Constructing The Ideal Computer Game

COMPUTE!

$2.50
July
1983
Issue 38
Vol. 5, No. 7
£1.85 UK $3.25 Canada
63379
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

Special Games Issue

Some Of The Finest Games Ever For VIC-20, 64, Atari, And Other Computers: *Roadblock*, *Castle Quest*, *Goblins*, And More!

Circles: A Machine Language Tutorial For Atari

Backing Up Your VIC-20 And 64 Disks

REM Revealed: A Tutorial For PET, VIC-20, And 64

PLUS:
Build Your Own Apple Data Manager, Stars On The Radio Shack Color Computer, Gold Miner Game For The TI-99/4A

Major Feature: New Products At The COMDEX Dealer Show

0 74470 63379 07

**BUY A BANANA.
SAVE A BUNCH.
MORE TO COME.**

FOR YOUR COMMODORE WORDPROCESSING NEEDS INVEST IN THE BEST

WORDPRO PLUS. IN A CLASS BY ITSELF.

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

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

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

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

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

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

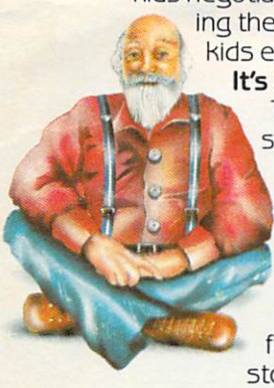
Call us today for the name of the WordPro Plus dealer nearest you.

Professional Software Inc.
51 Fremont Street
Needham, MA 02194
(617) 444-5224
Telex: 951579

This year, thousands of kids will be searching for the most amazing thing.

At Spinnaker, we don't believe in the "kill or be killed" concept behind most computer games. In fact, we believe computer games should be instructive. Not destructive. But just as importantly, they should be fun.

That's why IN SEARCH OF THE MOST AMAZING THING™ is designed to let your kids negotiate with aliens instead of destroying them. Because given the opportunity, kids enjoy using their minds.



It's Amazingly Fun.

The Most Amazing Thing is out there somewhere. Finding it won't be easy.

But relax, your kids will have the help of their old uncle Smoke Bailey. He'll give them a B-liner (sort of a cross between a hot air balloon and a dune buggy) to use on their journey. They'll have to learn how to fly the B-liner and navigate it through storms and fog. But before they do anything, your kids will have to talk to Old Smoke. He'll tell them about the Mire People and the strange language that they speak. He'll also tell them to avoid the dangerous Mire Crabs and how to get fuel for the B-liner.

Your kids will visit the Metallican Auction where they'll trade with the aliens for valuable chips. Your kids will then use these chips to buy things they'll need for their trip. And your kids will learn how to fly over the planet using their jet pack.

The Most Amazing Thing holds great powers, but it will take great skill, persistence and imagination to find it.

It's Amazingly Educational.

IN SEARCH OF THE MOST AMAZING THING is written by Tom Snyder, educator and author of the best-selling Snooper Troops™ Detective Series.

And like all Spinnaker games, IN SEARCH OF THE MOST AMAZING THING has real educational value. For instance, your kids will sharpen their ability to estimate distances and quantities. And since they'll be navigating their B-liner, they'll become aware of distance, direction and time. They'll also develop a knack for economic and monetary principles through trading with the aliens. And they'll solve problems through trial and error.

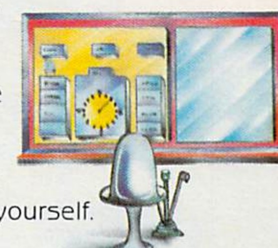
They'll learn all of these things, plus they'll learn that nothing is impossible if you put your mind to it.

A Novel Approach to Computer Games.

Besides offering your children all of the above, IN SEARCH OF THE MOST AMAZING THING gives them an opportunity to develop their reading skills. Because included with the game is Jim Morrow's new novel The Adventures of Smoke Bailey.* So your children will have hours of fun reading the book or playing the game. And they'll be learning at the same time.

Parental Discretion Advised.

If you're a parent who would rather see your kids reason with aliens than destroy them, you've got plenty of reasons to ask your local software retailer for IN SEARCH OF THE MOST AMAZING THING. It's compatible with Apple,® IBM,® Atari,® and Commodore 64™ computers. And it offers so much fun you'll probably be tempted to play it yourself. Or you can write us directly at: Spinnaker Software, 215 First Street, Cambridge, MA 02142.



You'll find this is one computer game that won't alienate you from your children.

SPINNAKER™
We make learning fun.

July 1983 Vol. 5, No. 7

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 132 before typing in programs.

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

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

EDITOR'S NOTES

The Magazine Epidemic

It's mildly distressing to observe the rash of new computer magazines in preparation or now being launched. While some appear to be the result of dedicated, sincere efforts at serving a market niche, and serving it well, many seem to be efforts to simply get something on the shelf.

It would seem that every publisher, large and small, in the country has suddenly discovered the personal computer marketplace. We welcome those of you providing genuine readership; we'll reserve comment on those of you who are slapping a computer label on inferior editorial matter as a medium for selling advertising. We're firm believers in the inherent decision-making strength of the free marketplace. Time will tell.

The New Computers

Will the surge of intelligent keyboards for game machines have a massive impact on personal computer sales? We think not. With Atari, Commodore, and TI battling it out in the price trenches, we expect to see the less than $100 market begin to expand in the features area. Principal change: more memory at less cost. And we'll just keep growing from there.

Random Bits

IBM's home computer (code name Peanut) is now rumored to appear by August. We expect this baby PC to come in as a mid-market machine with superb design, lots of support, and a slightly high price point in the $600 range. From a marketing/value added standpoint, the IBM name and reputation carries clout and has consumer impact. If and when it arrives, it will be an interesting competitor for the Commodore 64 and the soon-to-be-introduced Atari 600 and 800XL.

We hear that John Wiley, the book publishing house, is hard at work setting up a magazine staff to launch a personal computer magazine. Atari, Inc. has decided to accept advertising in their users magazine. As with Commodore publications, expect serious restraints on what type of advertising is allowed. Rumor has it that Atari won't be accepting game software advertising. We find *that* one hard to believe.

CBS is now looking for an entry into the computer magazine market. Rich Richmond, formerly Adventure International Marketing Manager, prepares to launch an Atari magazine (should we say "Another one..."?). All of this after unsuccessfully trying to raid **COMPUTE!**'s staff for several weeks.

Commodore, now in the publishing business, has become distant with **COMPUTE!** and *COMPUTE!'s Gazette* editors. We've always maintained that there's intrinsic value in independence.

Next month: The Consumer Electronics Show and a flock of exciting new products. We just returned from the National Computer Conference in Anaheim and, as far as the personal/home market goes, it simply makes us long for the arrival of CES. One point worth noting: several hundred exhibitors at this multimillion dollar show were housed in quasi-permanent, inflatable Quonset huts. Air conditioning failed and by late Monday, May 16, internal temperatures approached 115. So much for state-of-the-art technology at a state-of-the-art show.

Robert Lock

ATARI® DOES MORE THAN ANYONE ELSE TO GIVE YOU TWO HELPFUL KINDS OF COMPUTER SERVICE.

LOCAL. LONG DISTANCE.

If you need someone to fix your ATARI® Video Game or Home Computer, you'll find the best place is also the closest.

We have over 1,600 ATARI SERVICE℠ Centers coast to coast; just look in the Yellow Pages under Video Games or Computers.

And if you have any kind of question about your ATARI Home Computer—how to do something new with it, how to debug one of your own programs, what kind of peripherals are best—call the ATARI Help Line and talk to an ATARI expert. Our toll-free number is 1-800-538-8543.*

At ATARI SERVICE, we take care of you. As well as your ATARI system.

ATARI SERVICE
FACTORY AUTHORIZED NETWORK

WE ANSWER YOUR CALL FOR HELP.

*California: 1-800-672-1404

Coming In August

The Coming Year: Interviews With Industry Experts

CES: The Fall Computer Collection

Weather Forecasting On Several Computers

Neat Numbers For VIC

3-D Color Computer Art

Atari Verify

And Three Excellent Games

COMPUTE! Publications, Inc. publishes:

COMPUTE! The Journal For Progressive Computing

COMPUTE! Books

Telephone: 919-275-9809

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

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

Subscription Information

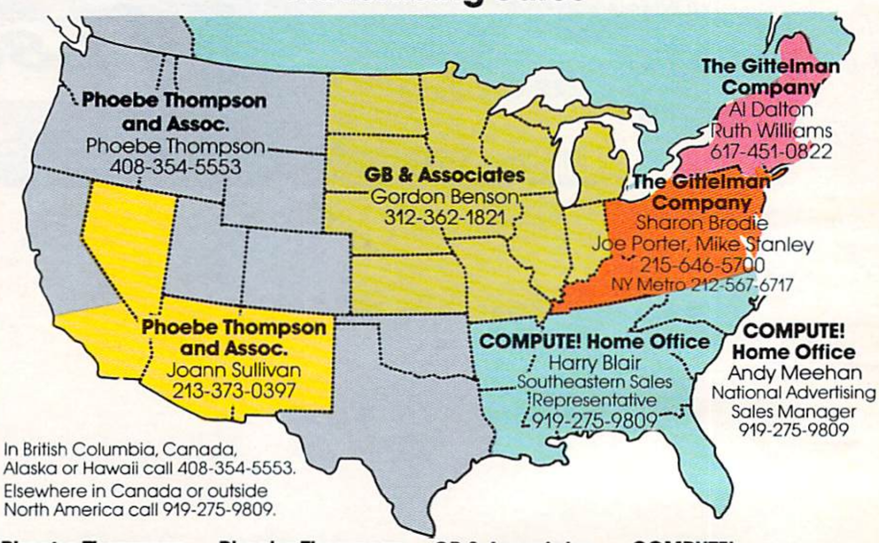
COMPUTE! Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403

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

COMPUTE! Subscription Rates (12 Issue Year):

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

Advertising Sales



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

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

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

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

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

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

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

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

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

64 Screen Problems

I own a Commodore 64 computer. While I am quite pleased with its performance, there is a problem. Any program that uses the BASIC commands GET or INPUT causes severe interference in the form of many white (silver) horizontal lines which shoot across the color TV screen.

I also purchased some software (namely *The Word Machine* and *The Name Game*) from Commodore and these programs exhibit that same, quite annoying, problem. I might add that three friends here in Albany who also own the C-64 have the exact same problem. I have heard that Commodore replaced a video chip in the later models (of which mine is one) and that there are problems with this new chip.

My question is: will Commodore solve what may be a very large (in number of computers) problem? My warranty is close to expiration.

Donald G. Weiser

This is probably the question we're asked most about the Commodore 64. The problem that you are referring to has come to be known as "sparkle."

The problem starts with the 64's character ROM, and the sparkle is caused by the way the 64 generates its characters to put onto the screen.

However, this problem can be more than a mere inconvenience in the early machines. It can cause difficulties with some programs, especially games. When utilizing the advanced 64 Sprite features (user defined, moveable objects), the sparkles can cause the computer to register a sprite collision when none has occurred.

There are some solutions. One is to make a few hardware modifications inside the 64, but this solution is frowned upon by Commodore, and may void your warranty. Another is screen relocation. It is said that if you relocate the screen memory into another area of RAM, the sparkle will disappear.

As for the number of units plagued by this problem, Commodore's estimate is five percent. It should be noted though, that almost all of the early models had sparkle, and as of this writing the problem is apparently still not solved. As a matter of fact, **COMPUTE!** *recently purchased two 64s for testing purposes, and one has a very severe "sparkle" problem.*

In answer to your question on repairs, Commodore has no set policy in this area. For units that are under warranty, Commodore says that it will attempt to repair anything with which the customer is dissatisfied.

Concerning units out of warranty, Commodore had no comment. However, a number of computer dealers and repair centers have stated that they will install the new or updated character ROMs if they can get them from Commodore. Commodore has said that they have not yet decided whether or not they will make the new character ROMs available to the service centers.

A Timex/Sinclair Tip

I have sometimes experienced problems on my ZX81 while changing line numbers. A line 30 that I am unable to delete, for instance, might appear after a line 2000. As a solution of sorts, I came up with the following short routine (also applicable to the T/S 1000) which allows me to locate the line in memory and POKE in a valid line number. This routine gives the location in memory of a program by line number. RUN it by typing GOTO 9500 . After INPUTing a particular line number, it will tell you the length of that line and how long the program is through the end of that line. As you can see in the sample run, the portion of the program considered here is 516 bytes long (incidentally, line 1 will not work in this test).

John B. Swetland

```
1 LET TEST=9500
10 LIST
9500 PRINT"ENTER LINE NUMBER"
9502 PRINT
9503 INPUT AQ
9504 LET N=16509
9505 LET N=N+2
9506 LET N=N+(PEEK N)+(PEEK(N+1)*256)
9507 LET N=N+2
9508 IF (PEEK N*256)+(PEEK(N+1))=AQ THEN
      PRINT;"LINE ";AQ;" STARTS AT ";N
9509 IF (PEEK N*256)+(PEEK(N+1))=AQ THE
      N GOTO 9511
9510 GOTO 9505
9511 PRINT
9512 LET I=(N+50)
9513 FORR J=N TO I
9514 IF PEEK J=118 THEN PRINT"LINE ";AQ;"
      ENDS AT ";J
9515 IF PEEK J=118 THEN GOTO 9517
9516 NEXT J
9517 PRINT
```

Look to the future with . . .

card? is the #1 selling printer interface for VIC20® and Commodore 64 computers because it works.

The "CARD/PRINT" works with all of these parallel printers without any modification to the printer: Epson MX-80 & EPSON MX-100 (with or without GRAFTRAX+), EPSON FX-80, EPSON FX-100, STAR MICRONICS DP 8480, STAR MICRONICS GEMINI 10, STAR MICRONICS GEMINI 15, PROWRITER, C-ITOH 8510, STARWRITER, SMITH CORONA TP-1, BYTEWRITER, MPI 88, MPI 99, DATA IMPACT D-92, OKIDATA 80, OKIDATA 82, OKIDATA 83, OKIDATA 84, OKIDATA 92, OKIDATA 93, TRANSTAR 130, TRANSTAR 140, IDS MICROPRISM, IDS PRISM (IDS printers require IDS cable P/N 603-673-9100), MENNESMANN TALLY MT-160, MANNESMANN TALLY 1000, NEC 8023, AXIOM GP-100, TANDY CGP-115 PRINTER/ PLOTTER and many more.

- INCLUDES ALL NECESSARY PLUGS CABLES AND HARDWARE
- REQUIRES NO SOFTWARE DRIVERS JUST PLUG IT IN AND PRINT
- ALL FUNCTIONS ARE UNDER USER OR PROGRAM SOFTWARE CONTROL
- ADDS SEVEN NEW COMMANDS TO INCREASE PRINTER FLEXIBILITY
- SOFTWARE SELECTABLE ASCII CONVERSION AND GRAPHICS MODES

NOTICE: CARDCO, Inc. is now offering the "PRINTER UTILITY PACKAGE" tape based program for use with our interface, a VIC 20® or a Commodore 64 and a Star Gemini series, and Epson MX or FX 80 or 100, or a PROWRITER printer. These programs include a machine language high-res screen dump, facilities to print Commodore graphics, and several sizes and styles of the "BANNER HEADLINE" type printing all for $19.95.

