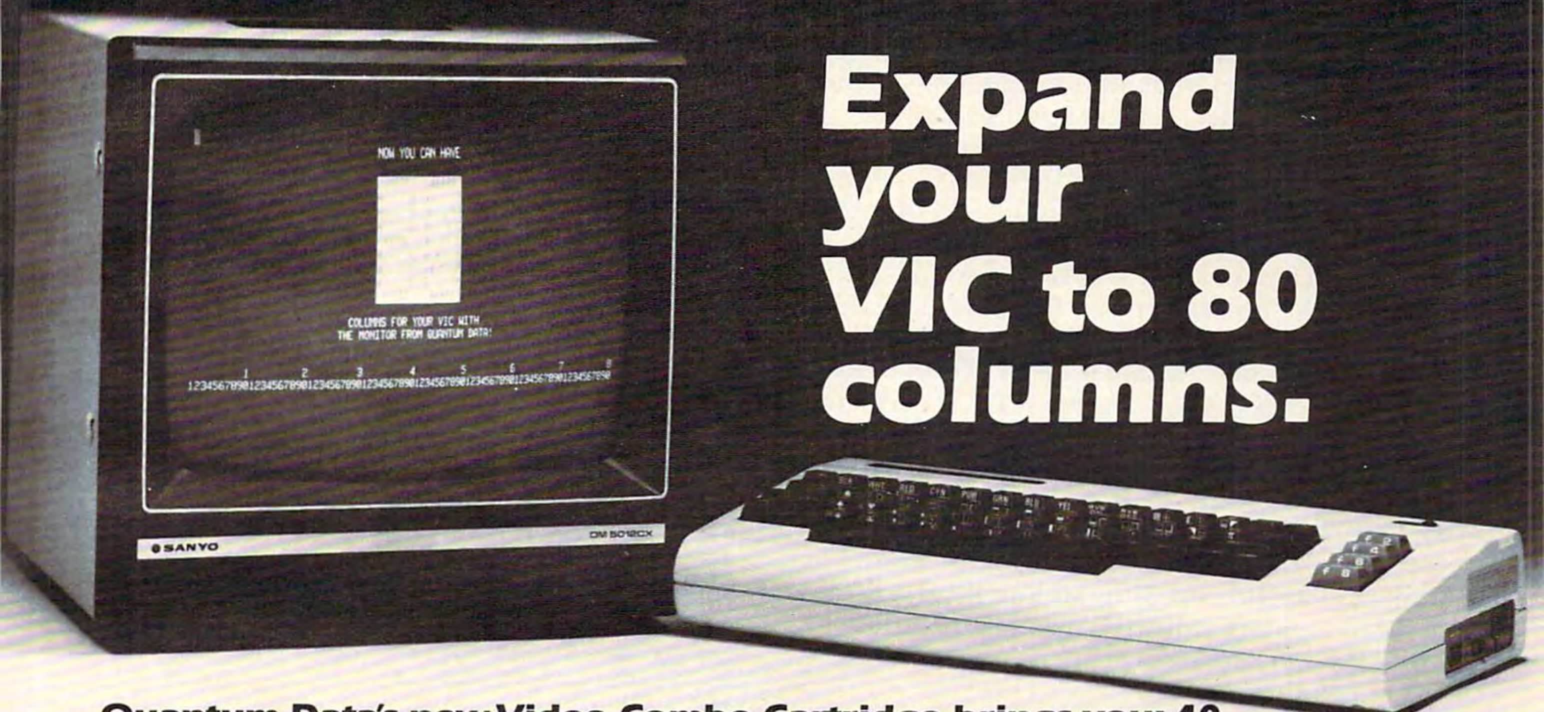
Pixdata is not as user-oriented as the other programs because it has been stripped down to bare essentials. You probably will have to modify some lines of Pixdata each time you run it. The values SR and LS initialized in line 30, for example, represent the first and last characters, respectively. If you have only 3.5K of free RAM, don't do more than 30 characters at a time, because you'll run out of memory.

What makes Pixdata interesting is that it self-destructs, saving you the chore of deleting it line by line to make room for your own program. (If you type Pixdata in by hand, be sure to save it on tape or disk before trying it.)

The secret of Pixdata lies in the way the VIC-20 stores BASIC lines. The first two bytes of a line represent the address of the *next* line. The third and fourth bytes are the line number. After that, the line consists of numbers that represent either tokens for BASIC commands (the token for DATA is 131) or the ASCII values of string characters. All numerals are treated as strings, so a DATA state-

www.commodore.ca

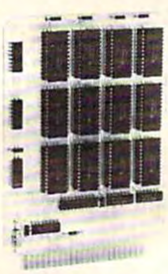
22-40-80 HIKE!



Expand your VIC to 80 columns.

Quantum Data's new Video Combo Cartridge brings you: 40 or 80 column display, plus 16K RAM and PROM socket. $299⁰⁰

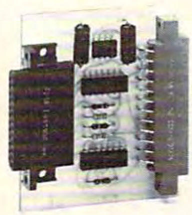
With the Video Combo Cartridge from Quantum Data you can now have 40 or 80 column display, 16K RAM and PROM all in one cartridge. It comes set for 40 column Display compatible with the VIC video modulator and your home T.V. Then, when you are ready to upgrade to 80 columns and a video monitor, just make a simple, no-cost change inside the cartridge. Instructions are provided. Also provided is a socket for a PROM, 16K of memory and AC adaptor. If you don't need memory, then 80 columns can be yours for only $199.50. A listing of the driver software is provided at no charge. A programmed PROM containing this software is also available for $19.95.



QDI expandor:

- Expands Basic user memory up to 24K in 8K steps
- PROMS may be mixed with RAM in 8K blocks
- 8K can be assigned to machine language area
- Plugs directly into VIC expansion port
- Low power, no additional power supply required
- Professional Quality, full buffering on all signals
- Small size: 6 x 4.5 inches.

16K expandor **$149.95**
24K expandor **$199.95**



QDI Printor RS-232 interface:

- Provides RS-232 voltage conversion for VIC serial port
- Allows use of a wide variety of RS-232 peripherals including printers, modems and voice synthesizers
- Low power CMOS circuitry requires no external power supply
- Small size: 2½ x 3 inches

Printor **$49.95**

QDI Minimother:

- Adds 3 slots to the memory expansion port
- Removable card guides allow either boards or cartridges
- Requires no additional power supply
- Fused to protect VIC power supply from overload
- Simple plug-in installation

Minimother **$69.95**

QUANTUM DATA, INC. (714) 966-6553
DEALER HOT LINE (714) 754-1945
3001 Redhill Bldg. 4, Suite 105, Costa Mesa, CA 92626

VISA and MasterCard accepted. $6 shipping for first item; $2 for each item thereafter. Prices subject to change without notice.

www.commodore.ca

ment may need as many as three bytes to represent a single numerical value. The number 128, for example, becomes 49, 50, and 56. Throw in a 44 for each comma, and you see why a DATA statement can use up more than four times the memory needed to store the numbers it represents.

Pixdata starts creating DATA statements at 5120, which is represented by the variable ZZ in line 40. Line 10 also sets 5120 as the end of BASIC memory, thereby protecting the DATA statements from the program itself. When Pixdata finishes creating DATA statements, it POKEs the low-high values of ZZ into the first and second bytes of line 1, the line that says "REM DELETE THIS LINE AFTER RUNNING." This causes BASIC to skip from line 1 to the first DATA statement, ignoring the rest of Pixdata in between. When you delete line 1 (simply type a "1" on a blank line and hit RETURN), the line editor compacts the DATA statements to the beginning of memory, destroying Pixdata in the process. If by adding RAM you have changed the start of BASIC memory, be sure to adjust the two addresses in line 170 accordingly before running Pixdata.

To use the DATA statements in a program, you will need a line like the following:

```
FORL=SR TO LS:FORM=0TO7:READ C:POKE
  XX+L*8+M,C:NEXTM:NEXTL
```

The values of XX (map memory), SR (first character screen value), and LS (last character) should be the same as they were in Pixdata.

Pixeloader and its companion programs should take some drudgery out of designing characters, but the programs themselves aren't much fun to type in from scratch. I will be glad to make cassette dubs of the programs for the standard fee of $3 a copy. Write "Pixelator" on a blank cassette and send it with a stamped, self-addressed mailer to James Calloway, Route 2, Box A-2, Morrisville, NC 27560.

The following articles in **COMPUTE!** provided valuable information and inspiration for the Pixelator: Jim Butterfield's "VIC Memory Map Above Page Zero" (January 1982); Doug Ferguson's "Large Alphabet for the VIC" and Butterfield's "More VIC Maps" (March 1982); Butterfield's "Browsing the VIC Chip" and Charles H. Gould's "Renumber VIC-20 BASIC Lines the Easy Way" (April 1982).

Program 1: Pixelator

```
20 XX=7168:SC=7680:CL=38400
30 POKE51,240:POKE52,XX/256-1:POKE55,240:POKE
   56,XX/256-1
40 FORLX=16TO1STEP-1:READXZ:POKEXX-LX,XZ:NEXT
   LX
50 POKEXX-10,SC/256:POKEXX-1,XX/256-1
60 PRINT"{CLEAR}{02 DOWN}";
70 FORY=1TO2:PRINT"{DOWN}{BLU}{02 RIGHT}/////
   ///{02 RIGHT}////////"
80 FORZ=1TO8:PRINT"{RIGHT}'{RED}          {BLU}
   %'{RED}          {BLU}%":NEXTZ
90 PRINT"{02 RIGHT}77777777{02 RIGHT}77777777
   {UP}":NEXTY
100 POKE36879,25:F=0:J=0:SYSXX-16:PRINT"{HOME}
    {GRN}{REV}F1{OFF}{BLU}-CREATE NEW CHA
    R."
110 PRINT"{CYN}{REV}F3{OFF}{BLU}-RETRIEVE MEMO
    RY"
120 GETS1$:IFS1$=""THEN120
130 IFS1$="{F1}"THENK=0:POKE36879,29:GOTO160
140 IFS1$="{F2}"THENPOKE36879,27:GOTO3500
150 GOTO120
160 IFJ=1THEN190
170 SYSXX-16:PRINT"{HOME}SELECT"SPC(4)"F1 F3":
    PRINT"FRAME:"SPC(4)"F5 F7";
180 GETS$:IFS$=""THEN180
190 IFASC(S$)>132THENONASC(S$)-132GOTO210,220,
    230,240
200 GOTO180
210 VV=3:HH=1:F=88:GOTO250
220 VV=3:HH=11:F=109:GOTO250
230 VV=13:HH=1:F=462:GOTO250
240 VV=13:HH=11:F=483
250 POKEF+SC,160:IFK>0THENPOKEF+CL,3:GOTO270
260 IFJ=0THENPOKEF+CL,5:GOTO280
270 IFJ>0THENC=CJ:C0=CG:GOTO320
280 SYSXX-16:PRINT"{HOME}SELECT CHARACTER";
290 GETC$:IFC$=""THEN290
300 GOSUB5000
310 IFCE=2ANDS2$="{F1}"THEN290
320 IFK=1ANDI=0ANDCE<>1THEN4000
330 IFCE>0THEN290
340 POKEF+SC,C:POKEF+CL,0:V=1:H=1:P=SC+23+VV*2
    2+HH:PA=P:PQ=PEEK(P)+72:PP=PQ
350 I=0:J=0:SYSXX-16:PRINT"{HOME}F1-STORE IN M
    EMORY"
360 PRINT"F3-ABORT"SPC(4)"F5-RENAME F7-STORE/I
    NCREMENT";
370 GETG$:POKEP,PQ:POKEPA,PP:IFG$=""THEN370
380 IFASC(G$)=32ORASC(G$)=160THENPOKEP,ASC(G$)
    :H=H+1:GOTO440
390 IFG$="{DOWN}"THENV=V+1:GOTO440
400 IFG$="{UP}"THENV=V-1:GOTO440
410 IFG$="{RIGHT}"THENH=H+1:GOTO440
420 IFG$="{LEFT}"THENH=H-1:GOTO440
430 IFASC(G$)<133ORASC(G$)>136THEN370
440 IFH>8THENH=1:V=V+1
450 IFH<1THENH=8:V=V-1
460 IFV>8THENV=1
470 IFV<1THENV=8
480 PP=PEEK(P):PA=P:IFPP=104ORPP=232THENPP=PP-
    72
490 IFG$="{F1}"THENK=0:POKEPA,PP:GOTO1000
500 IFG$="{F2}"THENK=0:POKEPA,PP:GOTO100
510 IFG$="{F3}"THENI=1:POKEPA,PP:POKEF+CL,PEEK
    (36879)-24:POKEF+SC,160:GOTO4120
520 IFG$="{F4}"THENJ=1:POKEPA,PP:GOTO1000
530 P=SC+(VV+V)*22+HH+H:PQ=PEEK(P)+72
540 GOTO370
1000 SYSXX-16:PRINT"{HOME}STORING ";:POKESC+8,C
1010 FORVE=1TO8:ZZ=0
1020 FORHY=1TO8:PO=SC+(VV+VE)*22+HH+HY
1030 IFPEEK(PO)=160THENZZ=ZZ+2^(8-HY)
1040 NEXTHY
1050 POKEXX+C*8+VE-1,ZZ:NEXTVE:IFJ=0THEN100
1060 GOTO2000
2000 CJ=C+1:CG=C0+1:S$=CHR$(ASC(S$)+1):IFASC(S$
     )>136THENS$="{F1}"
2010 IFCJ=64ANDXX=7168ANDSC=7680THENCE=2
2020 IFK=2THENK=1
2030 IFCG>127THENCG=0
2040 IFS2$="{F1}"ANDCE=2THENJ=0:GOTO100
```

www.commodore.ca

MIS VIC-20 SOFTWARE

TWO NEW ARCADE GAMES!
100% MACHINE 100% FAST

You were on a routine patrol through an uncharted asteroid belt when you were caught in a . . .

GALACTIC CROSSFIRE

Will you survive?

• **GALACTIC CROSSFIRE** is a multi-color hi-res arcade game that runs on the unexpanded VIC-20. JOYSTICK REQUIRED.

• **ALIEN SOCCER** is a multi-color hi-res arcade game that runs on the unexpanded VIC-20 using JOYSTICK, KEYBOARD, or PADDLES.

GALACTIC CROSSFIRE and **ALIEN SOCCER** are available on cassette with complete documentation, enclosed in an attractive vinyl binder for $14.95 each.

MIS produces the finest educational, recreational, and functional software available for the Commodore VIC-20 Personal Computer. Ask for MIS software at your local computer store or order direct from MIS. VISA, MASTER-CARD, phone orders, and C.O.D. accepted. California residents add sales tax.

ALIEN SOÇÇER

MIS
250 Fern Rock Way
Boulder Creek, CA 95006
(408) 338-9546

Life begins at 40!

MIDWEST MICRO associates
PO Box 6148
Kansas City, MO 64110

Include $1.25 for postage & handling. Missouri residents add 4.6% sales tax. Dealer inquiries invited.

Especially with TERMINAL-40, the NEW standard of excellence in telecommnications for the VIC-20. This unique program gives **40-character lines** and **smooth scrolling** entirely through software; no expensive hardware to buy! TERMINAL-40 also features a **4K** (or larger) **Receive Buffer** with screen review and printer output. Function keys are defined for instant access to frequently-used modes. Fully programmable Baud, Duplex, Parity, Wordsize, Stopbit, and Linefeed; supports control characters. Requires standard VIC-20 with 8K memory expansion and suitable modem. VIC printer optional. 24 p. manual.

TERMINAL-40 . . . $29.95

UN-WORD PROCESSOR 2 . . . $19.95

The improved UN-WORD retains the practicality and economy of the original. Easy-to-use text entry and screen editing. Use with **any size** VIC (5K to 32K). Supports VIC printer, RS-232 printers*, and now parallel printers*, too. Handy user Menu selects: single- or double-space, form feed, print width, number of copies. Supports printer control codes. With complete documentation.

Word Processing may be the single most-popular application for the VIC, and UN-WORD does so much for so little. *Don't miss out!*

Write for free brochure:

VIC-PICs ($19.95). Nineteen fascinating high-resolution *digitized* pictures. Color or B/W sets.

GRAFIX MENAGERIE ($14.95). 3-program set shows off VIC graphics. Contains dot- and line-plot routines.

GRAFIX DESIGNER ($14.95). Design your own graphics characters! Recall, erase, edit . . . save to tape for use in your own programs.

BANNER/HEADLINER ($14.95). Make GIANT banners on your printer. . . a 10-ft. long "Welcome Home"?

TICKERTAPE ($14.95). Glide *your* message across the screen in color . . . great for displays!

RS-232 INTERFACE ($49.95). Bi-directional communications with most serial printers & modems. Includes M/L handshake. 90 day warranty.

ORDER DESK OPEN
9 am - 4 pm
(816) 254-9600
MasterCard & VISA add 4%.
C.O.D. add $3.50.

VIC-20 software

VIC-20 is a trademark of Commodore Business Machines

*RS-232 and parallel printers require interface. All programs on high quality digital cassette tape.

www.commodore.ca

```
2050 IFK=0ANDCE=2THEN100
2060 GOTO190
3500 K=1:IFJ=1THEN3540
3510 SYSXX-16:PRINT"{HOME}F1-RETRIEVE FROM RAM"
3520 PRINT"F2-ROM GFX {REV}F4-REVERSE {OFF}F6-R
     OM TEXT {REV}F8-REVERSE{OFF}";
3530 GETS2$:IFS2$=""THEN3530
3540 IFS2$="{F1}"THENXR=XX:GOTO3580
3550 S2=ASC(S2$)-137:IFS2>-1ANDS2<4THENXR=32768
     +1024*S2:GOTO3570
3560 GOTO3530
3570 IFS2>1THENPOKE36869,242:GOTO160
3580 POKE36869,240:GOTO160
4000 IFJ=0THENC0=C
4010 SYSXX-16:PRINT"{HOME}":PRINT"LOOKING AT  "
     ;S5$:POKE7713,C0
4020 FORD=1TO8:DA=PEEK(XR+C0*8+D-1):DI=0
4030 FORDD=1TO8:DI=INT(DA/2^(8-DD)):DA=DA-DI*2^
     (8-DD)
4040 IFDI>0THENDO=160:GOTO4060
4050 DO=32
4060 IFDD=8ANDD<8THENZD=15:GOTO4090
4070 IFD=8ANDDD=8THENZD=-184:GOTO4090
4080 ZD=1
4090 ZF=SC+(VV+D)*22+HH+DD:POKEZF,DO:POKEZF+ZD,
     PEEK(ZF+ZD)+72:NEXTDD:NEXTD
4100 IFCE>0THENK=2:GOTO4120
4110 GOTO340
4120 SYSXX-16:PRINT"{HOME}RENAME":GOTO290
5000 C=ASC(C$):CE=0
5010 ONINT(C/32)GOTO5060,5040,5050,5020,5040,50
     30
5020 CE=1:RETURN
5030 C=C-64
5040 C=C-32
5050 C=C-32
5060 IFJ=1THENC0=CG
5070 IFXX=7168ANDC>63ANDSC=7680THENCE=2:RETURN
5080 RETURN
6000 DATA162,0,169,32,157,0,30,232,224,68,208,1
     ,96,76,244,27
```

Program 2: Pixaver

```
10 XX=(PEEK(56)+1)*256
3000 SYSXX-16:PRINT"{CLEAR}FIRST CHARACTER?";
3010 GETSR$:IFSR$=""THEN3010
3020 C$=SR$:GOSUB5000:SR=C:IFCE>0THEN3010
3030 PRINT"{HOME}"SPC(15)" "SR$;SPC(5)"LAST CHA
     RACTER? ";
3040 GETLS$:IFLS$=""THEN3040
3050 C$=LS$:GOSUB5000:LS=C:IFCE=1THEN3040
3060 IFSR>LSTHENSS=SR:SR=LS:LS=SS:SS$=SR$:SR$=L
     S$:LS$=SS$
3070 SYSXX-16:PRINT"{HOME}SAVING "SR$" TO "LS$;
3080 PRINT"{HOME}";:OPEN1,1,1,SR$
3090 SYSXX-16:PRINT"{HOME}SAVING "SR$" TO "LS$
3100 PRINT#1,SR
3110 PRINT#1,LS
3120 FORCZ=SRTOLS
3130 FORLL=0TO7
3140 PRINT#1,PEEK(XX+CZ*8+LL)
3150 NEXTLL
3160 NEXTCZ
3170 CLOSE1
3180 END
5000 C=ASC(C$):CE=0
5010 ONINT(C/32)GOTO5060,5030,5040,5020,5030,50
     50
5020 CE=1:RETURN
5030 C=C-64:GOTO5060
5040 C=C-32:GOTO5060
5050 C=C-128:GOTO5060
5060 IFXX=7168ANDPEEK(648)*256=7680ANDC>63THENC
     E=2:RETURN
5070 RETURN
```

Program 3: Pixeloader

```
10 XX=7168
20 OPEN1,1,0
30 INPUT#1,SR
40 INPUT#1,LS
50 FORS=SRTOLS
60 FORR=0TO7
70 INPUT#1,C:POKEXX+S*8+R,C:NEXTR:NEXTS
```

Program 4: Pixdata

```
1 REM DELETE THIS LINE AFTER RUNNING
10 POKE51,0:POKE52,20:POKE55,0:POKE56,20:REM ~
   MUST MATCH ZZ
20 XX=7168
30 SR=0:LS=26:REM FIRST AND LAST CHARACTERS
40 ZZ=5120:AA=ZZ
50 POKEZZ-1,0
60 FORL=SRTOLS
70 L2=INT((L*10+5000)/256):L1=(L*10+5000)-L2*
   256:POKEZZ+2,L1:POKEZZ+3,L2
80 POKEZZ+4,131:X=4
90 FORLL=0TO7
100 S$=STR$(PEEK(XX+L*8+LL)):S=LEN(S$)
110 FORLZ=2TOS:X=X+1:POKEZZ+X,ASC(MID$(S$,LZ,1
    )):NEXTLZ
120 IFLL=7THEN140
130 X=X+1:POKEZZ+X,44:NEXTLL
140 X=X+1:POKEZZ+X,0
150 X=X+1:Z2=INT((ZZ+X)/256):Z1=ZZ+X-Z2*256:PO
    KEZZ,Z1:POKEZZ+1,Z2:ZZ=ZZ+X:NEXTL
160 POKEZZ,0:POKEZZ+1,0
170 A2=INT(AA/256):A1=AA-A2*256:POKE4097,A1:PO
    KE4098,A2:POKE56,30
```

©

Expansion Board for the VIC 20

GOSUB International, Inc. presents

The CARDBOARD for the VIC 20! The CARDBOARD is an expansion motherboard for use with Commodore's VIC 20. It has six slots that will accept any VIC-compatible cartridge in any configuration.

Increase RAM up to 40K and use several utility ROMs plus have several games online, all switch selectable!

The CARDBOARD can be daisy-chained, giving the user an almost unlimited number of available expansion slots.

A system reset switch has also been added to the CARDBOARD allowing the user to select and/or restart games without turning off the computer.

All this for only $119.95 plus $1.50 S/H
VIC Light-Pen only $24.95

To order send check or money order to:

GOSUB International, Inc.
501 E. Pawnee, Suite 430
Wichita, Kansas 67211
(316) 265-9858

VISA and MasterCard phone orders also accepted
Send $1.00 for catalog of other VIC 20 items and software.

www.commodore.ca

VIC 20/PET/CBM OWNERS

WALLBANGER - Blast your way through the dodge'm, blast'm, and attack modes. If you destroy the bouncing balls before they destroy you, the walls close in for the next round. Wallbanger is written in machine language, has great sound, and encourages complex strategies.

CASS/5K/VIC 20/CBM 8032
CASS/8K/40 COL SCREEN/OLD-NEW ROMS/FAT FORTY . . . **$15.00**
(CALIF. RES. ADD 6% SALES TAX)

MILLIPEDE - Exterminate the oncoming millipedes and fleas as they descend through the mushroom patch. Blast giant bouncing spiders before they pounce on you. Shoot a millipede in the body and suddenly two millipedes descend toward your ship. Millipede is written in machine language, has excellent graphics. and great sound.

CASS/5K/VIC 20/CBM 8032
CASS/8K/40 COL SCREEN/OLD-NEW ROMS/FAT FORTY . . . **$15.00**
(CALIF. RES. ADD 6% SALES TAX)

Write for FREE game details:

ON LINE SOFTWARE
P.O.BOX 2044
ORCUTT, CA 93455

WARNING! These games cause high panic levels!

VIC 20/PET/CBM OWNERS

VIC RAPIDWRITER

Word Processor
for VIC-20 or 64

Fast flexible editing by character, line or block.
Full format control for I/O.
Mix calculations with text.
Unlimited document length.

Needs only VIC-20 with 8K, cassette.

Cassette (program & tutorial) and manual. **$39.95**
Specify VIC model and printer, VIC-1515 or Epson, others on request.

Complete Rapidwriter Systems from **$850.**

New & Powerful
Easy to Learn - Easy to Use

Send check or money order to:

RAPIDWRITER
91 Long Hill Rd.
Leverett, MA 01054
413-549-3744

Rapidwriter (c) H.D. Mfg. Inc. 1982
All rights reserved

World's Most Cost Effective Development System

- Type your programs directly from your Commodore VIC-20 keyboard into the built-in 4 kilobyte ROM emulator
- Jumper direct to target ROM socket
- Test programs in circuit
- Burn them direct to EPROM with built-in EPROM programmer and power supply
- Burns & runs EPROMS for the Commodore VIC-20, too
- Software on EPROM included
- Comprehensive 25 page manual
- Fits EXP PORT or SUPER EXP Leaves user PORT free

PROMQUEEN CARTRIDGE COMPLETE ONLY $169.50

manual available separately for $20.00 credited on cartridge purchase
Jumper Cable, 2 ft: $18.50 with cartridge; 2732A EPROMS $12.50 each with cartridge

Gloucester Computer Bus Co. 6 Brooks Rd., Gloucester, MA 01930 617-283-7719

Shipping charges for all items extra, Mass. residents add 5% tax.
VISA AND MASTERCARD ACCEPTED

NEW TOUCH-N-LIGHT PEN for the VIC 20

Imagine this . . .

1. draw pictures
2. play games
3. answer questions

and much, much more.

All without touching the keyboard. Just point the Touch-N-Light Pen at the screen and watch the VIC come alive. No skill necessary.

Game & Educational Programs available

Suggested list price $75.00

N.Y. STATE RESIDENTS ADD TAX

Send Check or Money Order to:

Sunshine Peripherals
1229 East 28th Street
Brooklyn, N. Y. 11210

Dealer Inquiries Invited

VIC is a registered trademark of Commodore Business Machines

www.commodore.ca

SPECIALS on INTEGRATED CIRCUITS

6502	7.45	10/6.95	50/6.55	100/6.15
6502A/6512A	8.40	10/7.95	50/7.35	100/6.90
6520 PIA	5.15	10/4.90	50/4.45	100/4.15
6522 VIA	6.45	10/6.10	50/5.75	100/5.45
6532	7.90	10/7.40	50/7.00	100/6.60
2114-L200		2.45	25/2.30	100/2.15
2114-L300		2.25	25/2.10	100/2.00
2716 EPROM		4.90	5/4.50	10/4.00
2532 EPROM		8.90	5/8.45	10/7.90
6116 Hitachi 2K×8 CMOS RAM		8.90	5/8.45	10/7.90
4116-200 ns RAM				8 for 15
Zero Insertion Force 24 pin Socket				2.00
S-100 Wire Wrap Socket				2.40

A P Products 15% OFF
A P Hobby-Blox 15% OFF

Anchor Automation Signalman Modems

All Signalman Modems are Direct Connect, and include cables to connect to your computer and to the telephone. Signalman Modems provide the best price-performance values, and start at less than $100. **Dealer and OEM inquiries invited.**

Mark I RS232
Mark II for Atari 850
Mark III for TI99/4A
Mark IV for CBM/PET with software
Mark V for Osborne (software available)
Mark VI for IBM Personal Computer
Mark VII Auto Dial/Auto Answer
Mark VIII Bell 212 Auto Dial/Answer

DC HAYES Smartmodem	**229**
RS232 MODEM – Accoustic	**119**
RS232 MODEM – CCITT frequencies	**175**

We carry Apple II+ from Bell & Howell

16K RAM Card for Apple	**65**
Apple LOGO	150
Video Recorder Interface	545
Super Serial Card	149
Thunderclock Plus	119
Z80 Softcard and CP/M	295
Parallel Printer Interface/Cable	85
Integer BASIC Card	150
Grappler Interface	139
Apple Paddle Pair	29
T G Products Joystick for Apple	**48**
T G Paddles	32
DC Hayes Micromodem II	299
Videx 80 Column Card	259
fullFORTH+ for Apple (fig-Forth)	69
Silentype Printer and Card	310
Graphics Tablet and Card	645
Apple PASCAL Language	195
Apple FORTRAN	160
We stock EDUWARE Software	
GENIS I Courseware Development System	185
Unicom Grade Reporting	250
Unicom School Inventory (resource mgmt)	250
Executive Briefing System with fonts	225
Apple Dumpling (Microtek) Printer Interface	**115**
Apple Dumpling with 16K Buffer	**160**

commodore

We stock the complete line. See us for Personal, Business, and Educational requirements.
Educational Discounts Available

PETSCAN I $345 base price.

Allows you to connect up to 35 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 3 computers.

VIC 20	229	VIC Sargon II Chess	32
VIC 1515 Printer	335	VIC GORF	32
VIC 3K RAM	32	Meteor Run (UMI)	39
VIC 8K RAM	53	VIC Radar Ratrace	24
VIC 16K RAM	99	Amok (UMI)	20
VIC Disk Drive	465	Snakman	15
VIC Pinball	32	Rubik's Cube	13
VIC Omega Race	32	Programmers Reference	15
Spiders of Mars (UMI)	39	Renaissance (UMI)	39
VIC Draw Poker	24	VIC Superslot	23

VICTORY Software for VIC

Street Sweepers	12	Maze in 3-D	12
Night Rider	11	Cosmic Debris	12
Treasurers of Bat Cave	17	Grave Robbers Adventure	11
Games Pack I	12	Games Pack II	12

TNW 488/103 with DAA	450
Compute!'s First Book of PET/CBM	11
WordPro 3+ – 32K CBM, disk, printer	195
WordPro 4+ – 8032, disk, printer	300
SPELLMASTER spelling checker for WordPro	**170**
VISICALC for PET, ATARI, or Apple	190
SM-KIT enhanced PET/CBM ROM Utilities	**40**
Programmers Toolkit – PET ROM Utilities	35
PET Spacemaker II ROM Switch	36
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET or CBM	8
IEEE-Parallel Printer Interface for PET	110
IEEE-RS232 Printer Interface for PET	120
The PET Revealed	17
Library of PET Subroutines	12
SADI Intelligent IEEE-RS232 or parallel	235
Programming the PET/CBM (Compute!) – R. West	**20**
Compute! First Book of VIC	11
Whole PET Catalog (Midnight Gazette)	**8**
4 Part Harmony Music System for PET	**60**
Color Chart Video Board for PET	125
PET Fun and Games (Cursor)	11

USI Video Monitors – Green or AMBER 20 MHz hi-res
Dealer and OEM inquiries invited.

REVERSAL (Spracklen) Apple or Atari	25
SARGON II – Apple or TRS-80	26
Data Manager (Lutus)	39
Apple II User's Guide (Osborne)	12
Introduction to Pascal (Sybex)	13
Pascal Handbook (Sybex)	16
Musical Applications of Micros (Chamberlin)	20

DISK SPECIALS

SCOTCH (3M) 5"	10/2.45	50/2.35	100/2.30
SCOTCH (3M) 8"	10/2.60	50/2.45	100/2.40

We stock VERBATIM DISKS
Write for Dealer and OEM prices.

BASF 5" or 8"	10/2.00	20/1.95	100/1.85
Wabash 5"	10/1.80	50/1.75	100/1.70
Wabash 8"	10/2.25	50/2.20	100/2.10

We stock MAXELL DISKS
Write for dealer and OEM prices.

Disk Storage Pages 10 for $4 Hub Rings 50 for $6
Disk Library Cases 8" – $3.00 5" – 2.25

CASSETTES—AGFA PE-611 PREMIUM
High output, low noise, 5 screw housings.

C-10	10/.56	50/.50	100/.48
C-30	10/.73	50/.68	100/.66

SPECIALS

EPSON MX-80 Printer with Graftrax+
EPSON MX-80 F/T Printer with Graftrax+
EPSON MX-100 Printer with Graftrax+

Zenith ZVM-121 Green Phosphor Monitor	109
Okidata and Prowriter printers available	
STARWRITER Daisy Wheel Printer F10	1445
We Stock AMDEK Monitors	
Watanabe Intelligent Plotter	990 6-Pen 1240
Staticide anti-static spray	6
dBASE II	445

ALL BOOK and SOFTWARE PRICES DISCOUNTED

Synertek SYM-1 Microcomputer	**SALE 189**
KTM-2/80 Synertek Video and Keyboard	349
KTM-3/80 Synertek Tubeless Terminal	385

Alspa Computer, Inc.

The price-performance leader. Includes Z80A, 1 or 2 full 8" drives (double density, double sided), 3 serial and 1 parallel port, and winchester port. Prices start at less than $2000. DEALER and OEM inquiries invited.

ZENITH data systems

Z90-80 64K	1995
Z90-82 64K, 1 double dens. drive	2245
Z37 1.3 Megabyte Dual Drive	1355
Z19 Video Terminal (VT-52 compatible)	695
All Zenith Hardware and Software Discounted.	
ZT-1 Intelligent Communications Terminal	**550**
Z100 16-bit/8-bit System	**CALL**

ATARI SPECIALS

800 Computer	675	Microsoft BASIC	72
400 – 16K	269	**MISSILE COMMAND**	**29**
810 Disk Drive	440	**ASTEROIDS**	**29**
825 Printer	625	**STAR RAIDERS**	**34**
850 Interface	170	Space Invaders	29
Inside Atari DOS	18	Music Composer	35
Joysticks or Paddles	19	Caverns of Mars	33
16K RAM (Microtek)	69	**PAC-MAN**	**36**
32K RAM (Microtek)	99	**CENTIPEDE**	**36**
Pilot	65	First Book of Atari	11
Super Breakout	29	**Anchor Modem – Atari**	**85**

Write for prices on other Atari items.

252 Bethlehem Pike
Colmar, PA 18915
215-822-7727
A B Computers

WRITE FOR CATALOG
Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular prices slightly higher. Prices subject to change.

KMMM Pascal for PET/CBM $85

A subset of standard Pascal with extensions.
- Machine language Pascal Source Editor with cursor oriented window mode
- Machine Language P-Code Compiler
- P-Code to machine language translator for optimized object code
- Run-time package
- Floating point capability
- User manual and sample programs

Requires 32K Please specify configuration.

EARL for PET (disk file based) $65

Editor, Assembler, Relocater, Linker

Generates relocatable object code using MOS Technology mnemonics. Disk file input (can edit files larger than memory). Links ,multiple object programs as one memory load. Listing output to screen or printer. Enhanced editor operates in both command mode and cursor oriented "window" mode.

RAM/ROM for PET/CBM

4K or 8K bytes of soft ROM with optional battery backup.

RAM/ROM is compatible with any large keyboard machine. Plugs into one of the ROM sockets above screen memory to give you switch selected write protectable RAM.

Use RAM/ROM as a software development tool to store data or machine code beyond the normal BASIC range. Use RAM/ROM TO LOAD A ROM image where you have possible conflicts with more than one ROM requiring the same socket. Possible applications include machine language sort (such as SUPERSORT), universal wedge, Extramon, etc.

RAM/ROM – 4K	$75
RAM/ROM – 8K	90
Battery Backup Option	20

SUBSORT by James Strasma $35

Subsort is an excellent general purpose machine language sort routine for PET/CBM computers. Sorts both one and two dimensioned arrays at lightning speed in either ascending or descending order. Other fields can be subsorted when a match is found, and fields need not be in any special order. Sort arrays may be specified by name, and fields are random length. Allows sorting by bit to provide 8 categories per byte. The routine works with all PET BASICs, adjusts to any memory size, and can co-exist with other programs in high memory.

SuperGraphics 2.0

NEW Version with TURTLE GRAPHICS

SuperGraphics, by John Fluharty, provides a 4k machine language extension which adds 35 full featured commands to Commodore BASIC to allow fast and easy plotting and manipulation of graphics on the PET/CBM video display, as well as SOUND Commands. Animations which previously were too slow or impossible without machine language subroutines now can be programmed directly in BASIC. Move blocks (or rocketships, etc.), or entire areas of the screen with a single, easy to use BASIC command. Scroll any portion of the screen up, down, left, or right. Turn on or off any of the 4000 (8000 on 8032) screen pixels with a single BASIC command. In high resolution mode, draw vertical, horizontal, and diagonal lines. Draw a box, fill a box, and move it around on the screen with easy to use BASIC commands. Plot curves using either rectangular or polar co-ordinates (great for Algebra, Geometry and Trig classes.)

The SOUND commands allow you to initiate a note or series of notes (or even several songs) from BASIC, and then play them in the background mode without interfering with your BASIC program. This allows your program to run at full speed with simultaneous graphics and music.

Seven new TURTLE commands open up a whole new dimension in graphics. Place the TURTLE anywhere on the screen, set his DIRECTION, turn him LEFT or RIGHT, move him FORWARD, raise or lower his plotting pen, even flip the pen over to erase. Turtle commands use angles measured in degrees, not radians, so even elementary school children can create fantastic graphic displays.

Specify machine model (and size), ROM type (BASIC 3 or 4)

SuperGraphics in ROM $45

Volume discounts available on ROM version for schools.

FLEX-FILE for PET/CBM Computers

Self Calculating DATA BASE
REPORT WRITER
MAILING LIST

FLEX-FILE is a set of flexible, friendly programs to allow you to set up and maintain a data base. Print files with a versatile Report Writer or a Mail Label routine. Programmers will find it easy to add subroutines to their own programs to make use of Data Base files.

RANDOM ACCESS DATA BASE

Record size limit is 250 characters. The number of records per disk is limited only by the size of each record and the number of records per disk is limited only by the size of each record and the amount of free space on the disk. File maintenance lets you step forward or backward through a file, add, delete, or change a record, go to a numbered record, or find a record by specified field. The Find command locates any record when you enter all (or a portion of) the desired key. Field lengths may vary from record to record to allow maximum packing of information. Files may be sorted by any field, and any field may be specified as a key. Sequential files from other programs may be converted to Flex-File format, and Flex-File records may be converted to sequential (WordPro, PaperMate, other word processors may also use Flex-File data). Maximum record size, fields per record, and order of fields may be changed at any time.

MAILING LABELS

With typical record size of 127 characters, each disk can handle over 1000 records (about 2800 with 8050 drive). Labels may be printed any number wide, and may begin in any column position. There is no limit on the number or order of fields on a label, and two or three fields may be joined together on one line (like first name, last name, and title). A "type of customer" field allows selective printing.

REPORT WRITER

Print any field in any column. For numeric fields, use decimal point justification (and round to any accuracy). Define any column as a series of mathematical functions performed on other columns. These functions include arithmetic operations and various log and trig functions. Pass results of operations such as running total from row to row. At the end of the report, print total and/or average for any column. Complete record selection, including field within range, pattern match, and logical functions can be specified individually or in combination with other parameters.

FLEX-FILE BY Michael Riley $80

Please specify equipment configuration when ordering.

PROGRAM YOUR OWN EPROMS

Branding Iron for PET/CBM $79

EPROM Programmer with software for all ROM versions. Includes all hardware and software to program or copy 2716 and 2532 EPROMs.

PORTMAKER DUAL RS232 SERIAL PORT 63

Two ports with full bipolar RS232 buffering. Baud rates from 300 to 4800. For PET/CBM, AIM, SYM.

CBM Software

TCL Pascal Version 1.6	135
Petspeed BASIC Compiler	225
Integer BASIC Compiler	110
CMAR Record Handler	110
UCSD Pascal (without board)	135
Wordcraft 80	300
BPI Accounting Modules	300
Professional Tax Prep Sys.	600
Intelligent Terminal Emulator	25
ASERT Data Base	375
Personal Tax Calc	55
Dow Jones Portfolio Mgmt.	110
Assembler Development	80
Legal Time Accounting	445
Medical Accounting System	445
Atlas 1200 Service & Mainten.	445
Titan Job Cost System	445
Freight Rating and Invoice	445
I.R.M.A.	370

FORTH for PET

BY L. C. Cargile and Michael Riley **$50**

Features include:
- full FIG FORTH model.
- all FORTH 79 STANDARD extensions.
- structured 6502 Assembler with nested decision making macros.
- full screen editing (same as when programming in BASIC).
- auto repeat key.
- sample programs.
- standard size screens (16 lines by 64 characters).
- 150 screens per diskette on 4040, 480 screens on 8050.
- ability to read and write BASIC sequential files.
- introductory manual.
- reference manual.

Runs on any 16K or 32K PET/CBM (including 8032) with ROM 3 or 4, and CBM disk drive. Please specify configuration when ordering.

Metacompiler for FORTH $30

simple metacompiler for creating compacted object code which can be executed independently (without the FORTH system).

PaperMate 60 COMMAND WORD PROCESSOR

by Michael Riley

Paper-Mate is a full-featured word processor for CBM/PET. Paper-Mate incorporates 60 commands to give you full screen editing with graphics for all 16K or 32K machines (including 8032), all printers, and disk or tape drives.

For writing text, Paper-Mate has a definable keyboard so you can use either Business or Graphics machines. Shift lock on letters only, or use keyboard shift lock. All keys repeat.

Paper-Mate text editing includes floating cursor, scroll up or down, page forward or back, and repeating insert and delete keys. Text block handling includes transfer, delete, append, save, load, and insert.

All formatting commands are imbedded in text for complete control. Commands include margin control and release, column adjust, 9 tab settings, variable line spacing, justify text, center text, and auto print form letter (variable block). Files can be linked so that one command prints an entire manuscript. Auto page, page headers, page numbers, pause at end of page, and hyphenation pauses are included.

Unlike most word processors, CBM graphics as well as text can be used. Paper-Mate can send any ASCII code over any secondary address to any printer.

Paper-Mate functions with 16/32K CBM/PET machines, with any printer, and with either cassette or disk.

To order Paper-Mate, please specify configuration.

Paper-Mate on disk or tape 40.00

PaperMate works on 16K VIC and Commodore 64 also.

BASIC INTERPRETER $200

Designed to support the CBM 8096 (8032 with add-on 64K board). A full interpreter implementation to automatically take advantage of the extra memory available to the 8032.

PEDISK II from cgrs Microtech

5" 40 track, 1 drive, 143K	$525
5" 40 track, 1 drive, 286K	690
8" IBM 3740 format, 77 track, 250K	995

JINSAM Data Base Management System for CBM. Comprehensive version available for most configurations.

COPY-WRITER Word Processor 159

Works like expensive word processors, plus has added features like double column printing, and shorthand generator. For PET/CBM and Apple.

CASH MANAGEMENT SYSTEM $45

Easy to use. Keeps track of cash disbursements, cash receipts, cash transfers, expenses for up to 50 categories.

252 Bethlehem Pike
Colmar, PA 18915 215-822-7727 **A B Computers**

WRITE FOR CATALOG

Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular price slightly higher. Prices subject to change.

www.commodore.ca

COMMODORE 64 MEMORY MAP

Compiled by Jim Butterfield, Associate Editor

SID (6581) Commodore 64

V1	V2	V3				V1	V2	V3
D400	D407	D40E	Frequency		L	54272	54279	54286
D401	D408	D40F	Frequency		H	54273	54280	54287
D402	D409	D410	Pulse Width		L	54274	54281	54288
D403	D40A	D411	O O O O	Pulse Width	H	54275	54282	54289
D404	D40B	D412	Voice Type: NSE PUL SAW TRI	Key		54276	54283	54290
D405	D40C	D413	Attack Time 2 ms - 8 sec	Decay Time 6 ms - 24 sec		54277	54284	54291
D406	D40D	D414	Sustain Level	Release Time 6 ms - 24 sec		54278	54285	54292

Voices
(Write Only)

D415	O O O O O	Filter Frequency	L	54293
D416	Filter Frequency		H	54294
D417	Resonance	Filter Voices: EXT V3 V2 V1		54295
D418	V3 Off; Passband: Hi Bd Lo	Master Volume		54296

Filter & Volume
(Write Only)

D419	Paddle X	54297
D41A	Paddle Y	54298
D41B	Noise 3 (Random)	54299
D41C	Envelope 3	54300

Sense

(Read Only)

Special voice features (TEST, RING MOD, SYNC) are omitted from the above diagram.

www.commodore.ca

CIA 2 (NMI) (6526) Commodore 64

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register	Decimal
$DD00	Serial In	Clock In	Serial Out	Clock Out	ATN Out	RS-232 Out			PRA	56576
$DD01	DSR In	CTS In		DCD* In	RI* In	DTR Out	RTS Out	RS-232 In	PRB	56577
	Parallel User Port									
$DD02	IN	IN	Out	Out	Out $3F	Out	Out	Out	DDRA	56578
$DD03	$06 For RS-232								DDRB	56579
$DD04	Timer A								TAL	56580
$DD05									TAH	56581
$DD06	Timer B								TBL	56582
$DD07									TBH	56583
~										
$DD0D				RS-232 In			Timer B	Timer A	ICR	56589
$DD0E								Timer A Start	CRA	56590
$DD0F								Timer B Start	CRB	56591

*Connected but not used by system.

Processor I/O Port (6510) Commodore 64

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register	Decimal
$0000	IN	IN	Out	IN	Out	Out	Out	Out	DDR	0
$0001			Tape Motor	Tape Sense	Tape Write	D-ROM Switch	EF.RAM Switch	AB.RAM Switch	PR	1

CIA 1 (IRQ) (6526) Commodore 64

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register	Decimal
$DC00	Paddle SEL A	Paddle SEL B			Joystick 0 R	L	D	U	PRA	56320
	Keyboard Row Select (Inverted)									
$DC01					Joystick 1				PRB	56321
	Keyboard Column Read									
$DC02	$FF – All Output								DDRA	56322
$DC03	$00 – All Input								DDRB	56323
$DC04	Timer A								TAL	56324
$DC05									TAH	56325
$DC06	Timer B								TBL	56326
$DC07									TBH	56327
~										
$DC0D				Tape Input			Timer B	Interr. A	ICR	56333
$DC0E					One Shot	Out Mode	Time PB6 Out	Timer A Start	CRA	56334
$DC0F					One Shot	Out Mode	Time PB7 Out	Timer B Start	CRB	56335

www.commodore.ca

Hex	Decimal	Description
0000	0	Chip directional register
0001	1	Chip I/O; memory & tape control
0003-0004	3-4	Float-Fixed vector
0005-0006	5-6	Fixed-Float vector
0007	7	Search character
0008	8	Scan-quotes flag
0009	9	TAB column save
000A	10	0=LOAD, 1=VERIFY
000B	11	Input buffer pointer/# subscrpt
000C	12	Default DIM flag
000D	13	Type: FF=string, 00=numeric
000E	14	Type: 80=integer, 00=floating point
000F	15	DATA scan/LIST quote/memry flag
0010	16	Subscript/FNx flag
0011	17	0=INPUT;$40=GET;$98=READ
0012	18	ATN sign/Comparison eval flag
0013	19	Current I/O prompt flag
0014-0015	20-21	Integer value
0016	22	Pointer: temporary strg stack
0017-0018	23-24	Last temp string vector
0019-0021	25-33	Stack for temporary strings
0022-0025	34-37	Utility pointer area
0026-002A	38-42	Product area for multiplication
002B-002C	43-44	Pointer: Start-of-Basic
002D-002E	45-46	Pointer: Start-of-Variables
002F-0030	47-48	Pointer: Start-of-Arrays
0031-0032	49-50	Pointer: End-of-Arrays
0033-0034	51-52	Pointer: String-storage(moving down)
0035-0036	53-54	Utility string pointer
0037-0038	55-56	Pointer: Limit-of-memory
0039-003A	57-58	Current Basic line number
003B-003C	59-60	Previous Basic line number
003D-003E	61-62	Pointer: Basic statement for CONT
003F-0040	63-64	Current DATA line number
0041-0042	65-66	Current DATA address
0043-0044	67-68	Input vector
0045-0046	69-70	Current variable name
0047-0048	71-72	Current variable address
0049-004A	73-74	Variable pointer for FOR/NEXT
004B-004C	75-76	Y-save; op-save; Basic pointer save
004D	77	Comparison symbol accumulator
004E-0053	78-83	Misc work area, pointers, etc
0054-0056	84-86	Jump vector for functions
0057-0060	87-96	Misc numeric work area
0061	97	Accum#1: Exponent
0062-0065	98-101	Accum#1: Mantissa
0066	102	Accum#1: Sign
0067	103	Series evaluation constant pointer
0068	104	Accum#1 hi-order (overflow)
0069-006E	105-110	Accum#2: Exponent, etc.
006F	111	Sign comparison, Acc#1 vs #2
0070	112	Accum#1 lo-order (rounding)
0071-0072	113-114	Cassette buff len/Series pointer
0073-008A	115-138	CHRGET subroutine; get Basic char

www.commodore.ca

007A-007B	122-123	Basic pointer (within subrtn)
008B-008F	139-143	RND seed value
0090	144	Status word ST
0091	145	Keyswitch PIA: STOP and RVS flags
0092	146	Timing constant for tape
0093	147	Load=0, Verify=1
0094	148	Serial output: deferred char flag
0095	149	Serial deferred character
0096	150	Tape EOT received
0097	151	Register save
0098	152	How many open files
0099	153	Input device, normally 0
009A	154	Output CMD device, normally 3
009B	155	Tape character parity
009C	156	Byte-received flag
009D	157	Direct=$80/RUN=0 output control
009E	158	Tp Pass 1 error log/char buffer
009F	159	Tp Pass 2 err log corrected
00A0-00A2	160-162	Jiffy Clock HML
00A3	163	Serial bit count/EOI flag
00A4	164	Cycle count
00A5	165	Countdown,tape write/bit count
00A6	166	Tape buffer pointer
00A7	167	Tp Wrt ldr count/Rd pass/inbit
00A8	168	Tp Wrt new byte/Rd error/inbit cnt
00A9	169	Wrt start bit/Rd bit err/stbit
00AA	170	Tp Scan;Cnt;Ld;End/byte assy
00AB	171	Wr lead length/Rd checksum/parity
00AC-00AD	172-173	Pointer: tape bufr, scrolling
00AE-00AF	174-175	Tape end adds/End of program
00B0-00B1	176-177	Tape timing constants
00B2-00B3	178-179	Pntr: start of tape buffer
00B4	180	1=Tp timer enabled; bit count
00B5	181	Tp EOT/RS232 next bit to send
00B6	182	Read character error/outbyte buf
00B7	183	# characters in file name
00B8	184	Current logical file
00B9	185	Current secndy address
00BA	186	Current device
00BB-00BC	187-188	Pointer to file name
00BD	189	Wr shift word/Rd input char
00BE	190	# blocks remaining to Wr/Rd
00BF	191	Serial word buffer
00C0	192	Tape motor interlock
00C1-00C2	193-194	I/O start address
00C3-00C4	195-196	Kernel setup pointer
00C5	197	Last key pressed
00C6	198	# chars in keybd buffer
00C7	199	Screen reverse flag
00C8	200	End-of-line for input pointer
00C9-00CA	201-202	Input cursor log (row, column)
00CB	203	Which key: 64 if no key
00CC	204	0=flash cursor
00CD	205	Cursor timing countdown

www.commodore.ca

00CE	206	Character under cursor
00CF	207	Cursor in blink phase
00D0	208	Input from screen/from keyboard
00D1-00D2	209-210	Pointer to screen line
00D3	211	Position of cursor on above line
00D4	212	0=direct cursor, else programmed
00D5	213	Current screen line length
00D6	214	Row where curosr lives
00D7	215	Last inkey/checksum/buffer
00D8	216	# of INSERTs outstanding
00D9-00F2	217-242	Screen line link table
00F3-00F4	243-244	Screen color pointer
00F5-00F6	245-246	Keyboard pointer
00F7-00F8	247-248	RS-232 Rcv pntr
00F9-00FA	249-250	RS-232 Tx pntr
00FF-010A	256-266	Floating to ASCII work area
0100-103E	256-318	Tape error log
0100-01FF	256-511	Processor stack area
0200-0258	512-600	Basic input buffer
0259-0262	601-610	Logical file table
0263-026C	611-620	Device # table
026D-0276	621-630	Sec Adds table
0277-0280	631-640	Keybd buffer
0281-0282	641-642	Start of Basic Memory
0283-0284	643-644	Top of Basic Memory
0285	645	Serial bus timeout flag
0286	646	Current color code
0287	647	Color under cursor
0288	648	Screen memory page
0289	649	Max size of keybd buffer
028A	650	Repeat all keys
028B	651	Repeat speed counter
028C	652	Repeat delay counter
028D	653	Keyboard Shift/Control flag
028E	654	Last shift pattern
028F-0290	655-656	Keyboard table setup pointer
0291	657	Keyboard shift mode
0292	658	0=scroll enable
0293	659	RS-232 control reg
0294	660	RS-232 command reg
0295-0296	661-662	Bit timing
0297	663	RS-232 status
0298	664	# bits to send
0299-029A	665	RS-232 speed/code
029B	667	RS232 receive pointer
029C	668	RS232 input pointer
029D	669	RS232 transmit pointer
029E	670	RS232 output pointer
029F-02A0	671-672	IRQ save during tape I/O
02A1	673	CIA 2 (NMI) Interrupt Control
02A2	674	CIA 1 Timer A control log
02A3	675	CIA 1 Interrupt Log
02A4	676	CIA 1 Timer A enabled flag
02A5	677	Screen row marker
02C0-02FE	704-766	(Sprite 11)

www.commodore.ca

0300-0301	768-769	Error message link	
0302-0303	770-771	Basic warm start link	
0304-0305	772-773	Crunch Basic tokens link	
0306-0307	774-775	Print tokens link	
0308-0309	776-777	Start new Basic code link	
030A-030B	778-779	Get arithmetic element link	
030C	780	SYS A-reg save	
030D	781	SYS X-reg save	
030E	782	SYS Y-reg save	
030F	783	SYS status reg save	
0310-0312	784-785	USR function jump	(B248)
0314-0315	788-789	Hardware interrupt vector	(EA31)
0316-0317	790-791	Break interrupt vector	(FE66)
0318-0319	792-793	NMI interrupt vector	(FE47)
031A-031B	794-795	OPEN vector	(F34A)
031C-031D	796-797	CLOSE vector	(F291)
031E-031F	798-799	Set-input vector	(F20E)
0320-0321	800-801	Set-output vector	(F250)
0322-0323	802-803	Restore I/O vector	(F333)
0324-0325	804-805	INPUT vector	(F157)
0326-0327	806-807	Output vector	(F1CA)
0328-0329	808-809	Test-STOP vector	(F6ED)
032A-032B	810-811	GET vector	(F13E)
032C-032D	812-813	Abort I/O vector	(F32F)
032E-032F	814-815	Warm start vector	(FE66)
0330-0331	816-817	LOAD link	(F4A5)
0332-0333	818-819	SAVE link	(F5ED)
033C-03FB	828-1019	Cassette buffer	
0340-037E	832-894	(Sprite 13)	
0380-03BE	896-958	(Sprite 14)	
03C0-03FE	960-1022	(Sprite 15)	
0400-07FF	1024-2047	Screen memory	
0800-9FFF	2048-40959	Basic ROM memory	
8000-9FFF	32768-40959	Alternate: ROM plug-in area	
A000-BFFF	40960-49151	ROM: Basic	
A000-BFFF	49060-59151	Alternate: RAM	
C000-CFFF	49152-53247	RAM memory, including alternate	
D000-D02E	53248-53294	Video Chip (6566)	
D400-D41C	54272-54300	Sound Chip (6581 SID)	
D800-DBFF	55296-56319	Color nybble memory	
DC00-DC0F	56320-56335	Interface chip 1, IRQ (6526 CIA)	
DD00-DD0F	56576-56591	Interface chip 2, NMI (6526 CIA)	
D000-DFFF	53248-53294	Alternate: Character set	
E000-FFFF	57344-65535	ROM: Operating System	
E000-FFFF	57344-65535	Alternate: RAM	
FF81-FFF5	65409-65525	Jump Table, Including:	

FFC6 - Set Input channel
FFC9 - Set Output channel
FFCC - Restore default I/O channels
FFCF - INPUT
FFD2 - PRINT
FFE1 - Test Stop key
FFE4 - GET

©

www.commodore.ca

With this short program for the 5K VIC, you can make any key on the keyboard represent any other key. This gives you the freedom to make an alphabetic keyboard, a numeric keypad, or any keyboard plan you need.

The VIC Keyboard Redefined

Amihai Glazer
Assistant Professor of Economics
University of California
Irvine, CA

You might need to use a numeric keyboard on your VIC. As it is, all numerals are situated on the top row of the keyboard instead of being conveniently arranged in a square pattern which makes data entry easy. This program creates just such a keypad in the center of the keyboard, as shown in Figure 1. Thus, for example, hitting the space bar will be equivalent to hitting "0," and hitting the "R" key will have the same effect as hitting the "7" key.

Not only will the screen show numerals each time the appropriate keys are pressed, but the computer will actually interpret these alphabetic keys as the corresponding numerals. The program also allows the user to redefine *any* key as any other key. You can, for example, rearrange your keys in alphabetical order, or create any keyboard you like.

Type in the program and RUN it. To enable the new interpretation of the keys, type SYS 7424 and hit RETURN. You now have a numeric keypad. To return to a normal keyboard, just hit the RUN and RESTORE keys simultaneously (alternatively, you can execute the statement POKE 655,220: POKE 656, 235). Executing a SYS 7424 will bring back the numeric keypad.

You can also redefine keys of your own choosing. Type GOTO 220 and hit RETURN. Now enter pairs of keys: the key you want changed, and then its new meaning. To stop the program, hit the F1 key. Thus, if you want the key labelled "=" to mean "*", hit the "=" key, then the "*" key, and then the "F1" key. To turn on these new definitions, type SYS 7424 and hit RETURN.

What's Happening

The program works as follows. Normally, during interrupt processing every sixtieth of a second, the VIC calls the decode logic machine language program, whose address ($EBDC) is in the jump vector in locations $028F-$0290. Our machine language program in locations $1D00-$1D14, however, sends the VIC to another machine language program we've put in locations $1D15-$1D24.

This program picks up the code for the key just pressed, given in location $CB. It then indexes into a recode table (beginning in location $1D27, decimal 7463), and puts the new code back into location $CB. Processing continues by jumping into the normal decode logic program in ROM, which is at location $EBDC.

The program's Lines 10-110 insert these two machine language programs into memory. Lines 120-140 initialize the recoding table, and lines 150-200 recode the keys in the form shown in Figure 1. Custom recoding by the user is provided for in lines 220-330. The recoding table is initialized in lines 230-250. CO$ and CN$ get the key that is being redefined, and its new definition. The codes the VIC uses for these keys are obtained from location 203 ($CB); CO and CN are assigned these values. A code of 39 (representing the "F1" key) stops the program. The appropriate changes in the recoding table, which will be used by the machine language program, are performed in lines 310-320.

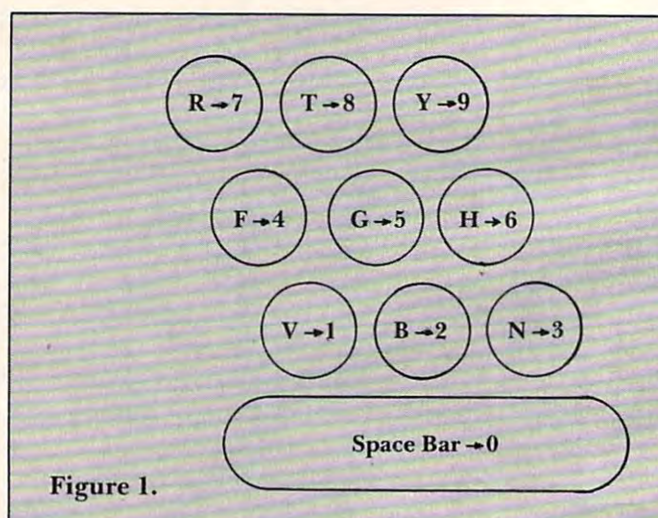


Figure 1.

```
10 REM CHANGE KEYBOARD
20 POKE 52,29: POKE 56,29:CLR
30 FOR I=7424 TO 7462
```

www.commodore.ca

PRACTICAL PROGRAMS FOR THE VIC 20* & COMMODORE 64

TOTL software

WORD PROCESSING

TOTL.TEXT 2.0 **$25.00**

Full capability word processing
- Margin and spacing control
- Centered title lines
- Indentation and tabs
- Upper and lower case and graphics
- Full screen editing
- Scrolling up and down
- No limit to document length

TOTL.TEXT 2.5 **$35.00**

Enhanced from TOTL. TEXT 2.0:
- Heading lines (up to 4)
- Footing line
- Footnotes
- Keyboard Input
- Justification to right margin
- Additional working memory

KEY WORD CROSS REFERENCE

RESEARCH ASSISTANT 2.0 **$25.00**
- Great for authors, students
- Compile reference notes

RESEARCH ASSISTANT, cont'd.
- Save related bibliographical data
- Quick cross reference by keyword

MAILING LIST and LABELS

TOTL.LABEL 2.0 **$20.00**

Easy editing
- User defines label size
- Automatically sorted
- Optional non-printing data line
- Select labels for printing

TIME MANAGEMENT— SCHEDULING, REMINDERS

TOTL TIME MANAGER 2.0 **$25.00**

Enter and edit activities
- Use for dates and times
- Inquiry by date, person, project
- Bar chart reports from 2 days (24 hours) to 4 years
- Print reports by activity, person or project
- Reports may also be sorted by dates
- 56 possible report formats

All programs require 8K expansion and cassette. Designed for tape and disk input/output and the VIC printer. Modification list available for RS232 printers. All programs shipped on cassette tape. Specify machine.