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

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

® VIC20 is a registered trademark of Commodore International

cardco, inc.

CAN A COMPUTER MAKE YOU CRY?

Right now, no one knows. This is partly because many would consider the very idea frivolous. But it's also because whoever successfully answers this question must first have answered several others.

Why do we cry? Why do we laugh, or love, or smile? What are the touchstones of our emotions?

Until now, the people who asked such questions tended not to be the same people who ran software companies. Instead, they were writers, filmmakers, painters, musicians. They were, in the traditional sense, artists.

We're about to change that tradition. The name of our company is Electronic Arts.

SOFTWARE WORTHY OF THE MINDS THAT USE IT. We are a new association of electronic artists united by a common goal — to fulfill the enormous potential of the personal computer.

In the short term, this means transcending its present use as a facilitator of unimaginative tasks and a medium for blasting aliens. In the long term, however, we can expect a great deal more.

These are wondrous machines we have created, and in them can be seen a bit of their makers. It is as if we had invested them with the image of our minds. And through them, we are learning more and more about ourselves.

We learn, for instance, that we are more entertained by the involvement of our imaginations than by passive viewing and listening. We learn that we are better taught by experience than by memorization. And we learn that the traditional distinctions — the ones that are made between art and entertainment and education — don't always apply.

TOWARD A LANGUAGE OF DREAMS. In short, we are finding that the computer can be more than just a processor of data.

It is a communications medium: an interactive tool that can bring people's thoughts and feelings closer together, perhaps closer than ever before. And while fifty years from now, its creation may seem no more important than the advent of motion pictures or television, there is a chance it will mean something more.

Something along the lines of a universal language of ideas and emotions. Something like a smile.

The first publications of Electronic Arts are now available. We suspect you'll be hearing a lot about them. Some of them are games like you've never seen before, that get more out of your computer than other games ever have. Others are harder to categorize — and we like that.

WATCH US. We're providing a special environment for talented, independent software artists. It's a supportive environment, in which big ideas are given room to grow. And some of America's most respected software artists are beginning to take notice.

We think our current work reflects this very special commitment. And though we are few in number today and apart from the mainstream of the mass software marketplace, we are confident that both time and vision are on our side.

Join us.
We see farther.

ELECTRONIC ARTS™

SOFTWARE ARTISTS? "I'm not so sure there *are* any software artists yet," says Bill Budge. "We've got to earn that title." Pictured here are a few people who have come as close to earning it as anyone we know.

That's Mr. Budge himself, creator of PINBALL CONSTRUCTION SET, at the upper right. To his left are Anne Westfall and Jon Freeman who, along with their colleagues at Free Fall Associates, created ARCHON and MURDER ON THE ZINDERNEUF.

Left of them is Dan Bunten of Ozark Softscape, the firm that wrote M.U.L.E. To Dan's left are Mike Abbot (top) and Matt Alexander (bottom), authors of HARD HAT MACK. In the center is John Field, creator of AXIS ASSASSIN and THE LAST GLADIATOR. David Maynard, lower right, is the man responsible for WORMS?

When you see what they've accomplished, we think you'll agree with us that they can call themselves whatever they want.

```
9518 PRINT"LINE ";AQ;" IS ";J-N;" BYTES L
      ONG"
9519 PRINT
9520 PRINT J-16509;" BYTE PROGRAM (PLUS D
      IMS)"
9600 STOP
9990 INPUT H$
9991 SAVE"TEST"
9992 GOTO 1
9509 IF (PEEK N*256)+(PEEK(N+1))=AQ THEN
      GOTO 9511
9513 FOR J=N TO I
```

Sample Run

```
ENTER LINE NUMBER
LINE 9600 STARTS AT 17020
LINE 9600 ENDS AT 17025
LINE 9600 IS 5 BYTES LONG
516 BYTE PROGRAM (PLUS DIMS)
```

Thank you for this handy tip. We can see where this program might also be useful in handling machine language routines.

VIC Memory Loss Cure

When using programmable characters, you lose some of your present memory. Is there any way to regain that memory without turning off the VIC?

Brian Gaetjens

Yes, and it can be done with a few easy POKEs. The most common way that memory is reserved for programmable characters is by POKEing locations 51 and 52 (the "pointer" for string storage in RAM), and locations 55 and 56 (the pointer for the limit, or "top," of memory). In the unexpanded VIC, the most common way to reserve character set space is to: POKE 51,0: POKE 55,0: POKE 52,28: POKE 56,28. This will reserve, or partition off, 512 bytes (enough for 64 programmable characters) at the top of BASIC RAM, leaving the programmer with 3069 bytes for BASIC programs. To reset the VIC to its original parameters, type: POKE 51,0: POKE 52,30: POKE 55,0: POKE 56,30. This will restore the VIC to its original configuration, and give you 3581 bytes for BASIC programming.

Monitor Sound

I currently have a 48K Atari 800 with a PERCOM disk drive. I would like to connect my computer to an RGB color monitor instead of a TV. But in doing so, I would lose all audio. Is there a way to have the sharpness of a color monitor and yet retain the sound capability necessary for the majority of Atari programs?

John C. Nardi

First of all, check the particular brand of color monitor you intend to buy. Some monitors do have a built-in audio capability. Other solutions would be to connect the audio output signal (pin 3 of the Atari's monitor plug) to your stereo system, or to an inexpensive, battery-powered amplifier available at most electronics supply houses.

An Atari/Commodore 64 Connection

Can an Atari 810 or other Atari disk drive be interfaced to a Commodore 64? I am thinking of buying a 64 as a second computer and would like to use my present Atari peripherals on the 64. Also, can the 64's SID sound chip be hooked up to an Atari?

David Lee

Both machines could communicate over a telephone modem hookup. Alternatively, you could hook them up directly using Commodore's RS-232 cartridge and Atari's 850 interface module (through its RS-232 port). You would likely be unsatisfied, though, at the slow rate by which data would be transferred between the two computers. Likewise, attempting to communicate to the SID chip from the Atari would be awkward. The whole would probably be less than the sum of the parts if you tried to gang these computers together and think of them as a team.

Atari and Commodore use very different peripheral buses (interface plugs). Although both have a serial bus, the 64 uses a variant of the popular RS-232C bus, while the Atari uses a complex serial standard.

VIC Disk Details

I own a Commodore VIC-20. I need a disk drive now, but I do not want to get a 1541 because I may upgrade to a PET in the future and do not want to buy a whole new drive. If I use a VIC to IEEE-488 interface to a 2031 drive, will I retain all the standard Commodore disk commands? Will I need DOS for the 2031 or the 1541? Please help.

Larry Abramowitz

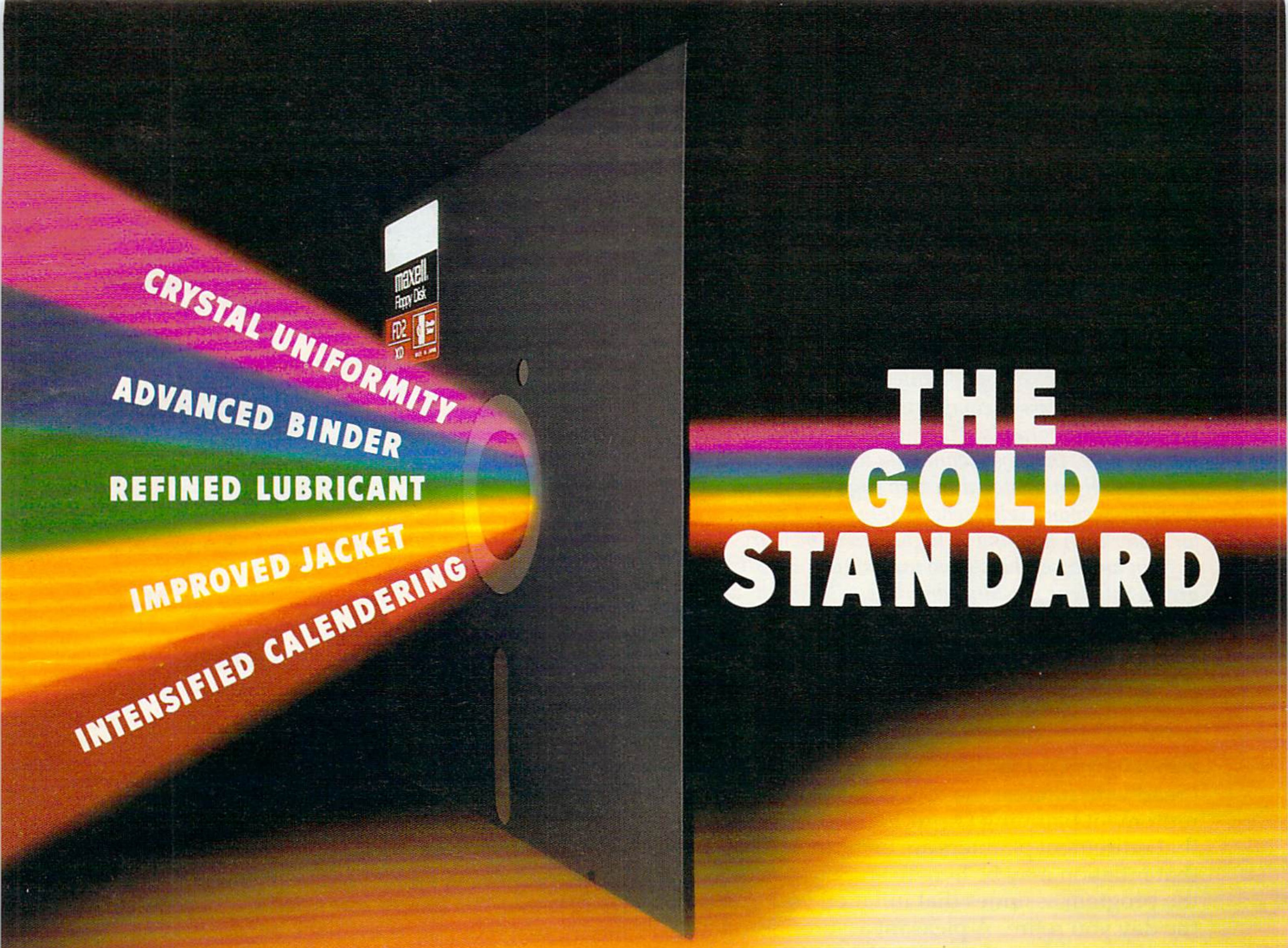
You will retain all of the standard commands. One of the main reasons for the manufacturing of an IEEE-488 interface is for upgrade adaptations like the one you're contemplating. There are several such interfaces on the market now. DOS is built into both the 2031 and the 1541.

Automatic BASIC To Machine Language Converter

Is there anything on the market that will convert standard BASIC programs into machine language? I need this for my Commodore 64 and its graphics.

Ben Savage

Your question is about speed: a program written in machine language can run a thousand times faster than the same thing programmed in BASIC. Some games,



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

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

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

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

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

maxell
IT'S WORTH IT.

Computer Products Division, Maxell Corporation of America, 60 Oxford Drive, Moonachie, N.J. 07074 201-440-8020 *large sorting tasks, and other kinds of computation require that the computer run at maximum velocity. That means machine language.*

There are large programs called compilers *which do something similar to what you want. They take a BASIC program apart and generate a high-speed version written in "P-code," a fast-running language similar to Forth. You can expect a "compiled" BASIC program to run anywhere from 10 to 40 times faster. One minor drawback is that the compiled program will usually be somewhat larger than the original BASIC version.*

There are also "optimizing" compilers which, during the process of compilation into P-code, also rearrange the program's structure to maximize efficiency. For example, the most commonly used variables in the program might be stored in zero page (the computer's first 256 memory cells) where storage and retrieval is far faster than it would be higher up in memory.

In any case, there is no way to turn BASIC programs into true machine language. You might want to use compilers for some programs, but also learn to program in machine language for those situations when speed is of the essence. Compiler programs for various computers are advertised in **COMPUTE!**.

Retirement Planning

I read with interest the article in **COMPUTE!** on retirement planning (April 1983). It is reassuring that retirement planners are finally acknowledging that inflation may be here to stay. Unfortunately, the program assumes that inflation will stop on the day you retire. A pleasant assumption, but one that could result in a lot of retirees who may not be able to afford subscriptions to **COMPUTE!** ten years down the road.

I have found the following program extremely useful for computing with my Atari 400, how much capital I would actually need in order to retire early. The program assumes:

1. That inflation will continue at a constant rate, and your yearly expenses will increase at this rate.

2. That you wish to spend your capital after retirement.

```
10 ?"ENTER CAPITAL AT RETIREMENT":
   INPUT A:?
20 ?"ENTER EXPECTED RATE OF INFLATION":
   INPUT B:?
30 ?"ENTER YEARLY ANTICIPATED RETURN
   ON INVESTMENTS AFTER TAXES AND
   INFLATION:INPUT C:?
40 ?"ENTER YEARLY EXPENSES LESS ANY
   INDEXED PENSION PLAN OR SOCIAL
   SECURITY BENEFITS":INPUT D:?
50 ?:?"YEAR";,"INCOME";,"CAPITAL"
55 Y=0
60 INC=INT(D*(1+B/100)^Y)
70 Y=Y+I
80 Z=B+C
90 A=INT(A*Z/100+A)-INC
100 ?Y,INC,A
110 IF A<0 THEN ?"CAPITAL EXHAUSTED":END
120 GO TO 60
```

Craig Cole

More Atari Automation

I'm writing in response to Joseph Wrobel's program, "Automate Your Atari" (January 1983). The following program neatly displays your disk directory (in two columns if necessary) each time you boot up your system. Just run "Automate" and enter each line below for each command. For example, command #1 would be 10 GR.0: DIM N$(17):T.60 and command #8 would be RUN without a line number. Since "Automate" counts characters, all spaces have been removed, end quotes are left off where possible, and abbreviations are used.

Rainer Forsch

```
10 GR.0:DIMN$(17):T.60
20 POS.2,3:PRINT"FILES CONTAINED ON THIS
  DISKETTE ARE:
30 O.#1,6,0"D:*,*":PRINT
40 I.#1;N$:PRINTN$:T=T+1:IFT=14THENGOS.70
50 G.40
60 POKE82,2:PRINT:POS.2,20:NEW
70 POS.2,4:POKE82,20:PRINT:RET.
RUN
```

PET Pause

While trying out one of Commodore's Model 8032 microcomputers, I stumbled upon a key function which would be handy for program debugging. I mentioned it to one of my instructors at Wake Forest, and he suggested that I share it with your readers.

Stopping program listing or execution can be useful for finding statement errors or viewing intermediate results of a calculation. Formerly, the only way to stop a program and the screen scroll was with the RUN/STOP key. This necessitates typing in the CONT command and pressing RETURN in order to resume execution. However, if the program is stopped by means of the colon key on the top row, scrolling may be resumed merely by tapping the back-arrow key, which also serves to slow the scroll if held down.

Interestingly enough, if a pure timing loop is running, the colon key will not halt execution. However, inclusion of a PRINT statement in the loop will enable the colon/halt function.

Jonathan Kerfoot

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

David D. Thornburg, Associate Editor

The Fifth Generation

I can hardly resist the temptation to point out that Orwell's vision for 1984 is (thankfully) not going to come true. It is interesting to note that, as with many other futurists, Orwell overestimated the amount of social change that would occur by 1984, and seriously underestimated the amount of technological innovation that will have been developed by then. While it is true that office workers in Orwell's novel dictate their letters into a "speak write," an automated stenographer/printer, much of the remaining technology is neither advanced nor inspiring.