Dealer Discounts Available

SHIPPING INCLUDED.
California Residents add 6% Sales Tax.
$3.00 charge for COD.

Send check or money order to:
TOTL Software
P.O. Box 4742
Walnut Creek, CA 94596
Call (415) 943-7877

*VIC 20 and Commodore 64 are trademarks of Commodore Business Machines.

ATTENTION COMMODORE 64 AND COMMODORE MAX PROGRAMMERS!

Academy Software is now actively seeking quality programs for these new computers. We are especially interested in educational, musical, utility and high quality original game programs. Submit samples attention New Program Mgr. All submissions will be held in strict confidence.

FOR THE VIC-20®

ALL PROGRAMS ON CASSETTE AND RUN IN UNEXPANDED VIC

- **TYPING TUTOR - $12.95**
  If you've ever wanted to learn touch typing, this is for you! Makes learning the keyboard easier. 4 programs on one tape teach the keys in the correct progression. Automatically advances to new keys as your skills develop. Highly praised by customers. "Fantastic", "Excellent", "High quality".
- **WORD INVADERS - $10.95**
  Sharpen up your touch typing skills by blasting the invading words out of the sky before your base is destroyed. Four levels of difficulty match the letters as learned on our TYPING TUTOR program. Typing can be fun!
- **FLASHCARD MAKER & FLASHCARD QUIZ - $10.95**
  2 programs on one tape allow you to prepare your own study material and make it easier to learn. Quiz program has options for study, full test and easy learning mode. Keeps score and allows re-test of missed questions or entire set. Used by school systems. Includes sample data tape with 50 states and their capitals.

Shipping & handling $1.00 per order
California residents add 6% sales tax

Visa and Mastercard orders must include full name as shown on card, card number, and expiration date.

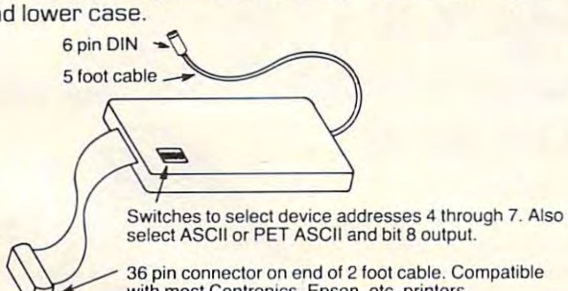
ACADEMY SOFTWARE
P.O. BOX 9403
SAN RAFAEL, CA 94912 **(415) 479-4703**

NEW CBM-64 AND VIC-20 PRODUCTS

*** CBM-64/VIC-20 PRINTER INTERFACE ***

Model MW-302 $119.95

This unit will interface your VIC-20 or CBM-64 to standard parallel printers such as Epson, Centronics, C. Itoh, and many others. Allows printing of full upper and lower case.



MW 302 PRINTER INTERFACE

*** SYSTEM 310 *** $1195.00

Consists of CBM-64, MX-80, Printer interface MW-302, and C2N cassette
(CBM-64 sold separately for $595)

*** DIGITAL TO ANALOG CONVERTER OR 8 CHANNEL ANALOG TO DIGITAL CONVERTER ***

Model MW-304v $129.95

Allows your VIC or CBM to output analog signals or digitize up to 8 analog channels.

MICRO WORLD ELECTRONIX INC
6340 W. Mississippi Ave.
Lakewood Colorado 80226 **(303) 934-1973**

VIC-20® SOFTWARE

AT A PRICE YOU'LL LOVE

You broke the price barrier when you bought your VIC-20* now we break the software barrier by offering...

3 GAMES **PLUS** a detailed instruction booklet **FOR ONLY $9.95**

Ontario residents add 7% sales tax **ADD $.75 FOR SHIPPING**

SPECIFY JOYSTICK OR KEYBOARD VERSION AND SEND TO...

SUPERCHOMPER - Munch your way around
REMEMBER - Version of Simon, 5 skill levels
SEAWOLFE - Sink various enemy ships

HARLI SOFTWARE
1740 GARDEN BRIAR COURT
#RR2 THUNDER BAY
ONTARIO, CANADA P7C 4V1

"An outstanding example of the excellent hi-res graphics and realistic sounds possible on the unexpanded VIC-20."*

Dealer inquiries welcome *Trademark of Commodore

Intelligent Software

For the Commodore VIC-20, 64, PET, and CBM

Honestly, Bill, you've gone too far. Now who is going to believe that **Mail List** allows random access to 1150 files per disk, interfaces with **Word Processor Plus** to provide mass mailing capabilities, runs on ALL Commodore machines (V2 BASIC; min. 16K, disk drive, printer req'd.), and (you can't be serious!) costs just $50, INCLUDING the word processor?

Well, we'll just have to see.

Also available: **Copycalc** (an electronic spreadsheet), $20 ($15 w/another program); **W/P** (12K, printer req'd.), $30; **BB MGR** (12K req'd.), $20; more. All prices include shipping; Calif. residents add 6%. **Please** specify hardware configuration. Catalog available.

William Robbins, Box 3745, San Rafael, CA 94912

. . . when you're through playing games.

www.commodore.ca

```
40 READ D
50 POKE I,D
60 NEXT I
70 REM MACHINE LANG.
80 REM PROGRAM
90 DATA 120,8,72,138, 72,169,21,141,143,2,169
   ,29,141,144,2,104,170,104
100 DATA 40,88,96,8,72,138,72,166,203,189,39,2
    9,133,203,104,170,104,40
110 DATA 76,220,235
120 FOR I=0 TO 64
130 POKE 7463+I,I
140 NEXT I
150 FOR I=1 TO 10
160 READ CO,CN
170 POKE 7463+CO,CN
180 NEXT I
190 REM RECODED KEYS
200 DATA 32,60,27,0,35,56,28,1, 42,57,19,2,43,
    58, 10,3,50,59,11,4
210 END
220 REM CUSTOM RECODE
230 FOR I=0 TO 64
240 POKE 7463+I,I
250 NEXT I
260 PRINT "INPUT OLD, NEW"
270 GET CO$:IF CO$="" THEN 270
280 CO=PEEK(203):IF CO=39 THEN STOP
285 PRINT CO$;" ";
290 GET CN$: IF CN$="" THEN 290
300 CN=PEEK(203)
310 PRINT CN$
320 POKE 7463+CO,CN
330 GOTO 270
```

©

VIC PET APPLE SOFTWARE

GRAPHVICS - super graphics package adds 18 commands to VIC BASIC. Plot 152 x 160 points. Hires & Multicolor modes on same screen! Text & graphics screens. Save/load pictures to/from tape or disk. Req. 3K/8K expander. W/sample programs & user's manual. $25 [$30].

VIC/PET VIGIL - Interactive Games Language - Program your own or play the 9 games included. With 60+ powerful commands. Easy to learn. VIC version has color and sound (requires 3K/8K expander). Complete with user's manual. $35 [40].

VIC/PET PIPER THE MUSIC MACHINE - Simplest way to compose, conduct and play music. Complete control of notes, rests, volume, repeats, tempo. W/sample compositions and user's manual. $25 [$30].

VIC HIRES/MULTICOLOR GRAPHICS UTILITIES - Add graphics to VIC BASIC. Requires NO extra memory. Plot points, lines and boxes in fine detail. 104 x 152 points. W/sample programs and manual. $20 [$25].

NEW **VIC/PET TINY BASIC COMPILERS** - Produces true 6502 code. Subset of BASIC supports all floating point operations. Compiler listing optional if you have memory (16K PET, 8K expander VIC). For OLD, NEW, 4.0, 8032 PET or VIC with 3K/8K expander. $25 [$30].

NEW **VIC JOYSTICK DRAWING** - Paintbrush for VIC MULTICOLOR mode pictures. Requires 3K/8K expander and a joystick. $15 [$20].

NEW **I-CHING for VIC** - colorful fortune teller gives you insite into your life from an Oriental perspective. Includes manual and 275 page guide. Requires 8K expander. $30 [$35].

NEW **VIC BASIC Reference Card** - Only $2.50 [$3.00].

PET MACHINE LANGUAGE GUIDE - hidden talents in your Old, New or 4.0 ROM PET/CBM. 30+ routines fully detailed. $9 [$11].

PET TINY Pascal PLUS+ - structured language. Editor, Compiler and Interpreter. All programing constructs and graphics. For NEW 4.0/8032 ROMS & 32K memory. Disk $50 [$55], cassette $55 [$60].

NEW **APPLE II DYNASOFT PASCAL** - complete dev't system. Editor, Compiler, Interpreter & Supervisor. Data types: scalars, char, array, pointer, integer. Hires, Lores, machine language interface, sample programs, user's manual. Disk $50 [$55]; W/optional source code $85 [$90].

VIC BUDGETEER - Get control of your expenses with this visual planner. Requires 3K/8K expander. Available September. $25 [$30].

VIC MACHINE LANGUAGE GUIDE - Available September. $10 [$12].

ABACUS SOFTWARE
P.O. Box 7211
Grand Rapids, Michigan 49510
616/241-5510

master charge | VISA

ORDERING INFORMATION:
FREE POSTAGE. Unless noted, prices are for cassette. **Add $3.00 per DISK package.** Foreign prices in [ ]. Manuals available separately for inspection, creditable towards purchase of software $5.00 each [$7.00 foreign]. All orders must be prepaid in US Dollars via International Money order or by VISA, MC, ACCESS, Eurocard

ComputAbility

presents the Newest in Joystick Products

Starfighter™ & SLIK STIK™

Starfighter™

- 2 Year Warranty
- Last Longer
- More Accurate
- Less Fatiguing
- Easier to Hold

$16.95

LEFTY JOYSTICK ADAPTOR

Adapts to *any* Atari type joystick. Moves fire button to top right.

$9.95

EXTENSION CABLE (5 ft.)

Adapts to *any* Atari controller.

$6.95

ALSO CARRY A LARGE SELECTION OF ATARI AND COMMODORE SOFTWARE.

MasterCard/Visa

Order Toll Free
800/558-0003

In Wisconsin Call
414-351-2007

SLIK STIK™

- 90 Day Warranty
- Same as Starfighter except made of softer materials

$9.95

ALL JOYSTICKS WORK WITH:

- Atari VCS
- Sears Telegame
- Commodore VIC-20
- Atari 400/800

All products above have registered Trademarks

Send Check or Money Order to ComputAbility, add $2.50 shipping and Handling, WI residents add 5% sales tax. Price and availability are subject to change without notice. 30 Day Money Back Guarantee.

Order Hours: 12 pm – 9 pm
Mon.-Fri.
CST

ComputAbility
P.O. Box 17882
Milwaukee, WI 53209

www.commodore.ca

Atari Rainbow: Colors By Page Flipping

Robert W. Myers
Charlotte, NC

Have you ever wanted more colors than are provided on your Atari? Here's how you can mix colors to produce new colors. The demonstration program uses four colors in Graphics mode 2, which are mixed two at a time to produce a total of ten different colors.

Blending Colors

All this color, like most everything on the TV screen, is really an illusion. The blending of colors takes place because the displays are changed back and forth so fast that our eyes cannot keep up with the changes. Therefore, we see only one color, which is a mixture of the colors in all the different displays. You can mix more than two colors at a time, but as the number of displays increases, the amount of flicker on the screen increases too. The practical limit is four displays mixing at once. But the ten colors that my program produces seem like a rainbow compared to the four colors normally allowed by the CTIA chip.

This mixing is done by using multiple screen RAM areas and changing the Load Memory Scan (LMS) bytes in the display list during the Vertical Blank Interrupt. I realize that this sounds like a very complicated thing to do, but it's not.

Understanding The Display List

The Display List is a program for the ANTIC chip, which is a microprocessor that controls the TV screen so that the 6502 can be free to spend more of its time doing computational chores. The Display List is in RAM, and the first byte of the Display List can be found at PEEK(560)+256*PEEK(561).

Usually you will find that the first three bytes are the code that causes the black area at the top of the screen (to insure that nothing is lost due to overscan of the TV). The next byte is the LMS byte which sets the D6 bit (64 decimal). Added to this 64 is the ANTIC Graphics mode number, which is given in Table 1.

The LMS is a three-byte instruction. The 64 + mode# is the first byte; the second and third bytes are the address of the beginning of screen RAM.

This address is what we are interested in here. Rapidly changing it allows us to switch from one picture to another and back. We cannot do this address swapping from BASIC; it is far too slow. The LMS bytes are changed by a short machine language routine that is run 60 times a second while the picture is blanked out as it returns to the top of the screen to begin the next frame. This is *Vertical Blank Interrupt.*

The routine loads the LMS bytes with the address of the first (normal) screen RAM, then it does an exclusive-or with one of the memory locations. This causes the memory location to toggle between 0 and 1. This 0 or 1 is used to determine whether a branch will be taken or not. If the branch is taken, the next instruction is JMP $E462, which puts the interrupt back in normal operation. If the branch is not taken, then the LMS bytes are changed to the address of the other (alternate) screen RAM. Then comes the JMP $E462.

Using VBI

The VBI is amazingly easy to use. All you do is write your routine that is to run during the interrupt. Then write a machine language program that puts the high byte of your routine's address into the X-register, the low byte into the Y-register, and the number seven into the accumulator. Finally you JSR $E45C. This second machine language program is at lines 160, 170, and 180 of my program.

After setting up your VBI to change the LMS, you print or plot and move one set of your screen RAM to the other (alternate) location that you have specified to the LMS. This technique should be usable with any multicolor display mode or any combination of display modes not only to mix colors, but also to mix text and graphics, to display mixed resolutions, etc.

ANTIC Graphics Mode Numbers

BASIC mode#	0	1	2	3	4	5	6	7	8
ANTIC mode#	2	6	7	8	9	10	11	13	15

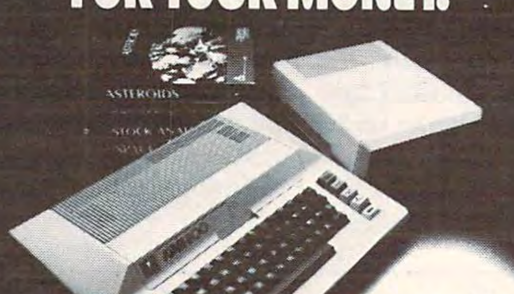
```
1 REM ***************************
2 REM *{25 SPACES}*
3 REM *  MIXING COLORS TO MAKE  *
4 REM *{4 SPACES}AN ATARI RAINBOW
  {5 SPACES}*
5 REM *{10 SPACES}by{13 SPACES}*
6 REM *{4 SPACES}ROBERT W. MYERS
```

www.commodore.ca

```
   {6 SPACES}*
7 REM *{25 SPACES}*
8 REM ***************************
9 REM
10 GRAPHICS 2+16:BREAK=1000
15 REM MACHINE LANGUAGE TO BE RUN DURING
    VERTICAL BLANK{9 SPACES}INTERRUPT
20 FOR I=0 TO 36:READ A:POKE 1536+I,A:NE
   XT I
30 DATA 173,39,6,141,49,6,173,40,6,141,5
   0,6,173,51,6,73,1,141,51
40 DATA 6,240,12,173,41,6,141,49,6,173,4
   2,6,141,50,6,76,98,228
45 REM FIND DISPLAY LIST IN RAM
50 DLIST=PEEK(560)+256*PEEK(561)
55 REM MODIFY MACHINE LANGUAGE PROGRAM B
   Y POKEING IN ADDRESSES  FROM DISPLAY
   LIST
60 BYTE=DLIST+4:GOSUB BREAK:REM LOAD MEM
   ORY SCAN LOW BYTE
70 POKE 1540,LOW:POKE 1562,LOW
80 POKE 1541,HIGH:POKE 1563,HIGH
90 BYTE=DLIST+5:GOSUB BREAK:REM LOAD MEM
   ORY SCAN HIGH BYTE
100 POKE 1546,LOW:POKE 1568,LOW
110 POKE 1547,HIGH:POKE 1569,HIGH
120 BYTE=DLIST+20:GOSUB BREAK:REM NORMAL
     SCREEN RAM
130 POKE 1576,HIGH:POKE 1575,LOW
140 BYTE=DLIST-250:GOSUB BREAK:REM ALTER
    NATE SCREEN RAM
150 POKE 1578,HIGH:POKE 1577,LOW
155 REM MACHINE LANGUAGE PROGRAM TO INIT
    IALIZE VERTICAL BANK{4 SPACES}INTERR
    UPT
160 FOR I=0 TO 10:READ A:POKE 1600+I,A:N
    EXT I
170 DATA 104,162,6,160,0,169,7,32,92,228
    ,96
180 X=USR(1600)
220 REM DRAW FIRST SCREEN
240 POSITION 0,4
250 PRINT #6;"ATARi coMPUTER Club"
260 PRINT #6
270 PRINT #6;"{4 SPACES}OF CHARLottE"
275 REM MOVE FIRST SCREEN TO ALTERNATE S
    CREEN RAM
280 FOR I=0 TO 240
290 POKE DLIST-250+I,PEEK(DLIST+20+I)
300 NEXT I
305 REM SETCOLORS AND DRAW SECOND SCREEN
312 SETCOLOR 0,12,6
313 SETCOLOR 1,4,6
314 SETCOLOR 2,15,8
315 SETCOLOR 3,8,6
320 POSITION 0,4
330 PRINT #6;"AtARi COMPUTeR CLUB"
340 PRINT #6
350 PRINT #6;"{4 SPACES}OF CHArLOttE"
359 REM HOLD IMAGE ON SCREEN
360 GOTO 360
999 REM SUBROUTINE TO BREAK DOWN NUMBER
    INTO HIGH AND LOW BYTES
1000 HIGH=INT(BYTE/256)
1010 LOW=BYTE-HIGH*256
1020 RETURN
```

©

NOW ATARI GIVES YOU MORE COMPUTER FOR YOUR MONEY.



Any way you look at it, the affordable ATARI 800™ Home Computer offers you the best value around. With its expandable memory capacity (up to 48K), the ATARI 800 Home Computer offers you a full range of sophisticated programs spanning everything from home finance to home entertainment.

A variety of easy-to-connect accessories lets you expand your ATARI 800 Home Computer as your requirements grow.

ATARI $699

Plan and analyze your budget, organize your files or balance your books. The ATARI 800 Home Computer is so versatile and easy to use, the sky's the limit.

But why stop there?

Enjoy Star Raiders,™ Asteroids,™ Missile Command™ and the sensational home version of Pac-Man,™ computer games that make the most of the ATARI 800 Home Computer's exceptional color, graphics, sound and animation capabilities.

Whatever your computer needs, the ATARI 800 Home Computer has what you're looking for.

ATARI HOME COMPUTERS
We've Brought The Computer Age Home.™

Available now at: ICC INTERNATIONAL COMPUTER CENTER

MEADOW PARK PLAZA / 22811 HAWTHORNE BLVD., TORRANCE, CA 90505
(213) 373-6771 / TELEX: 194561

RAM

For ATARI

48K RAM BOARD FOR THE 400 with Lifetime Warranty

- Highest quality available
- Reduces power consumption
- Reduces heat

48K Board (400) **$240**

32K Board (400/800) **$100**

FREE SHIPPING ANYWHERE IN U.S.A.

INTEC PERIPHERALS CORP

906 E. Highland Ave.
San Bernardino, CA 92404
(714) 881-1533

MasterCard VISA

ATARI, 400, 800 are Trademarks of ATARI, Inc.

www.commodore.ca

ATARI 400/800

This disk and program
was destroyed by a simple spill of milk.

NOW YOU CAN PREVENT A TOTAL LOSS

The Replica 1+1 Backup and Utilities Systems Package makes it possible for the user to build a complete reserve library to replace a damaged or faulty disk when necessary.

PROGRAM FEATURES

DISK BACKUP: Copies the sections of the disk which contain data, and also permits read and write without verify, speeding up the backup process considerably. Copies 'bad sectors' with ease. UTILITIES PACKAGE: Scans any ATARI disk, providing the user with information as to which sectors contain data, which are empty and which are 'bad'. Includes a sector editor which allows the user to see and modify data within a sector. Disassembles a sector into machine language op codes. Searches for a series of bytes or a string within a section of a disk. Includes a custom format scan and patch analysis enabling the user to backup any presently available* ATARI computer disk. And much more. . . .

To receive the Replica 1+1 package and documentation fill out the coupon below and send $50 to:

C1

NUDMEHI SOFTWARE
495 Old York Road Suite 425
Jenkintown, Pa. 19046

Visa and Mastercard Phone Orders: 215-635-2722

NAME ____________________

ADDRESS ____________________

CITY ____________ STATE ________ ZIP ________

☐ MY CHECK FOR $50 IS ENCLOSED
☐ BILL BY VISA # ____________________
EXPIRATION DATE ____________________
☐ BILL MY MASTERCARD # ____________________
EXPIRATION DATE ____________________

____
CARDHOLDER'S SIGNATURE

*As of 8/15/82. Subsequent updates will become available at a nominal fee.
Please Note: It may be necessary to adjust your drive speed when backing up certain disks.

THE REPLICA 1+1 IS NOT INTENDED FOR THE ILLEGAL DUPLICATION OF COPYRIGHTED MATERIALS

ATARI is a trademark of ATARI, INC.

www.commodore.ca

There's nothing wrong with the way the CBM/PET/VIC writes data to sequential files. But sometimes it can be useful to pack the data in order to save space or aid certain types of processing.

PACK UP YOUR DATA

Jim Butterfield
Associate Editor

If your program contains a statement like PRINT#1,V and if you execute that statement when V contains a value of, say, 159, five characters will be placed on the file: Space, 1, 5, 9, and RETURN. Caution: if you don't have a 4.0 BASIC, one more character will be put to the file – a Line Feed – and it may give you problems. In this case, your program should say PRINT#1, V;CHR$(13); and be sure to include both semicolons. This applies to the VIC as well as to earlier PET/CBM units.

This is ideal for many purposes. An INPUT#.. statement executed at a later time will receive the characters just as if you had typed them on the keyboard, and the value of 159 will be input. All neat and orderly. What's more, the file is made up of conventional ASCII characters: it may be manipulated by text editors, sent to a communications line, or handled in a number of conventional ways.

But occasionally – rarely! – we might find a need to change the rules. We might have a utility program (notably a sort routine) that wants to handle the data in "columns" as if it were on a punched card. In this case, we would want to organize our data more formally. On the opposite side of the coin, we might need to crunch our data – it's very large and the file size is becoming a problem.

Formatted Data

Normally we would write the various fields of a computer record as individual items. To write name, initials, address, and balance, we might write:

```
PRINT#1,N$
PRINT#1,I$
PRINT#1,A$
PRINT#1,B
```

and it's written. Corresponding INPUT# statements would bring it back when needed. It's fairly compact and not hard to handle.

If we wanted to go into "fixed column" format, we'd need to make decisions. The name might be fitted into columns 1 to 15; the initials into columns 16 to 18; the address into columns 19 to 40; and the balance into columns 41 to 46. Now that we've made the decisions, we must pack the data that way.

Each field of data must be fitted to the fixed size. If the name were too long, we would need to trim it back with LEFT$(N$,15); if it were too short, we'd need to extend it with spaces by coding N$+"   ". We can do both together by writing LEFT$(N$+"   ",15). We must be sure to allow enough spaces to fill needed space; it's most convenient to define a lot of spaces as S$, which will make our coding more compact.

Names must align on the left, so that the *B* of BUTTERFIELD will fall into the same column as the *P* of PUNTER; in this way, a column sort will place the two names in correct alphabetic order. Numeric values must go the other way: 123 and 45 must be placed so that the *3* and the *5* digits are lined up. This is called "right justification" and is done with the RIGHT$ function: RIGHT$("   "+STR$(B),6). One caution on numerics: be careful with fractions; it's usually better to change everything to integer values, such as cents rather than dollars-and-cents.

The whole record then becomes:

```
S$="                    " (spaces)
R$=LEFT$(N$+S$,15)+LEFT$(I$+S$,3)+LEFT$
   (A$+S$,22)+RIGHT$(S$+STR$(B),6)
```

Note that, in this case, every record will be exactly 46 characters long.

When we read this record (one INPUT# statement will do the job), we must extract the various fields. This is quite easy if we use the MID$ statement:

```
N$=MID$(R$,1,15)
I$=MID$(R$,16,3)
A$=MID$(R$,19,22)
B=VAL(MID$(R$,41,6)
```

The strings will be their original values, except that they will be padded out with extra spaces to make up the specified length.

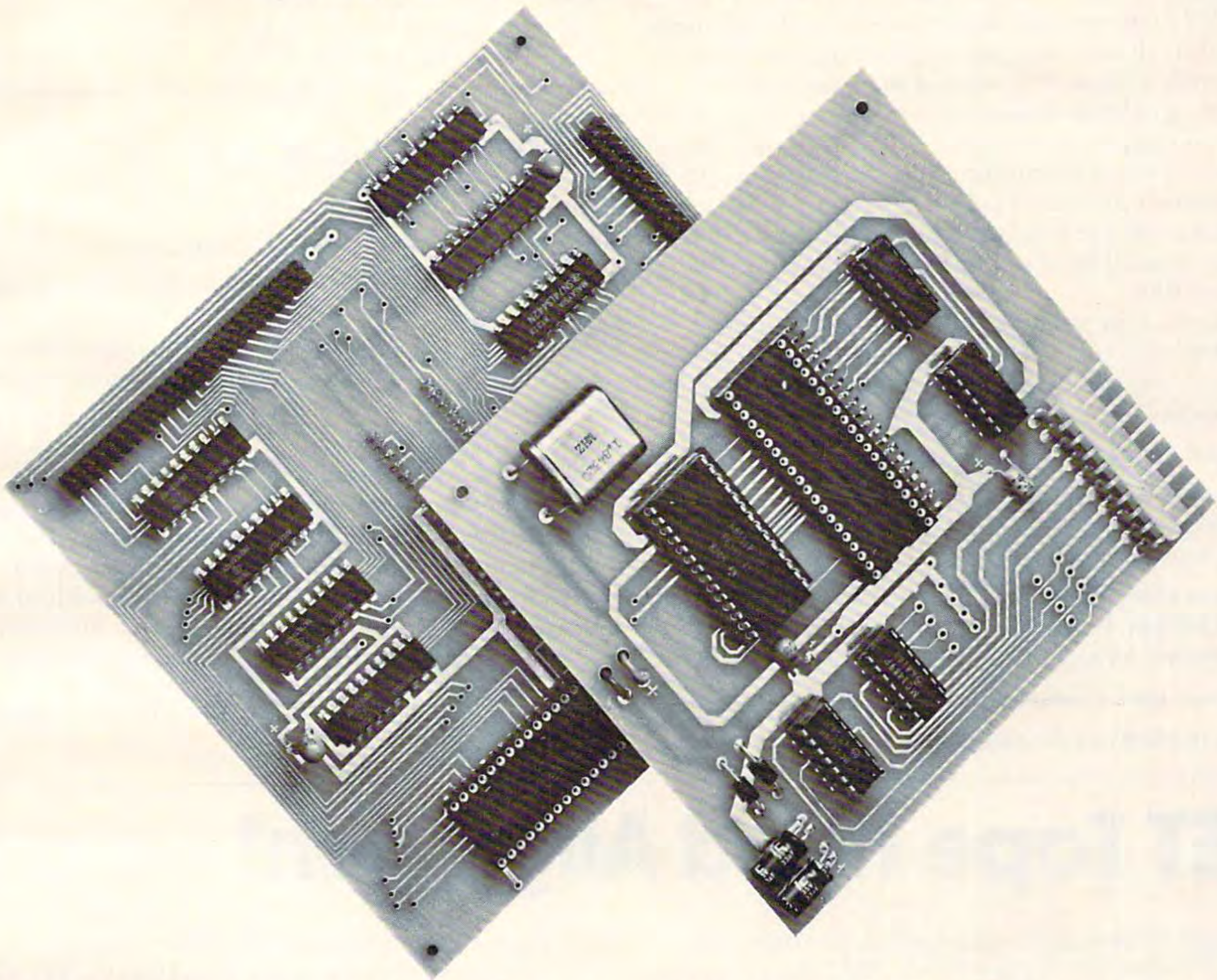
Packing Them In

In contrast to the previous formatting, binary packing saves space. It makes the information almost indecipherable, however, unless you have the key. Also, as we crunch the information together, we lose the capability to manipulate the data with other programs, since what we are writing is not readable ASCII.

The principle is this: why store a value like 169 in five bytes of storage when the binary value of 169 will fit into one byte? It's a dangerous road. We must be sure to leave enough space for the size of the number we plan to hold. Two bytes, for example, will hold an integer value from zero

www.commodore.ca

NEW FROM KILO



PET/CBM Expansion Adaptor *\$139.00 (order KEACom)*
is a true expansion BUS that allows:
- Direct memory access
- Multi-processing
- I/O expansion
- Memory expansion
- Alternative operating systems

Write or call for application notes explaining full capabilities. Great for those PET/CBM's with soldered-in ROMS.

Upgrade to KTERM + from KTERM
\$47.00 (order KTERM-U)

Other software for PET/CBM users
Run time linking loader *\$33.95 (order KRTLL)*
Allows: Linked loading of BASIC programs anywhere in memory; runtime linking of BASIC program segments; saving BASIC machine code loads during BASIC execution. Creates M.C. virtual memory

Screen form *\$49.00 (order KSF)*
Create invoice and other forms by using the cursor to layout CRT input/output display fields. Avoid input statement problems. Avoid programming get commands formatted printing. Programmed controlled status line similar to WORDPRO'S.®

RS-232 Adaptor for PET®/CBM® *\$89.00 (order KUART)*
- Bidirectional, full duplex serial port
- Uses 6551 ACIA baud rates 50 to 19200, software selectable
- Plugs into processor socket

Special Offer
Order RS232 adaptor by November 30 and we will ship KTERM for the price of KTERM+

Full use of 6551 features including
- Interrupts
- Modem control lines

Serial port and modem control lines use RS232 voltage levels.