I was reminded of the impact of technological advances as I created the first draft of this month's column on my Brother EP-20 battery-operated electronic typewriter. This marvel of design is quite compact, fits on an airplane tray table, and is almost silent. Since it retails for about $200 and allows the user to correct up to 16 characters of text before it is printed, I would not be surprised to see this device open up whole new markets for typewriters. I never used a typewriter for rough drafts before, simply because they were too bulky. Now, this device has become my portable workstation (sadly missing the storage that would make it a terminal for my word processor), and I take it everywhere.

Is it significant that this innovation was developed by a Japanese company? As we look at the computer industry, it is clear that it is taking on a decidedly international flavor. And yet, so far, the big names in personal computers are definitely American (TI, Commodore, Atari, Apple, IBM, etc.).

KIPS Super Computer

A recently published book, *The Fifth Generation* (Addison-Wesley, $15.95), suggests that we must be much more aware of Japanese advances in computer technology if we are to survive as a technological nation. Far from being a "scare" book designed to erect protectionist trade barriers, *The Fifth Generation* is more a call to arms. Its authors are Edward Feigenbaum, a pioneer in the field of artificial intelligence, and Pamela McCorduck, a science writer who has written extensively on computers and intelligent behavior in machines. The authors say that Japan has embarked on a ten-year crash program to develop a new type of super computer – a "fifth generation" machine that is called a Knowledge Information Processing System (KIPS). The KIPS is expected to be markedly different in architecture from the computers in use today. Furthermore, it is expected that users of the KIPS will interact with it very differently from the way people use computers today.

What is a KIPS? While most of today's computers are used for data processing and, with the exception of languages like LISP and Logo, most computer languages are geared towards data processing tasks, the KIPS is an optimized blend of hardware and software, tailored to perform general symbol manipulation and symbolic inference. This shift in emphasis recognizes that most of our work is nonmathematical in nature. Much of our work involves reasoning, not calculating.

A Reasoning Machine?

Can one build a "reasoning" machine? According to Feigenbaum and McCorduck, the Japanese lack our preoccupation with this question. From their perspective, it is sufficient to note that computer systems powerful enough to be fifth generation machines will function at a level far beyond that with which we are presently familiar.

Modest projects in the development of systems that outperform human "experts" are an important result of research in artificial intelligence. For example, programs that perform certain types of medical diagnoses, analyze and propose synthetic pathways in the creation of new chemical compounds, and predict the location of geological deposits have already been implemented on existing commercial computers using languages such as LISP. Such programs must operate with both a "knowledge base" and a set of "inference procedures." To read a map, for instance, one

must have both maps to read and a procedure for reading them.

Intuitive Solutions

The fifth generation KIPS will be built around the collection of vast amounts of data and the collection of problem-solving techniques that range from rigid deterministic methods to those that mimic the human ability to act on "hunches." You need not become embroiled in the machine intelligence controversy to appreciate that such systems have the potential to completely redefine computers, their use, and their place in society.

In order to create the KIPS, advances are required in both computer hardware and software. The computers we are familiar with operate in serial fashion. Instructions are executed one at a time. This type of computer architecture was developed by John von Neumann, and speed limitations in such computer systems are caused by the "von Neumann bottleneck" – processing instruction by instruction, byte by byte. In order to create faster computers, the fifth generation machines may favor a system using many processors in parallel.

A Billion Inferences Per Second

To appreciate the need for this approach, you should remember that the KIPS is to be used primarily for the linking of a knowledge base by symbolic representations (e.g., a *sparrow* is a *kind of bird*), or for the representation of rules (e.g., *if* the temperature is over 400 degrees, *then* the boiler must be turned down). To be used effectively, a problem-solving program must scan its library of "IFs" to find one relevant to the problem at hand. Finding this needle in the knowledge-based haystack of the size anticipated by the Japanese will require much more computational horsepower than we have seen to date. For example, today's big computers are capable of executing no more than 100,000 logical inferences per second (LIPS). (One logical inference corresponds to one IF/THEN statement.) A personal computer such as an Apple II might execute (depending on the language chosen) about 100 LIPS. The KIPS will be designed to execute up to a *billion* LIPS.

Such achievements are not the result of hardware alone. Interestingly, the language of present interest to the KIPS project leaders has already been developed by the Europeans – PROLOG.

How feasible is this project? There is much diversity of opinion on this topic, but there is consensus that, even if the project goals are not met in the allotted ten years, the interim results will most certainly change the nature of computers and computing. As Feigenbaum and McCorduck say:

Word literacy has given us power, access to an opulent, soaring world of mind – an alteration of thought processes – that is denied the illiterate. Computing literacy, even in its present form, opens still another world, one that all eventually may enter as routinely as they enter the world of letters, and it will confer perhaps even more power than the mighty pen and press have already given us. This is not idle promotion. As human muscle-power has been amplified by many special-purpose machines, so human mind-power will be amplified. The computer will change not only what we think, but how. ©

GO WITH THE WINNER

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

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

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

WINNING FEATURE #1
An Unbelievable Program

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

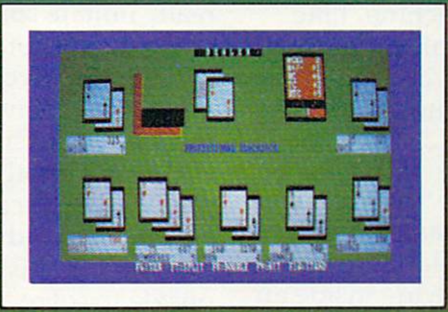
A Teaching System for Winners

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

Ken Uston's PROFE$$IONAL BLACKJACK™

WINNING FEATURE #2
An Unbelievable Free Offer

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



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

WINNING FEATURE #3
An Unbelievably Low Price

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

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

Be a winner with Intelligent Statements™ software.

intelligent statements™

GROWN-UP GAMEWARE

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

Richard Mansfield, Senior Editor

Writing A Simulation Game

There are three basic types of computer games: arcade, adventure, and simulation games. Let's briefly look at the characteristics of arcade and adventure games and then write a simulation.

Realtime Action

Arcade games feature what's called *realtime* action. Unlike chess or bridge, things happen fast. You can't sit back and plan your next move; you must react immediately to the space invaders. In other words, events take place at the same speed as they would in reality: realtime.

Arcade games also have a strong appeal to the eye and ear. There is much animation, color, and sound. In fact, your ability to respond quickly and effectively depends in part on all the clues you get from the graphics and sound effects. Strategy, while often an aspect of arcade play, is clearly secondary. These games are a new kind of athletics: the fun of man versus machine. Like auto racing, arcade games are essentially isometric exercises – you don't run around; you just stay in one place flexing and unflexing your muscles, tensing and relaxing.

Story And Strategy

Strategy, however, is more important in "adventure" games. The emphasis is on planning ahead and solving riddles. It can be like living inside an adventure novel. There is drama, characterization, and plot. You might start out, for example, in a forest with a shovel and a trusty, if enigmatic, companion parrot. As you try to figure out what to do next, the parrot keeps saying "piny dells, piny dells." After wandering aimlessly through the trees, it suddenly comes to you that the bird is saying "pine needles" and you dig through them and find a treasure map.

Your "character" will travel, meet friends and enemies, and have the opportunity to pick up or ignore potentially useful items such as food, magic wands, and medicine. It's customary that you cannot haul tons of provisions. You'd have to decide whether or not to leave the shovel in the forest. Yet you might be sorry that you'd dropped it if you're involved in a cave-in later in the game.

In any case, adventure games are fundamentally verbal. The computer displays the words:

```
YOU ARE IN A BOAT ON A LAKE. NIGHT IS
FALLING.
```

to which you can respond in any number of ways. You might type:

```
DIVE OFF BOAT.
```

and the computer would reply that you now see an underwater cave or whatever. You move through the scenes the way a character moves through a novel. There is generally no penalty if you take time to plan your next move. It's not *realtime*.

Imitations Of Life

The third category, simulation, is the least common kind of computer game. This is because to really imitate something, to *simulate* it effectively, you need lots of computer memory to hold lots of variables. However, memory has recently become far less expensive so we can expect to see increasingly effective simulation games. *Star Trek* and *Hammurabi*, both simulations, have long been popular home computer games. Although they are similar to adventure games, simulations are random. That is, there is no secret to discover, no puzzle to solve, no plot. Like real life, things happen with unpredictable, complex results.

Here's a program which simulates investing. The key to simulating is to arrange realistic *interactions* between variables. Look at line 600. If there is "international unrest," the price of gold (PGLD) goes up and the price of Bundtfund stock (PB) goes down. This relationship between gold, stock, and an international crisis is true to life. Alternatively, stock goes up and gold goes down in line 700 during a "market rally."

The game allows you to make investment decisions, and then a "month" passes during which the value of your investments will go up or down. In line 510, three variables are given ran-


FRANK CASEY
DATE
DOLLARS
3181
POWER

dom values. Stock can gain or lose up to 10 points (variable X), and gold can change by $20 an ounce (Y). Variable Z will be used to simulate flipping a coin. Also notice lines 520 and 525. In 520, we determine whether or not there will be unrest. The variable CH is just a counter. Each "month," CH is raised by one. Two conditions are required for unrest to happen: in a given month, CH must be greater than 4 and it must be less than whatever X turns out to be. If both these conditions are met, CH is reset to zero and we've got international unrest. This has the effect of creating unrest roughly every four to six months. Likewise, another rhythm is set up in line 525 to cause market rallies. In both cases, however, you cannot be certain exactly when to invest in gold or in stocks.

The decision to raise or lower stock prices is made in line 530 and based on the coin toss variable, Z. Again, stocks move in opposition to gold. Prices will rise about 50 percent of the time, but you can never know what will happen in a given month.

Suggested Complications

This is the core, a rough sketch, of an investment simulation game. There is much you can do to make it a more effective simulation and thereby a more enjoyable game. The more variables in a simulation, the better. For example, add leverage and additional "incidents" which affect prices, improve the randomizing, and include other types of investments. You could even use a separate counter which, every five years, causes the X and Y variables to swing more widely to reflect recession/recovery cycles.

As you can see, a simulation should be lifelike. It has interdependent cycles and a degree of unpredictability. Its realism derives from including a sufficient number of variables. And those variables must interact in plausible ways and with just the right amount of randomness. A simulation is a little world you create. You can define cause and effect and then fine-tune the whole thing until it seems well-balanced. Adventure and arcade games are certainly enjoyable, but this investment simulation can be built up to the point where it's just as much fun as any other kind of game.

Mixing Styles

Of course, these three categories – arcade, adventure, and simulation – are somewhat arbitrary. Some of the best games contain elements of each. There are adventure games with graphics – you see the forest, the shovel, the pine needles. After you say DIVE, your character jumps into a lake and the screen transforms into an underwater scene. Likewise, arcade games can include the different "settings" so characteristic of adventure games. Popular arcade games such as *Tron* and *Donkey Kong* change the playfield as you earn more points.

There are several ways to add to the appeal of our investment simulation, beyond just making it a more complex, more accurate simulation. You could add the visuals and sound of arcade games. Try creating a tickertape across the top of the screen to show price changes and news events. Maybe add a bell sound to indicate the end of further transactions. If your computer has a voice synthesizer, news events could be announced over the "radio." And from adventure games you could borrow two elements: riddles and the necessity of planning ahead. One easy way to incorporate these two elements would be to make paying taxes a part of the game. After all, the closer it is to real life, the better the simulation.

Special Program Notes: If you have an ***Atari****, you'll need to add semicolons (;) between the variable names and the PRINT statements to make everything print on a single line. If you have a* ***TI****, put each statement on its own separate line. In other words, you cannot use colons (:). Line 10 would be CASH=100000 and you'd need to add a line: 11 PGLD=400. If you have a* ***Timex/Sinclair****, use LET whenever a variable is defined. For example, line 10 would start: 10 LET CASH=100000. If you have a* ***TRS-80 Color Computer****, add the following line: 5 RANDOMIZE.*

Investment Simulation

```
10 CASH=100000:PGLD=400
20 PB=80
31 PRINT: PRINT" BUNDTFUND IS $"PB" PER
   SHARE.YOU HAVE "B" SHARES. -- $"PB*B
33 PRINT" GOLD IS{3 SPACES}$"PGLD" PER O
   UNCE.{2 SPACES}YOU HAVE "GLD" OUNCES.
    -- $"GLD*PGLD
34 T=PB*B+GLD*PGLD
35 PRINT"{31 SPACES}TOTAL INVESTMENTS --
    $"T
36 PRINT"{31 SPACES}YOU HAVE $"CASH" TO
   SPEND."
40 PRINT"{24 SPACES}GRAND TOTAL (INVESTM
   ENTS + CASH) $"T+CASH
45 IFCK=1THEN500
50 PRINT: PRINT"1.BUY{2 SPACES}2.SELL
   {2 SPACES}3.DONE"
60 INPUTA:IFA=3THENCK=1:GOTO31
100 PRINT"WHICH?{3 SPACES}1.GOLD
    {2 SPACES}OR{2 SPACES}2.STOCK"
110 INPUTF
120 PRINT"HOW MANY (SHARES OR OUNCES)?"
130 INPUTN
140 IFF=1THEN160
150 PRICE=PB*N:IFA=1THENCASH=CASH-PRICE:
    B=B+N:GOTO400
155 CASH=CASH+PRICE:B=B-N:GOTO400
160 PRICE=PGLD*N:IFA=1THENCASH=CASH-PRIC
    E:GLD=GLD+N:GOTO400
```

```
170 CASH=CASH+PRICE:GLD=GLD-N
400 GOTO50
500 CK=0:PRINT:PRINT" ONE MONTH LATER ..
    .":FORT=1TO700:NEXTT:PRINT
510 X=INT((RND(1)*100)/10):Y=INT((RND(1)
    *200)/10):Z=RND(1)
520 CH=CH+1:IFCH>4ANDCH<XTHENCH=0:GOTO60
    0
525 IFCH=2GOTO700
530 IF Z>.5THENPB=PB+X:PGLD=PGLD-Y:GOTO3
    1
540 PB=PB-X:PGLD=PGLD+Y:GOTO31
600 PRINT"INTERNATIONAL UNREST...":PGLD=
    PGLD+2*Y:PB=PB-2*X:GOTO31
700 PRINT"MARKET RALLY ...{2 SPACES}":PG
    LD=PGLD-2*Y:PB=PB+3*X:GOTO31
```

©

Questions Beginners Ask

Tom R. Halfhill, Features Editor

Are you thinking about buying a computer for the first time, but don't know anything about computers? Or maybe you just purchased a computer and are still a bit baffled. Each month, **COMPUTE!** *will tackle the questions most often asked by beginners.*

Q Are there any problems I could cause while using a computer that could permanently damage it? How about any commands used in the wrong way? In other words, what are the chances that I could do real damage to the computer by not knowing how to use it right?

A There's an old saying in computing that goes something like this:

"The only way you can hurt a computer through its keyboard is to hit it with a hammer."

Of course, this isn't completely true; spilling liquids into a computer keyboard isn't too healthy for it, either. But the general thrust of that adage *is* pretty certain – aside from physical abuse, a computer can't be damaged by anything you can type on its keyboard.

There's only one rare exception we've ever heard of. A certain POKE command on one Commodore PET computer (PET/CBM's with 4.0 BASIC) can drastically speed up the process by which the computer creates the screen display. If this command is left running wild, the computer keeps speeding up until it eventually self-destructs. The chances of this POKE happening by accident are extremely remote. There are 65536 memory locations in a PET that can be POKEd, and there are 256 possible numbers that can be POKEd in each location (0 to 255). Therefore, the chances of accidentally typing in that fatal POKE command are only one in 16,777,216.

Other than this rare example, you really don't have to worry about damaging the hardware of your computer system by experimenting with commands or programs. The same pretty much holds true for the devices attached to the computer. At worst, you might cause an error which traps a device in an endless loop – for example, the disk drive might keep spinning, or the printer might keep spewing forth paper. Conceivably, if the system were left unattended, the device could eventually overheat or suffer excessive wear. But if you're there, you can always stop such "runaway" events by switching off the power. Anytime you switch off a computer or device and then switch it back on again, it resets itself.

Remember, though, we're talking about *hardware* damage. There are lots of ways you can cause permanent *software* damage. Simply typing NEW on the keyboard and pressing RETURN will wipe out any BASIC program in memory. If the program has not been saved on disk or tape, it will be lost. Likewise, certain commands can erase a program from a disk or tape, or overwrite it with something else. A wrong command, a program bug, or a typing error when entering a program listing can cause a *system crash* – your computer "locks up" (refuses to accept commands). Since the only way to recover, usually, is to switch the computer off and on again, the program in memory will be lost. But you can rest assured that the computer itself is always safe from permanent damage.

Q Can I do word processing with a tape recorder, or must I have a disk drive?

A It is quite possible to do word processing with a tape recorder.

Make sure, however, that the word processing program you buy or use is designed to work with tape. Some programs are for disk only; still others work with both.

The peripheral device which is most essential for word processing is a printer. Without a printer, you won't be able to generate a paper printout of your writing. And since the whole object of word processing is writing, a printer is indispensable. If you want to do word processing and must choose between buying a disk drive first or a printer, opt for the printer.

For casual word processing (average letter-writing, etc.) you may find that a tape recorder is a sufficient storage device. However, for more serious applications, you'll probably discover that a disk drive is necessary. Tape recorders can be reliable, but they are very slow compared to disk drives. Also, a disk drive adds flexibility to word processing. Depending on the word processing program, a disk drive can make it possible to easily store frequently used paragraphs on disk for merging with other files; to link several files together for very long documents; to merge files of names and addresses with form letters; and other advanced functions. ©

PET/CBM/COMMODORE 64

PaperClip

Professional Word Processor at a Breakthrough Price

PaperClip™ performs all the advanced features found in Word Processors costing much more. . .

1) Full screen editing. 2) Copy/Transfer sentences and paragraphs. 3) Insert/Delete sentences and paragraphs. 4) Headers/Footers/Automatic page numbering. 5) Justification/Centering. 6) User defineable keyphrases. 7) Supports both cassette and disk. 8) Variable data – Form letters. 9) Horizontal scrolling up to 126 characters. 10) Insert/transfer/erase columns of numbers. 11) Add/subtract columns of numbers. 12) Supports most dot matrix and letter quality printers. In fact, a printer set-up routine is supplied to take the best advantage of the printer at hand. 13) French and Math technical character sets available.

Also available for Commodore 64

Requires Basic 4.0, 32K memory.

$125.00 U.S.

Call or write for the store nearest you.

BATTERIES INCLUDED

71 McCaul Street
Toronto, Ontario
Canada M5T 2X1
(416) 596-1405

Part I

Constructing The Ideal Computer Game

Orson Scott Card, COMPUTE! Books Editor

In this first article of a two-part series, the author examines currently available types of home computer games and suggests a new approach: a game where the player creates his or her own world. As an example, he describes the elements of a "game-building game" called "Railroader." It's something of a cross between traditional entertainment and computer programming.

Next month, the article concludes with advanced applications and specific techniques for programming Railroader on your computer.

When I first bought an Atari 400, I told my wife all kinds of stories. About how computers were the wave of the future. About how our kids had to become computer literate. About how useful the computer would be.

I didn't fool her. I didn't even fool myself. I knew I was getting the machine because of the games.

And I've done my time. My *Super Breakout* game regularly tells me "Wow!" My *Centipede* scores are respectable, and my wife and I make a great team playing *Ghost Hunter*.

But now, after a couple of years with the ultimate game machine, I've discovered a dreadful secret: true *home* computer games are rare.

Look at your games for a minute. What are they actually doing? Most of them are doing what pinball machines are designed to do – enticing you to try to beat the machine, with the odds hopelessly stacked against you. That makes sense for arcade games. They are supposed to make money, and the only way to make money is to force you to play against the clock, pumping in as many quarters per hour as possible. When the local wizards started playing 30 minutes per quarter on the Dig-Dug machine in the corner Seven-Eleven, they flipped a switch inside it and suddenly the old patterns stopped working. I stopped getting 250,000 points a game – and the company started getting a lot more quarters. That's business.

But why do home games have to play that way? The arcade games *are* fun on the home machine, at first. But they can get frustrating or boring. After a while I begin not to care anymore whether I get above 70,000 on *Centipede*. I'm never going to "win," and I don't lose a quarter when I *don't* win.

There's something worse than boredom. Something a little pernicious. Teenagers who come to my house to play my games have a great time. But when my four-year-old son and I sit down to a few games of *Salmon Run* or *Picnic Paranoia*, he almost always ends up in tears. Not because I always win – I'm a nicer father than *that* – but because the *machine* always wins. He doesn't stand a chance. He can never finish. He can never accomplish anything.

Why should all those wonderful graphics, all those fantastic imaginary worlds, be devoted to either frustrating my son or programming him until he learns how to do his part perfectly?

Because that's what all but three computer games I've tried end up doing – programming the *player*. Rewarding and punishing *me* until I learn to display the correct behaviors. What are the arcade wizards, except human beings who have learned to obey the demands of a computer program?

Don't get me wrong. I still love a new arcade game. I'm in there flapping away at *Joust*, making hamburgers and McMuffins with *Burgertime*, and mastering the art of swinging on chains and ropes in *Donkey Kong Junior*. I'm as eager as anyone to find out what the next screen will look like, to find out what the programmer has created in his or her little world. But it's still the programmer's

Brøderbund
AMERICA'S FAVORITE COMPUTER GAMES

Now for the Commodore 64

CHOPLIFTER!™ It's the best selling game ever for the Apple. In its Atari 400/800 version, it zoomed to the top of the charts. Now CHOPLIFTER!™ is available for the Commodore 64.

Your challenge is to rescue peace-conference delegates held hostage behind enemy lines. Amid heavy fire from tanks, missiles and planes, you heroically maneuver your daredevil chopper. You'll need all your courage and skill to survive against treacherous odds. The game's remarkable graphics and realistic joystick control won't let you give up. After all, world peace rests in your hands!

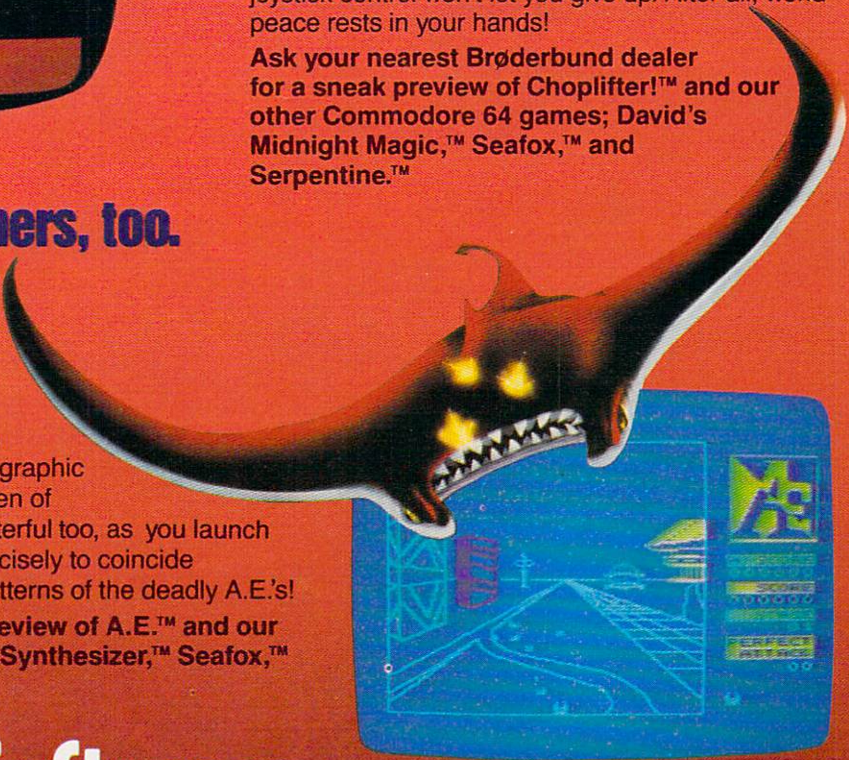
Ask your nearest Brøderbund dealer for a sneak preview of Choplifter!™ and our other Commodore 64 games; David's Midnight Magic,™ Seafox,™ and Serpentine.™

And for VIC-20 owners, too.

A.E.™ A runaway best-seller for the Apple II and the Atari 400/800, is a winner for VIC-20.

Giant robot stingrays designed to fight pollution have run amok and are attacking in waves from the sky. A.E.™ ("stingray" in Japanese) is a graphic masterpiece, with screen after screen of spectacular 3-D effects. The arcade-style action is masterful too, as you launch your remotely-triggered missiles and detonate them precisely to coincide with the swooping, diving, constantly changing flight patterns of the deadly A.E.'s!

Ask your nearest Brøderbund dealer for a sneak preview of A.E.™ and our other VIC-20 games: Martian Raider,™ Multi-Sound Synthesizer,™ Seafox,™ Shark Trap,™ and Sky Blazer.™



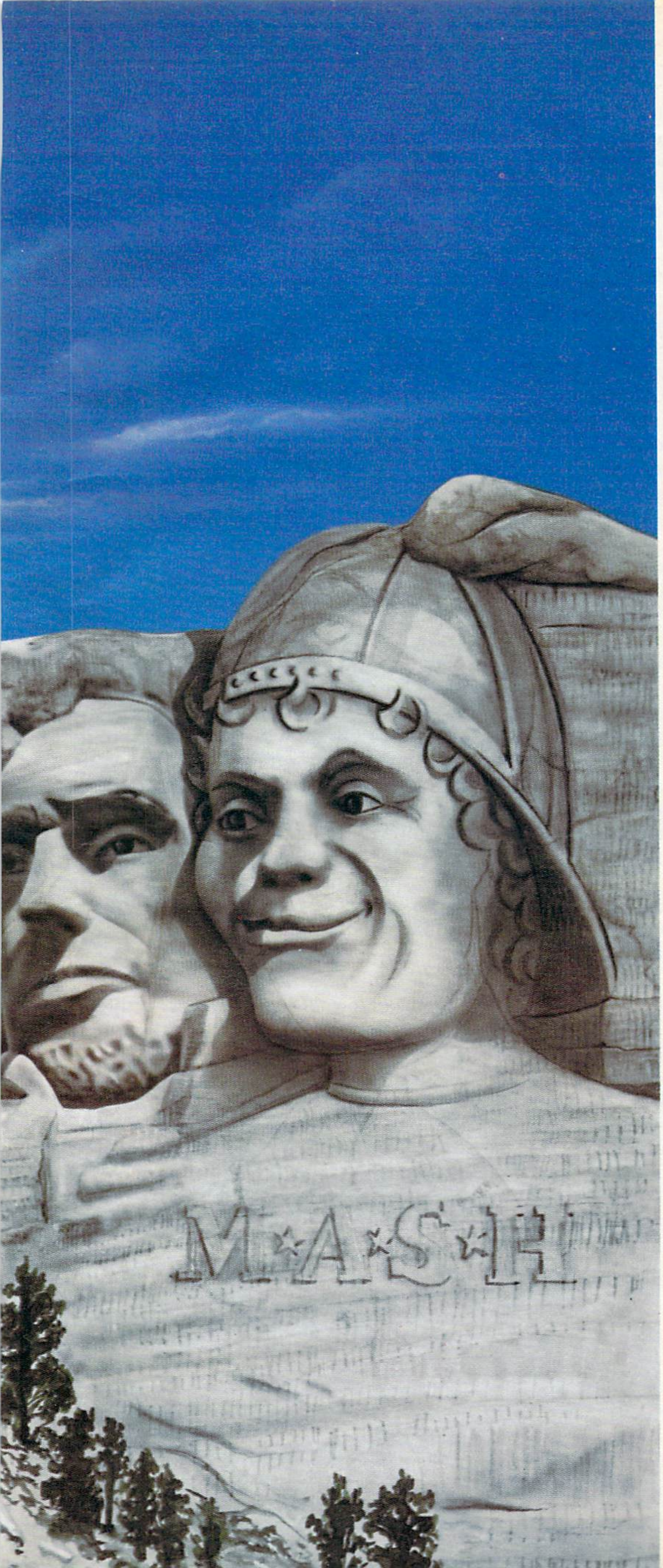
Brøderbund Software

Brøderbund Software 1938 Fourth Street San Rafael, CA 94901

Atari, Apple, VIC-20, Commodore 64 and IBM PC, are registered trademarks of Atari, Inc., Apple Computer Inc., Commodore Electronics, Ltd., and International Business Machines Corp. respectively.

(cartridge versions shown)

Design the best new M*A*S*H™ video game, and you'll get something money can't buy.



Fame.
That's right. F-A-M-E.
Because if the game you design is good enough, we'll make it and put your name on every one we sell.

And remember, you don't have to be a computer programmer to come up with a great game. You just have to have a great imagination, and give us a short description of your idea.

IF FAME ISN'T ENOUGH, HOW ABOUT A FORTUNE?

$25,000 to be exact. That's how much the game contract will bring you. Guaranteed. What's more, we're giving away a whole list of prizes for the other great game ideas we get.

- Four First Prize AMC Jeeps®
- 400 Second Prize Texas Instruments 99/4A™ Home Computers.
- 4000 free Fox Video Games.

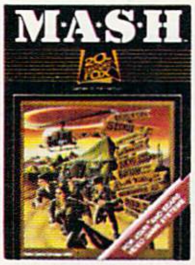
EVERYBODY WINS.

Because the new M•A•S•H game comes packaged with a smashing M•A•S•H T-shirt.
Absolutely free.

HOW DO YOU GET STARTED?

Just buy the original M•A•S•H game. The contest rules and entry blanks are in every package.

And if you win, you might say the rewards are monumental.



20th Century Fox

Games of the Century

THE M*A*S*H™ CONTEST. IT'S A SMASH.

world, not mine.

Even the adventure games, both text and graphics, usually boil down to puzzle-solving, out-guessing the programmer.

But in the home, where the family has unlimited access to the computer, there can and should be a different type of game. A different kind of play.

What Is Really Fun?

I've noticed a couple of important things in my family. First, about myself. I almost never stay up late playing computer games. But I have been known to stay up until three or four or six o'clock in the morning working on a program. You might say that, in a way, programming is much like arcading: after all, BASIC is forcing me to react in certain patterns, and I'm only just now beginning to learn when to PEEK and when to POKE. I have been trained, right?