Cable – 2 foot-PC board connector to RS232
\$19.00 (order KUART-C)

Software support for RS232 adaptor
KTERM – simple terminal program *\$14.00 (order KTERM)*
Use your PET/CBM and a modem to The Source and other networks

KTERM in ROM (specify address) *\$27.00 (order KTERM-R)*

KTERM + – Use your PET/CBM as your own information management system. Send and receive disk files over telenetworks. Supports ASCII, WORDPRO, MAE and PETSKI file formats. Use to print to serial printers. Convert PETSKI to ASCII. Print a file or transmit it as background task under interrupt control while using your computer. *\$57.00 (order KTERM +)*

KILO CORPORATION **P.O. BOX 7530** **ANN ARBOR, MI 48107**
Visa and Mastercard accepted • Phone orders welcomed (313) 668-1566

www.commodore.ca

to 65535.

When we print binary values to a file, we must abandon all our "normal" formatting rules. For example, a value of 13 stored in binary will be indistinguishable from a RETURN character, so we won't be able to use the INPUT statement to read it. A word of caution to cassette tape users: two characters cannot be written to tape files: CHR$(10) (Line Feed) and CHR$(0) (Null). This makes cassette tape of limited use in building packed files.

Let's write some packed numbers to a file. We'll assume that the numbers will fit into two bytes, so the values will range from zero to 65535. We'll write ten numbers to a binary file:

```
100 OPEN 1,8,2,"0:DATBIN,U,W"
```

Note that we designate the file as type USR (User). This is the same as Sequential. We just want to mark it as being in unusual format.

```
110 FOR J=1 TO 10
120 INPUT V:IF V<0 OR V>65536 GOTO 120
130 V%=V/256:L=V%*256
```

We have split V into low and high bytes.

```
140 PRINT#1,CHR$(L);CHR$(V%);
```

Don't forget the semicolons.

```
150 NEXT J
160 CLOSE 1
```

Ten numbers have been written into 20 bytes. Now let's read them.

```
100 OPEN 1,8,2,"DATBIN,U,R"
110 FOR J=1 TO 10
120 GET#1,A$,B$
```

We must use GET; INPUT can't cope.

```
130 PRINT ASC(A$+CHR$(0))+ASC(B$+CHR$(0))*
    256
```

The CHR$(0) is needed to allow for zeros; they will be received by the GET statement as a null string.

```
140 NEXT J
150 CLOSE 1
```

We've just coded numbers very compactly. One hundred numbers would fit into 200 bytes or one disk sector. Similar numbers in conventional sequential files would take up three or four sectors.

Most of the time, you'll want to stay with ordinary data files. They are more orderly and easier.

But you can build special types of files if you wish. Formatting and compacting are perfectly logical manipulations. Use them with care – when you need them.

PET Tape Head Alignment

Louis F. Sander
Pittsburgh, PA

Though Commodore tape systems are quite reliable, if you're having problems LOADing programs which were *not SAVEd on your cassette drive,* the culprit is probably head alignment. Here's a simple technique to eliminate most of the trial and error from the process of PET head alignment.

Connect an audio amplifier between the READ and GND pins on your tape player. (See Figure 1.) This will allow you to listen to your tapes as they are loaded, or even when the tape is running under manual control.

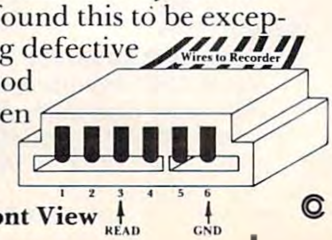
Then do your head alignment *while the tape is running.* As you turn the adjustment screw, the sound quality will make a definite transition from mushiness to crispness and back to mushiness. At the center of the crisp area, the tape and the head are in excellent alignment, and LOADing should be easy. I have used this method for several months and have found it to be useful for aligning my head to *any* foreign tape; then by using one of my own previously recorded tapes, I can easily return my head to its original state.

If you are using an amplifier to produce sound with your PET, that amplifier will work perfectly for head alignment. I use a Radio Shack #277-1008 cigarette-pack-size unit that I bought for $11.95, but almost any audio amplifier will do the job.

The cassette connector's READ and GND pins are identified in Figure 1. They are also plainly labeled on most PET printed circuit boards, right at the place where the recorder connector plugs in. You can make your connections to these pins by temporarily inserting wires or straightened paper clips into the back of the connector, where they will be able to touch the metal pins, or by contacting the appropriate points on the printed circuit board itself. Connect the shielded or grounded amplifier input wire to GND, and the other one to READ.

The voltages on these pins are very low, so there is no electrical shock hazard, but for PET's sake you should be very careful not to touch READ and GND together, or to apply any external voltages to these points.

An extra benefit of connecting the audio amplifier to the tape deck is that it lets *you* hear whatever PET hears. I've found this to be exceptionally useful for detecting defective tapes, for guaranteeing good LOADs, and for telling when PET is searching a blank area on the cassette.



PET Cassette Connector, Front View

www.commodore.ca

SYSRES™

THE ULTIMATE RESIDENT PROGRAM MANIPULATION SYSTEM FOR PET™/CBM™ MICROCOMPUTERS

SO COMPLETE, EVEN THE BEST OF THE COMPETITION DOESN'T COMPARE!

EXTENDED DOS SUPPORT

@ (type "N" keyboard)
< (type "B" keyboard)
! (original keyboard)
> (for 'wedge' users)

These commands may be used interchangably, to perform the following dos support functions.

Command	Function
@	Display disk status / send command
@N	Format (header) a new diskette
@I	Force initialize diskette
@V	Validate diskette (collect)
@D	Duplicate diskette
@C	Copy or concatenate disk file(s)*
@R	Rename file
@S	Scratch file(s)*
@$	List directory*
@U:	Reset disk drive
@L	List disk file or BASIC program*

* Added/enhanced disk command.

EXTENDED EDITOR

Command	Function
/	Quick load from disk
↑	Quick load from disk with auto run
APPEND	Append from disk to end of current program
AUTO	Auto line number (allows header)
BLOAD	Load machine language (binary) file
BRUN	Load and execute machine language program
CHANGE	Change pattern to another pattern
CLOSE	Close one or all files
CMD	Set output to file (does not send "READY.")
DELETE	Delete a range of lines from program
DUMP	Dump all scalar variables to screen or file
EXEC	Execute a file as keyboard commands
FIND	Find occurances of a pattern
GET	Read a sequential file into editor
KEY	Define a key as a special function
KEYS	Turn key functions on
KILL	Disable SYSRES™
KILL*	Disable SYSRES™ and unreserve memory
LIST	Improved BASIC LIST command
LOAD	Defaults to disk drive
MERGE	Merge from disk into current program
MON	Break to current machine language monitor
OLD	Restore program after "NEW"
PUT	Send program to disk as text file
RENUMBER	Renumber all or part of program
RUN	Run current program, ignores screen garbage
SAVE	Defaults to disk drive, allows replace
SETD	Set disk device #, allows multiple drives
SETP	Set printer channel, format mode, paging
TRACE	Select 1 of 3 trace/step modes and speed
VERIFY	Compare current program against disk/tape
WHY	Print position of last error
WHY?	List line of break or error
*	Send output to printer
#	Display current version of SYSRES™

COMPARE SPECIFICATIONS!

	SYSRES™	POWER™
Number of ADDED commands	33	13
Number of IMPROVED BASIC commands	7	none
Number of DOS SUPPORT commands	11	none
Approximate added syntax options	1200	60
Instruction manual length	86 pages	75 pages
Instruction manual style	structured	conversational
Re-loactable?	yes	no
Use on more than one (any) PET/CBM™	yes	no
Upgradable	yes	no

COMPARE FEATURES!

	SYSRES™	POWER™
Automatic printer output?	yes	no
Selectable ASCII conversion?	yes	no
List programs without loading them?	yes	no
Formatted program listings?	yes	no
Dump SEQuential/RELative files?	yes	no
Edit data files?	yes	no
True program merge?	yes	no
Auto number with **AUTO TEXT**?	yes	no
Load machine language programs?	yes	no
Auto-execute machine language programs?	yes	no
Directory (menu) file commands?	yes	no

COMPARE "EQUIVALENT" FUNCTIONS!

Function: Change occurances of one pattern to another.

Feature	SYSRES™	POWER™
Command word	CHANGE	@
'Wild cards' in search string?	yes	yes
'Wild cards' in replace string?	yes	no
Selectable range?	yes	yes
Match in entire text?	yes	yes
Match in commands only?	yes	no
Match exact variable names?	yes	no

Function: Define special one-key functions.

Feature	SYSRES™	POWER™
Command word	KEY	REM"
Requires BASIC program changes?	no	yes
Destroys variables?	no	yes
Re-define any key?	yes	no
Maximum string length	255	73
Quotes and carriage-return allowed	yes	no
Re-define any token key?	yes	no
Retain user keys from program to program?	yes	no

JUST A FEW OF THE FEATURES OF SYSRES™

* Fast up/down scrolling which works!
* Advanced repeat-key routine!
* Re-define any or all keys as any keyword (full or short form) or as any string up to 255 charactors long!
* Auto line numbering which can feed a string of up to 127 charactors as well!
* Extended DOS support (requires DOS 2A or greater)!
* Never enter another file name! All file commands work from the directory!
* Supports multiple disk drives!
* List BASIC programs, sequential and relative files without loading them into memory!
* TRUE PROGRAM MERGE (overlay). Supports subroutine libraries!
* Load and run machine language programs with parameter passing!
* Supports multiple printers!
* Automatic printer output with paging plus formatted listings with full ASCII code conversion including cursor control and special charactors for non-CBM™ printers!
* Edit text files and assembler source code without leaving BASIC!
* Renumber part of a program or even change the order of lines!
* Over 700 FIND/CHANGE commands including variable names ("A$" will not match "BA$"), pattern matching with "wild-cards", and even commands to remove spaces and REM's!
* Three TRACE modes including trace variables!
* Does not affect BASIC program operation!
* One AUTO-BOOT DISKETTE works for ALL PET™ or CBM™ computers (BASIC 2.0 or greater with at least 16k of RAM.). SYSRES™ requires NO ROM SPACE or extra boards, so you can take it with you if you want to use another computer. It may be put above the screen if you have RAM there. It boots automatically without disturbing any program in RAM!
* If, **for any reason**, you are not satisfied with the SYSRES™ system, you may return it along with any back-up disks (within 30 days) for a **full refund.** Your disks will be erased and returned to you.
* Diskette and Extensive Manual – only $95 (Please specify disk drive model when ordering.)

CALL US FOR THE NAME OF YOUR NEAREST DEALER

We are pleased to announce the aquisition of the author, Don Lekei, and the rights to SYSRES™. Don is now hard at work producing versions of our **STOCKFILE™** series of integrated **INVENTORY CONTROL, POINT OF SALE, ORDER ENTRY,** and **BILL OF MATERIALS** packages for the PET/CBM™ computers. The best inventory control system for the APPLE][™ will soon be available for the CBM™. See your local dealer for details!

CANADA
Tel: (604) 984-0477
#6, 144 West 15th St.
North Vancouver, B.C.
Canada V7M 1R5

UNITED STATES
Tel: (206) 734-3744
#204, 4202 Guide Meridian
Bellingham, WA
U.S.A. 98226

SYSRES™ is a trademark of Solidus International Corp.
(POWER™ is a trademark of Professional Software Inc.)

SOLIDUS INTERNATIONAL CORPORATION

www.commodore.ca

The two programs offered here use only the three Atari console keys to input answers. These programs should be useful for children as well as for programmers interested in learning more ways to use their machines. Requires at least 16K of memory.

Adding By Counting: Atari And Pre-schoolers

Stephen Levy
Bowie, MD

Using computers to teach young children can be fun and challenging. The Atari's design makes it extremely easy for young children to use. The Atari offers numerous ways other than the keyboard for a child to input answers.

Program 1, called "Add," simply teaches a very young child to add by presenting a simple addition problem and an equivalent number of symbols for each number in the problem. By counting the symbols, the child can decide on the answer. The child then presses the SELECT key until his/her answer (with the appropriate number of symbols) appears on the screen. In addition, the word for each number in the problem appears.

```
  5 ***** FIVE
+ 2 ##  TWO
  7 ^^^^^^^
```

To find out if an answer is correct, the child holds down the OPTION key. Program 3, which must be added to Program 1, contains the subroutines for the computer's response for correct and incorrect answers, as well as some music and sound. If the child gives an incorrect response, he/she is given another try. If the answer is correct, the child hears a song and then is given the option of another problem.

"Match," Program 2 (which also must be merged with Program 3), is designed similarly to Add, except in this case the child must match the word for a number with the correct number.

The Merge Timesaver

Neither Program 1 (Add) nor Program 2 (Match) will RUN alone. Each needs to be merged with Program 3. This was done in order to avoid having to type program lines unnecessarily. Program 3 should be typed and LISTed to disk (LIST"D1:‹filename›") or to tape (LIST"C:") first. Then type Program 1 and ENTER Program 3 to merge Programs 1 and 3 (Disk: ENTER"D1:‹filename›"; Tape: ENTER"C:"). You can repeat this operation to use Program 2. This avoids your having to type Program 3 twice. Also, all three programs can be merged into one by adding a menu option for the user at line 35 and changing lines 40 and 50 to IF-THEN statements based on the selection from the menu:

```
35 POSITION 3,2: PRINT "Press |SELECT| for
   ADD": POSITION 3,5: PRINT "Press |OPTION|
   for MATCH"
40 IF PEEK(53279)=5 THEN GOTO 300
50 IF PEEK(53279)=3 THEN GOTO 1400
60 GOTO 40
```

These two programs (Add and Match) have been written so that they can be merged easily.

Line 30 contains a long string which is used to print the word for the number chosen at random. Lines 105 and 115 in Add and line 230 in Match are used to select the appropriate characters from the string NUMBER$(line 30); this is more easily studied in line 230.

Here is a summary of the major sections of the program.

100-115	print the proper number of symbols and the word for the numbers.
120-130	select numbers for problems.
200-220	position answer and symbols on screen.
230	selects proper characters from NUMBER$.
240-260	select symbols to be used.
300	begins main program for Add.
318-360	print problem and go to subroutine to print symbols.
380-430	are routine to select answer.
450-460	check for correct answer.
520(5000-5520)	correct response.
600-695	incorrect response.
1400	begins program for Match (select a number).
1430-1445	print the word for the number.
1450	prints number.
1460-1520	select answer.
1610-1620	check for correct answer.

Program 1.

```
1 REM ADD
2 REM BY STEPHEN LEVY
3 REM BOWIE, MARYLAND
20 DIM CLEAR$(1),NUMBER$(51),C$(1),NU
   M$(6)
30 NUMBER$="ZERO ONE  TWO  THREEFOUR
   FIVE SIX  SEVENEIGHTNINE ":CLEAR$=
   CHR$(125):C$=CHR$(94)
40 GOTO 300
100 FOR AA=1 TO NUM1:POSITION AA+5,4:
    PRINT #6;C$:NEXT AA
105 POSITION AA+6,4:PRINT #6;NUMBER$(
    NUM1+1+(NUM1*4),NUM1+5+(NUM1*4)):
    RETURN
```

www.commodore.ca

The New Standard

The following is from a review by Analog 400/800 magazine comparing programs for personal finance for Atari* computers...

"The programs we will discuss are *Personal Financial Management System* from Atari, *A Financial Wizard* from Computari, and *Budgetmaster* from Sunrise Software.

All three programs begin with the basic premise of setting up a budget, helping you follow it, and giving you an idea of where you are spending your hard earned dollars.

***A Financial Wizard* from Computari is by far the best of these programs, and will be *the standard of comparison* for the others. There are 26 expense categories available that are easily adapted to your personal requirements; 21 are regular expense accounts, one is reserved for salary and four are usage categories for record keeping, such as gas and electric usage. You then input your budgeted amounts.**

The check entry mode is very simple to use. After asking you what month you are entering, the program prompts you to enter the check information including whether or not it is tax deductible. Come tax time, you will *really* appreciate this function and the Check Search mode which will search by Name, Category, Check # or tax deductible checks.

The way *A Financial Wizard* displays and handles your tabulations is excellent. You can chart your actual expenses vs. your budget by month, by category or Year to Date. Tabulations by month give you a list of all categories, how much you spent, how much you budgeted, the dollar amount plus or minus your budget and the percentage of your total income you are spending on each category. The tabulation by category also gives you actual expense vs. budget, the difference, and the average amount you are spending. Besides the charts, you can also look at your expenses vs. budget in bar graph form, again by month or by category. There it is in black & white (and blue and gold). The amount you budgeted vs. the amount you spent.

Everything about this program is excellent, but where it *really* outshines the rest is in the Check Reconciliation. In effect, it gives you your bank statement on the screen, a complete list by month of all your checks and deposits.

Graphics, while really not a factor in the quality of programs of this type, do make your budgeting chores a little more pleasant. Again *A Financial Wizard* comes out on top.

The version of *A Financial Wizard* that was reviewed is version 1.3. We have been told that a version 1.5 is coming out. This newest version will be enhanced in a few ways. There will be a check writer option. You enter your checks as if they have already been written, the program will perform all of the previously mentioned functions and if you have a printer, will print out your checks, you just sign and mail. Bank compatible checks will be available from Computari; ordering information will be in the package. There will also be an audit feature. User compatibility is excellent, and is set up with most of the instructions on the screen so you are not constantly referring to the instruction manual.

We strongly recommend this program."

A Financial Wizard 1.5 — **The Ultimate System**

- Budget–forecast 26 expense categories
- Check Entry–easy data entry–scan & correct 26 major & 36 sub-categories–information block
- Check Search–single or multiple parameters–(up to seven) to search entries
- Tabulations–detailed expense vs. budget by month, year-to-date, category
- Bar Graphs–screen displays in graph form expenses vs. budget–by month or category–printing with graphic capable printers
- Check Reconciliation–fast clearing of resident checks & deposits, complete summary report
- Checkwriter–print your custom checks
- Printouts–most popular printers • Audit Report
- Multi-Colored Graphics • Audio Enhancements
- 7 Utility Programs • User-Friendly Operation
- Easy To Use Instruction Manual
- Tinted Plastic Storage Case

The system is designed for Atari computers having a minimum of 24K and operating from a disk drive. The price is only $59.95 plus $3 for handling/postage.
If your dealer does not have **A Financial Wizard**...
Telephone orders are accepted on Mastercharge or Visa credit cards. Mail order must be accompanied by check or money-order or credit card #. Dealer Inquiries invited. (405)751-2783.

*trademark of Atari Inc. OK residents ad 4% Tax.

A Financial Wizard Exclusively thru ON LINE COMPUTER CENTERS of OKC™ 10944 North May Oklahoma City, OK 73120 (405) 751-2781

☐ CHECK ENCLOSED ☐ VISA ☐ MASTERCHARGE

NAME (Print) ____________________

Address ____________________

City ____________ State ________ Zip ________

Card # ____________ Exp. ________ Sig. ________

www.commodore.ca

```
110 FOR AA=1 TO NUM2:POSITION AA+5,6:
    PRINT #6;C$:NEXT AA
115 POSITION AA+6,6:PRINT #6;NUMBER$(
    NUM2+1+(NUM2*4),NUM2+5+(NUM2*4)):
    RETURN
120 NUM1=INT(RND(0)*10):RETURN
130 NUM2=INT(RND(0)*10):RETURN
140 FOR WAIT=1 TO 500:NEXT WAIT:RETURN
200 IF AA=19 THEN 315
205 IF AA<11 THEN POSITION 5+AA,8:PRI
    NT #6;C$:POSITION 3,8:PRINT #6;AA
206 SOUND 0,75,10,8
207 IF AA=10 THEN POSITION 2,8:PRINT
    #6;"10 "
210 IF AA>10 THEN POSITION 5+(AA-10),
    9:PRINT #6;C$:POSITION 2,8:PRINT
    #6;AA
215 SOUND 0,0,0,0
220 RETURN
240 CHAR=INT(RND(0)*8)+36:GOTO 260
250 CHAR=INT(RND(0)*5)+60
260 C$=CHR$(CHAR):RETURN
300 REM ADDING
310 GOSUB 120:GOSUB 130
315 GRAPHICS 18:SETCOLOR 4,14,12:SETC
    OLOR 0,8,18
318 POSITION 3,4:PRINT #6;NUM1
320 IF NUM1=0 THEN POSITION 5,4:PRINT
    #6;"zero":GOTO 340
330 GOSUB 240:GOSUB 100
340 POSITION 3,6:PRINT #6;NUM2
345 IF NUM2=0 THEN POSITION 5,6:PRINT
    #6;"zero":GOTO 360
350 GOSUB 250:GOSUB 110
360 POSITION 2,7:PRINT #6;"■■■":POSIT
    ION 1,5:PRINT #6;"+"
370 AA=0:POSITION 3,8:PRINT #6;"0"
375 GOSUB 240
380 POSITION 0,0:PRINT #6;"press sele
    ct to{12 SPACES}change answer":GOS
    UB 140
385 IF PEEK(53279)=5 THEN AA=AA+1:GOS
    UB 200
390 POSITION 0,0:PRINT #6;"press opti
    on when   you like your answer"
    :GOSUB 140
400 IF PEEK(53279)=5 THEN AA=AA+1:GOS
    UB 200
420 IF PEEK(53279)=3 THEN 450
430 GOTO 380
450 IF AA=NUM1+NUM2 THEN GOSUB 520
460 IF AA<>NUM1+NUM2 THEN GOSUB 600:G
    OTO 315
470 SETCOLOR 4,14,12:SETCOLOR 0,8,18
480 POSITION 0,0:PRINT #6;"press sele
    ct for{4 SPACES}another problem
    {8 SPACES}":GOSUB 140:GOSUB 140
485 IF PEEK(53279)=5 THEN 300
490 IF PEEK(53279)=3 THEN END
500 POSITION 0,0:PRINT #6;"press opti
    on to end{18 SPACES}":GOSUB 140
510 GOSUB 140:GOTO 480
```

Program 2.

```
1 REM MATCH
2 REM BY
3 REM STEPHEN LEVY
4 REM BOWIE, MARYLAND
20 DIM CLEAR$(1),NUMBER$(51),C$(1),NU
   M$(6)
30 NUMBER$="ZERO ONE  TWO  THREEFOUR
   FIVE SIX  SEVENEIGHTNINE ":CLEAR$=
   CHR$(125):C$=CHR$(94)
50 GOTO 1400
140 FOR WAIT=1 TO 500:NEXT WAIT:RETURN
230 NUM$=NUMBER$(COUNT+1+(COUNT*4),CO
    UNT+5+(COUNT*4)):RETURN
1400 REM SELECT A NUMBER
1403 COUNT=INT(RND(0)*9):GOSUB 230
1405 GRAPHICS 18:SETCOLOR 4,5,9:SETCO
     LOR 0,7,5
1410 POSITION 1,0:PRINT #6;"MATCH UP
     THE word":POSITION 2,1:PRINT #6;
     "WITH THE number"
1412 POSITION 0,8:PRINT #6;"PRESS sta
     rt TO BEGIN"
1415 AA=1
1416 GOSUB 140
1417 IF PEEK(53279)<>6 THEN 1417
1420 GRAPHICS 18:SETCOLOR 0,1,13:SETC
     OLOR 4,5,9
1430 POSITION 8,7:PRINT #6;NUM$
1440 POSITION 2,3:PRINT #6;"select to
      change":POSITION 1,4:PRINT #6;"
     option if you like"
1445 POSITION 4,5:PRINT #6;"YOUR ANSW
     ER"
1447 GOSUB 140
1450 POSITION 1,10:PRINT #6;"0 1 2 3
     4 5 6 7 8 9"
1460 IF PEEK(53279)=5 THEN AA=AA+2:SO
     UND 0,75,10,8:FOR W=1 TO 10:NEXT
      W:SOUND 0,0,0,0
1470 IF PEEK(53279)=3 THEN 1600
1480 IF AA>19 THEN AA=1:POSITION 19,9
     :PRINT #6;" "
1490 IF AA=1 THEN 1510
1500 POSITION AA-2,9:PRINT #6;" "
1510 POSITION AA,9:PRINT #6;C$
1515 GOSUB 140
1520 GOTO 1460
1600 ANS=((AA+1)/2)-1
1610 IF ANS=COUNT THEN GOSUB 520
1620 IF ANS<>COUNT THEN GOSUB 600:GOT
     O 1420
1630 GOSUB 140
1635 GRAPHICS 18:SETCOLOR 4,8,12:SETC
     OLOR 0,8,2
1637 POSITION 1,3:PRINT #6;"VERY GOOD
     ":POSITION 2,5:PRINT #6;NUM$;" I
     S ";COUNT
1639 GOSUB 140:GOSUB 140
1640 POSITION 2,5:PRINT #6;" OPTION T
     O END"
1645 POSITION 0,3:PRINT #6;"
     {13 SPACES}"
1650 POSITION 1,1:PRINT #6;"SELECT FO
     R ANOTHER{6 SPACES}PROBLEM
     {7 SPACES}"
1660 IF PEEK(53279)=3 THEN END
1670 IF PEEK(53279)=5 THEN 1403
1680 GOTO 1660
```

Program 3.

```
520 POSITION 2,11:PRINT #6;"correct":
    GOSUB 5000:RETURN
600 REM WRONG ANSWER
610 POSITION 2,11:PRINT #6;"sorry"
615 FOR S=1 TO 2
620 SOUND 0,120,2,8
625 GOSUB 695
635 SOUND 0,29,10,12
636 FOR WAIT=1 TO 40:NEXT WAIT
640 GOSUB 690:NEXT S
650 FOR S=1 TO 3
660 SOUND 0,180,2,8
670 GOSUB 695:GOSUB 690
672 NEXT S
673 FOR S1=1 TO 2
```

www.commodore.ca

Memory Expansion Boards for the ATARI* Computer

Completely compatible with ATARI hardware and software
No modifications necessary
Fully assembled and tested
User installable · simply plug it in
Gold connector tabs and sockets
One year warranty

16K Memory board — $ 49.95
32K Memory Board — $ 99.95
48K Memory Board — $199.95

Build Your Own Memory

16K board · no components · $12.50
32K board · no components · $30.00
48K board · no components · $50.00
Add $2 Shipping and handling per board
*ATARI is a trademark of Atari Inc.

Dealer Inquiries Welcome

Tiny Tek, Inc.
P.O. Box 820249 • Dallas, TX 75382-0249
214-373-8926

PRETZELLAND™ SOFTWARE PRESENTS:

ATARI®

AFFORDABLE ATARI(R) SOFTWARE

ANDROID ATTACK™ The nuclear reactor in our Top-Secret installation has malfunctioned and is in danger of melting down! You've been chosen to go in and manually release the cooling water. Now only one problem remains- there isn't time to deactivate the Security Androids guarding the unmanned installation, so you'll have to fight your way in, release the water, then escape before the entire underground laboratory fills with water! If you survive this mission, we've got another for you, only this time there are more levels to get through and the Androids are meaner!

AAAAAAGGG™ The fabric of space has been weakened by Atomic bomb testing! Strange little creatures are popping out of hyperspace all over Earth. It's up to you to catch as many of these little critters as you can, before they overrun the world! The creatures come in five types, each with its own point value. The higher its value, the quicker it pops back into hyperspace, so you've got to work fast. Every now and then, one little creature appears which is so charged with energy that it zips all around the screen briefly, then disappears. If you can catch this SUPER AAARRRGGG, there are bonus points and time for you, but don't get POISONED by little green men!

STARBASE ASSAULT™ A fleet of alien warships has surrounded your Starbase, firing deadly plasma torpedoes in an attempt to knock out its protective force field rings and destroy it! You've got to man the station's laser cannon, rotating and firing it rapidly, to blast the aliens and their torpedoes. As the battle continues, the aliens attack in ever greater numbers- up to 5 at a time and from 8 different directions! Every hit by an alien drains your force field. A high score bonus will partially replenish your energy supply, but when the last energy ring is gone, your Starbase is destroyed!

FOR 16K CASSETTE OR 24K DISK EACH...$18.95

ASK FOR PRETZELLAND SOFTWARE AT YOUR LOCAL DEALER, OR ORDER DIRECT. PLEASE ADD $2.00 SHIPPING FOR MAIL ORDERS. C.O.D. ORDERS ACCEPTED BY PHONE. CALL OR WRITE FOR OUR ILLUSTRATED LISTING OF AFFORDABLE GAMES FOR YOUR ATARI 400/800 (R).

PRETZELLAND SOFTWARE™
2005 A WHITTAKER RD. YPSILANTI, MI. 48197 (313) 483-7358


```
674 SOUND 0,29,10,11
676 FOR WAIT=1 TO 40:NEXT WAIT
677 GOSUB 690:NEXT S1
680 RETURN
690 SOUND 0,0,0,0:FOR WAIT=1 TO 40:NE
  XT WAIT:RETURN
695 FOR WAIT=1 TO 80:NEXT WAIT:RETURN
5000 REM INTRO MUSIC
5005 S3=2
5010 MUSIC=INT(RND(0)*2)+1
5020 RESTORE 5300+(MUSIC*100)
5030 READ S1,TIME
5040 IF S1=-1 THEN SETCOLOR 4,8,3:RETURN
5050 SOUND 0,S1+3,10,7:SOUND 1,S1,10,11
5055 SETCOLOR 4,S3,8
5060 FOR WAIT=1 TO TIME*20:NEXT WAIT
5070 SOUND 0,0,0,0:SOUND 1,0,0,0:FOR
  WAIT=1 TO 5:NEXT WAIT
5075 S3=S3+1:IF S3>15 THEN S3=1
5080 GOTO 5030
5400 DATA 122,2,122,2,82,2,82,2,73,2,
  73,2,82,4,92,2
5410 DATA 92,2,97,2,97,2,109,2,109,2,
  122,4
5420 DATA 82,2,82,2,92,2,92,2,97,2,97
  ,2,109,4
5430 DATA 82,2,82,2,92,2,92,2,97,2,97
  ,2,109,4
5440 DATA 122,2,122,2,82,2,82,2,73,2,
  73,2,82,4
5450 DATA 92,2,92,2,97,2,97,2,109,2,1
  09,2,122,4,-1,-1
5500 DATA 122,2,109,2,97,2,122,2,122,
  2,109,2,97,2,122,2,97,2,92,2,82,
  4,97,2,92,2,82,4
5510 DATA 82,1,73,1,82,1,92,1,97,2,12
  2,2,82,1,73,1,82,1,92,1
5520 DATA 97,2,122,2,122,2,82,2,122,4
  ,122,2,82,2,122,4,-1,-1
```

©

ATARI
A Warner Communications Company

800 (16K)	**$649.00**
400 16K	**279.00**
400 YOURS to 32K or 48K	**CALL**
410 RECORDER	**79.00**
810 DISK DRIVE	**449.00**
850 INTERFACE	**165.00**
830 MODEM	**149.00**
825 PRINTER	**575.00**
481 ENTERTAINER KIT	**79.00**
484 COMMUNICATOR KIT	**309.00**
PRINTERS — Atari, Epson, Smith Corona	**CALL**

Prices subject to change without notice.
Shipping extra. No tax out of state.
Ca. residents add appropriate taxes.

WE ARE AN AUTHORIZED ATARI SALES AND SERVICE CENTER

COMPUTERTIME, INC.