There *is* a difference – all the difference in the world. When I program, I can save the result on something a lot more permanent than a vanity board. And I'm not just charting through someone else's program – I'm creating something that never existed before, at least not in the exact form I'm giving it. When I'm through, there's a lasting result. And I can take all the time in the world. I can take the time to do it *right*.

A second thing I've noticed is the way my children play when they *aren't* using the computer. They do like a shoot- em-up game as much as other kids. But games like that are only a minority of the things they do.

They also like solving puzzles, and spend much more time doing mazes or putting together picture puzzles than they ever spend on fast-action games.

Most of all, though, their playing time is spent making things or pretending things. They spend hours with wooden or plastic building blocks, making castles or spaceships or houses or anything they can imagine. They draw and color, write stories or act out plays, dress up in costumes or read aloud from books – whether they understand the actual words on the page or not.

In fact, they do exactly what I like to do with the computer: create their own small world that works just the way they want it to work. They don't want anyone to tell them that they can't make a castle that way, or to insist that six legs are too many for a horse. "You made your *twos* backward," we tell our son, and he looks at us impatiently and says, "Let me do it my way."

How many hundreds of dollars have we sunk into our home computer? We own it, don't we? Why, then, do we have so many programs that tell *us* what to do? Why can't my children – or my wife and I, for that matter – play games that let us tell the *computer* what to do, that let us create something that will last, that let us use the magic of the computer to make things we could never make before?

The Few Games That Work At Home

I've found three games that approach the sort of play that only the home computer can allow – games that are neither elaborate puzzles nor quarter-stealing pinballs.

Galahad and the Holy Grail. At first glance, this Atari (APX) adventure game looks pretty much like other realtime graphics adventures. Only after you've played it awhile do you begin to realize that this is the first game to give you the freedom to play your own game. True, there are fast-moving knights and spiders and a persistent, maddening moth to kill you when your reflexes are too slow, and there are puzzles to solve. But there are no win conditions. The program never congratulates you and says, "That's it, you've solved it all." It's fun simply to explore the dozens of different rooms and find out what secrets they hold. It's no coincidence that my son loves to play it, and has never found it frustrating, though it is always challenging.

Eastern Front. This APX game isn't for children, and there are definite win conditions, but it is a war game that gives you freedom to plan your own moves, to develop your own strategy, and there are hundreds and hundreds of possible ways to play, none of them "wrong." Your decisions are shaped by events, but the events do not control you any more than you control them. (*ATARI Program Exchange, 155 Moffett Park Drive, B-1, P.O. Box 427, Sunnyvale, CA 94086.*)

Facemaker. It runs slowly, but I find that my son never gets impatient with the game from Spinnaker. The choice of facial features is very limited, but the important thing is the way the program and the child interact. It allows a child whose drawings are still very primitive to make faces that actually resemble real faces, and program them to perform a series of actions. When my son plays with *Facemaker*, he is creating something, and doing things with it that could not possibly be done without the computer. (*Spinnaker, 215 1st St., Cambridge, MA 02142.*)

The Five Types

There are probably other games that make use of the special advantages of the home computer, but the point is that they are distressingly rare. Most of the games coming out today are variations on the same old themes:

- *Target Shoot.* The targets move, they dance, they are cute or they are menacing, but the game always consists of shooting them down.
- *Tag.* The same old targets, but you have to


The Official
ZAXXON™
by SEGA®

catch them instead of shooting at them.

- *Coming at You.* Tag, only they're trying to catch *you* or shoot *you*, and you spend a lot of time running and dodging.
- *Scramble.* You have to get from here to there, and there are things in the way.
- *Maze.* Like scramble, only there are several routes you can follow, and you have to figure out the best one.

Have I missed anything? Even sports simulations, like the sports that inspired them, are combinations of these elements. Football is tag plus scramble – or coming at you, depending on whether you're playing offense or defense. Hockey is target shoot plus scramble. Baseball is scramble, target shoot, and tag. Lots of fun, but all these wonderful new games are just combinations of the same old things.

New, creative game elements are getting rarer. An arcade game like *Joust*, which really does introduce a whole new way to move a player on a screen, still turns into tag-plus-scramble once you master wing-flapping. *Donkey Kong Junior* has that wonderful swinging motion and the difference between two-handed and one-handed climbing, but it's still a maze with things coming at you.

What else *is* there?

Games That Let You Create

What I want to see are games that let the player create things. BASIC and LISP and PASCAL and PILOT all fit the bill – but they also require mastering some pretty sophisticated concepts. They're fun, but they aren't exactly *play*. What I would like to see is something as simple as building with wooden blocks, while exploiting all the strengths of the home computer.

And what are those strengths?

1. *Time.* Running out of quarters doesn't mean you have to quit. Nobody's rushing you to finish. You can think, instead of letting the computer train your reflexes.

2. *Permanence.* You can save the result of what you've done, change it, re-use it, limited only by the number of cassettes or diskettes you have on hand.

3. *World creation.* You're manipulating numbers, it's true; but the result can be visible and audible, and it can move. You can create worlds the way fiction writers create them, and bring them to life as, until now, only movie-makers could.

4. *Individuality.* It's your computer. Why shouldn't the results of your play, and your children's play, be uniquely your own? Why should the only difference between you and any other player be your score?

Let's Design A Game

It's easy to talk about this kind of game. It's only a little harder to design it. So I'll give you a detailed game design that you can program. But after what I've said about individuality and creativity, there's no way I could provide you with a complete program listing. I'll just offer detailed documentation for the game, then a few hints on how to program it, and let you design the way the program works yourself. It can easily be executed in BASIC, though at some points you may be happier with machine language subroutines.

(The documentation that follows is long and detailed, but when you're designing a computer game, it's usually a good idea to figure out exactly what the player's experience of playing the game will be like. This is especially true if you aren't as conversant with your programming language as you are with English. By writing out the instructions and rules first, as I have done here, you can save yourself debugging and revising time later.)

Railroader

You are building a network of railroads. When it's all built, you control the switches and make your train run on the tracks wherever you want.

The game, though simple enough for a preschooler to master, is really an introduction to programming. Model railroaders were designing *loops* and *branches* long before electronic computers were a twinkle in Sperry-Rand's eye. If the player does not close all the loops and resolve all the branches, the program will provide a few reminders. If the player still refuses to tie up loose ends, the program will do it.

And, for those who have the most fun playing cooperatively with someone else, the program allows two players to design railroads on the same screen, and run their trains at once (with sometimes disastrous effects).

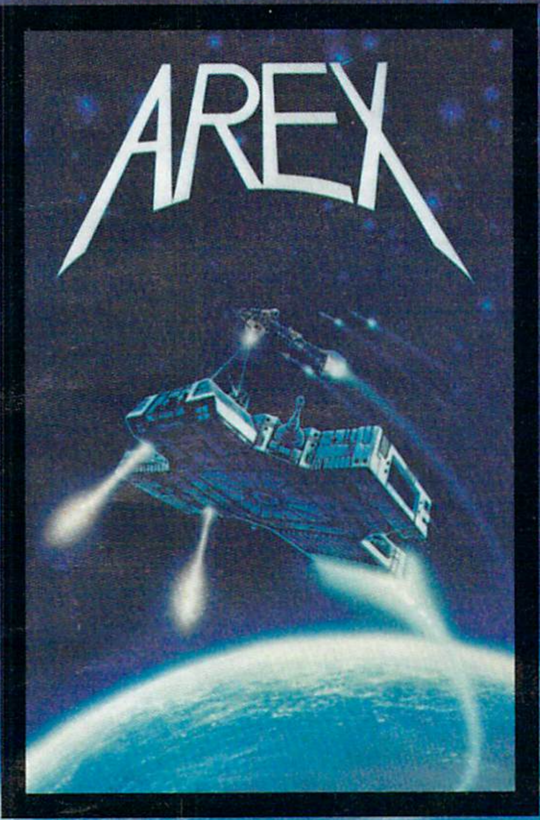
The Track-Laying Stage

"Railroader" begins by announcing its name and finding out the answers to a few questions. Do you want to lay track or run a train on an already-created track layout? Will there be one or two players? Do you want to lay track at the beginner or expert level? Do you want to save the track layout you create, and if you plan to save it in a disk file, what should the file be named?

When you have made your selections and pressed START, the screen displays a list of instructions:

> "Use joystick and joystick button to lay track units."
>
> "Type 1 to go on to the next track unit."
>
> "Type 2 to choose which railroad spur to complete."

BECOME AN INTREPID SPACE ADVENTURER . . .



by William Muk
CoCo version by Roger Schrag
Atari version by John Anderson

Far beyond the known galaxies, you venture deep into the vast reaches of outer space. But you are not alone! In a flash, without so much as a how-do-ya-do, they're in hot pursuit and you're left to do before you're done unto. **Can** you elude your pursuers? **Will** you elude your pursuers? And who ***are*** these guys anyway? Find the answers to these and other compelling questions in AREX. See your dealer today!

AREX . . . Coin-op arcade realism at home for 1 to 2 players.

AREX

CoCo 16K TAPE	060-0172	$34.95
TRS-80 Model 1 & 3 16K TAPE	010-0172	$34.95
TRS-80 Model 1 & 3 32K DISK	012-0172	$34.95

. . . OR FLY HIGH IN THE WORLD OF HIGH FINANCE

by George Schwenk
TRS-80 version by Dave Simmons
CoCo version by Roger Schrag

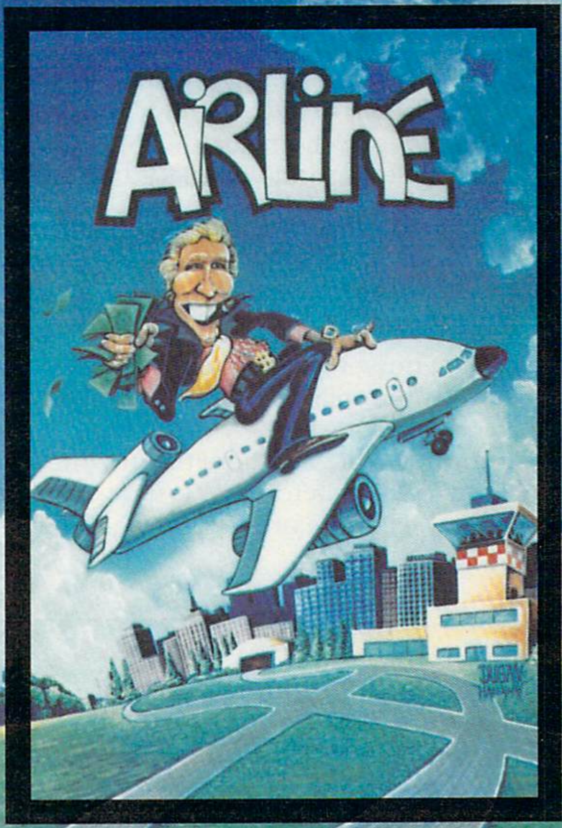
"Yas, after purchasing diamond mines in South Africa, oil wells in Saudi, and rare beer cans in Walla Walla, Washington, I had begun to wonder what other trendy commodities remained to be added to my swelling portfolio. Then a snip of a ticket girl dared to tell me (ME, Hartley J. Wormsflather III!) that my flight was overbooked. To avoid future misunderstandings, I bought the airline."

"I think I'm on to something profitable here." Hartley J. Wormsflather III

AIRLINE . . . A no-holds-barred strategy game for 1 to 4 players.

AIRLINE

ATARI 400 & 800 / CoCo / Model 1 & 3 16K TAPE . . 140-0169 $24.95



Published by

Adventure INTERNATIONAL

a subsidiary of Scott Adams, Inc.
BOX 3435 • LONGWOOD, FL 32750 • (305) 862-6917
Prices Subject To Change Without Notice

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

143

"Type 3 when your layout is complete. At this point, if you haven't brought every spur back to the main line, Railroader will do it for you, and if you want to save the layout, Railroader will save it."

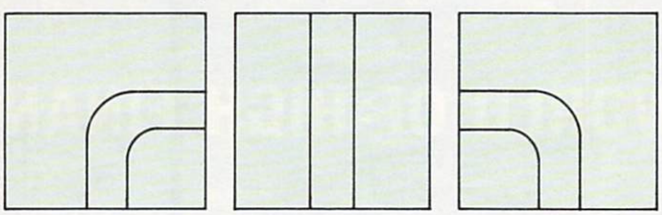
"Press any key or joystick button to begin."

When you give the signal, a light green screen appears. If there is one player, a single orange square appears about one-quarter of the way in from the left on the bottom of the screen. If there are two players, a second square appears a quarter of the way in from the right. These squares work like cursors – they mark the area where you are laying track.

Laying Simple Track Units (The Beginner Level)

To lay track, use your joystick. Push forward to make a straight vertical track unit appear in the square. Push left for a track that curves to the left, right for a track that curves to the right. If you change your mind, push a different direction, and the track unit changes. However, the first track unit always starts at the bottom center of the square.

These simple track units look like this:

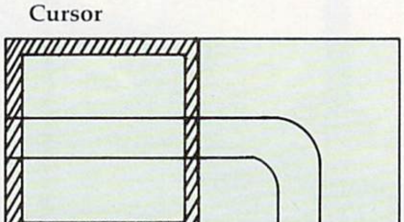


When you are satisfied with your choice, you reach over to the computer and type 1. Your square now moves to the blank area just beyond the end of the track unit you placed on the screen. If you put on a straight track, your square will appear just above it; if you curved left, your square will appear to the left.

If you are playing alone, you may immediately lay the next unit of track; if there is another player, you must wait your turn to lay track again; when the other player types 1, it will be your turn.

The next time you lay track, your new track unit will begin where the old one left off. If you curved left before, your new track unit will start in the middle of the right-hand edge of your cursor, like this:

Cursor



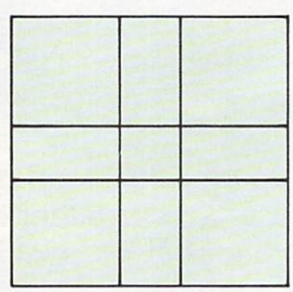
Again, to lay simple track units you have three choices. Let's say that you curved left on your first track unit. Now if you push the joystick left, a straight horizontal track unit will appear. If you pull the joystick toward you, the track will make another curve, this time downward. If you push the joystick away from you, the track unit will curve upward.

With every simple track unit you lay, the track will always begin where the last square left off, and will end up heading in one of the three valid directions you can push the joystick.

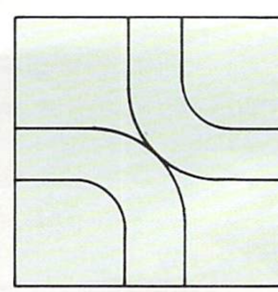
If you cause the track to end at the edge of the screen, your cursor will appear at the opposite edge. This means that track that ends on the left side of the screen is continued on the right side; track that ends at the top is continued at the bottom.

Erasing. If you want to go back and change the last track unit you completed, push the joystick in the direction of that track unit. Any track unit you laid in the new position will be erased, and your cursor will move to the former square, where you can either lay a track segment or go still farther back, erasing each track segment as you leave it behind. You may erase as many track units as you like, or stop at any point and lay a new track segment. But remember, if you are playing with another player, your turn ends when you type 1. You can erase as many units as you like, but you can lay only *one* track unit.