P.O. Box 216
Kentfield, CA 94914

CALL TOLL-FREE 800-227-2520
In California 800-772-4064
For product and price list: send $2.00 for shipping.

www.commodore.ca

For PET/CBM, Upgrade or 4.0 BASICs, with disk, this program will make the changes necessary so that any program will start itself RUNning. It need not be the first program on the disk – the program itself, when LOADed, will take control of the computer.

PET Self-starting Programs

Richard Mansfield
Senior Editor

There are some programs which are used so often that it is convenient to put them first on a disk if you use BASIC 4.0. Located on the disk as the first program, it will then automatically LOAD and RUN if you hit the RUN key. (Upgrade BASIC's RUN key tries to load from the tape drive.) "Bootfixer," however, will make *any* program self-starting.

If you have a disk of frequently used utilities, it might be worthwhile making every one of them self-starting. With 4.0 BASIC, you could have any one of them up and running with a simple dL"name. Also, people who have never used a computer would find this method of RUNning simpler to learn. All they would need to do is turn on the machine, insert a disk, and type in the name of the program they wanted: then the computer takes over. Built-in disaster prevention, such as a program with a disabled STOP key, should eliminate many of the start-up crashes experienced by novices.

Making The Mock Stack

Before Bootfixer can change another, target program, you must first prepare a special, slightly longer version of the target. It will include page one (memory from 256 to 511), which is the secret of automatic RUNs. It's easy. If your target program is called "HEXDUP," LOAD it normally and then type SYS 4 to get into the monitor. Type: .M 00C9 00C9 (RETURN) (this shows you the program's highest location in memory), and you will see something like:

```
00C9 C5 04 00 00 00 00 00 00
```

We only care about those first two hex numbers. To reSAVE the program with a different name (HEXDUP1), switch the two hex numbers and add one. In this case, HEXDUP ended in memory at 04C5, so we make it 04C6 during the monitor SAVE. Normally, a BASIC program starts at 0401 hex, but we are going to SAVE this special version from 0100, the bottom of the stack. So, type in the following fashion (substituting your program's new name and the correct end address plus one found at 00C9):

```
.S "HEXDUP1",08,0100,04C6
```

That's it. We now have a version of HEXDUP which contains a false page one, a mock stack, which will be loaded in whenever HEXDUP1 is loaded. The computer puts all of its machine language RTS addresses (the same as BASIC's RETURN) on the stack. Bootfixer will now do two things to HEXDUP1. It will replace part of the false stack (on HEXDUP1 while it sits on the disk) with 60 03 60 03, etc. This has the effect of sending control of the computer to address 0361 when HEXDUP1 is loaded into the machine. Second, a little machine language routine is inserted into HEXDUP1 at 0361 to make it start a BASIC RUN when control is sent via the false stack to 0361.

To transform HEXDUP1, just LOAD and RUN Bootfixer. It will ask you for the name of the program you want fixed and then move into the disk and make the necessary changes. If you accidentally give it the name of a program not yet prepared to be fixed, it will report that to you and close all files without doing any damage. Replace lines 480 and 490 with 481 and 491 if you use Upgrade BASIC.

Machine language programs can be made self-starting too. Find out the starting address of the machine language program, and replace the CHR$(96) in line 420 with the least significant byte *plus one* and the CHR$(3) in line 440 with the most significant byte. This will send control directly to the machine language program following a LOAD.

```
100 PRINT"{CLEAR} BOOTFIXER ":T=18:S=1:D$="0":
    OPEN15,8,15,"I"+D$
110 OPEN2,8,2,"#"+"0":REM     OPEN CHANNEL 2
120 REM **** LOCATE TARGET
130 INPUT"FILENAME";NA$:LN=LEN(NA$)
140 GOSUB210:GOSUB300
150 IFT=0THENPRINTNA$" NOT FOUND":GOTO540
160 GOTO140
170 GOTO540
180 REM ***POINT TO BYTE AND GET IT INTO X.
190 PRINT#15,"B-P:"2,L:GET#2,A$:IFA$=""THENA$=
    CHR$(0)
200 X=ASC(A$):RETURN
210 PRINT"TRACK"T" SECTOR"S:REM *** CHANGE TRA
    CK/SECTOR
220 PRINT#15,"U1:2,"D$;T;S:REM    PUT T/S INTO
     DISK BUFFER
230 L=0:GOSUB180:T=X:L=1:GOSUB180:S=X:RETURN
240 REM *** CHECK FOR FULL MATCH
250 FORJ=ITOI+LN:L=J:GOSUB180:IFX=0ORX=160 THE
    N270
260 X$=X$+CHR$(X):NEXTJ
270 IFX$<>NA$THENX$="":RETURN
280 L=I-2:GOSUB180:TT=X:L=I-1:GOSUB180:SS=X:PR
    INT
```

www.commodore.ca

CFi computer solutions
201 WEST 92 STREET • NEW YORK • NY 10025 • 212 • 724 • 1711

OFFICIALLY APPROVED BY COMMODORE

PROFESSIONAL TAX PREPARATION SYSTEM

WILL PROCESS THE FOLLOWING SCHEDULES AND FORMS:

1040
LETTER SCHEDULES A, B, C, D, E, F, G, SE
NUMBERED FORMS: 3903, 2106, 2441, 3468, 5695, 2210, and 1040-ES

STATE TAXES FOR: PA, CA, NY, NJ, MA, IA, and FL property.

Husband, Wife, and Joint forms prepared simultaneously.
Prints directly onto tax forms.

ASERT

Aid for Search and Retrieval of Text

To the usual Data Manager record format ASERT adds an area of FREETEXT, for all those "odd" bits of information that don't quite fit anywhere; and 180 SEARCHWORD fields, a unique system of coding with powerful search capabilities. Now used by Real Estate Agencies, Employment Agencies, Schools, Travel Services and Libraries across the U.S. and in Europe, ASERT makes the filing cabinet obsolete!

11,000 CHARACTER "VIRTUAL" RECORD LENGTH
INTERFACE WITH POPULAR WORD PROCESSORS
COMPILED FOR SPEED

OTHER CFI SOFTWARE
Personal Tax Calculator • Emergency Control Program • VIC Animation Tutorial

AVAILABLE FROM YOUR COMMODORE DEALER AND CFI.

www.commodore.ca

```
290 GOTO340
300 REM *** CHECK THROUGH ONE BLOCK FOR NAME M
    ATCH
310 FORI=5TO230STEP32
320 L=I:GOSUB180:IFCHR$(X)=LEFT$(NA$,1)THENGOS
    UB240
330 NEXTI:RETURN
340 REM *** ACCESS 1ST SECTOR OF TARGET PROGRAM
350 T=TT:S=SS:GOSUB210
360 L=2:GOSUB180:AL=X:L=3:GOSUB180:AH=X:SA=AL+
    AH*256
370 IFSA<>256THENPRINT:PRINTNA$" IS NOT PREPAR
    ED FOR BOOTFIX":GOTO540
380 REM *** ESTABLISH FALSE STACK
400 PRINT#15,"U1:2";DR;TT;SS:PRINT
410 FORPB=173TO254STEP2:PRINT#15,"B-P:2";PB
420 PRINT#2,CHR$(96);
430 PRINT#15,"B-P:2";PB+1
440 PRINT#2,CHR$(3);:PRINT"*";:NEXT:PRINT
450 PRINT#15,"U2:2";DR;TT;SS
460 GOSUB210:PRINT
470 REM ***PUT AUTOBOOT CODE ONTO PAGE THREE
480 DATA 165, 202, 133, 43, 165, 201, 133, 42,
     32, 233, 181, 32
481 REM FOR UPGRADE DATA 165, 202, 133, 43, 16
    5, 201, 133, 42, 32, 114, 197, 32
490 DATA 182, 180, 76, 74, 183
491 REM FOR UPGRADE DATA 66, 196, 76, 196, 198
500 PRINT#15,"U1:2";DR;T;S
510 FORPB=105TO121:READBY:PRINT#15,"B-P:2";PB
520 PRINT#2,CHR$(BY);:PRINT". ";:NEXT:PRINT:PR
    INTNA$" CAN NOW BOOT ITSELF"
530 PRINT#15,"U2:2";DR;T;S
540 CLOSE2:CLOSE15
```

©

TAYLORMADE SOFTWARE • TAYLORMADE SOFTWARE

for the COMMODORE VIC-20

- TOUCH TYPING TUTOR 2.0 (TTT-5K) $15.95
  Learn to type with all fingers by following the keyboard and finger placement pictured on the TV screen. 19 lessons fully described in manual. Using computer generated pseudo words learn your rate and errors.
- STAKEOUT (SO-JS) $14.95
  5K strategy game for 2 players using joystick. Object is to fence in the most territory. With 3K or more memory expansion you may play against VIC.
- VIC LEMONADE (VL-5K) $14.95
  Classic economics game for 2 players. Tunes and color graphics. Maximize your profits by choosing the best prices, advertising, & quantity to match randomly varying weather conditions. Make a million dollars starting with $2 if you can!
- MORSE CODE TRAINER (MC-5K) $19.95
  Learn Morse Code – speed 1-35 wpm. Character speed may be set faster than word speed. Random code groups or your own practice message.
- AEROBICS POINTS CALCULATOR (APC) $24.95
  Disk Version – (APC-D) $29.95
  Finds & keeps track of weekly aerobics points for running, walking, swimming, cycling for any time or distance. With 3K or more memory expansion 25 additional activities. Menu selected program plus 12 page manual.

Shipping/handling per order:
U.S. or Canada $1.50
Foreign U.S. $ + $3.00

TAYLORMADE SOFTWARE
P.O: Box 5574
Lincoln, NE 68505
(402) 464-9051

VIC is a trademark of Commodore Business Machines, Inc.

TAYLORMADE SOFTWARE • TAYLORMADE SOFTWARE

PROOFREADING SOFTWARE

NEW ... THE MOST ADVANCED SPELLING CHECKER SOFTWARE AVAILABLE FOR COMMODORE WORD PROCESSORS.

Now you can rapidly eliminate misspellings from your word processing text. Spellmaster can quickly pay for itself in reduced clerical time spent on correcting misspellings.

SPELLMASTER
by Dwight Huff & Joe Spatafora

Features include:

40,000 Word Capacity. Spellmaster (CBM 8050 version) is delivered with a 19,000 word dictionary, allowing the user to add up to 21,000 words.
Direct Screen Editing of Mistakes. Words "suspected" to be incorrect are displayed in "reverse video" on the screen; simply correct the mistakes and resave your corrected file ... ready to be printed.
Add Words with a Single Keystroke! Spellmaster makes it easy to Permanently Add any correctly spelled word in your text to your User Dictionary.
Menu-Driven and User Friendly. The average user can learn to operate Spellmaster in 30 minutes.
Complete Documentation is provided to guide you through Spellmaster on a "step by step" basis.

Machine Language Speed allows a large Wordpro4+ textfile to be Proofread in 2 minutes or less. Proofreading of linked files is easy and automatic.
Legal and Medical Dictionaries are available to add 4500 technical terms to the dictionary.
Compatible with the Commodore 8032, 4032, 8096, SuperPET, & the Commodore 64. Operates with the Commodore 2040, 4040 and 8050 Disk Drives. WordPro, Wordcraft, & Silicon Office versions available.

Spellmaster $199 Legal/Medical Modules $75

SPELLMASTER SYSTEMS SOFTWARE
6219 Thirteenth Avenue South
Gulfport, Florida 33707
(813) 347-6733
Dealer inquiries are invited.

WordPro is the registered trademark of Professional Software, Inc. and Pro MicroSoftware Ltd. Wordcraft is registered trademark of Dataview, Inc. Silicon Office is the registered trademark of Brystol Software Factory.

www.commodore.ca

For VICs without memory expansion, these gaming routines will help speed up BASIC considerably.

VIC Joystick And Keyboard Routine

Michael Kleinert
Nanuet, NY

In the Fall 1982 issue of *Home and Educational COMPUTING!* was an article by David Malmberg entitled "Using the VIC Joystick," which demonstrated a short BASIC routine for reading from the joystick. After adding that routine to one of my game programs, I discovered that BASIC can just be too slow for some games. My attempts to speed up that routine were unsuccessful, so I decided to write one in machine language for reading from the joystick. I designed the routine to be most suitable for game purposes, especially those in which you must guide an object around the screen by using the joystick.

Entering The Machine Coding

Type in the BASIC loader provided in Program 1. For those who may not have a joystick or might like to use the keyboard, I have included an identical routine for the keyboard in Program 2.

Using The Routines

Both routines are very similar. Each checks for up, down, left, and right. Accounting for diagonal directions would require longer and more complex programming. The keyboard version will look for the depressing of four keys, which I have defined as I (up), M (down), J (left), and K (right).

I designed the routines for controlling the movement of an object on the screen, and I suggest the following format:

```
10 POKE A,B: SYS 7168: POKE A,32: A=A+PEEK(1)
-PEEK(2): GOTO 10
```

In the above line, A is the memory location of a character's position on the screen, and B is the character code of the desired character. First the character is POKEd onto the screen, and then the subroutine is called with SYS 7168. The subroutine checks for any movement of the joystick (or for keys being pressed). If it detects the joystick being pushed in any direction, it places an appropriate numerical value into location 1 or 2. These values will be used to update the position of the character being moved. First, the old character must be erased. This is accomplished by the command POKE A,32. The character is erased by POKEing a space onto the same screen position (A). After it has been erased, its position can be updated by adding the contents of memory location 1 and subtracting the contents of memory location 2. Do this as shown above, with the command A = A + PEEK(1)-PEEK(2).

If the routine does not detect the joystick or keyboard being depressed, the values in these two memory locations will be set to zero, and the variable A (character's position) will remain the same.

Avoiding Leaving The Screen

If the joystick is pushed up (or the "I" key is pressed on the keyboard), the routine will place a value of 22 into memory location 2. This causes the number 22 to be subtracted from the current screen address contained in variable A, and is the basis for accomplishing upward movement of a character on the screen. Similarly, a character is moved right, left, and down in this fashion.

In order to keep the character from going off the top or the bottom of the screen, more complex programming is required. An appropriate method is illustrated in Program 3. The program is not a game, but simply a demonstration for the use of the routines. It will scatter several boxes, as obstacles, on the screen and will enable you only to move your "player" around the screen with the joystick or keyboard. It is the basic structure for a game.

If you are going to use the joystick, enter in lines 10 to 40 from Program 1. If you are using the keyboard, copy the lines from Program 2.

When you are ready to use one of the routines in your own BASIC program, do the following. Place lines 10 to 40 from Program 1 or lines 10 to 30 from Program 2 at the beginning of your program. Then, wherever you wish to utilize the routine in your program, give the command SYS 7168. To update the character's position, use the method which I described above.

Other Applications

There are many other uses for these routines. You may use them in simple delay loops to temporarily stop the program and wait until something is pressed.

To check for a desired direction on the joystick or a key on the keyboard, use the values from Figures 1 and 2. For example, if you are using the

www.commodore.ca

keyboard subroutine and want the program to wait until the letter "I" is pressed on the keyboard, you PEEK location 2 as follows:

100 SYS 7168: IF PEEK (2)<>22 THEN 100

This will call the subroutine, and the program will not proceed until the value in location 2 is equal to 22.

If you are using the joystick and want to wait until it is pushed to the right, you follow the same basic format: PEEK memory location 1 for a value of one. For example:

100 SYS 7168: IF PEEK(1)<>1 THEN 100

The Firing Button

A "firing" button is not accounted for in either of the two routines, since it would require a line of BASIC. If you would like to check for the firing button, you would place the following step into your program:

200 IF PEEK(37137)>69 THEN GOSUB (Line number)

After the GOSUB, you would place the line number to which you wish to send the program if it finds the firing button depressed.

If you wish to check for a "firing" button on the keyboard, you may use the following line, which checks for any depressing of the SPACE BAR (the one I usually use).

200 IF PEEK(197)=32 THEN GOSUB (LINE #)

The Demo Program

Briefly, here's a description of the function of each line in the demonstration, Program 3.

5 Limits the end of BASIC to protect the machine language routine, clears variables, and sets "A" equal to 7800 (the character's memory location on the screen).

10 READs the machine code from the DATA statements and POKEs the values into memory, starting at 7168.

20-40 Contain the machine code for the routine in DATA statements.

50 Clears the screen and then POKEs the color red onto each screen location.

60 Puts obstacles on the screen in 30 random screen locations.

70 POKEs the character onto the screen, calls the subroutine, and then sets "B" equal to the updated address.

80 If the new address is found to be off the screen, or if it is occupied by a box, the character remains stationary and the program goes back to line 70.

90 The new screen position has been accepted, so the old character is erased. The program goes back to line 70 to go through the same process.

Both routines can be used on a VIC with any amount of memory and can be placed anywhere in the user's RAM. In order to keep things relatively simple, I wrote the demonstration program for a 3.5K VIC; it will not work on a VIC with any memory expansion. These routines help speed up programs a great deal.

Program 1: Joystick Reader

```
10 FORM=ØTO65:READN:POKE7168+M,N:NEXT
20 DATA169,128,141,19,145,169,Ø,133,1,133,2,1
   69,127,141,34,145,162,119,236,32,145
30 DATA2Ø8,4,169,1,133,1,169,255,141,34,145,1
   62,118,236,17,145,2Ø8,4,169,22,133,1
40 DATA162,11Ø,236,17,145,2Ø8,4,169,1,133,2,1
   62,122,236,17,145,2Ø8,4,169,22,133,2,96
```

Program 2: Keyboard Reader

```
1Ø FORA=ØTO4Ø:READB:POKE7168+A,B:NEXT
2Ø DATA169,Ø,133,1,133,2,165,197,2Ø1,12,2Ø8,4
   ,162,22,134,2,2Ø1,36,2Ø8,4,162,22,134
   ,1
3Ø DATA2Ø1,44,2Ø8,4,162,1,134,1,2Ø1,2Ø,2Ø8,4,
   162,1,134,2,96
```

Program 3: Demonstration

```
5 POKE56,28:POKE52,28:CLR:A=78ØØ
1Ø FORM=ØTO65:READN:POKE7168+M,N:NEXT
2Ø DATA169,128,141,19,145,169,Ø,133,1,133,2,1
   69,127,141,34,145,162,119,236,32,145
3Ø DATA2Ø8,4,169,1,133,1,169,255,141,34,145,1
   62,118,236,17,145,2Ø8,4,169,22,133,1
4Ø DATA162,11Ø,236,17,145,2Ø8,4,169,1,133,2,1
   62,122,236,17,145,2Ø8,4,169,22,133,2,
   96
5Ø PRINT"{CLEAR}":FORX=384ØØTO389Ø5:POKEX,2:N
   EXT
6Ø FORX=1TO25:Y=INT(RND(1)*5ØØ)+1:POKEY+768Ø,
   16Ø:NEXT
7Ø POKEA,42:SYS7168:B=A+PEEK(1)-PEEK(2)
8Ø IFB>8185ORB<768ØORPEEK(B)=16ØTHEN7Ø
9Ø POKEA,32:A=B:GOTO7Ø
```

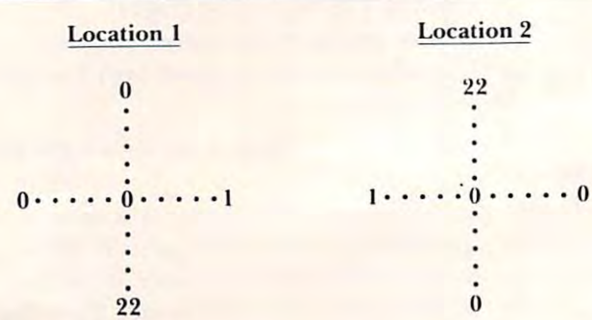
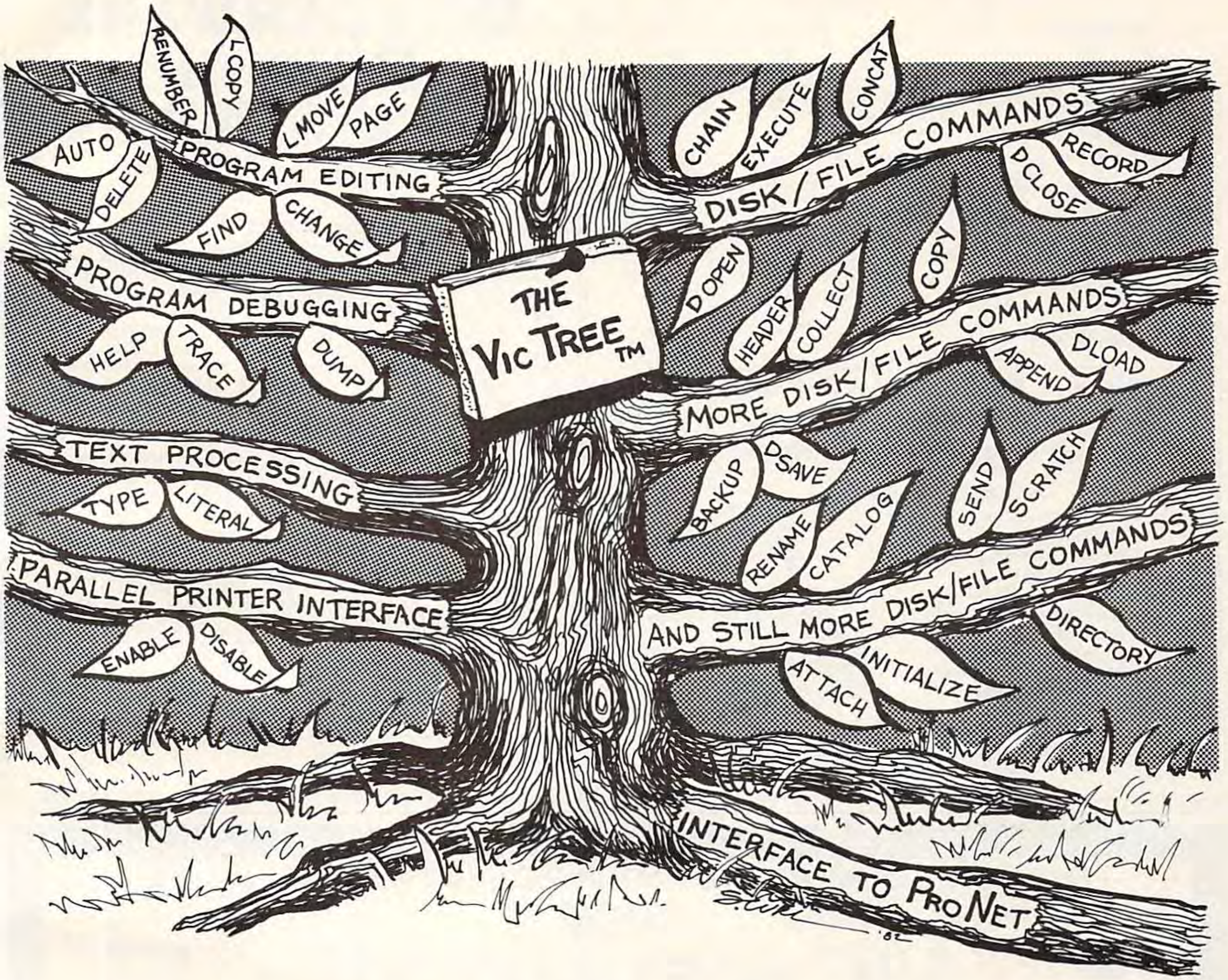


Figure 1: Joystick

www.commodore.ca

Skyles Electric Works Presents



The VicTree™

...Leaves your new Vic (or CBM 64) with 35 additional commands.
...Branches out to most BASIC 4.0 programs.
...Roots into most printers.

New from Skyles: the VicTree, a coordinated hardware and software package that allows your Vic to branch out in unbelievable directions and makes it easier than ever to do BASIC programming, debugging and to access your disk. And the new VicTree provides routines to interface the Vic to the powerful ProNet local network. 8kb of ROM — 4kb for the BASIC commands, 4kb for disk commands and interfacing to ProNet — plus 4kb of RAM for miscellaneous storage. Perfect not only for the new Vic but also for the Commodore 64. Unbelievably simple to use and to install, the VicTree gives you all the additional BASIC 4.0 commands to allow most BASIC 4.0 programs to work on your new Vic or CBM 64.

Now only $89.95...or **$99.95** complete with Centronics standard printer cable. (Cable alone **$19.95.**) Available now from your local dealer or order through your Visa or MasterCard toll free: (800) 227-9998 (California, Canada, Alaska, Hawaii: (415) 965-1735) or send check or money order directly to:

Skyles Electric Works

231E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

www.commodore.ca

```
     Location 1                     Location 2
         0                              22
         .                              .
         I                              I
         .                              .
0 . . J . . . . 0 . . . . K . . 1   1 . . J . . . . 0 . . . . K . . 0
         .                              .
         M                              M
         .                              .
        22                              0
```

Figure 2: Keyboard

©

INTRODUCING THE

VIC-20≡MX-80

CONNECTION

- HARDWARE NEEDS
- HOOK UP DIRECTIONS
- HI-RES SCREEN PRINTS
- TABULATION
- ALL EXPLAINED SO YOU CAN CREATE YOUR OWN GRAPHICS
- 2 AND 3-D GRAPHICS! (THIS AD IS A SAMPLE OF VIC-EPSON OUTPUT)

LABEL GRAPHS WITH SIDEWAYS PRINTING

SEND $15.00 CHECK OR MONEY ORDER
R.E.HUFFMAN, 9607 DOGWOOD
MUNSTER, INDIANA 46321

WUNDERWARE PROUDLY PRESENTS

State of the Art
Games for your VIC 20

Send for game catalog. Including these and more.

The Mad Painter—*This game is a little unique and a lot of fun. You control a paint brush, moving it around a colorful maze. Your job is to paint the entire maze. This is not as easy as it sounds, because in the maze with you are two voracious Bristle Biters (they love paint brushes). Occasionally you will receive a visit from an Invisible Stomper who leaves footprints in your fresh paint. Requires joystick. $9.95 cassette, $12.95 disk*

Bank Robbers!—*This is a lot of fun and slightly different. Your two partners in crime have made it to the roof of the VIC NATIONAL BANK and they're throwing the money down to you. Catch as much of it as you can but it's not easy, the way it drifts around. Stash it in the trunk of your getaway car and go back for more. In the end, the police will arrive and you will discover that crime, indeed, does not pay—but it's kind of fun! Specify keyboard or joystick. $9.95 cassette, $12.95 disk.*

Munchmaid—*Dots...monsters...a maze...and munchmaid! Guide munchmaid around the maze eating dots and avoiding monsters. If she eats a special dot, the tables turn, and she can chase the monsters! Munchmaid keeps top score and initials on the screen. Requires joystick and 3k memory expansion cartridge. $12.95 cassette, $15.95 disk.*

•Price includes Postage & Handling. •Catalog is included with order. •Foreign orders & COD's; Please add $3.00. •Prices are subject to change without notice. •At your dealer or send check or money order to:

wunderware®, P.O. Box 1257,
Jacksonville, OR 97530
☎503-899-7549

"Coming Attraction"
Barbarian Quest, 16K, graphic, adventure game.

VIC-20 is a registered trademark of Commodore Business Machines.

LET COMPUTERMAT TURN YOUR VIC INTO A HOME ARCADE

NEW
VIC SOFTWARE VIC

VISA · MasterCard

ALIEN INVASION — Arcade style excitement for your VIC. Look out here they come. Aliens are descending from the sky. Move your laser into position and defend the earth. The attacks are unending — can you survive or will Vader rule the galaxy. Many extras on this one. 20 levels of play. $12.95

CATTLE-ROUNDUP — The cows are loose in the maze. You have 2 minutes to get each cow back into the corral. You can push, coax and call the cows. Some cows are not very smart and some are very stubborn. You will have to help them. Be careful that you don't leave the corral gate open. Color graphics and sound. Eight levels of play and a time limit. $12.95

HEAD ON — Your car moves forward around the race track. You can move up, down, right and left. Try to score points by running over the dots on the track. Watch out for the crusher — if you crash you lose a car. Four cars and bonus levels. Full color graphics and sound. Fast action and very addicting. 9 levels of play. $12.95

SNAKEOUT — Blocks appear on the screen at random. You move up, down, right and left and try to move your snake over the blocks. Each block that you get raises your score. Keep building your score but watch out because the escape routes keep getting smaller. Time limit, color graphics and sound. 3 games on this cassette. Snakeout — 2 player Snakeout and Trapper. 9 Levels of Play. $12.95

BOMB'S AWAY — Can you stop him? The crazy bomber drops the bombs from the top of the screen. You get 3 buckets to catch them. Before you know it bombs are falling so fast you wonder when he will stop. Just when you think you have him under control your bucket gets smaller. Is your hand quicker than your eye? $12.95

TARGET COMMAND — Move your laser into position and get ready for some quick action. Different types of missiles are dropping. How many can you shoot down. They all travel at different speeds and different levels. You must be fast on the trigger to get them all. Time limit, bonus points and very addicting. Color graphics and sound. Arcade style fun. 10 levels. $12.95

BUG BLAST — Now its your turn to get even. Bug blast makes it easy. Blast thru the cactus and wipe them out. But watch out for the higher levels. Just when you think everything is under control the attackers seem to be everywhere. Will they ever stop. Now you can get even — if you dare. If you like Centipede you are gonna love BUG BLAST. $12.95

CRICKET — Can you help Cherp get across the roadway and river. He has to dodge trucks and cars to get to the center. Then help him get across the river. He can jump floating logs and turtles to get across. Watch out for gators — they love crickets for lunch. How many times can you get across? $12.95

VIC AND PET ARE TRADEMARKS OF CBM
ALL PROGRAMS ON CASSETTE

ALL VIC SOFTWARE RUNS IN STANDARD VIC.

WRITE FOR FREE CATALOG OF VIC SOFTWARE
PLEASE ADD $1.00 PER ORDER FOR SHIPPING

COMPUTERMAT • BOX 1664, DEPT C
LAKE HAVASU CITY, ARIZONA 86403

PET OWNERS — Write for free catalog
Ask for catalog P

www.commodore.ca

ML Software

CASSETTES *for the* **VIC-20***

BOMBER You're in command of a plane bombing enemy terrain. Enemy guns try to shoot you down. Try to bomb the guns to rack up points and make yourself safer, but it's not easy! Complete with color, sound, and a chance to make a buried oil tank explode — if you can get at it. Specify keyboard or joystick **$11.95**

REBOUND A game of skill and strategy. Try to make the little ball (thingamajig, whatever) bounce off the walls and barriers so that it hits the jackpot and not you. The wall configuration changes, and sixty points get deducted from your score for every second you think, so this one is no pushover. Specify keyboard or joystick **$11.95**

SUPERTENNIS *Not* another computer ping-pong game. This one uses a joystick, not a paddle. Let's face it — Bjorn Borg and John McEnroe don't move in one dimension, do they? Your speed is not infinite and the ball bounces and goes out of bounds. Come to the net and hit forehands and backhands. A true simulation. Joystick required.......**$14.95**

All 3 programs on one cassette for only $23.95.
Graphic cover program free with any purchase.
No expansion or accessories required (except joystick for SUPERTENNIS).