When Tracks Touch. At the beginner level, if you cause the track to touch an existing track segment, either your own or the other player's, Railroader will automatically create the following valid patterns:



Crossover



Curved by-pass

At the beginner level, and whenever you are touching the other player's track units, you may not cause the two tracks to join. If you are about to cross a curved track, you can choose to curve *only* in the opposite direction. If you are about to cross a straight track, you can lay only a straight track across it, not a curve that would join it. And if a new track unit would cause your track to run into the end of another player's spur, you will be allowed to lay only curved tracks that turn away from the other player's track:

COMM*DATA SOFTWARE

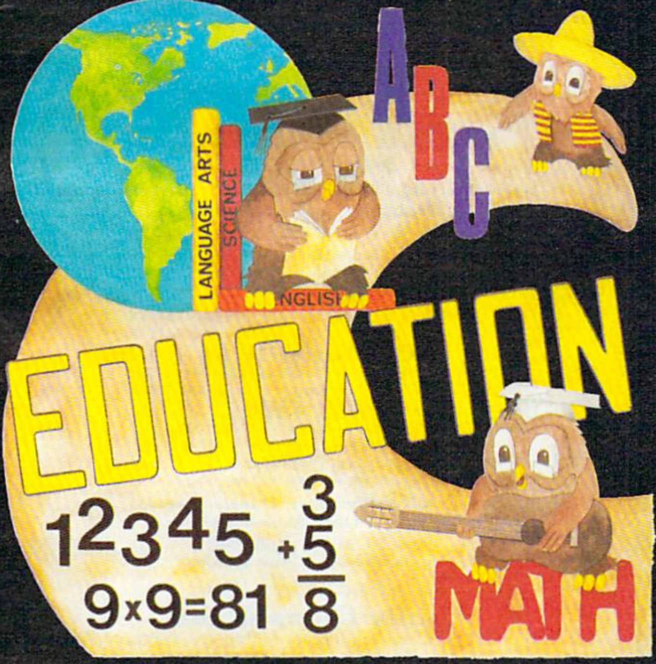
FOR THE COMMODORE 64®, PET®, AND VIC 20®

TAKE AN EXCITING TRIP DOWN AVENUES OF ADVENTURE WITH:

- Pakacuda*
- Escape*
- Logger*
- Ape Craze*
- Centropods*
- Supercuda*
- Street Maze
- Caves of Annod
- Capture the Beast
- Market

THROUGH TRAILS OF CREATIVITY WITH:

- Sketch and Paint
- Music Mentor



ALONG THE PATH TO KNOWLEDGE WITH:

- Wordspot
- Math Tutor Series
- Alphabet Tutor
- Geography Smash
- Gotcha Math
- English Invaders
- Math Invaders Series

ASK FOR COMM*DATA COMPUTER HOUSE SOFTWARE AT YOUR LOCAL DEALER.

ARRANGE PASSAGE TODAY!

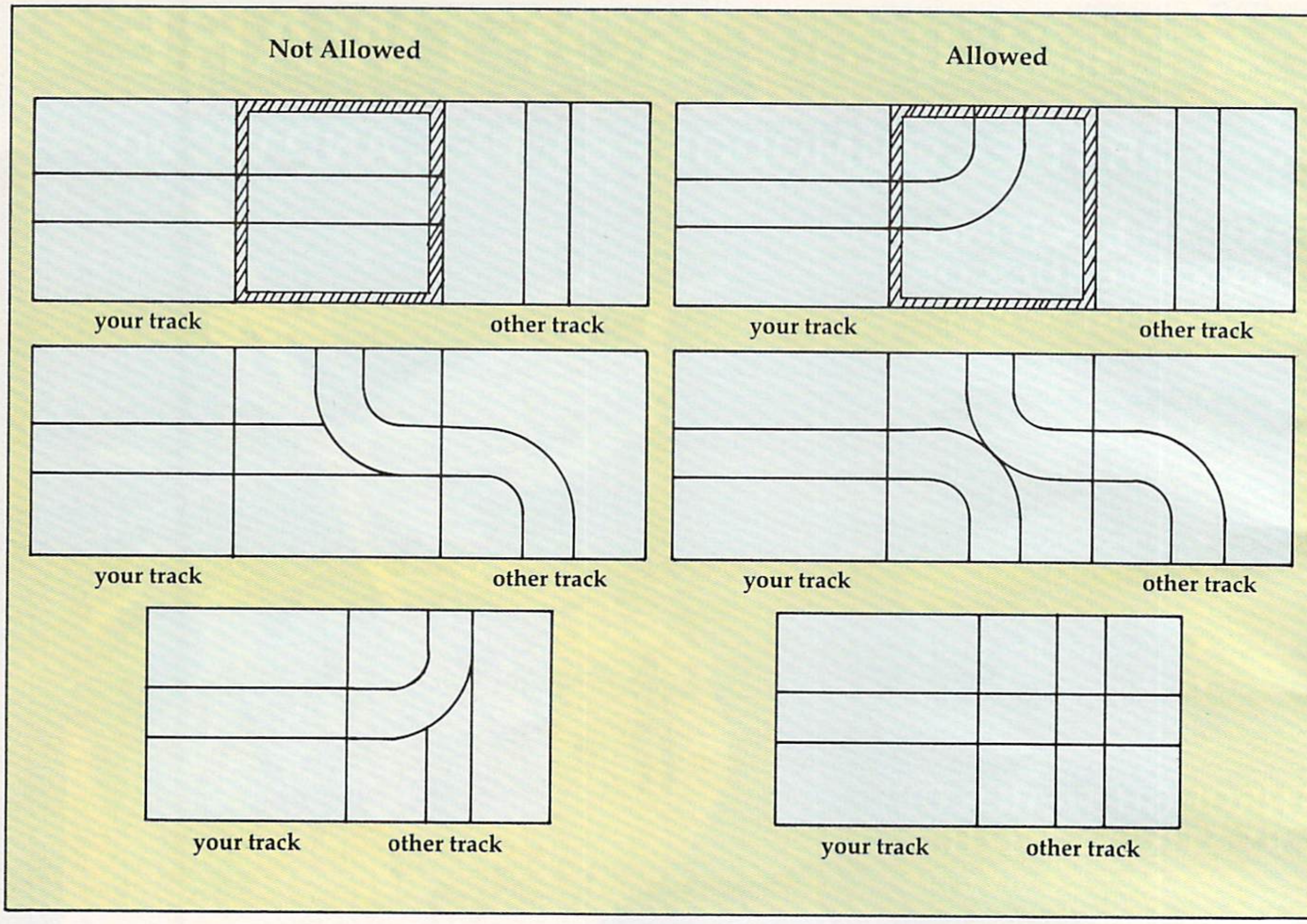
Or Send for FREE Complete Catalog:
COMM*DATA COMPUTER HOUSE
320 Summit Avenue
Milford, Michigan 48042
(313) 685-0113

Dealer Inquiries Welcome.

MasterCard

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

VISA



Erasing Crossovers and By-passes. If you are erasing and come to a unit where your track by-passes or crosses over another track, either your own or the other player's, Railroader will leave the other track intact, and remove only the track from the line you are erasing.

Ending the Track-laying Session. With the beginner-level game, that is the whole track-laying sequence. You just keep laying track until you match up the end of your track with the beginning at the bottom of the screen. When you are ready to quit, type 3. If you haven't linked your track with the beginning track unit, Railroader will automatically lay track from the last unit you created until it links with the first unit, so that the track always makes a closed loop.

Next month we'll go on to the Expert Level Game. ©


STAR SENTRY
FOR THE ATARI 400/800/1200XL

NEW PRODUCTS AT THE COMDEX/SPRING COMPUTER SHOW

Tom R. Halfhill, Features Editor

New products displayed at the Comdex/Spring conference, held in Atlanta during late April, show a trend toward still more home computers, lower-priced home peripherals, and increasing support for the popular home computers already on the market.

This year's Comdex/Spring show was more interesting than most for home computerists. Known officially as the "National Spring Conference Exposition for Independent Sales Organizations," Comdex is primarily a show for computer dealers, manufacturers, and businessmen. Consequently, almost all the wares on display at this large show are for the more expensive personal and business systems.

At the show this year, however, there seemed to be more than the usual number of exhibitors displaying products for lower-priced home computers. Two new home computers were shown – both imports; several low-cost printers and other peripherals made impressive appearances; and software started catching up with hardware (at least a little) as new programs were introduced for all the popular home computers. Most of these products should be on the market by the time this article appears. Here's a rundown:

New Computers

It's hard to imagine how the low-end home computer market can absorb many more machines, especially with such leading contenders as Commodore, Texas Instruments, Atari, and Tandy engaged in runaway price wars. But the home market is expanding so fast that no one wants to be left out, least of all the Japanese and the British.

That's why you can expect to see more imports invading the U.S. market. The British success with the Timex/Sinclair isn't easily ignored.

The newest British entry is the Oric-1, manufactured by Oric Products International Ltd., of Berkshire, England. Reputedly the second best-selling micro in Britain and Europe (next to the Sinclair), the Oric-1 appears to be a good computer in search of a good U.S. distributor. An Oric representative said the company experimented with mail order sales, but quit in favor of setting up a more conventional distribution network. Oric hopes to have one in place by midsummer.

The standard Oric-1 includes: 16K of Random Access Memory (RAM); a 57-key keyboard, with moving keys arranged typewriter-style; full repeat on all keys; standard ASCII character set with upper/lowercase; 96 redefinable characters; 16 colors; 40-column by 28-row screen display in text mode; and a 240- by 200-pixel high-resolution graphics mode. For sound there is a three-channel sound synthesizer with a seven-octave range and programmable envelopes, similar to the Commodore 64, an internal speaker, and connections for external speakers.

A cassette interface works at 300 baud or a very fast 2400 baud, and interfaces include a built-in Centronics-standard parallel printer interface; an expansion port for RAM and Read Only Memory (ROM) cartridges; and a Red-Green-Blue (RGB) interface for high-resolution color video monitors. The built-in BASIC programming language includes such interesting commands as INK and PAPER (for color control), DOUBLE, FLASH, and INVERSE (for character control), DRAW, CIRCLE, and PLOT (for graphics), and even SOUND, MUSIC, PLAY, PING, SHOOT, EXPLODE, and ZAP (for sound control).

The Central Processing Unit (CPU) is the 6502A microprocessor, basically the same chip found in Apple, Atari, and Commodore computers. While this doesn't mean the Oric-1 is


"You'll find my KoalaPad™ Touch Tablet, and KoalaWare™ software a natural profit builder."
Available for Apple,® Atari,® Commodore,® and IBM® personal computers.

Look what for your VIC 20.

Fast action. Complex strategies. Interesting characters. Superior sound effects. Multiple levels of play.

These are the things you want from your VIC 20.™

They're also the things you get from Tronix. From the people who brought you *Swarm!*, *Sidewinder* and *Galactic Blitz.*

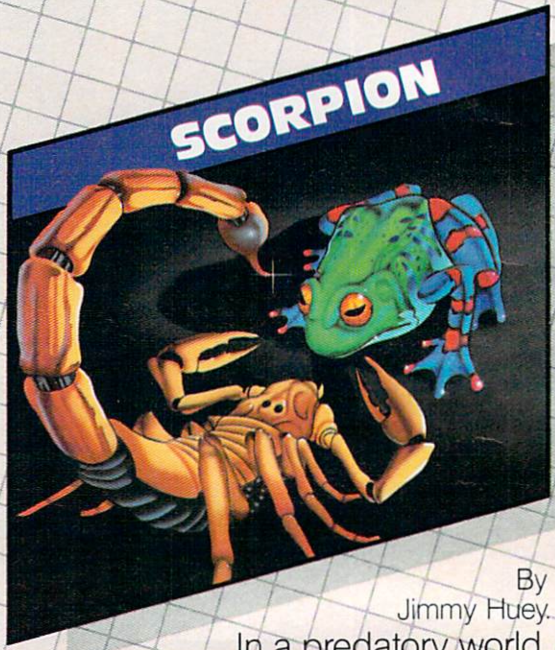
And now, there's more.

Now Tronix brings you the same rewarding rapid-fire excitement in three brand-new game cartridges.

Each one is something different. Something new. But they all have one thing in common.

They're all designed to bring out the best in your VIC 20.

You shouldn't settle for anything less.



By Jimmy Huey.

In a predatory world of killer worms, dragons, stalkers, pods and fly traps, the scorpion prowls the maze in search of sustenance. Frogs and their eggs mean survival to the scorpion. But they can also mean instant death! (Suggested retail $39.95)

compatible with these computers, it does mean that machine language programmers could adjust to it fairly easily.

The standard Oric-1 will sell for about $120 in U.S. funds. For about $240, there's a 64K RAM version with 16K of overlaid ROM, similar in arrangement to the Commodore 64.

Oric also makes a full line of peripherals for the Oric-1. At Comdex, Oric was showing prototypes of a microfloppy disk drive using the Hitachi 3-inch disks. The microfloppy is expected to sell for about $240.

If Oric succeeds in setting up a good U.S. distribution network, the Oric-1 could prove competitive in this country, especially if its overseas software base is also brought to America.

The Japanese Sord

Of course, the Japanese aren't standing idly by, either. Their newest export to the U.S. is the Sord M5, a $199 computer with impressive graphics and three different plug-in BASICs. The M5 is made by Sord Computer Systems, the fastest-growing microcomputer company in Japan. Founded in 1970 with $2500 by 26-year-old Takayoshi Shiina, Sord now commands about 15 percent of the Japanese business microcomputer market. Sord is exporting a line of high-end personal and business computers to the U.S., and the M5 is its first home computer.

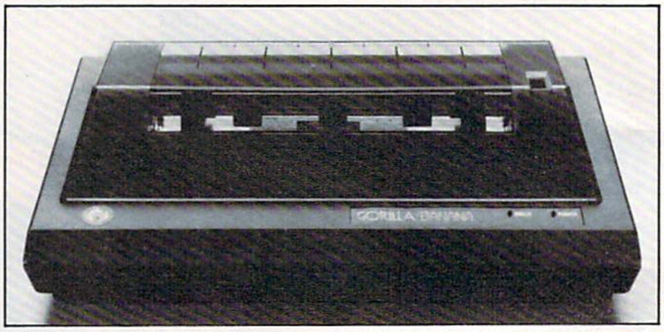
The M5 will be sold in two different configurations: the M5 Fun Computer and the M5 Multi-Computer. The basic specifications are the same: 20K of RAM expandable to 32K (although 16K is used for the screen); 8K of ROM with a machine language monitor; 16 colors; a 55-key keyboard with moving rubber keys; upper/lowercase and graphics characters; a flip-up top that conceals a cartridge slot for games, programming languages, and other plug-in "firmware"; built-in Centronics-standard parallel printer interface; cassette interface for standard tape recorders; sound generator; Z80A CPU; and a Texas Instruments video chip which allows up to 32 sprites (screen objects which can be created and animated by your own programs).

The two packages do vary, however, in terms of included accessories. The M5 can accept any of three BASIC language cartridges – BASIC-I (Introductory), BASIC-G (Graphics), and BASIC-F (Floating Point). BASIC-I is for beginners and children, BASIC-G is for general home use and graphics programming, and BASIC-F is a full-fledged floating-point BASIC for business, science, and math applications. The M5 Fun Computer comes with BASIC-I and a game cartridge. The M5 Multi-Computer comes with BASIC-G, an interesting dialect with special commands for the graphics and sprites. The Multi-Computer also has a carrying case and the FALC cartridge, a home data base program adapted from Sord's business software.

The M5 will be distributed through local dealers by Sord Computer of America, New York.

The Gorilla Banana

When personal computers cost $1000 or more, it seemed reasonable that printers sold for around $500 or $600. But now that full-featured home computers are widely available for under $100, the same printers can seem disproportionately expensive. That's why manufacturers are rushing to produce printers (and other peripherals) that are priced for the hundreds of thousands of people who are buying inexpensive mass-market computers.



The Gorilla Banana is the first in a new line of low-cost peripherals from Leading Edge.

Several new low-cost printers were seen at Comdex. Probably the one which attracted the most attention was the Gorilla Banana, the first in an upcoming line of low-cost peripherals from Leading Edge Products, Inc., of Canton, Massachusetts (best-known for Elephant Memory disks). Due this summer at $249.95, the Banana is an 80-column, tractor-feed, unidirectional, dot-matrix printer capable of 50 characters per second. It has four character sets (U.S., British, Swedish, and German), a double-width print mode, and upper/lowercase (although without true descenders). There's also a dot-addressable graphics mode with a density of 63 x 60 dots per inch.

The Banana attaches directly to any computer with a Centronics-standard parallel printer interface. Computers without a parallel port will need an interface at extra cost. An interface for Commodore 64 and VIC-20 computers will be available for $29.95, and an optional cartridge for the same price will allow the Banana to print the special Commodore graphics characters.

Another interesting 80-column dot-matrix printer is the STX-80 from Star Micronics, Inc., of Dallas, Texas. Suggested retail is $199. Although the STX-80 is a thermal printer – it uses a special print head and heat-sensitive paper to form its type instead of an inked ribbon – you wouldn't

HES
Gridrunner
Cartridge for VIC 20
HES WRITER
Shamus
Cartridge for VIC 20
Protector
Cartridge for VIC 20
HES MON
Cartridge for VIC 20
HES MON is a 6502 machine language monitor with a mini-assembler. It is an indispensable accessory for all assembly language programmers.
HES MON is a very powerful tool with many features not found in other monitors (see back).
RAM expansion is optional.
Complete instruction manual is included.
Cartridge for Commodore 64
Turtle Graphics II

guess it from the printouts. The thermal paper looks and feels much like standard typing paper. Unlike most thermal paper, which is silver, this paper is white with crisp black lettering. The STX-80 is a unidirectional printer that works at 60 characters per second, has upper/lowercase with true descenders, a double-width text mode, block graphics characters, European characters, a dot-addressable graphics mode, and a Centronics parallel interface.

Star Micronics also offers a 40-column, inked ribbon, dot-matrix printer for $250. The DP-8240 prints at 50 characters per second, has friction or tractor feed, upper/lowercase without true descenders, graphics characters, scientific and European characters, and a dot-addressable graphics mode.

The lowest-priced printer exhibited was the $129.99 Impact Printer from Fidelity Electronics, Ltd., of Miami, Florida. The Impact Printer works with the VIC-20 and Commodore 64 with no additional interface. Printing at 30 characters per second, it has a 24-column line and uses standard adding machine roll paper. Other features include upper/lowercase, graphics characters, inverse characters, and dot-addressable graphics.

Custom Joysticks

Since the "feel" of a joystick is highly subjective, many independent companies are introducing "custom" joysticks for those who dislike the standard models (for an overview of custom game controllers, see "The Joy Of Joysticks," **COMPUTE!**, February 1983). A few more new joysticks surfaced at Comdex.

Suncom, Inc., of Northbrook, Illinois, makers of the Slik Stik and Starfighter joysticks for Atari-compatible computers, came out with a Starfighter model for the Apple. The Starfighter is about the same size and shape as a standard Atari joystick, but with comfortably rounded edges. Overall, it's a luxurious controller with right- and left-handed fire buttons; an alternate fire button for games that require two buttons; a centering adjustment to fine-tune the stick's neutral position to each Apple; a switch to select either a long or short throw of the stick; and a high-low sensitivity switch to further tune the stick's response. Also, Suncom guarantees the Starfighter for two years. Suggested retail is $49.95 for the Apple IIe version (a $5.95 adapter is needed for the Apple II/II+).

Suncom also introduced two new controllers for Atari-compatible machines (Atari 400/800/1200XL, Commodore 64 and VIC-20, Atari VCS 2600, Sears Telegame). The most unique is the Joy-Sensor, a stickless joystick. The Joy-Sensor is a hand-holdable box with a flat disc where the stick should be. Instead of flexing a stick, you rock the disc. It lists for $34.95.

Suncom's other new joystick is the TAC-2 (Totally Accurate Controller). This looks like an adaptation of the Starfighter, with the addition of a longer, ball-tipped stick, and both right- and left-handed fire buttons. The TAC-2 is guaranteed for two years and lists for $19.95.

For users of Texas Instruments computers, Suncom introduced a $12.95 adapter so that Atari-style joysticks will work on the TI-99/4A, and a $13.95 dual cassette recorder adapter.

Since the "feel" of a joystick is highly subjective, many independent companies are introducing "custom" joysticks for those who dislike the standard models.

Two new joysticks were also introduced by the Kraft Systems Company of Vista, California. The Kraft Joystick is a lightweight Atari-compatible controller with an unusually short, flexible stick designed for fingertip action. It includes an extra-long eight-foot cord, a one-year warranty, and retails for $16.95. Another joystick, the Switch-Hitter, has two fire buttons for use by right- or left-handed players. Otherwise identical to the Kraft Joystick, it retails for $19.95.

Accessories And Peripherals

Numerous other add-ons were introduced at Comdex/Spring, too. Here are some which deserve special note:

- A low-cost modem for the Apple. The $119 Networker modem, by Zoom Telephonics, of Boston, Massachusetts, plugs into a single expansion slot and requires no other connections or external power source. It's a 300-baud direct-connect modem that hooks up to any modular phone jack. It has an originate/answer switch, a carrier detection LED, and is compatible with any standard telecommunications software. For $169, the Networker comes with *Netmaster*, a terminal program with upload/download, and a 40K text buffer (on a 64K system).
- Plug-in boards for Commodore and Texas Instruments computers. Microtek, Inc., of San Diego, California, introduced a $299 64K memory board for the TI-99/4A which fits into the expansion box. A 32K board also is planned. For the VIC-20, Microtek introduced VIGOR (VIC's Grand

Old RAM-cage). This is a $39.95, three-slot expansion board. For both the VIC and Commodore 64, there's the CC-2064, a $70 interface cable which allows the computers to drive parallel printers.

• New disk drive for Atari. The Rana 1000 Atari-compatible disk drive, by Rana Systems, of Carson, California, also was shown at the West Coast Computer Faire in San Francisco a few weeks before Comdex/Spring. Due on the market this summer, the Rana 1000 is switchable single/double density and will retail for $449 ($49 extra for the double-density Disk Operating System).

The Rana 1000 disk drive for Atari offers single and double density for $449.

It has some unique features not found on other drives: a write-protect button, a unit ID button (which tells you the drive's position in the daisy chain if you have several), an error button (which returns an error code), and a button which lets you know which track the head is reading or writing. What's more, the drive runs very quietly and is only about a third the size of a standard Atari drive.

• Network systems for Atari. These systems look like they'd be ideal for classrooms, computer camps, and even users groups. With the Quick Share, you can hook up to four Atari computers to a single disk drive, 850 Interface Module, and printer. The Quick Share continuously scans the four computers for input/output commands and lets them access the devices on a first-come, first-served basis. Four blinking LEDs let users know when the devices are busy. It costs $595 and is available from Wolsten's Computer Devices, Inc., of East Orange, New Jersey. The company also introduced a similar, but larger system primarily for classroom use. Called the Network 216 and Monitor 16, it allows up to 16 Ataris to connect to a single drive and printer. In addition, the master station hooks up to a TV so the operator can see what's happening on any one of the 16 computer monitors. A headset with a microphone plugs into the station so the operator can converse privately with any of the 16 students (the operator's voice comes through the TV speaker). This looks like a great way for teachers to make sure their students aren't playing *Centipedes* on the sly. It will sell for $1995, cables extra.

• Supermother for VIC-20. What's a Supermother? It appears to be the largest expansion board available for the VIC. This huge board has eight switch-selectable slots for memory and program cartridges, a system reset button, a pause button that freezes games or other programs, and a switch that lets you back up cartridges on tape or disk. It retails for $149.95, from Compuscope, Inc., of Tillamook, Oregon.

Educational Software

Now that more schools are acquiring computers for their students, and more parents are buying home computers for their children, the demand for good educational software is becoming almost unquenchable. Fortunately, some companies with background in other educational fields are starting to get involved in software.

Among these is Scholastic, Inc., of Englewood Cliffs, New Jersey. Remember the *Weekly Reader*? Scholastic is now introducing Wizware, a line of programs for Apple, VIC-20, Atari, and Texas Instruments computers. The first samples are entertaining and colorful and make good use of each computer's special features. Among the interesting programs at the show were *Turtle Tracks*, which uses turtle graphics to teach programming by creating drawings and songs; *The Square Pairs*, a memory game; and *Your Computer*, a how-to introduction to computers with a robot narrator.

Another line of educational software was displayed by Edu-Ware Services, Inc., of Agoura Hills, California. Most were for the Apple, with a few for the Atari. Ranging from preschool to college level, the programs cover basic math, algebra, spelling, reading, perception, and SAT/PSAT preparation. One of the most interesting packages was *Hands On BASIC Programming*, an introduction to Applesoft BASIC with additional instruction on more advanced BASICs. It includes a 185-page manual and two disks of sample programs.

Microfloppy Update

More shots were fired during the show in the continuing microfloppy wars (see "Mass Memory Now And In The Future," **COMPUTE!**, March 1983). Since nobody has agreed yet whether to adopt the 3-inch, 3¼-inch, or 3½- inch standard, everyone seems to be going their own way.

Thus Verbatim Corp. of Sunnyvale, California, widely known for its larger diskettes, unveiled a prototype of a 3½-inch microfloppy disk. The 3½-inch size is backed by Sony, and Verbatim's microfloppy will be manufactured under license from Sony. However, Verbatim is varying a bit even from Sony's standard in order to conform with recommendations of the Microfloppy In-

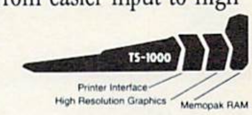
dustry Committee. Verbatim's microfloppy will have 80 tracks instead of 70, an automatic shutter which covers the head window when the disk is removed from a drive, and a thinner magnetic coating.

Meanwhile, across the convention hall, another company was introducing a 3¼-inch microfloppy drive while distributing photocopies of news articles about a rejection of the 3½-inch size. The 3¼-inch drive, hooked up to a Radio Shack TRS-80 Color Computer, was exhibited by Tabor Corp., of Westford, Massachusetts. It's based on the Dysan 3¼-inch microfloppy, a challenger to Sony's 3½-inch disk. Instead of selling directly to the public, Tabor plans to supply the drive to other companies for private labeling. The photocopied article was from *Computer Systems News*, reporting on the recent vote by the American National Standards Institute not to adopt a working paper submitted by Verbatim and Shugart pushing the 3½-inch size.

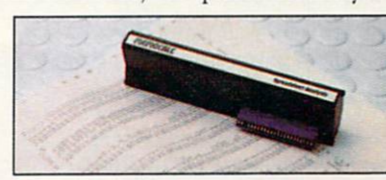
The decision was far from final, however, and all three sizes are still very much alive. And just to make things more interesting, IBM recently unveiled a 4-inch microfloppy disk drive. It appears it will be quite a while before the various factions within the microcomputer industry agree on how much to shrink disks. ©

TIMEX MAKES THE COMPUTER, BUT WE MAKE IT TICK.

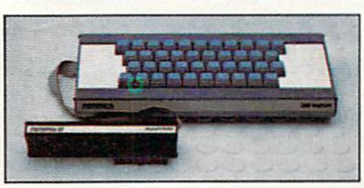
If you own a TS-1000 or ZX-81 computer and want to bring out the power within it, you'll want Memotech. From easier input to high quality output and greater memory, Memotech makes the add-ons you demand. Every Memotech peripheral comes in a black anodized aluminum case and is designed to fit together in "piggy back" fashion enabling you to continue to add on and still keep an integrated system look.



MEMOPAK RAM All Memopak RAMs are directly addressable, user transparent, are neither switched nor paged and no additional power supply is required. You can also choose the Memopak RAM which is just right for your needs. From economy to power. **16K RAM** The Memopak 16K RAM is the most economical way to add memory to your TS-1000. It is fully compatible with the Timex or Memotech 16K RAMs to provide you with up to 32K of RAM. The 16K RAM also offers additional add-on capabilities through its "piggy back" connection. **32K RAM** The 322K Memopak enables you to execute sophisticated programs and store large data bases and like the 16K RAM is fully compatible with Timex's or Memotech's 16K RAMs to give you a full 48K of RAM. **64K RAM** The 64K Memopak is powerful enough to turn your TS-1000 into a computer with capabilities suitable for business and educational use. It accepts such BASIC commands as 10 DIM A (9000). **MEMOCALC** Memocalc, our spreadsheet analysis software, enables TS-1000 users to perform complex number crunching routines with ease. With the 64K RAM a table of up to 7000 numbers with up to 250 rows or 99 columns can be specified. Quick revisions can be achieved by entering new data to your formula.

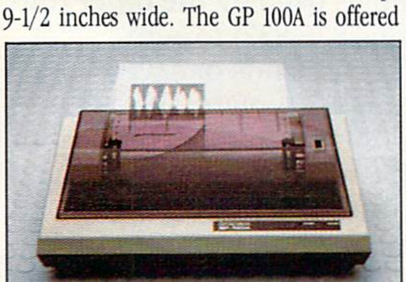


MEMOTECH KEYBOARD For ease of operation, the Memotech keyboard is a high quality standard typewriter keyboard, with TS-1000 legends. The keyboard is cable connected to a buffered interface which is housed in a standard Memopak case and plugs directly into the back of the TS-1000 or other Memopaks. **MEMOPAK HRG** The Memopak High Resolution Graphics, with up to 192 by 248 pixel resolution, enables display of high resolution "arcade game" style graphics through its resident 2K EPROM, programmed with a full range of graphics subroutines.



CENTRONICS PARALLEL AND RS232 INTERFACES Memotech's Interfaces enable your TS-1000 to use a wide range of compatible printers. The resident software in the units gives the complete ASCII set of characters. Both Memopak Interfaces provide lower case character capabilities and up to 80 column printing. The RS232 Interface is also compatible with modems and terminals.

SEIKOSHA GP 100A PRINTER The Seikosha GP 100A uses a 5x7 dot matrix printing format with ASCII standard upper and lower case character set. Printing speed is 30 characters/second with a maximum width of 80 characters. The printer uses standard fanfold paper up to 9-1/2 inches wide. The GP 100A is offered as a package including cable and interface. Other printer packages are also available through Memotech.



ORDER AT NO RISK. All Memotech products carry our 10 day money back guarantee. If you're not completely satisfied, return it within ten days and we will give you a full refund. And every Memotech product comes with a **six month warranty.** Should anything be defective with your Memotech product, return it to us and we will repair or replace it free of charge. **Dealer inquiries welcome.** To order any Memotech product use the order coupon or call our toll-free number **800/662-0949.**

TS-1000 is a registered trademark of Timex Corp.