Postage and handling included.
California residents add 6% sales tax.

M.L. SOFTWARE 3324 Kempton Ave.
Oakland, CA 94611

**Vic-20 is a trademark of Commodore Business Machines, Inc.*

$19.95

21101 S. Harvard Blvd.
Torrance, CA 90501
(213) 328-9422
Dealer Inquiries Invited

VICWARE

by MicroSpec Ltd.

Colorful, Exciting Software for the Commodore VIC Computer!

Utility applications make your VIC earn its living while the games allow you and your family to enjoy it!

Moon Lander	9.95	Home Finance	12.95
Reversal 1	9.95	*Reversal 2	13.95
Bricks	9.95	Wrap	9.95
Blackjack	9.95	Capture	9.95
Tennis	9.95	Target Moon	9.95
Slots	9.95	Math Drill	9.95
Bombs Away	9.95	Alien Raiders	9.95
*Biorhythm	9.95	*Home Inv.	16.95
*Data Manager	19.95	*Portfolio Mgr.	19.95
Starfighter	13.95	*Hangman	12.95
*Auto Expense	19.95	Spelling 2-6	9.95
Typing Drill	12.95	(Specify Grade)	

*Requires Memory Expansion

Write for FREE Catalog

2905 Ports O'Call Court
Plano, Texas 75075
(214) 867-1333

VISA and MASTERCARD Accepted

BEING LIFEGUARD AT GATOR BEACH TAKES MORE THAN JUST A GOOD TAN! SHARKS ARE SCARY, BUT THESE 'GATORS ARE SOMETHIN' ELSE!
GATOR BEACH--A CHALLENGING, HAIR-RAISING GAME WITH MACHINE LANGUAGE FOR LOTS OF FAST ACTION AND THRILLS!

VIC-20 WITH AT LEAST 3K EXPANSION, JOYSTICK.

ON SALE AT YOUR LOCAL COMPUTER STORE OR SEND CHECK FOR **$24.95** TO:

VOYAGER SOFTWARE . P.O. BOX 1126 . BURLINGAME, CA 94010

ALLOW 21 DAYS FOR DELIVERY. CALIFORNIA RESIDENTS ADD 6½% SALES TAX.
DEALER ENQUIRIES INVITED

www.commodore.ca

A BASIC game is translated into machine language. The comments in the program will teach you how to PLOT, DRAWTO, COLOR, etc., in your own machine language games.

Insight: Atari

Bill Wilkinson
Optimized Systems Software
Cupertino, CA

Last month marked the first anniversary of this column in **COMPUTE!**, and I didn't even notice it. Which tells you how busy I am. We, like almost everyone in the software industry, are beginning to realize that survival comes only to those who diversify. So we are busily introducing new products and concepts. We think the net effect is beneficial to everyone: for us it means a chance to grow and try new approaches; for the user it means newer and better products with a wider choice than ever.

Of course, with the wider choice comes the obvious problem: which one of several competing packages should the user buy? I think I am asked that question only slightly less often than its predecessor: which computer should I buy? I usually sidestep the issue by saying something like this: "Find a software package that seems to do exactly what you want it to do. Ask for references from satisfied customers. When you are convinced that the software will suit your needs, buy the computer that is needed to run the particular software."

The most common problem I see is people buying too little computer for the problem they want to tackle. And, while the problem is sometimes related to the speed of the chosen machine (let's face it, you shouldn't be doing realtime voiceprint analysis with an Atari), the more common problem is simply lack of memory – both kinds of memory, RAM and disk.

This month, I have several topics of interest to Atari aficionados. And, of course, the monster listing of the assembly language version of the "Boing" game (the BASIC version was published last month). Please – hear my disclaimer: I am not nor do I claim to be a game programmer. I am quite aware that Boing is not the epitome of the gamer's art. Rather, I am here attempting to show the fundamentals of writing graphics games in assembly language. So don't type this game in expecting a miracle program; use it for instructional purposes only. Add to it, experiment with it, and chalk it up to experience.

A Boo-Boo

Well, so far we've encountered only one substantial mistake in our book, *Inside Atari DOS* (published by **COMPUTE!**). The error occurs in the text on page 11 and in the diagram (Figure 2-3) on page 14. Both correctly indicate the contents of the last three bytes of a data sector (the "link" information), but both assign the wrong order to these bytes. The byte containing the "number of bytes used in sector" is the *last* byte of the sector (byte 127 in single density sectors), *not* byte 125 as shown. Then the bytes shown as 126 and 127 move up to become 125 and 126, respectively.

Our apologies for the misinformation; we hope it didn't affect too many of you adversely. I think the mistake came about because of the comment in the listing at line 4312 on page 87, where the file number and sector link bytes are called "bytes 126, 127." Well, they are, if you are numbering from 1 to 128. The tables, etc., in the book are all numbered from 0 to 127; but recall that sectors on the disk are numbered from 1 to 720 (instead of 0 to 719). I don't know why we humans have such a hard time counting from zero, but we do. And computers have a hard time counting from any other number. Oh well.

Incidentally, the only other error in the diagrams that I have found occurs on page 21, where the labels "SABUFH" and "SABUFL" at the heads of the two columns are reversed.

CP/M For Atari?

I often get asked whether OS/A + will run CP/M programs on the Atari (since externally OS/A + looks very, very similar to CP/M — not an accident). But, you simply can't run CP/M on a 6502 (the heart of any Atari or Commodore or Apple). So how do Apple II owners run CP/M? Simple. They plug a card into their machine that essentially disables the 6502 and runs a Z-80 CPU instead. Why not do the same with an Atari?

First, let me say that I don't think that, as a practical matter, it is possible to *replace* the 6502 in the Atari 400/800 with another CPU (e.g., a Z-80). The reasons are many, but the primary one is the fact that the Atari peripheral chips (particularly Antic) seem somewhat permanently married to the 6502. However, there is no real reason that one could not put a co-processor board in the third slot of an 800 (the co-processor would probably have to have its own memory, though, to avoid interfering with the Atari's DMA and interrupt processing). This is essentially how some manufacturers have added 8086 capability to Apple II's. But it is expensive, since we now must pay not only for a CPU but also for 65K bytes of RAM and some sort of I/O to talk to the "main" 6502 CPU.

www.commodore.ca

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 5406
Greensboro, NC 27403

or call the Toll Free number listed below.

Change Of Address. Please allow us 6-8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $20.00 (2 years, $36.00; 3 years, $54.00. For subscription rates outside the US, see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
800-334-0868
In NC 919-975-9809

VERVAN

10072 Balsa St., Cucamonga, Ca 91730

Utility Programs for the ATARI 400/800

CASDIS - machine language program that allows you to transfer most "boot" tapes and cassette data files to disk. No special programming knowledge required. Programs that normally read cassette files during execution may be up-loaded to disk and operate normally without software modifications. Only difference is the program is on disk and loads at disk transfer speeds **Diskette only $25**

CASDUP - machine language program that allows you to copy most "boot" tapes and cassette data files as easily as you normally copy a BASIC tape **Cassette only $20**

FULMAP - (avail. late '82) machine language utility package for developers of BASIC programs Features: variable cross reference generator lists all program variables alphabetically with line numbers which reference them; line number cross reference generator which tells you how and where all line numbers are used; address utility which lists all indirect address references and tells you where they are used. This program resides in high memory and accessed from BASIC by entering "MAP". All outputs can be dumped to a printer **Cassette or Diskette $40**

DISASM - machine language program that allows you to disassemble machine language programs. You can load a file from cassette or disk and the program will display all the file's addresses and their contents in hex, ASCII (if any), and 6502 op code mnemonics. All outputs can be dumped to a printer **Specify Cassette or Diskette $25**

DISDUP - machine language program for sector level copying of disk information. User may specify single sector, range of sectors or all sectors on a disk to be copied. Copies may be made with or without a read verify. Sectors which cannot be read from or written to are displayed on the screen and optionally to a printer **Diskette only $25**

Please add $2 shipping & handling per program. California residents add 6% sales tax

Back-up policy: Our disks are protected against casual copying, but we appreciate your potential need for a back-up copy. If you add $10 to your order for a program, we will send you two (2) copies of the program disk. Our normal replacement guarantee applies to both copies. This offer does not apply to dealer sales.

ATARI 400/800 is a trademark of Warner Communications, Inc. *Dealer Inquiries Welcome*

GETTING SERIOUS ABOUT PROGRAMMING YET?!?

ATARI®

UNLOCK YOUR ATARI

WITH THE WORLD'S MOST COMPLETE POCKET PROGRAMMING AID.

16 comprehensive pages

- error codes
- basic commands with abbreviations
- peek and poke locations
- internal codes
- machine language aides
- much, much more!!

Get Yours Now only $9.95

(dealer ad space available)

ORDER NOW-CASH, CHARGE OR C.O.D.

ADVANCED COMPUTING ENTERPRISES
5516 ROSEHILL C-1
SHAWNEE MSN., KS 66216
(913) 262-2875 • (913) 631-4180

ACE

ATARI IS A REGISTERED TRADEMARK OF ATARI INC.

FIRST BORN IN 1978!

the original & continuously updated

CCA Data Management System

Now Available For Atari Computers		$ 99.50
	For Apple Computers	150.00
	For CPM Based Computers	225.00

CCA Data Management System

Uses

- Business
  Accounts Receivable
  Accounts Payable
  Inventories
  Billing
  Lists and Rosters
- Home Phone Lists
  Budgets, Hobbies

Features And Capabilities

- Long record lengths
- Up to 24 fields per record
- **Not Copy Guarded**
- Alpha numeric items
- Numeric only items
- Add, update, scan, etc. files
- 10-Level sort ascending, descending, allows alphabetizing data file.
- Contact your local dealer for details or write us for our catalog

DIVISION OF CUSTOM ELECTRONICS, INC.
CE SOFTWARE

238 Exchange St., Chicopee, Massachusetts 01013
(413) 592-4761
Mastercard & VISA Accepted
• Dealer And Distributor Inquiries Invited
• Closed Mondays — Open Daily 'Til 5:30 — Fridays 'Til 8

www.commodore.ca

But doing this leaves you stuck with using the Atari serial bus to get data on and off a disk. And, aside from the slow speed, in my opinion an Atari 810 is really too small for practical CP/M work. So, what's the solution, if any? Actually, I've heard of a couple and know of one that is now working.

The first CP/M solution is to simply treat the Atari as an intelligent terminal and hook it up to a CP/M system. While this sounds like overkill, remember that most CP/M systems do not come with a terminal (screen and keyboard), and none can offer the color graphics capabilities of the Atari. But Vincent Cate (alias USS Enterprises) of San Jose, California, has come out with a hardware/software package that does more than make an Atari into an intelligent terminal. His package also allows most CP/M based computers with a 19,200 baud serial port to effectively replace the disk(s) and printer of an Atari computer.

The CP/M system is turned on and started up first, and it fools the Atari into believing that it is an 810 disk drive (just as does the 850 Interface Module in diskless systems). It thus boots a mini-pseudo-DOS into the Atari which simply passes file requests over the serial bus to the CP/M system. A great idea for someone who has a CP/M system and wants either to get a graphics terminal or to justify buying a game machine.

The primary limitation of this system is simply that you won't be able to read or write Atari-formatted diskettes, though it may be possible to CLOAD from an Atari cassette and then SAVE to the CP/M disk. You won't be compatible with the rest of the Atari world, but for games you probably don't care. At $150, this is the cheapest CP/M to Atari connection, but it does presume the prior purchase of a CP/M-based system.

L. E. Systems (alias David and Sandy Small, et al.) has another method of doing co-processing: remove the cover of your 800 and replace it *and* the OS ROM board with an extension of the Atari's internal computer bus. On this bus one can stick more memory cards, disk controllers, and (of course) a Z80 card with its own 65K of memory. If your goal is to build a super powerful graphics machine, with access to the vast CP/M library, this is a workable approach (about $1900 with two disk drives, *plus* the cost of the Atari 800).

However, for about the same money, you could buy a *real* CP/M machine (such as the Cromemco C-10) with 80-column screen, full function keyboard, built-in printer interface, bigger disks, etc. And then, if you wished, you could hook up your Atari via Vincent Cate's interface. The L. E. Systems' approach, though, assures lightning fast data and control flow between the Z80 and the 6502. More importantly, it allows you to continue to buy and use Atari-compatible disk-based software.

Finally, my rumor mill says that by the time you read this there will be a product available which will function as a more or less conventional Atari-compatible disk controller (à la Percom). But, at the flip of a switch, it will instead boot up and run CP/M (internal to the controller box), treating the Atari as an intelligent terminal, much as Vincent Cate's system does with more conventional CP/M computers.

Do I have any recommendations? Not really. Personally, I like my 128K Byte Cromemco (with 10 Megabyte hard disk and dual 1 Megabyte floppies) for serious software development. But when I think about it, I realize that the thing that makes this system so nice is *not* the CP/M compatibility (I almost never use CP/M, preferring to stick with Cromemco's Cromix). Rather, it is simply nice to have all that disk space available on command. So why get CP/M? Because you want to get into exotic compiler languages or because you need some very sophisticated business packages. Fine. But for games? Home finances? Learning how to program in BASIC? Graphics? I suggest you avoid CP/M.

Going With Boing

At last, we have here the complete listing of Boing as written in assembly language. As much as practicable, I have done a direct one-for-one translation from BASIC to machine code, without taking advantage of most of the foibles of the machine. Perhaps the only major change I have introduced is also the most unnoticeable from a casual reading of the source: I have made all the variables (which are six-byte floating point numbers in BASIC) into single bytes. This is *not* always possible. Sometimes, when writing in assembler, one needs numbers greater than 255; then one "simply" uses two-byte integers (or three or four-byte integers, or floating point even).

Except that, on a 6502, that "simply" isn't so simple. There are no 16-bit (or larger) instructions on a 6502, and one must simulate them using series of eight-bit loads, adds, stores, etc. For example, if this program were using Mode 8 graphics, where the horizontal position can vary from 0 to 319 (thus requiring a two-byte number to hold it), all of the code involving the "X..." variables would be larger and more complex. Lesson to be learned: use byte-size numbers whenever possible on a 6502.

Anyway, with regard to the listing of Boing, please note that I didn't leave enough space between my BASIC line numbers to allow my assembly language to share the numbering scheme. So I have put the BASIC lines into the listing in a way that makes them stand out for ease of reading.

www.commodore.ca

FIRST and FINEST

In Systems Software for Atari and Apple

MAC/65

First we delivered Atari's Assembler/Editor (the cartridge).

Then we produced our enhanced "EASMD."

Now OSS is introducing the **finest** integrated assembly language development system yet!

In addition to being ideal for writing small, "quick and dirty" subroutines and programs, **MAC/65** shows its full power and speed when used with even the most complex of large assembly language source files.

Naturally, **MAC/65** is upward compatible with both EASMD and the Atari cartridge. And, of course, the object code output is also compatible with OS/A+, Atari DOS, and/or Apple DOS, as appropriate.

MAC/65 .... **$80.00***

OS/A+

Optimized Systems Software — the group that produced **both** the **first** Apple DOS **and** the **first** Atari DOS — now brings you OS/A+, which combines the **finest** features of these and other successful personal computer operating systems.

OS/A+ is the **first** and **finest** operating system available for **both** Apple II and Atari computers and features a keyboard-driven, easy-to-use command processor. In addition to several simple resident commands, OS/A+ allows logical and readable requests for even the most sophisticated utility commands. In fact, the user can even add system commands as desired.

But the real power and flexibility of OS/A+ is its ability to easily interface to devices and **disk drives** of virtually any kind and size. File compatibility (with Apple DOS or Atari DOS, as appropriate), device independence, batch processing, easy of use — OS/A+ truly brings the **finest** in operating systems to **your** computer.

AND NOW OS/A+ (for standard Atari or Apple drives) is included as a part of **every** standard OSS language package. Versions of OS/A+ for some higher capacity drives available at extra cost.

Unless otherwise noted, all OSS products require 48K and at least one disk drive. We recommend 64K for the Apple version of OS/A+.

SpeedRead+

The **first** and still **finest** speed reading tutor designed for **you** to use on **your** computer is available **only** from OSS.

SpeedRead+ uses time-proven techniques to train **you** to instantly recognize words and phrases, and yet it goes far beyond what mere mechanical devices are capable of.

SpeedRead+ exercises your peripheral vision, improves your eye movement and timing, and generally works **with you** at your pace... now and in the future.

NOTE: The Atari version of SpeedRead+ needs only 16K of RAM.

SpeedRead+ **$59.95**

tiny c

As a product of Tiny C Associates, **tiny-c** was the **first** structured language interpreter for microcomputers. Now OSS brings this innovative interpretive language to **your** home computer. While not having the speed and power a true **C** compiler, **tiny-c** is an excellent choice for the programming student who is ready to begin learning the valuable techniques of structured languages.

tiny-c provides an easy-to-use, easy-to-modify environment that encourages experimentation while promoting proper programming style. The **tiny-c** package includes not only a comprehensive and instructional user manual but also **complete source.**

tiny-c **$99.95***

C/65

NOW AVAILABLE!

The **first** native mode **C** compiler ever produced for Atari **and** Apple computers.

C/65 supports a very usable subset of the extremely powerful and popular **C** language. Just as **C** is used by the most sophisticated programmers from the professional and academic communities, so shall **C/65** prove to be a powerful and much-needed tool for 6502 software developers.

C/65 supports integer and character types (and arrays), pointers, fully recursive functions, and much more.

NOTE: C/65 requires MAC/65 or an equivalent assembler. Two disk drives recommended but not required.

C/65 .... **$80.00***

BASIC A+

"From the authors of Atari BASIC..."

It's a fact! OSS gave you that **first** and **most popular** language for Atari Home Computers. But why be content with the **first** when you can have the **finest**?

BASIC A+ is the **only** logical upgrade available to the Atari BASIC programmer. While retaining **all** the features which make Atari BASIC so easy to use, we've **also** given BASIC A+ features that place it at the forefront of modern interpretive languages. BASIC A+ will let **you** explore the worlds of structured programming, superior input/output, helpful programming aids, and even a very comprehensive PRINT USING command. **And**, exclusively for the Atari computer, an almost unbelievable array of PLAYER/MISSILE GRAPHICS commands and functions.

BASIC A+ **$80.00***

***REMEMBER:** Standard OS/A+ is **included** at no extra charge with BASIC A+, MAC/65, C/65, and tiny-c.

ATARI, APPLE II, and TINY C are trademarks of Atari, Inc., Apple Computer, Inc., and Tiny C Associates, respectively. ***SpeedRead+***, MAC/65, C/65, BASIC A+ and OS/A+ are trademarks of Optimized Systems Software, Inc.

Optimized Systems Software, Inc., 10379 Lansdale Ave., Cupertino, CA 95014, (408) 446-3099

www.commodore.ca

Presuming that you have read my August and September columns, you will recognize the style and conversions that I have done. Statements such as PLOT, DRAWTO, COLOR, and others have been translated into JSRs to routines in my graphics package. (Note that the listing of the package has been omitted for space considerations. Simply include lines 9000 through 9999 of the listing in my August article.) I would, however, like to discuss a few points of interest.

Notice the coding of lines 2600 and 2700, where the BASIC program had used PTRIG(x)-PTRIG(x+1) to obtain a +1, 0, or -1 value from the joystick. But that requires turning the joystick 90 degrees from normal to play the game. As long as we are coding in assembly language, let's do it right!

What we have here, then, is essentially the code that BASIC A+ uses for its HSTICK(n) function. I think the code is easy to follow if you remember that the switches in the joystick force a zero bit in locations STICKn when they are pushed. By masking to only the bits we want, and by then inverting the bits, we are able to treat an "on" bit in a more or less normal fashion.

By the way, note that here, as elsewhere in the code, we are also using one-byte numbers to hold both positive and negative values. This works only so long as the absolute value of the signed numbers does not exceed 127, so be careful when using this technique.

Note the simulation of the array YP(n). First, look at how easy it is to handle array elements with constant subscripts, as in BASIC line 1010 (listing lines 1210 to 1230). Even variable subscripts aren't too hard when the array is byte sized and byte dimensioned. Look at BASIC line 4210 (listing lines 6030 and 6040). Admittedly, a true assembly language simulation of the BASIC line would probably go more like this:

```
LDX    HITP
LDA    SCORE,X
CLC
ADC    #1
LDX    HITP
STA    SCORE,X
          ;SCORE (HITP)=SCORE(HITP)+1
```

But why not be a *little* smart when making conversions? Besides, if we were writing in some higher level languages, we could have written "INCREMENT SCORE(HITP)".

Finally, the hardest part of this conversion needs some analysis. As we noted last month, in order to provide better movement and bounce characteristics for the ball, we allowed it to have movements (and positions!) of -1, -0.5, 0, +0.5, and +1. But now we're in assembly language using byte integers. How do we implement fractional movements? We can't really, so we must choose an equivalent scheme.

Notice the variables in the program called "Q.Yxxx". These variables all are used to hold values that represent *half* movements or positions. Example: if Q.YNEW contains 17, that means it is really representing position 8.5! Notice, then, that before plotting any point that is represented in this fashion, we must divide its value by 2 (by using a LSR instruction, c.f., listing lines 3820, 3930, etc.). Choosing this scheme has some interesting consequences: the last statement of BASIC line 3080 (listing lines 4500 through 4650) is, in some ways, the hardest part of this listing to understand, simply because of the implied "mixed-mode" arithmetic that is used. But it works!

Foibles Of The Assembler/Editor

Writing this article caused me to rediscover some of the foibles of the Atari Assembler/Editor cartridge (and EASMD, for that matter). For many of you, these quirks may seem normal, especially if you haven't used several different assemblers on various machines. But, to others, these eccentricities can be annoying or puzzling.

First, beware of the "*=" pseudo-operator. It is *not* an origin operator ("ORG" in many assemblers), even though it is used as such! Any label associated with this pseudo-op will take on the value of the instruction counter *before* the operator is executed. This is necessary since "*=" is *also* used to reserve storage ("DS" or "RMB" in some assemblers).

Examples:

```
LABEL1 *= *+5
        ; reserves five bytes of storage
        ; and assigns the label "LABEL1"
        ; to the five bytes
   *= $4000
        ; sets the instruction counter
        ; to 4000 hex
LABEL2 *= $5000
        ; assuming this line followed one
        ; above, assigns 4000 hex to
        ; "LABEL2" and sets instruction
        ; counter to 5000 hex!
```

Second, examine any references to location "CLOCK.LSB" in the Boing listing (e.g., line 5870). Notice that, even though CLOCK.LSB is in zero page, the assembler produced a three-byte instruction for all references to it. This is because the *definition* of CLOCK.LSB did not occur until *after* the first *reference* to it! Actually, the assembler/editor is being remarkably clever here. Remember that the cartridge is, like most assemblers, a two-pass program. It reads the source once to determine where things are and will be, and then it reads the

www.commodore.ca

COMPUTE!'s Second Book Of Atari

After only three years on the market, the Atari 400/800 microcomputers have become among the most popular personal computers ever made. So it was no surprise when *COMPUTE!'s First Book of Atari*, a collection of the best Atari articles published during 1980-81 in **COMPUTE!** Magazine, also became a "bestseller" with Atari enthusiasts. The first printing sold out in just a few months.

That's why we've followed up with *COMPUTE!'s Second Book of Atari*. Available immediately, the *Second Book of Atari* continues **COMPUTE!**'s tradition for personal computer users.

But the *Second Book of Atari* differs from the *First Book* in one important respect – all the articles are totally new and previously unpublished. The *Second Book of Atari* includes such interesting articles as "Page Flipping," "Fun With Scrolling," "Perfect Pitch," "Player-Missile Drawing Editor," and "TextPlot Makes a Game." Whole chapters are devoted to subjects such as "Advanced Graphics and Game Utilities," "Programming Techniques," and "Beyond BASIC." With 250 pages – more than 25 percent thicker than the *First Book* at the same price – the *Second Book of Atari* is crammed with information and ready-to-type program listings. And the book is spiral-bound to lie flat and is fully indexed for quick reference.

Best of all, *COMPUTE!'s Second Book of Atari*, like **COMPUTE!** Magazine itself, is written and edited to appeal to all computer enthusiasts – beginners and experts alike. Priced at only $12.95.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2.00 shipping and handling. Outside the U.S., add $4.00 for air mail, $2.00 for surface mail. All orders prepaid, U.S. funds only.

www.commodore.ca

source again to produce the listing and code. But, during the first pass through the source, it can't possibly know whether CLOCK.LSB is in zero page or not, so it chooses the safe route and assumes non-zero page. Then, lo and behold, it discovers that we really wanted the label to be in zero page. What to do?

If we now assign that label to zero page, the second pass of the assembler will produce only two bytes of code here, and all references to labels past that point will be off by one byte. We will have the infamous "phase error." So the assembler has a rule that states "once non-zero page, always non-zero page," and it continues to generate three-byte references. For a simple assembler like the Atari cartridge, this is a big step. It is still possible to produce phase errors with the cartridge, but it is more difficult than with many 6502 assemblers.

Third and last, there is a problem with the assembler/editor when it comes to multiple forward references. Consider the following code fragment:

AAA = BBB
BBB = CCC
CCC = 5

There is no way for a two-pass assembler to determine what the value of AAA is! On the first pass, it says "AAA is undefined, because BBB hasn't been defined yet." And then it thinks "BBB is undefined, similarly because of CCC." On the second pass, it *should* say "ERROR!!AAA is undefined, because BBB still hasn't been defined yet." But it can then produce "BBB is equal to 5 because that's what CCC is equal to."

Unfortunately, the assembler/editor doesn't keep a separate flag meaning "label as yet undefined." The "BBB = CCC" line is sufficient, from the assembler's viewpoint, to establish the existence of "BBB." So, on the second pass, it blindly puts the value of BBB (presumably zero) into AAA. Watch out for this trap! It has snared many a good programmer! I hope you realize that there would be no problems if you had coded that sequence in this order:

CCC = 5
BBB = CCC
AAA = BBB

That's it for this month. Next month we will investigate the many languages available to the Atari programmer. We will discuss and fix the major bug in Atari's 850 interface handler (the "Rn:" drivers). And maybe, just maybe, we will try to add cassette tape verification to BASIC.

```
0000        1000        .PAGE "     === GAME STARTUP ==="
            1010 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            1020 ;
            1030 ; This is the startup of BOING
            1040 ;
            1050 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            1060 ;
            1070 ; CAUTION:  set memory origin according to
            1080 ;           your system needs!
            1090 ;
            1100 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            1110 ;
0000        1120        *=   $6000
            1130 ;
            1140 BOING
            1150 ;

:BASIC: 1010 DIM YP(1),SCORE(1):SCORE(0)=0:SCORE(1)=0

6000 4C0760 1160        JMP  AROUND.DIM
6003 00     1170 YP     .BYTE 0,0      ; y-position
6004 00
6005 00     1180 SCORE  .BYTE 0,0      ; and score
6006 00
            1190 ;
            1200 AROUND.DIM
6007 A900   1210        LDA  #0
6009 8D0560 1220        STA  SCORE+0   ; SCORE(0)=0
600C 8D0660 1230        STA  SCORE+1   ; SCORE(1)=0
            1240 ;
            1250 ;

:BASIC: 1020 SINGLE=PEEK(53279)<>7)

600F AD1FD0 1260        LDA  53279     ; peek at console switches
6012 4907   1270        EOR  #$07      ; A=7? Then A=0.  A<>7? Then A<>0.
6014 8DE062 1280        STA  SINGLE    ; set up our flag
            1290 ;

:BASIC: 1100 LASTWIN=1:IF RND(0)>=0.5 THEN LASTWIN=-LASTWIN

6017 A001   1300        LDY  #1        ; use y as temp for lastwin
6019 AD0AD2 1310        LDA  RANDOM    ; get a random byte
601C 1002   1320        BPL  HALFCHANCE
601E 88     1330        DEY            ; 50-50 chance that we do this
601F 88     1340        DEY            ; ...makes Y = $FF, same as -1
            1350 HALFCHANCE
6020 8CE162 1360        STY  LASTWIN   ; store temp in final place
            1370 ;

:BASIC: 2000  REM prepare for a serve

            1380 LINE2000
            1390 ;

:BASIC: 2010 GR.3 : COLOR 2 : PLOT 0,0 : DRAWTO 39,0

6023 A903   1400        LDA  #3
6025 20F362 1410        JSR  GRAPHICS  ; GR.3
            1420 ;
6028 A902   1430        LDA  #2
602A 202063 1440        JSR  COLOR     ; COLOR 2
            1450 ;
602D A900   1460        LDA  #0
602F A8     1470        TAY
6030 AA     1480        TAX
6031 202B63 1490        JSR  PLOT      ; PLOT 0,0
            1500 ;
6034 A900   1510        LDA  #0
6036 A227   1520        LDX  #39
6038 A8     1530        TAY
6039 204463 1540        JSR  DRAWTO    ; DRAWTO 39,0
            1550 ;

:BASIC: 2020 PLOT 0,19 : DRAWTO 39,19

603C A900   1560        LDA  #0
603E AA     1570        TAX
603F A013   1580        LDY  #19
6041 202B63 1590        JSR  PLOT      ; PLOT 0,19
            1600 ;
6044 A900   1610        LDA  #0
6046 A227   1620        LDX  #39
6048 A013   1630        LDY  #19
604A 204463 1640        JSR  DRAWTO    ; DRAWTO 39,19
            1650 ;

:BASIC: 2030 .... NOTE: We don't print the scores in this version ....

            1660 ;
            1670 ;
```

www.commodore.ca

```
:BASIC: 2040 COLOR 3:PLOT 0,9:DRAWTO 0,11:PLOT 39,9:DRAWTO 39,11

604D A903          LDA  #3
604F 202063        JSR  COLOR      ; COLOR 3
              ;
6052 A900          LDA  #0
6054 AA            TAX
6055 A009          LDY  #9
6057 202B63        JSR  PLOT       ; PLOT 0,9
              ;
605A A900          LDA  #0
605C AA            TAX
605D A00B          LDY  #11
605F 204463        JSR  DRAWTO     ; DRAWTO 0,11
              ;
6062 A900          LDA  #0
6064 A227          LDX  #39
6066 A009          LDY  #9
6068 202B63        JSR  PLOT       ; PLOT 39,9
              ;
606B A900          LDA  #0
606D A227          LDX  #39
606F A00B          LDY  #11
6071 204463        JSR  DRAWTO     ; DRAWTO 39,11
              ;
              ;

:BASIC: 2050 IF SINGLE THEN COLOR 2:PLOT 39,0:DRAWTO 39,19

6074 ADE062        LDA  SINGLE
6077 F016          BEQ  NOTTHEN2050 ; not single player mode
              ;
6079 A902          LDA  #2
607B 202063        JSR  COLOR      ; COLOR 2
              ;
607E A227          LDX  #39
6080 A900          LDA  #0
6082 A8            TAY
6083 202B63        JSR  PLOT       ; PLOT 39,0
              ;
6086 A227          LDX  #39
6088 A013          LDY  #19
608A A900          LDA  #0
608C 204463        JSR  DRAWTO     ; DRAWTO 39,19
              ;
              NOTTHEN2050
              ;
              ;

:BASIC: 2060 YP(0)=10:YP(1)=10

608F A90A          LDA  #10
6091 8D0360        STA  YP         ; YP(0)=10
6094 8D0460        STA  YP+1       ; YP(1)=10
              ;

:BASIC: 2070 IF SINGLE THEN LASTWIN=1

6097 ADE062        LDA  SINGLE
609A F005          BEQ  LINE2100   ; NOT SINGLE
609C A901          LDA  #1
609E 8DE162        STA  LASTWIN    ; LASTWIN=1 BECUZ SINGLE<>0
              ;

:BASIC: 2100 REM SET UP BALL

              LINE2100
              ;
              ;

:BASIC: 2110 XMOVE=LASTWIN:YMOVE=INT(3*RND(0))-1:Y=INT(12*RND(0))+4

60A1 ADE162        LDA  LASTWIN
60A4 8DE362        STA  XMOVE      ; XMOVE=LASTWIN
              ;
60A7 A902          LDA  #2
60A9 206263        JSR  RND        ; get random number from 0 to 2
60AC 8DE662        STA  Q.YMOVE
60AF CEE662        DEC  Q.YMOVE    ; then do the '-1'
60B2 0EE662        ASL  Q.YMOVE    ; and convert to "half-moves"
              ;
60B5 A90B          LDA  #11
60B7 206263        JSR  RND        ; get random number from 0 to 11
60BA 18            CLC
60BB 6904          ADC  #4         ; '+4' as above
60BD 0A            ASL  A          ; double number of moves to get
                                     half-moves
60BE 8DE562        STA  Q.Y        ; Again, this is a 'half-position'
                                     variable
              ;
              ;

:BASIC: 2120 YNEW=Y : X=19-5*XMOVE:XNEW=X

60C1 ADE562        LDA  Q.Y
60C4 8DE762        STA  Q.YNEW     ; YNEW=Y
              ; Here, we take advantage of the fact that XMOVE
              ;    can only have values -1 or +1
60C7 A9FB          LDA  #0-5       ; assume XMOVE = +1
60C9 ACE362        LDY  XMOVE      ; does XMOVE = +1?
60CC 1002          BPL  XMOVEPLUS  ; yes
60CE A905          LDA  #5         ; no...so -5*-1 = +5
              XMOVEPLUS
60D0 18            CLC
60D1 6913          ADC  #19        ; 19-5 OR 19+5
60D3 8DE262        STA  X
              ;
60D6 ADE262        LDA  X          ; but you can see we don't really
                                     need this
60D9 8DE462        STA  XNEW       ; XNEW = X
              ;
              ;

:BASIC: 2500 REM MAIN PLAYING LOOP

              ;
              ;

:BASIC: 2600 V0=PTRIG(0)-PTRIG(1):IF NOT V0 THEN 2700

              ; note that what we really want is V0=+1 if
              ;    stick is pushed one way and V0=-1 if
              ;    stick is pushed the other.
              ;
              LINE2600
60DC AD7802        LDA  STICK0     ; OS shadow location
60DF 2903          AND  #3         ; look at just fwd and backwd
                                     switches
60E1 4903          EOR  #3         ; invert the sense
60E3 F006          BEQ  GOTV0      ; if zero, stick not pushed
60E5 C901          CMP  #1         ; FWD pushed?
60E7 F002          BEQ  GOTV0      ; good...what we wanted
60E9 A9FF          LDA  #0-1       ; must be pulled back
              GOTV0
60EB 8DEB62        STA  V0         ; ta-da
              ;
60EE ADEB62        LDA  V0         ; so is stick pushed?
60F1 F03E          BEQ  LINE2700   ; IF NOT V0 THEN 2700
              ;
              ;

:BASIC: 2610 VP0=YP(0)-V0:IF VP0<2 OR VP0>17 THEN 2700

60F3 AD0360        LDA  YP+0       ; YP(0)
60F6 38            SEC
60F7 EDEB62        SBC  V0
60FA 8DED62        STA  VP0        ; VP0=YP(0)-V0
              ;
60FD C902          CMP  #2
60FF 9030          BCC  LINE2700   ; IF VP0<2 THEN 2700
6101 C912          CMP  #18
6103 B02C          BCS  LINE2700   ; or IF VP0>17 THEN 2700
              ;
              ;

:BASIC: 2620 COLOR 0:PLOT 0,YP(0)+V0:COLOR 3:PLOT 0,VP0-V0:YP(0)=VP0

6105 A900          LDA  #0
6107 202063        JSR  COLOR      ; COLOR 0
              ;
610A AD0360        LDA  YP+0
610D 18            CLC
610E 6DEB62        ADC  V0         ; YP(0)+V0
6111 A8            TAY             ; is y position
6112 A900          LDA  #0
6114 AA            TAX
6115 202B63        JSR  PLOT       ; PLOT 0,YP(0)+V0
              ;
6118 A903          LDA  #3
611A 202063        JSR  COLOR      ; COLOR 3
              ;
611D ADED62        LDA  VP0
6120 38            SEC
6121 EDEB62        SBC  V0
6124 A8            TAY
6125 A900          LDA  #0
6127 AA            TAX
6128 202B63        JSR  PLOT       ; PLOT 0,VP0+V0
              ;
612B ADED62        LDA  VP0
612E 8D0360        STA  YP+0       ; YP(0)=VP0
              ;
              ;
```

www.commodore.ca

```
:BASIC: 2700 V1=PTRIG(2)-PTRIG(3):IF SINGLE OR V1=0 THEN 3000

              3150 LINE2700
              3160 ; note that what we really want is V0=+1 if
              3170 ;   stick is pushed one way and V1=-1 if
              3180 ;   stick is pushed the other.
              3190 ;
6131 AD7902   3200        LDA  STICK1    ; OS shadow location
6134 2903     3210        AND  #3        ; look at just fwd and backwd
                                             switches
6136 4903     3220        EOR  #3        ; invert the sense
6138 F006     3230        BEQ  GOTV1     ; if zero, stick not pushed
613A C901     3240        CMP  #1        ; FWD pushed?
613C F002     3250        BEQ  GOTV1     ; good...what we wanted
613E A9FF     3260        LDA  #0-1      ; must be pulled back
              3270 GOTV1
6140 8DEC62   3280        STA  V1        ; ta-da
              3290 ;
6143 ADE062   3300        LDA  SINGLE
6146 D045     3310        BNE  LINE3000  ; IF SINGLE THEN 3000
6148 ADEC62   3320        LDA  V1        ; so is stick pushed?
614B F040     3330        BEQ  LINE3000  ; or IF V1=0 THEN 3000
              3340 ;
              3350 ;


:BASIC: 2710 VP1=YP(1)-V1:IF VP1<2 OR VP1>17 THEN 3000

614D AD0460   3360        LDA  YP+1      ; YP(1)
6150 38       3370        SEC
6151 EDEC62   3380        SBC  V1
6154 8DEE62   3390        STA  VP1       ; VP1=YP(1)-V1
              3400 ;
6157 C902     3410        CMP  #2
6159 9032     3420        BCC  LINE3000  ; IF VP1<2 THEN 3000
615B C912     3430        CMP  #18
615D B02E     3440        BCS  LINE3000  ; or IF VP1>17 THEN 3000
              3450 ;
              3460 ;


:BASIC: 2720 COLOR 0:PLOT 39,YP(1)+V1:COLOR 3:PLOT 39,VP1-V1:YP(1)=VP1

615F A900     3470        LDA  #0
6161 202063   3480        JSR  COLOR     ; COLOR 0
              3490 ;
6164 AD0460   3500        LDA  YP+1
6167 18       3510        CLC
6168 6DEC62   3520        ADC  V1        ; YP(1)+V1
616B A8       3530        TAY            ; is y position
616C A900     3540        LDA  #0
616E A227     3550        LDX  #39
6170 202B63   3560        JSR  PLOT      ; PLOT 39,YP(1)+V1
              3570 ;
6173 A903     3580        LDA  #3
6175 202063   3590        JSR  COLOR     ; COLOR 3
              3600 ;
6178 ADEE62   3610        LDA  VP1
617B 38       3620        SEC
617C EDEC62   3630        SBC  V1
617F A8       3640        TAY
6180 A900     3650        LDA  #0
6182 A227     3660        LDX  #39
6184 202B63   3670        JSR  PLOT      ; PLOT 39,VP1+V1
              3680 ;
6187 ADEE62   3690        LDA  VP1
618A 8D0460   3700        STA  YP+1      ; YP(1)=VP1
              3710 ;
              3720 ;
              3730 ;


:BASIC: 3000 REM *** BALL CONTROL ***

              3740 LINE3000
              3750 ;
              3760 ;


:BASIC: 3010 COLOR 0 : PLOT X,Y

618D A900     3770        LDA  #0
618F 202063   3780        JSR  COLOR     ; COLOR 0
              3790 ;
6192 AEE262   3800        LDX  X
6195 ADE562   3810        LDA  Q.Y
6198 4A       3820        LSR  A         ; Divide half-position by 2 to
                                             get real pos'n
6199 A8       3830        TAY
619A A900     3840        LDA  #0
619C 202B63   3850        JSR  PLOT      ; PLOT X,Y
              3860 ;
              3870 ;


:BASIC: 3020 COLOR 1:PLOT XNEW,YNEW

619F A901     3880        LDA  #1
61A1 202063   3890        JSR  COLOR     ; COLOR 1
              3900 ;
61A4 AEE462   3910        LDX  XNEW
61A7 ADE762   3920        LDA  Q.YNEW
61AA 4A       3930        LSR  A         ; Divide half-position by 2 to
                                             get real pos'n
61AB A8       3940        TAY
61AC A900     3950        LDA  #0
61AE 202B63   3960        JSR  PLOT      ; PLOT XNEW,YNEW
              3970 ;
              3980 ;


:BASIC: 3030 X=XNEW:Y=YNEW

61B1 ADE462   3990        LDA  XNEW
61B4 8DE262   4000        STA  X         ; X=XNEW
              4010 ;
61B7 ADE762   4020        LDA  Q.YNEW
61BA 8DE562   4030        STA  Q.Y       ; Y=YNEW
              4040 ;
              4050 ;


:BASIC: 3040 XNEW=XNEW+XMOVE:YNEW=YNEW+YMOVE

61BD ADE462   4060        LDA  XNEW
61C0 18       4070        CLC
61C1 6DE362   4080        ADC  XMOVE
61C4 8DE462   4090        STA  XNEW      ; XNEW=XNEW+XMOVE
              4100 ;
61C7 ADE762   4110        LDA  Q.YNEW
61CA 18       4120        CLC
61CB 6DE662   4130        ADC  Q.YMOVE
61CE 8DE762   4140        STA  Q.YNEW    ; YNEW=YNEW+YMOVE
              4150 ;
              4160 ;


:BASIC: 3050 IF XNEW<38 AND XNEW>1 THEN 3200

              4170 ;
61D1 ADE462   4180        LDA  XNEW
61D4 C926     4190        CMP  #38
61D6 B004     4200        BCS  NOTTHEN3050
61D8 C902     4210        CMP  #2
61DA B04C     4220        BCS  LINE3200  ; XNEW<38 AND XNEW>1, SO GO
              4230 ;
              4240 NOTTHEN3050
              4250 ;
              4260 ;


:BASIC: 3060 HITP=(XNEW>20):XHIT=39*HITP

61DC A200     4270        LDX  #0
61DE A000     4280        LDY  #0
61E0 ADE462   4290        LDA  XNEW
61E3 C914     4300        CMP  #20       ; XNEW>20 ?
61E5 9004     4310        BCC  XNEWLT20  ; NO
61E7 A001     4320        LDY  #1        ; YES...SO 'TRUE' IS 1
61E9 A227     4330        LDX  #39
              4340 XNEWLT20
61EB 8CE962   4350        STY  HITP
61EE 8EEA62   4360        STX  XHIT
              4370 ;
              4380 ;


:BASIC: 3070 IF SINGLE THEN IF HITP THEN 3100

61F1 ADE062   4390        LDA  SINGLE
61F4 F005     4400        BEQ  LINE3080  ; NOT SINGLE
61F6 ADE962   4410        LDA  HITP
61F9 D024     4420        BNE  LINE3100  ; YES, SINGLE AND HITP
              4430 ;
              4440 ;


:BASIC: 3080 YMSAVE=YMOVE:YNEW=INT(YNEW):YMOVE=(YNEW-YP(HITP))/2

              4450 ;
              4460 LINE3080
61FB ADE662   4470        LDA  Q.YMOVE
61FE 8DE862   4480        STA  Q.YMSAVE  ; YMSAVE=YMOVE
              4490 ;
              4500 ; REMEMBER: we are using half move increments in Q.Y...
              4510 ;   variables...so we really simply want to get
              4520 ;   rid of the lowest bit (the half step)
              4530 ;
6201 ADE762   4540        LDA  Q.YNEW
```

www.commodore.ca

```
6204 29FE          AND  #$FE       ; mask off last bit
6206 8DE762        STA  Q.YNEW     ; YNEW=INT(YNEW)
              ;
6209 AEE962        LDX  HITP       ; so X is either 0 or 1
620C 4A            LSR  A          ; Q.YNEW / 2 gives the true YNEW
620D 38            SEC
620E FD0360        SBC  YP,X       ; YNEW-YP(HITP)
              ; we don't need to divide by 2, because Q.YMOVE wants
                 half-moves
6211 8DE662        STA  Q.YMOVE    ; done
              ;
              ;
```

:BASIC: 3090 IF ABS(YMOVE)>1 THEN 4000

```
6214 ADE662        LDA  Q.YMOVE
6217 C903          CMP  #3         ; halfsteps, remember
6219 9004          BCC  LINE3100   ; 0,1, or 2 halfsteps
621B C9FE          CMP  #$FE
621D 902C          BCC  LINE4000   ; aha...>2 halfsteps, <-2
                                     halfsteps
              ;
              ;
```

:BASIC: 3100 XMOVE= -XMOVE

```
              LINE3100
621F A900          LDA  #0
6221 38            SEC
6222 EDE362        SBC  XMOVE
6225 8DE362        STA  XMOVE      ; xmove = -xmove
              ;
              ;
```

:BASIC: 3200 IF YNEW=1 OR YNEW=18 THEN YMOVE= -YMOVE

```
              LINE3200
6228 ADE762        LDA  Q.YNEW
622B C902          CMP  #1+1       ; remember: half moves
622D F004          BEQ  THEN3200
622F C924          CMP  #18+18
6231 D009          BNE  NOTTHEN3200
              ;
              THEN3200
6233 A900          LDA  #0
6235 38            SEC
6236 EDE662        SBC  Q.YMOVE    ; 0-YMOVE
6239 8DE662        STA  Q.YMOVE    ; is obviously the same as -YMOVE
              ;
              NOTTHEN3200
              ;
              ;
```

:BASIC: 3290 GOTO 2600

```
              ;
              ; if we simply jumped back to LINE2600 here, the game
              ;   would play impossibly fast...
              ;   so we put in a delay
              ;
623C A900          LDA  #0
623E 8D1400        STA  CLOCK.LSB ; the 60th of a second ticker
              DELAY1
6241 AD1400        LDA  CLOCK.LSB
6244 C902          CMP  #2         ; a 30th of a second?
6246 D0F9          BNE  DELAY1
              ;
6248 4CDC60        JMP  LINE2600
              ;
              ;
```

:BASIC: 4000 REM *** the LOSE routine ***

```
              LINE4000
              ;
              ; we will score the misses, even though we don't
              ;   display the results
              ;
              ;
```

:BASIC: 4010 COLOR 0:PLOT X,Y

```
624B A900          LDA  #0
624D 202063        JSR  COLOR      ; COLOR 0
              ;
6250 AEE262        LDX  X
6253 ADE562        LDA  Q.Y        ; the half step
6256 4A            LSR  A          ; becomes an integral step
6257 A8            TAY
6258 A900          LDA  #0
625A 202B63        JSR  PLOT       ; PLOT X,Y
              ;
              ;
```

:BASIC: 4020 COLOR 1:PLOT XNEW,YNEW

```
625D A901          LDA  #1
625F 202063        JSR  COLOR      ; COLOR 1
              ;
6262 AEE462        LDX  XNEW
6265 ADE762        LDA  Q.YNEW
6268 4A            LSR  A          ; again, half step to full step
6269 A8            TAY
626A A900          LDA  #0
626C 202B63        JSR  PLOT       ; PLOT XNEW,YNEW
              ;
              ;
```

:BASIC: 4030 FOR I=1 TO 10:NEXT I

```
              ; shoddy, shoddy -- using a for/next loop for timing!
              ;
              ; here, we do it right
626F A900          LDA  #0
6271 8D1400        STA  CLOCK.LSB
              ;
              DELAY2
6274 AD1400        LDA  CLOCK.LSB
6277 C902          CMP  #2         ; tick tock yet?
6279 D0F9          BNE  DELAY2     ; nope, maybe just tick
              ;
              ;
```

:BASIC: 4040 COLOR 0:PLOT XNEW,YNEW

```
627B A900          LDA  #0
627D 202063        JSR  COLOR
              ;
6280 AEE462        LDX  XNEW
6283 ADE762        LDA  Q.YNEW     ; starting to look familiar?
6286 4A            LSR  A
6287 A8            TAY
6288 A900          LDA  #0
628A 202B63        JSR  PLOT       ; PLOT XNEW,YNEW
              ;
              ;
```

:BASIC: 4050 COLOR 2:PLOT XNEW+XMOVE,YNEW+YMSAVE

```
628D A902          LDA  #2
628F 202063        JSR  COLOR      ; COLOR 2
              ;
6292 ADE462        LDA  XNEW
6295 18            CLC
6296 6DE362        ADC  XMOVE
6299 AA            TAX             ; x register = XNEW+XMOVE
629A ADE762        LDA  Q.YNEW
629D 18            CLC
629E 6DE862        ADC  Q.YMSAVE
62A1 4A            LSR  A          ; integerize the sum
62A2 A8            TAY             ; y register = YNEW+YMSAVE
62A3 A900          LDA  #0
62A5 202B63        JSR  PLOT       ; PLOT it
              ;
              ;
```

:BASIC: 4130 SOUND 0,132,12,12:POKE 20,0

```
62A8 A984          LDA  #132
62AA 8D00D2        STA  SOUND.FREQ ; implicitly channel 0
62AD A9CC          LDA  #12*16+12
62AF 8D01D2        STA  SOUND.CONTROL ; ,12,12 also for channel 0
62B2 A900          LDA  #0
62B4 8D1400        STA  CLOCK.LSB ; finally, BASIC did it right!
              ;
              ;
```

:BASIC: 4140 SETCOLOR 1,0,PEEK(20)*4:IF PEEK(20)<32 THEN 4140

```
              LINE4140
62B7 AD1400        LDA  CLOCK.LSB ; same as PEEK(20)
62BA 0A            ASL  A
62BB 0A            ASL  A          ; * 4
62BC 8DC502        STA  SETCOLOR1 ; control register number 1
              ;
```

www.commodore.ca

```
62BF C980   5920        CMP  #32*4     ; a little tricky...can you
                                            follow it?
62C1 90F4   5930        BCC  LINE4140  ; it works...really
            5940 ;
            5950 ;

:BASIC: 4150 SOUND 0,0,0,0

62C3 A900   5960        LDA  #0
62C5 8D00D2 5970        STA  SOUND.FREQ
62C8 8D01D2 5980        STA  SOUND.CONTROL
            5990 ;
            6000 ;

:BASIC: 4200 REM *** SCORE IT ***

            6010 ;
            6020 ;

:BASIC: 4210 SCORE(HITP)=SCORE(HITP)+1

62CB AEE962 6030        LDX  HITP
62CE FE0560 6040        INC  SCORE,X   ; isn't assembler easy?
            6050 ;
            6060 ;

:BASIC: 4220 LASTWIN=1 : IF HITP THEN LASTWIN=-LASTWIN

62D1 A901   6070        LDA  #1
62D3 AEE962 6080        LDX  HITP      ; if HITP?
62D6 F002   6090        BEQ  NOT.HITP  ; no
62D8 A9FF   6100        LDA  #0-1      ; yes...so make it -1
            6110 NOT.HITP
62DA 8DE162 6120        STA  LASTWIN   ; that's all that is needed
            6130 ;
            6140 ;

:BASIC: 4990 GOTO 2000

            6150 ;
62DD 4C2360 6160        JMP  LINE2000
            6170 ;

BOING — not quite up to PONG
GENERAL RAM USAGE

62E0        6180        .PAGE "GENERAL RAM USAGE"
            6190 ;
62E0 00     6200 SINGLE BRK            ; flag for one-player game
62E1 00     6210 LASTWIN BRK           ; who won last time?
            6220 ;
            6230 ; the x moves
            6240 ;
62E2 00     6250 X      BRK            ; current x position
62E3 00     6260 XMOVE  BRK            ; current x movement
62E4 00     6270 XNEW   BRK            ; new x position
            6280 ;
            6290 ; and the y positions and moves
            6300 ;
            6310 ; remember: the Q.Yxxx locations reference positions
            6320 ;   or movements in terms of half steps
            6330 ;
62E5 00     6340 Q.Y    BRK            ; current y position
62E6 00     6350 Q.YMOVE BRK           ; current y movement
62E7 00     6360 Q.YNEW BRK            ; new y position
62E8 00     6370 Q.YMSAVE BRK          ; saved for LOSE routine only
            6380 ;
            6390 ; other misscellany
            6400 ;
62E9 00     6410 HITP   BRK            ; the HIT Person...who missed
62EA 00     6420 XHIT   BRK            ; where the miss occurred
                                           (x position)
            6430 ;
62EB 00     6440 V0     BRK            ; just a temporary
62EC 00     6450 V1     BRK            ; ditto
            6460 ;
62ED 00     6470 VP0    BRK            ; Vertical position of Paddle 0
62EE 00     6480 VP1    BRK            ; Vertical position of Paddle 1
            6490 ;
            6500 ; system equates
            6510 ;
0012        6520 CLOCK   =   18        ; the system clock
0014        6530 CLOCK.LSB = CLOCK+2   ; the 60th of a second ticker
0278        6540 STICK0 =   $278       ; OS shadow read of first stick
0279        6550 STICK1 =   $279       ; ditto for second stick
D200        6560 SOUND.FREQ = $D200    ; port which controls channel
                                           0 freq
D201        6570 SOUND.CONTROL = $D201 ; and control
            6580 ;
02C5        6590 SETCOLOR1 = $2C5      ; also known as COLPF1
```

```
BOING — not quite up to PONG
The GRAPHICS subroutines

62EF        6600        .PAGE "The GRAPHICS subroutines"

6372        6630        .OPT LIST
6372        6640        .END
```

[Put the graphics subroutines from line 9000 on up (pg. 150, **COMPUTE!**, *August 1982) here.]*

ATARI™ 400/800 OWNERS!

Discover

DATAPORT

YOUR SOURCE FOR THE BEST IN ENTERTAINMENT AND EDUCATIONAL SOFTWARE! OUR NEW CATALOG LISTS PAGE AFTER PAGE OF GAMES ON BOTH CASSETTE AND DISK! SEND $1.00 NOW TO:

DATAPORT
P.O. BOX 975
NORTHBROOK, IL 60062

GAMES

FOR THE

ATARI 400/800

OVER 100 GAMES, SIMULATIONS, ADVENTURES AND MORE!!

20% OFF LIST PRICE!

Artworx · Adventure International · On Line Systems · CE
Automated Simulations(EPYX) · Arcade Plus
Gebelli · Avalon Hill · Crystal · Broderbund · IDSI
Budgeco · Datasoft (and more!!)

FREE CATALOG, NEWSLETTER

TO ORDER CALL
(412) 235-2970
OR WRITE

mideastern software

BOX 247 NEW FLORENCE, PA. 15944

add $2.00 shipping/handling per order
PA residents add 6% sales tax

www.commodore.ca

Telecommunications

TELEGAMING

Michael E. Day
Chief Engineer
Edge Technology
West Linn, OR

Telegaming brings to mind many things, from simple games played via a telephone link to interactive games such as chess, and on to multiple-participant *macrogames*. Indeed, the farther you go when thinking along these lines, the more difficult it becomes to separate gaming from real life, simulations from the events they imitate.

Actually, telegaming has been around for a long time. Probably the earliest form of telegaming was the use of couriers to carry letters between two or more individuals noting the moves of the particular game in progress. The official postal service eventually replaced the couriers. Later, with the development of the ability to communicate via electrical means, telegaming as we would normally consider it – via electrical communication devices – came about.

One game that has received notice in this regard is chess, which lends itself easily to telegaming since strategy is of greater importance than speed. There are many chess games in progress at this very moment by mail, by telephone, by radio, and yes, even by computer.

Telegaming is certainly not just for computers though. Airborne television and cable television can (in some locales do) support telegaming. In Britain, one major system is the Prestel Videotex system, which uses the television in conjunction with the telephone to provide its services. The Prestel system currently supports approximately 16,000 users. While the system normally provides the usual fare of stock reports, news, etc., it also provides for telegaming.

Even something as simple as gaming can sometimes run afoul of politics, however. Last April, during the Falkland Islands problem, Prestel added a video game called "Obliterate." The object was to sink an Argentine flagship. A good shot brought the comment, "Well done, sir! You are a national hero. Horatio would be proud of you," while a poor shot would elicit, "Your poor judgment is endangering the reputation of your country and giving the enemy a chance to retaliate." A rather loud protest from the House of Commons scotched the game after only a week.

Five Adventurers, Three Maps

Telegaming is, of course, not limited to television. In fact, gaming via a terminal to a master computer at some remote location – which allows access to complex games not normally available to the game player – is more common. These games are often provided by timeshare computer networks such as "CompuServe." While other special-purpose computer systems for public use can support gaming, they seldom make it available, largely because there is only a single phone line to the system. In order to allow high volume use of the single line, such systems necessarily limit gaming activities.

While many private systems do not restrict use, systems available for free public use are mostly privately supported. The timeshare networks, which have the multiple communications capability already installed, do not have the restriction problem of the smaller private systems. In fact, they charge for the use of their facilities and, to increase revenues, tend to provide games which are oriented to lengthy line times and, if possible, more than one user.

One of the more popular games is the multiuser adventure, which allows more than one person to play at once. This adds interest: there is now competition for the available resources of the simulation. There might be five adventurers but only three treasure maps.

One problem with the current telegaming structure is response time. In order to have the fast response time needed for interactive gaming, you must be in direct contact with the gaming computer. This means line charges are accumulated even when you're not actively communicating with the system. In games such as chess where the response time is not critical, you can avoid these charges by not staying in direct communication, but instead breaking the link and calling back at a later time after the next move has been planned.

Having a reasonable response time while not actually using the communications link would lead to increased telegaming by reducing the connect time and its associated cost. Some interactive cable systems come close to this. While many still require that the communications be done via the telephone, some provide the ability to interrogate the "black box" on the TV set which attaches the set to the cable, providing a lower cost means of returning information to the cable system. The limitation here is that the system must interrogate each set on the line to get information, and this can slow performance in interactive game uses.

An interactive telegaming system of this sort could be of immense use to the general telecommunications market. Widespread use of interactive data systems is now impeded by requirements to

www.commodore.ca

get on to the system and by the charges generated once there. The usual method of operation is to plan for the activity ahead of time, call up the data base, get the desired information, and get off as quickly as possible. This means that the information is being inefficiently used since only the known information is being retrieved.

Metagames

Ideally, both a proper information retrieval system and a good interactive telegaming system should be easy to access and inexpensive.

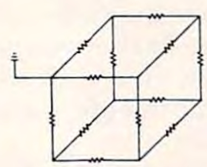
There is one company around that could build such a system. Bell Telephone already has a communications network in place that is easy to use and relatively inexpensive to operate. One major problem is that it still can take ten to 30 seconds or more to establish a connection to another phone. This means that the information retrieval/gaming system would have to have a different means of access if it was intended to be disconnected between operations. This could be done, but would be more expensive than the current method of telephone interconnect since more equipment would be involved. If the data access/gaming computer is not located inside the local exchange, high priority lines to the computer will also have to be accounted for. All this, of course, adds to the cost.

In the end, a quick retrieval data system will be implemented one way or another, simply because there is a need for it. How it will be implemented is yet to be seen. Once the system has been implemented, telegaming will quickly follow. If, however, telegaming comes into being first, a data system will soon follow it. Both systems require the same type of telecommunications capabilities; it is simply a matter of which will be first.

It is perhaps not too far off when we will be able to join in metagames – simulations so large that they are, in effect, hard to distinguish from reality. If memory becomes very cheap and computer switching becomes very fast, games might be built which contain so many variables that nearly any decision (or move) could be accommodated by the game. Add telecommunication to this metagame, and you have historical re-enactments or imaginary events taking place all over the world simultaneously (on videoscreens or in "environmental rooms"). An adventure game could take months or years to reach its conclusion.

You could join an army as a private and, after months of part-time "playing," you could work your way up to become a general or a spy or whatever. All the players would join or leave this network simulation as their time and interests permitted. Imagine a computer-controlled, world-wide simulation so full, so convincing that millions of players could experience (and influence) a make-believe first contact with aliens. You might be assigned to the team which decodes their language, or you might choose to just watch the event unfold on the Simulated Evening News. Whatever happens, the coming marriage of games and telecommunications will bring about some surprises. [For additional thoughts about gaming in the coming years, see "Future Games" elsewhere in this issue.] ©

CIRCUIT SOLVER I



You don't have to be an engineer to make productive use of this program. A large clear manual leads you through the use of Circuit Solver using sample problems which have practical application.

Features

- Simple Circuit Entry
- Circuit Listing
- Circuit Storage & Retrival
- Easy Circuit Editing
- Supports Op-Amps
- Many Useful Sample Circuits

SOLUTIONS
1430 N. LATRUBE
SUITE 2A
CHICAGO, IL 60651

ORDER #

CSI-100A	PET/CBM*	$34.95
CSI-101A	APPLE*	34.95
CSI-102A	TRS-80*	34.95

Include $2.50 shipping and handling.

* PET/CBM, APPLE and TRS-80 are ® Trademarks of Commodore, Apple Computer and Tandy Pesp.

MasterCard VISA

PION introduces the INTERSTELLAR DRIVE.

A solid state disk emulator for your APPLE*, TRS80, S100, or SS50 computer**

- A FAST mass storage device. Speeds up any program requiring disk access.
- No head seek time, no motor startup time, no moving parts.
- Standard 256K bytes of storage expandable to 1 megabyte.
- Independent regulated power supply.
- Automatic power failure detect and battery backup.
- Hardware error detection and write protect.
- Only 4 bytes-ports of address I/O space used
- Hardware optimized for block transfers and access.
- Drivers, diagnostics, and utilities software provided.
- Certified to comply with Class B limits Part 15 of FCC rules.

Introductory Price $1095.
plus tax and shipping

MasterCard VISA

SEND FOR FREE BROCHURE
Identify your model.

PION, INC. Tel. (617)648-1717
74 Appleton St., Arlington, MA 02174

*Trade Mark Apple **Trade Mark Tandy Corp.

www.commodore.ca

A Monthly Column

Machine Language: The Beginner's Dilemma

Jim Butterfield
Associate Editor

The beginner in machine language programming is faced with a three-way task. It's not enough to learn about machine code itself; the beginner must also develop skills on the particular machine that has been chosen. These extra skills fall into two general categories: using the tools that are provided and finding your way around the architecture.

Machine Language Itself

The machine code is the easiest part. There are numerous books and reference sources that will supply this information.

The matter is confused slightly by approaches and titles. Some teach machine language, some teach assembly language, and others identify themselves as books on "programming" or "program design." There isn't really much difference; they all develop the same skills.

I tend to favor learning the machine itself first – hexadecimal codes and such – and working up to the more general assembly language level later. It seems to me that if you can retain a firm image of the instructions as they lie in memory, you will always have a strong feeling for the real nature of the machine. For a beginner, assemblers do too much; it's easy to lose touch with how the machine is really doing the job. Later, assemblers will prove to be a powerful aid to programming, but they may be too powerful for the beginner.

But the neophyte may find himself blocked at the start. It's all very well to read about these codes, but how do you get them into the machine? And how do the codes create output to screen or printer?

Machine Language Tools: Monitors And Interfaces

The user needs some understanding of the *monitor* before he can do anything useful. This is the tool that allows him to enter code into the machine; to check code for correctness; to initiate a program test run; and to intercept a program during the run in order to investigate its performance.

The monitor for a given machine may come in many forms. It may be built in, or loaded from tape or disk, or plugged in as a ROM cartridge. A given system may have one monitor, or a choice of several, or even extensions that can be added to a built-in monitor. Variety may be the spice of life, but it makes things difficult for textbooks. It's easy to show how to add two numbers together with a 6502; the coding is the same for all systems. But an outline of how to put this addition program into the computer must vary from machine to machine, from monitor to monitor.

There's another problem that needs to be solved. Different machines call for different interfaces to input and output. As a result, a general textbook can't complete the picture, since the input and output mechanics vary from machine to machine. On Commodore products, output (print) is generated by a call to $FFD2; but the identical activity on Apple, KIM, Atari, AIM, or OSI is coded in a manner unique to that machine. Pity the poor machine language book author: he/she can't complete the picture without either tying himself to a specific machine or attaching a long rambling list of interfaces.

Architecture

Even identifying the tools specific to your machine isn't enough. We need to know how the machine is structured: in particular, what parts of memory are used for what purposes.

Where is the screen? It's often memory-mapped, but might be one place on a PET and another place on an Apple. Machines like Atari and VIC have "mobile" screens. There are several places in memory which might reflect the screen, depending on circumstances.

More importantly: what space is available on your computer, and what is in use? It's hard to enter a program into your computer if you don't know how to find or create a safe place in RAM for the program to go.

Again, it's hard for the textbook. Either it specializes in your machine, or leaves the poor beginner without the information he needs to fit the program to his machine.

You cannot effectively learn machine language in a vacuum. Each learner must have a chance to try his hand at coding the things he learns. Yet it seems to the beginner that he's being prevented from doing this: his books don't tell him enough.

Try to gain information on your machine. It may come from various sources: manufacturer's

www.commodore.ca

literature, books, magazine articles, clubs, or examination of other people's programs.

Learn how to use the tools, especially the monitor. Find the best input/output interfaces for your machine. Study the mapping to find safe places to put your programs.

You'll find that all three skills will develop together. You'll learn machine language, machine tools, and machine architecture at the same time. Later you may want to transfer your skills to another machine and may need to learn new tools and architecture. By that time, you'll know enough about the whole machine environment to pick up very quickly.

For the beginner, machine language programming often seems to be an insurmountable obstacle. No single book gives all things needed to make a decent start. But a minimum set of skills can be developed, and after that the path becomes much easier.

An old joke tells of a drunk who falls down an open elevator shaft and then calls back to his friend, "Watch that first step – it's a big one!" The first step in machine language learning is a big one, too, but it sets the stage for unlimited further development – painlessly. ©

JINSAM™

★ FAST/EASY/MENU DRIVEN
★ UNLIMITED FIELDS & RECORD LENGTHS
★ MERGE (add) OR GENERATE FILES
★ MULTIPLE SEARCH CRITERIA
★ STATISTICAL PACKAGE
★ INTERFAÇE VISICALC™
★ INTERFACE WORDPRO™ & WORDCRAFT™
★ HELP COMMANDS
★ PRIVATE ACCESS CODES
★ WILD CARD SEARCH

Data Management Systems
For CBM™ & IBM PC™

Includes Free: **REPORT GENERATOR**
MAIL LABEL PRINTER

"Much more powerful than you can imagine"
Kilobaud Microcomputing Magazine
"JINSAM is thorough, totally flexible, well documented, well programed, and sophisticated."
Educational Computer Magazine

Send ☐ Information ☐ Newsletters ($5) ☐ Manual ($40)
☐ Demo Disk (CBM or IBM PC $15 ea. Circle One)
— DEALER INQUIRIES WELCOMED —

JINI MICRO SYSTEMS, INC.
P.O. Box 274 • Riverdale, NY 10463
(212) 796-6200

YES! We said "SORT VISICALC"!

versacalcT.M.

A UNIQUE VISICALC(tm) ENHANCEMENT

NOW AVAILABLE FOR PET

If you use Visicalc(tm) but are bumping into its limitations, you need Versacalc(tm)! Versacalc runs within Visicalc but uses no extra memory; in fact, it effectively increases memory by letting you call in modules from disk as needed.

A Tutorial section makes clear such features as @LOOKUP, DIF, @NA, @ERROR, which are not well explained in the Visicalc manual.

A Utilities section makes it easy to create your own menu-driven modules which condense hundreds of commands into four keystrokes. You can build in sophisticated error checking (e.g. Is the input value between certain limits?). Now it is possible for people untrained in Visicalc to perform the weekly updating without constant instruction.

for Commodore Pet & CBM 2040 - DOS 1.0
available for 8050 on request

Retail $125

SORT Visicalc, and Other Useful Wonders!

NOW YOU CAN:
SORT a Visicalc screen on any column, ascending or descending, and all related formulas and labels are sorted too.
put the entire disk CATALOG on the screen at once!
easily do Year-To-Date accumulations!
"pound" formulas to expose the full formulas in place on the screen!
append two Visicalc files!
print the contents of a /SS file!
print the contents of a /PF file!
AND our EASEL BINDER is so nice that you will put your other manual in it!

Distributed by:
Anthro-Digital Software
P.O. Box 1385
Pittsfield, MA 01202
413-448-8278

Versacalc is a trademark of Versacalc Enterprises, Inc.
Visicalc is a trademark of Personal Software, Inc.

www.commodore.ca

With this technique, your PET/CBM (4.0 BASIC or Upgrade, 16 or 32K) can store and play digitized speech. No special hardware is required. This program lets the PET digitize, store, playback and monitor speech or other audio signals from the tape deck. It also is a beginning for the processing of the digitized audio signals and can be used for a rudimentary voice-print analysis allowing you to discriminate between different people's voices.

Digital Speech

Kenneth Finn
Bedford, NY

The machine language of the program (from $033A to $03BD) is called Voice-Rec. Its job is to take the information from the cassette tape and store it in memory from $1000 to $4000. (Note: If you change locations $035E and $0391 to $30 [which is machine language BMI] the memory will be saved from $1000 to $8000.)

A 20K Hz Sampling Rate

This program is interesting in several ways. The first is that the sampling time for the audio signal has been reduced to about 41 microseconds. This corresponds to a sampling rate of better than 20 KHz/second. One of the ways this was accomplished was by taking the program and practically duplicating it for the high-low and low-high transitions on CA1, which is the cassette read head. The sections from $0349-$037B and $037C-$03AE are almost identical. This was done to make the sampling rate the fastest possible.

Another peculiarity of the program is that the data is packed. Each memory byte contains either a number or $FF, which means an overflow. The number corresponds to how many 40 microsecond loops went by before the signal changed from high to low or vice versa. This packing method allowed us to store about 20 seconds of audio in the 12K of memory allocated. While this does not seem like much time, remember that about 20K samples are taken every second. Without this packing, the entire 32K PET would fill up in about one and a half seconds. This packing is made possible by the silent periods between words and the presence of other low frequency components of human voice.

A third peculiarity of this program is that the paths taken by the program for the three possible conditions – no transition, overflow, transition – have all been equalized to within four or five microseconds. This is evident by the number of NOPs or ($EA) in the program.

The second section of the program is called Voice. It goes from $03B0 to $03F6. It has been previously published in the November 1981 issue of **COMPUTE!** but has been modified here so that it can be co-resident with Voice-Rec; the two programs go well together. Its job is to allow you to position a voice tape by monitoring or listening to what is on it. It is very useful when you are trying to get the tape set up to record a specific segment of it.

A couple of things about it are interesting. First of all, it shows you how the stop key, the CB2 line, the tape read line, and the cassette motor can all be used from machine language. Second, it has an even higher sampling rate than Voice-Rec. Both this program and Voice-Rec can be in the second cassette buffer without any trouble, or they can be separated easily. Both are also relocatable.

The third part of the program is called Voice-Play. It goes from $033A to $03B3, and it can play back the recorded speech from memory $1000 to $4000. (This program also can be modified by changing $036C and $0397 to $30 or BMI, and then it will play from $1000 to $8000.)

It has been designed to work with Voice-Rec in a similar way. Its timing loops at 43 microseconds match closely the loops of Voice-Rec; the playback is at least uniform, if not good.

Now let's examine the process that we have been using and see what we can now do with our digitized voice. What we have been doing is making the PET into a one-bit analog to digital converter. Another way of describing the process is saying that we have been making a record of an infinitely clipped signal. While this method is not quite as good as using an eight-bit ADC, it at least has the benefit of allowing us to get some experience cheaply and can be improved by the use of a good amplifier with tone controls on the PET's CB2 line. Since we are not capturing the signal in a very sophisticated way, I have chosen to make the sampling rate as high as possible to make up for it. That is why the first two program sections were not merged.

Let's begin by looking at the digitized data that we made and seeing how densely it has been packed.

```
10 POKE53,13:POKE52,0:CLR
20 FORI=4096TO16384
30 S=S+PEEK(I)
40 NEXTI:PRINTS/12288
```

www.commodore.ca

This little program will produce the average byte value in the program. When I ran it, I got about 32, the average number of samples packed into each byte. This is why we can compress 20 seconds of information at a 20K Hz sample rate into only 12K bytes of memory.

Voice Analysis

A second analysis of the program was to produce a histogram of the signal. Remember that each byte represents a sort of instantaneous frequency. Thus, we want to examine what amounts of each frequency were present.

```
10 POKE53,13:POKE52,0:CLR:DIMA%(256)
20 FORI=4096TO16384
30 A%(PEEK(I))=A%(PEEK(I))+1:NEXTI
40 OPEN4,4,0
50 FORI=1TO70
60 PRINTA%(I),A%(I+70),A%(I+140),A%(I+210)
70 NEXTI:CLOSE4:END
```

This little program will produce a histogram, running down the page, on the PET printer. For the sample that I used, the majority of the important information was contained in the first 50 or so numbers running down. This is not too surprising, since the average value of the sample was 32. (Note, please, that overflow samples of 256 or $FF were not really treated correctly in this little analysis. They should have been added to the next following byte to get the correct frequency.) This data is a kind of voice-print for a person's speech. If you have different people say the same thing into a tape recorder and then analyze each voice with our system, you will get a separate voice-print. Women's voices, since they tend to be higher, will have higher amounts of lower numbers, which correspond to the higher frequency. While this system is crude, it does provide a departure point.

A third analysis of this data is to transform the signal via differentiation. Before you wring your hands in despair, remember that we are dealing with digitized information, and all we have to do is to transform the data by taking the difference between each number in our stored data base. The ease with which we can manipulate a signal once it is in memory is why we started this project in the first place.

Another thing we can do quite easily is to filter the signal any way we like. Try adding two or three numbers to each datum, and see how each modification changes the signal. While this technique is not strictly a filter, it illustrates the idea that digital processing of speech data is useful.

Remember that once the rough parts of the work have been done in machine language, the fun parts can be done in BASIC. This makes it simple to process the data.

One final point. Up to now we have been working in the time domain. We have a representation of how the voice looks at each point in time. There are other ways we can present this signal. While the other methods cannot mathematically tell us more about the signal, they can give us other ways to look at it.

One famous method is to transform the signal into the frequency domain by using a fourier transform. This analysis gives an altogether different type of histogram of the signal.

How To Use The PET/CBM Software Voice Synthesizer

The program is a combination BASIC loader and a runtime helper. When RUN, it loads the machine language programs from the DATA statements. Type each number carefully, and save the program before you run it, in case you've made an error (remember to change the indicated lines if you have Upgrade ROMs or 16K memory).

The program presents you with three options: Monitor Tape, Record, and Play. The Monitor program simply plays the tape. Press the RUN/STOP key to stop monitoring. You must press RUN/STOP while the tape is playing something audible, or the program won't acknowledge you. If you press it quickly, without holding down, you'll be returned to the menu of options; otherwise you'll see the message: BREAK AT LINE XXX. You can type RUN to restart the program.

When you're ready to record the tape into your computer's memory, press PLAY on the tape player first, then press R for Record. The tape will run for about 20 seconds. You can then listen to the digitized voice or sound with Play. The quality is best with an external CB2 speaker (some 4032's and all 8032's have a built-in piezoelectric "bell" that can produce low volume, high pitched CB2 sound). You can attach an amplifier to pins M and N on the user port if you want to add CB2 sound.

Change these lines for a 16K PET/CBM

```
1090 DATA 36, 37, 112, 45, 234, 234
1160 DATA 234, 36, 37, 112, 2, 80
1290 DATA 112, 80, 160, 6, 136, 208
1370 DATA 184, 36, 37, 112, 29, 160
```

Change these lines for Upgrade ROM PET/CBM

```
1170 DATA 173, 32, 123, 252, 88, 96
1420 DATA 16, 219, 48, 153, 32, 123
```

www.commodore.ca

Z-RAM™ Opens Up The World Of CP/M™ To The Commodore Computer!

Z-RAM, a circuit board that fits inside the Commodore 4000 and 8000 computers, adds a Z-80A microprocessor and 64K of Random Access Memory [RAM], tripling the current maximum user memory!

Also provided with Z-RAM is the CP/M operating system, adding a new dimension to the Commodore computer by finally making the vast world of CP/M software available to its users.

With the addition of the Z-RAM board, Commodore computers will operate with 96K RAM and use both the 6502 and Z-80 processors. Several modes of operation are possible:

- With Z-RAM installed, all earlier versions of Commodore computers [regardless of memory capacity] can function as 32K machines. All current programs will run using 32K memory and the standard 6502 processor.
- Z-RAM permits the 6502 to use the full 96K of memory to operate the new expanded versions of Wordcraft Ultra, WordPro -Plus™ and other expanded Commodore programs.
- The Z-80A processor will operate in 64K memory and will run the industry standard CP/M operating system. WordStar™, Super-Calc™, and Accounting II Plus™ are only a sample of the fantastic CP/M programs now available to run on your Commodore computer!

Contact Your Nearest Commodore Dealer Today . . .
You'll Be So Glad You Did!

Distributed by:

COMPUTER MARKETING SERVICES INC.

[609] 795-9480
300 W. Marlton Pike, Cherry Hill, New Jersey 08034

Z-RAM is a trademark of Madison Computer
CP/M is a trademark of Digital Research

www.commodore.ca

```
100 POKE53,16:POKE52,Ø:CLR:GOSUB1ØØØ
105 PRINT"{CLEAR}{REV}VOICE SYNTHESIS{DOWN}":P
    OKE59468,12
110 PRINT"{REV}M{OFF}ONITOR TAPE{DOWN}":PRINT"
    {REV}R{OFF}ECORD{DOWN}":PRINT"{REV}P{
    OFF}LAY{Ø2 DOWN}"
120 PRINT"{UP}CHOICE?":GETA$:IFA$=""THEN12Ø
130 ON -(A$="M")-(A$="R")*2-(A$="P")*3 GOSUB 1
    5Ø,16Ø,17Ø
140 GOTO1Ø5
150 PRINT"{DOWN}PRESS {REV}RUN/STOP{OFF} TO QU
    IT":SYS958:RETURN
160 PRINT"{DOWN}NOW RECORDING...":SYS826:RETUR
    N
170 PRINT"{DOWN}PLAYING...":SYS634:RETURN
1000 FOR ADRES=634TO759:READ DATTA:POKEADRES,DA
    TTA:NEXT ADRES
1010 DATA 12Ø, 2Ø6, 19, 232, 169, 16
1020 DATA 133, 37, 165, Ø, 133, 36
1030 DATA 168, 17Ø, 177, 36, 17Ø, 197
1040 DATA 255, 2Ø8, 5, 32, 225, 2
1050 DATA 8Ø, 244, 169, 2Ø4, 141, 76
1060 DATA 232, 2Ø2, 24Ø, 7, 16Ø, 1Ø
1070 DATA 136, 2Ø8, 253, 24Ø, 246, 23Ø
1080 DATA 36, 2Ø8, 2, 23Ø, 37, 184
1090 DATA 36, 37, 48, 45, 234, 234
1100 DATA 234, 177, 36, 17Ø, 197, 255
1110 DATA 2Ø8, 5, 32, 225, 2, 8Ø
1120 DATA 244, 169, 236, 141, 76, 232
1130 DATA 2Ø2, 24Ø, 7, 16Ø, 1Ø, 136
1140 DATA 2Ø8, 253, 24Ø, 246, 23Ø, 36
1150 DATA 2Ø8, 2, 23Ø, 37, 184, 234
1160 DATA 234, 36, 37, 48, 2, 8Ø
1170 DATA 173, 32, 192, 252, 88, 96
1180 DATA 234, 2Ø2, 24Ø, 7, 16Ø, 1Ø
1190 DATA 136, 2Ø8, 253, 24Ø, 246, 23Ø
1200 DATA 36, 2Ø8, 2, 23Ø, 37, 184
1210 DATA 96, 88, 169, 52, 133, 249
1220 FOR ADRES=826TO1Ø14:READ DATTA:POKE ADRES,
    DATTA:NEXT ADRES
1225 RETURN
1230 DATA 12Ø, 169, 16, 133, 37, 169
1240 DATA Ø, 133, 36, 17Ø, 169, 53
1250 DATA 141, 19, 232, 173, 16, 232
1260 DATA 169, 6Ø, 141, 17, 232, 44
1270 DATA 17, 232, 48, 21, 232, 224
1280 DATA 255, 24Ø, 12, 184, 36, 37
1290 DATA 48, 8Ø, 16Ø, 6, 136, 2Ø8
1300 DATA 253, 24Ø, 234, 234, 234, 234
1310 DATA 234, 138, 145, 36, 162, Ø
1320 DATA 23Ø, 36, 2Ø8, 2, 23Ø, 37
1330 DATA 44, 17, 232, 16, 219, 234
1340 DATA 173, 16, 232, 169, 62, 141
1350 DATA 17, 232, 44, 17, 232, 48
1360 DATA 21, 232, 224, 255, 24Ø, 12
1370 DATA 184, 36, 37, 48, 29, 16Ø
1380 DATA 6, 136, 2Ø8, 253, 24Ø, 234
1390 DATA 234, 234, 234, 234, 138, 145
1400 DATA 36, 162, Ø, 23Ø, 36, 2Ø8
1410 DATA 2, 23Ø, 37, 44, 17, 232
1420 DATA 16, 219, 48, 153, 32, 192
1430 DATA 252, 88, 169, 52, 133, 249
1440 DATA 169, 61, 141, 19, 232, 96
1450 DATA 12Ø, 169, 53, 141, 19, 232
1460 DATA 169, 249, 141, 16, 232, 169
1470 DATA 16, 45, 18, 232, 24Ø, 224
1480 DATA 173, 16, 232, 169, 62, 141
1490 DATA 17, 232, 44, 17, 232, 16
1500 DATA 251, 169, 2Ø4, 141, 76, 232
1510 DATA 173, 16, 232, 169, 6Ø, 141
1520 DATA 17, 232, 44, 17, 232, 16
1530 DATA 251, 169, 236, 141, 76, 232
1540 DATA 2Ø8, 2Ø6, Ø, Ø, Ø, Ø
```

©

CONVERT YOUR PET INTO A TERMINAL
$129.95

RS232 Hardware and cable, and sophisticated terminal software. Upload and Download, communicates in ASCII, status line, built-in file translator. A complete package, all you need is a modem and we sell them too.

Communicate with Compuserve, Source, etc. Upload/Download to/from 4040 or 8050. Drives ASCII or PET Printer. Comm. in ASCII. Status line, Toll Timmer.

Super Saver Package Deals:
STCP (129.95) and Hayes Smart Modem (279) - $365.00
STCP (129.95) and Signalman Modem (99) - $215.00

Eastern House
3239 Linda Dr
Winston Salem, N.C. 27106
(919) 924 2889 (919) 748 8446

Verbatim
flexible disks

Call Free **(800) 235-4137** for prices and information. Dealer inquiries invited. C.O.D. and charge cards accepted.

PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. In Cal. call
(800) 592-5935 or
(805) 543-1037.

VISA

COMMODORE USERS

Join the largest, active Commodore users group in North America and get—

— Access to club library of over 3000 programs.

— Informative club newsletter.

— Access to the combined talents of some of the most knowledgeable people on PET/CBM/VIC/C-64.

Send $20.00 ($30.00 overseas) for Associate Membership to:

Toronto Pet Users Group
P.O. Box 100 Station S
Toronto, Ontario, Canada M5M 4L6

www.commodore.ca

Programming The PET/CBM

by Raeto Collin West

The book described by Jim Butterfield as

"...unquestionably the most comprehensive and accurate reference I have seen to date..."

The Reference Encyclopedia for Commodore 2000, 3000, 4000, and 8000 series computers and peripherals.

Here's just a sample of reviewer and reader reaction:

From reviewers:

Educational Computing Review by Stephen Potts
"Of all the books I have read on the PET this book *Programming the PET/CBM* by Raeto West must rank as one of the most comprehensive and readable accounts on the PET that I have ever had the pleasure to see...

"If you wish to get more from your PET than arcade games and simple teaching programs then this book is a must for your bookshelf. It does not matter whether you run on BASIC 1, BASIC 2, or BASIC 4 since all routines are supplied with addresses and changes to make them run on any machines wherever possible...

"...this book, with its lucid explanations of the PET, its useful routines and programming hints, is an essential purchase."

IPUG Magazine Review (British PET User Group) by Ron Geere
"This publication represents over a year's intensive research ... and the resulting product is a valuable work of reference. A tremendous amount of useful information has been packed in this 500+ page work at which I was so over-awed that I did not know how to start this review at first...

"This book is a must for every CBM/PET user."

From readers:

"...a book the average to advanced user cannot afford *not* to possess..."

"My copy of your *'Programming the PET/CBM* has been in daily use for nearly a month and I am finding it totally addictive, suffering severe withdrawal symptoms whenever I try half-heartedly to move on to other reading matter. It is without doubt the best book on its subject available today..."

"I have recently acquired a copy of your book *Programming the PET/CBM* and must congratulate you on its concept and on packing in so much detail. It's so very much better than anything I have had up to now that it'll be my constant reference manual."

"I have received my copy of *Programming the PET/CBM* by Raeto West and I have recommended it to several of my students. This book is so valuable that I cannot now afford to be without it."

Published exclusively in North America by **COMPUTE! Books**. The book is an astonishing reference manual of useful information. Contents include this and much more:

1 **Introduction and overview:** Plan of the book, sources of information, features and chronology of CBM hardware.
2 **BASIC and how it works:** Storage of BASIC and its variables; tokens, pointers, syntax; optimising BASIC.
3 **Program and system design:** Capabilities of the equipment; charts, algorithms, space, timing.
4 **Effective programming in BASIC:** Seventeen examples, including subroutines, dates, DATA, INPUT, rounding.
5 **Alphabetic reference to BASIC keywords:** Full descriptions, with examples, of all keywords, with methods for adding additional commands not present in CBM BASIC, e.g. AUTO, DEL, OLD, POP, PRINT USING, SORT, VARPTR.
6 **Disk drives:** Descriptions of operation and workings of disk drives, with BASIC and machine-code examples; bugs.
7 **Alphabetic reference to disk BASIC commands:** BASIC 4 disk commands with examples and notes.
8 **Other peripherals and hardware:** Tape storage and handling; printers; modem; keyboard; EPROMs; reset switches.
9 **Graphics and sound:** Tables of CBM characters; CRT chip; animation, bar plots, 80 by 50 etc.; user-port sound.
10 **The transition to machine-code:** Introductory concepts; a BASIC monitor; use of MLM, Supermon, Extramon; easy examples.
11 **More 6502 machine-code:** 6502 hardware features; eighteen common problems in programming; debugging.
12 **Alphabetic reference to 6502 opcodes:** Examples, notes, and explanations on each opcode from ADC to TYA.
13 **Using ROM routines:** IRQ, NMI, RESET; the Kernel; examples – modifying LIST; ordinary and relocating loaders.
14 **Effective 6502 programming:** Assemblers; CHRGET and wedges; PIAs, VIA, IEEE; common mistakes.
15 **Index to BASIC ROMs and RAM:** Memory map of RAM and ROM, detailing and comparing BASICs 1, 2, and 4.
16 **Mathematical programming:** Precision; equations; statistics; simulation; finance; calculus; machine-code.
17 **Programming in business and education:** Examples, applications and pitfalls in business and education.
Appendices: 6502 reference charts; Supermon listings; ASCII; glossary

Plus many programs, diagrams and charts. Paperback, 504 pages. ISBN 0 942386 04 3. **$24.95.**

To Order
Programming The PET/CBM
Call
TOLL FREE 800-334-0868
In NC Call 919-275-9809
Or send coupon to
COMPUTE! Books, P.O. Box 5406, Greensboro, NC 27403

In England, order from Level Limited, P.O. Box 438, Hampstead, London, NW3 1BH. Price in England is £14.90, including P & P.

Please send ________ copy (copies) of **Programming The PET/CBM** at $24.95 each. (In the US and Canada, add $3.00 shipping and handling. Outside North America add $9.00 for air mail delivery, $3.00 for surface delivery).

All orders must be prepaid in US funds (money order, check, or charge).

☐ Payment Enclosed
Please charge my ☐ VISA ☐ MasterCard ☐ Am. Express
Account No. ________ Expires ____ / ____
Name ________
Address ________
City ________ State ________ Zip ________
Country ________

Allow 4-6 weeks for delivery. Foreign surface delivery allow 2-4 months.

www.commodore.ca

VIC Ringer is a utility that should be in any programmer's bag of tricks. Those who are used to wide screen computers will find it especially helpful in working with VIC's screen wraparound.

VIC Ringer

Thomas Henry, Mankato, MN

The Commodore VIC-20 computer has got to be one of the most pleasant computers to program. However, you might find the 22-column screen a little disconcerting, especially if you were brought up on other computers. Of course, the VIC-20 does have screen wraparound, meaning that your BASIC lines can be a full 88 characters wide, including the line numbers. A BASIC line can actually occupy four normal screen display lines. This is a great scheme, but if your mind is on programming you may forget to watch for the end of this four-line limit. This is especially true if you are adding a line and the screen is already full of other statements.

Let's add an end-of-the-line bell. This bell should ring when the cursor is within, say, eight spaces of the end of a BASIC line. This gives enough warning so that you can finish up the line or make the necessary changes to start a new line. The Commodore CBM-8032 has such a bell, and so does every typewriter; why shouldn't the VIC-20?

"VIC Ringer" is written in machine language and sits at the top of memory. The top of memory pointers at $37 and $38 are automatically adjusted so that they point just below the Ringer. Thus the program is "locked in" and safe from BASIC program interference. It can then peacefully coexist with any other programs you may care to enter in.

The principle of operation is quite simple. On the VIC-20 every 1/60 of a second, the computer stops doing whatever it may have been doing and goes to an *Interrupt Request* service routine. This routine scans the keyboard for key closures, refreshes the display, updates the realtime clock, and so forth. The VIC Ringer program is inserted into this Interrupt Request routine. Essentially, when the computer receives the interrupt signal, it will jump to the VIC Ringer routine and check to see if the cursor is at the 80th position on a BASIC line. If it is, and if this is the first time the 80th position has been found, then the bell rings. The computer then jumps to the normal Interrupt Request routine. If the cursor is not on the 80th position, the bell ringer routine is skipped.

ATTENTION VIC-20 USERS

Put YOUR BASIC programs on a cartridge!

The **ROMPACKER CARTRIDGE SYSTEM** lets you copy programs onto an EPROM USER Cartridge as easily as copying them onto a tape.

FEATURES

- BASIC or Machine Language Programs
- Up to 24K of Programs per Cartridge
- Automatic Program Menu on Power Up
- Single Keystroke Program Selection
- Uses Popular 2532 EPROMs

***Special* — ROMPACKER STARTER SYSTEM**

Includes: Firmware cartridge, User Cartridge (with two 2532 EPROMs), EPROM programmer and manual.
$149.95 till 11/30/82 (Regular $179.95)

Extra **ROMPACKER** User Cartridges — $39.95
2532 EPROMs $9.50 each / 4 for $36.00

Orders: Prepaid orders shipped free — others add $4.00
Mass. residents add 5% sales tax
Visa and MasterCard accepted

Business Computer Systems of New England
P.O. Box 2285
Springfield, MA 01101
Phone (413) 567-8584

COMPU SENSE

USE YOUR OWN CASSETTE PLAYER/RECORDER WITH YOUR VIC-20®

With the new CARDETTE from Cardco, Inc. you can interface any standard cassette player-recorder to save programs and load them with unerring ease into your VIC-20.

No longer are you restricted to using only the VIC Datasette. Just add a CARDETTE and you can use the tape unit of YOUR choice.

The CARDETTE comes with all necessary cables and wires to complete the hook-up. There is nothing else to buy. All you need to supply is YOUR tape player/recorder and your tapes.

Price: Just $29.95

To order:
812 S. LIGHTNER
WICHITA, KS. 67218
(316) 684-4660

PERSONAL CHECKS ACCEPTED
(ALLOW 3 WEEKS) OR C.O.D.
HANDLING CHARGES $1.50

VISA

www.commodore.ca