MEMOTECH
CORPORATION

7550 West Yale Avenue, Denver, Colorado 80227, 303/986-1516, TWX 910-320-2917

Mail To: Memotech Corporation, 7550 West Yale Ave., Denver, CO 80227
Code: CO-7

	Price*	Qty.	Total
16K RAM	$ 49.95		
32K RAM	99.95		
64K RAM	149.95		
Memocalc	49.95		
Keyboard with Interface	99.95		
High Resolution Graphics	99.95		
Centronics Parallel Interface	74.95		
RS232 Interface	99.95		
Printer Cable	19.95		
GP 100A Printer Package**	399.00		
Shipping and Handling	4.95		$ 4.95
Tax (Colorado residents only)			
TOTAL			$

*All prices quoted in U.S. dollars. Prices and specifications subject to change without notice.
**Please add an additional $5.00 for printer shipping charges.

☐ Check ☐ MasterCard ☐ Visa

Account No. __________ Exp. ______

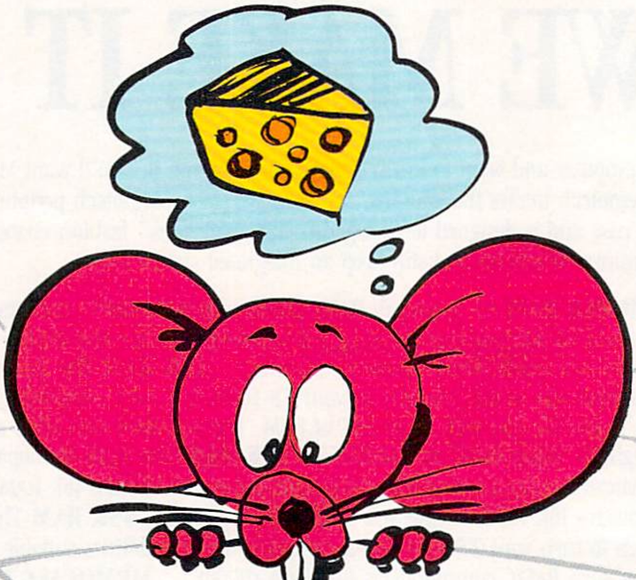
Name __________ ( ) Phone number

Address __________

City ______ State ____ Zip ____

RATS!

Mike Steed



Harry Blair

This impressive game makes you feel that you are inside a maze, not just seeing it from above. Three dimensional views appear as hallways, doors, and corners as you struggle to find the way out. It's for Upgrade or 4.0 BASIC PETs and Commodore 64.

You must find your way through a maze displayed from a rat's eye view. After you have solved the maze, the program displays the top view and traces your steps.

First, you are asked what maze size you want, up to 15 by 15 (you may wish to change the DIM statement in line 49 – add two to the largest dimension you want – and line 43). Line 45 checks to see if the machine code has been POKEd in, so you have to wait for that only the first time.

The space bar is used to move forward, and the "J" and "L" keys are used to turn left and right, respectively (turning doesn't change your location; it just gives you the view in another direction). The "M" key will display the top view of the maze, mark your position, and tell you in which direction you are headed.

There are four machine language routines in RATS! (they will all work as is with Upgrade or 4.0 ROMs). LINE, as its name implies, draws a line; this routine is similar to Applesoft's HPLOT TO or Atari BASIC's DRAWTO command. PLOT sets the "hi-res cursor" to the position from which the next line is to be drawn, and plots that point on the screen.

INIT removes everything that is not a letter or number from the screen (thus the quarter-square graphics are erased, but not the "MOVE XX" at the bottom of the screen), and sets all the variables used by the other routines (locations 826-837) to zero.

SCR either loads or saves something to or from the screen. This routine is used to save the screen to memory after the top view of the maze has been displayed the first time, and from then on is used to display the maze almost instantly, so you have to wait only once.

Readers who want a copy of the program (PET version only) without having to type it in may send a blank tape or 8050 disk, an SASE mailer, and $3 to: *Mike Steed*

712 W. 1280 S.
Provo, UT 84601

Program 1: RATS! PET Version

```
3 POKE 59468,12:PRINT CHR$(142):GOTO 38
4 REM DRAW 3-D VIEW
5 N=2:A=H:B=V:FF=2↑(F-1):SYS IN
6 Z=M%(A,B)*FF:IF ((Z/16) AND 1)=1 THEN ~
    RL=-1:GOSUB 25:GOTO 8
7 W=M%(A+S,B-R)*FF:IF ((W/128) AND 1)=1 ~
    THEN RL=-1:GOSUB 21
8 IF ((Z/64) AND 1)=1 THEN RL=1:GOSUB 25
   :GOTO 10
9 W=M%(A-S,B+R)*FF:IF ((W/128) AND 1)=1 ~
    THEN RL=1:GOSUB 21
10 IF ((Z/128) AND 1)=1 THEN 14
11 N=N+1:IF N>8 THEN 15
12 A=A+R:B=B+S:IF B<2 THEN 15
13 GOTO 6
14 GOSUB 17
15 RETURN
16 REM DRAW CENTER BACK
17 POKE HX,VX+DX(N):POKE HY,YU(N):SYS PL:
    POKE HY,YD(N):SYS LI
18 POKE HX,VX-DX(N):SYS LI:POKE HY,YU(N):
    SYS LI:POKE HX,VX+DX(N):SYS LI
19 RETURN
20 REM DRAW BACK SIDE
21 POKE HX,VX+RL*DX(N-1):POKE HY,YU(N):SY
    S PL:POKE HX,VX+RL*DX(N):SYS LI
22 POKE HY,YD(N):SYS LI:POKE HX,VX+RL*DX(
    N-1):SYS LI
23 RETURN
24 REM DRAW RIGHT OR LEFT SIDE
25 POKE HX,VX+RL*DX(N-1):POKE HY,YU(N-1):
    SYS PL:POKE HX,VX+RL*DX(N)
26 POKE HY,YU(N):SYS LI:POKE HY,YD(N):SYS
     LI:POKE HX,VX+RL*DX(N-1)
27 POKE HY,YD(N-1):SYS LI:POKE HY,YU(N-1)
   :IF N>2 THEN SYS LI
28 RETURN
29 REM GET KEYBOARD CHARACTER
30 GET A$:IF A$="" THEN 30
31 RETURN
32 REM ERROR SOUND
33 POKE 59467,16:POKE 59466,51:POKE 59464
   ,80
34 FOR L=1 TO 50:NEXT
35 POKE 59467,0:POKE 59466,0:POKE 59464,0

36 RETURN
37 REM INITIALIZE
38 HX=828:HY=829:LINE=12288:PLOT=12665:IN
   IT=12685:SCR=12725
39 FL=12726:FH=12730:TL=12734:TH=12738
40 PRINT "{CLEAR}{05 DOWN}{17 RIGHT}RATS!

41 PRINT "{02 DOWN}{03 RIGHT}SOLVE A MAZE
     FROM A RAT'S EYE VIEW
42 INPUT "{03 DOWN}{07 RIGHT}MAZE SIZE (H
   ,V)   3,3{05 LEFT}";H,V
43 IF H<3 OR H>15 OR V<3 OR V>15 THEN 40
44 PRINT "{CLEAR}{DOWN}PLEASE WAIT...
45 IF PEEK(LI)=32 AND PEEK(LI+1)=33 AND P
   EEK(LI+2)=48 THEN 48
46 CK=0:FOR L=12288 TO 12761:READ A:POKE ~
   L,A:CK=CK+A:NEXT
47 IF CK<>45230 THEN PRINT "{DOWN}ERROR I
   N DATA STATEMENTS":STOP
48 N=H*V-1:H=H+1:V=V+1:D=1
49 DIM M%(17,17),WALK(100),CUT(5),DX(8),Y
   U(8),YD(8)
50 FOR J=1 TO V+1:M%(1,J)=4:M%(H+1,J)=1:N
   EXT
51 MX=79:MY=49:VX=39:VY=24:X=VX
52 FOR J=1 TO 8:DX(J)=X:YU(J)=INT(VY-X*VY
   /VX):YD(J)=INT(VY+X*(MY-VY)/VX)
53 X=INT(X*7/10):NEXT
54 FOR I=2 TO H:M%(I,V+1)=8:M%(I,1)=2:FOR
    J=2 TO V:M%(I,J)=15:NEXT:NEXT
55 R=INT(H/2)+1:S=INT(V/2)+1:M%(R,S)=15
56 PRINT "{CLEAR}{DOWN}GENERATING MAZE...
   ";:GOSUB 33
57 REM GENERATE RANDOM MAZE (ALGORITHM FR
   OM ROGERS AND STRASSBERGER)
58 FOR IWALK=1 TO N
59 I=Z
60 IF M%(R-1,S)>14 THEN I=I+1:CUT(I)=1
61 IF M%(R,S-1)>14 THEN I=I+1:CUT(I)=2
62 IF M%(R+1,S)>14 THEN I=I+1:CUT(I)=3
63 IF M%(R,S+1)>14 THEN I=I+1:CUT(I)=4
64 IF I=0 THEN 75
65 IF I<>1 THEN I=INT(RND(1)*I)+1
66 ON CUT(I) GOTO 67,69,71,73
67 M%(R,S)=M%(R,S)-(M%(R,S) AND 1):R=R-1
68 M%(R,S)=M%(R,S)-((M%(R,S)/4) AND 1)*4:
   GOTO 86
69 M%(R,S)=M%(R,S)-((M%(R,S)/8) AND 1)*8:
   S=S-1
70 M%(R,S)=M%(R,S)-((M%(R,S)/2) AND 1)*2:
   GOTO 86
71 M%(R,S)=M%(R,S)-((M%(R,S)/4) AND 1)*4:
   R=R+1
72 M%(R,S)=M%(R,S)-(M%(R,S) AND 1):GOTO 8
   6
73 M%(R,S)=M%(R,S)-((M%(R,S)/2) AND 1)*2:
   S=S+1
74 M%(R,S)=M%(R,S)-((M%(R,S)/8) AND 1)*8:
   GOTO 86
75 IF D=-1 THEN 79
76 IF R<>H THEN 83
77 IF S<>V THEN 82
78 R=2:S=2:GOTO 84
79 IF R<>2 THEN 83
80 IF S<>V THEN 82
81 R=H:S=2:GOTO 84
82 S=S+1:D=-D:GOTO 84
83 R=R+D
84 IF M%(R,S)=15 THEN 75
85 GOTO 59
86 NEXT IWALK
87 MH=H:MV=V:I=INT(RND(1)*(MH-1))+2
88 M%(I,1)=0:M%(I,2)=M%(I,2)-((M%(I,2)/8)
    AND 1)*8
89 H=INT(RND(1)*(MH-1))+2:H1=H:V1=V
90 PRINT "{CLEAR}{DOWN}MAZE COMPLETED.":G
   OSUB 33:GOTO 105
91 REM DISPLAY TOP VIEW OF MAZE
92 HZ=INT(79/MH):VZ=INT(49/MV)
93 SYS IN:POKE 216,24:PRINT TAB(25);"{UP}
         {HOME}";
94 POKE HX,1+HZ:POKE HY,1+VZ:SYS PL:POKE
   HY,MV*VZ+1:SYS LI
95 FOR J=1 TO MV:FOR I=2 TO MH:N=M%(I,J):
   X=I*HZ+1:Y=J*VZ+1
96 IF ((N/2) AND 1)=1 THEN POKE HX,X:POKE
    HY,Y:SYS PL:POKE HX,X-HZ:SYS LI
97 IF ((N/4) AND 1)=1 THEN POKE HX,X:POKE
    HY,Y:SYS PL:POKE HY,Y-VZ:SYS LI
98 NEXT:NEXT
99 RETURN
100 REM MARK PLAYER'S POSITION
101 X=H*HZ-1:Y=V*VZ-1:POKE HX,X+1:POKE HY,
    Y+1:SYS PL
```

```
102 POKE HX,X-HZ+2:POKE HY,Y-VZ+2:SYS LI:P
    OKE HY,Y+2:SYS PL
103 POKE HX,X+2:POKE HY,Y-VZ+2:SYS LI
104 RETURN
105 FOR X=1 TO MH:FOR Y=1 TO MV:M%(X,Y)=M%
    (X,Y)+M%(X,Y)*16:NEXT:NEXT
106 REM PLAY
107 F=INT(RND(1)*4)+1:ON F GOTO 108,109,11
    0,111
108 R=0:S=-1:GOTO 112
109 R=+1:S=0:GOTO 112
110 R=0:S=+1:GOTO 112
111 R=-1:S=0
112 PRINT "{CLEAR}{DOWN}PRESS {REV}J{OFF}
    TO TURN LEFT
113 PRINT "{DOWN}PRESS {REV}L{OFF} TO TURN
     RIGHT
114 PRINT "{DOWN}PRESS {REV}SPACE{OFF} TO
    GO FORWARD
115 PRINT "{DOWN}PRESS {REV}M{OFF} TO DISP
    LAY TOP VIEW OF MAZE
116 PRINT "{03 DOWN}{REV} PRESS ANY KEY TO
     CONTINUE "
117 GOSUB 30:PRINT "{CLEAR}";:GOSUB 5
118 REM GET KEYSTROKE
119 GOSUB 30
120 ON -(A$="J")-2*(A$="L")-3*(A$=" ")-4*(
    A$="M") GOTO 122,124,131,136
121 GOSUB 33:GOTO 112
122 F=F-1:IF F<1 THEN F=4
123 GOTO 125
124 F=F+1:IF F>4 THEN F=1
125 ON F GOTO 126,127,128,129
126 R=0:S=-1:GOTO 130
127 R=+1:S=0:GOTO 130
128 R=0:S=+1:GOTO 130
129 R=-1:S=0
130 GOTO 135
131 Z=M%(H,V):T=Z*2^(F-1):T=(T/128) AND 1:
    IF T=1 THEN GOSUB 33:GOTO 119
132 NM=NM+1:POKE 216,24:PRINT TAB(25);"{UP
    UP}MOVE";NM;"{HOME}";
133 IF NM<100 THEN WALK(NM)=F
134 H=H+R:V=V+S:IF V<2 THEN 147
135 GOSUB 5:GOTO 119
136 IF NOT MS THEN 138
137 POKE FL,218:POKE FH,49:POKE TL,0:POKE
    TH,128:SYS SC:GOTO 139
138 GOSUB 92:POKE FL,0:POKE FH,128:POKE TL
    ,218:POKE TH,49:SYS SC:MS=-1
139 GOSUB 101:PRINT "{HOME}YOU ARE FACING
    ";: ON F GOTO 140,141,142,143
140 PRINT "NORTH";:GOTO 144
141 PRINT "EAST";:GOTO 144
142 PRINT "SOUTH";:GOTO 144
143 PRINT "WEST";
144 PRINT ".  PRESS ANY KEY TO":PRINT "CON
    TINUE":GOSUB 30
145 PRINT "{HOME}
                   ":PRINT "         "
146 GOSUB 5:GOTO 119
147 GOSUB 33:V=V1:H=H1:IF MS THEN POKE FL,
    218:POKE FH,49:POKE TL,0:POKE TH,
    128
148 IF MS THEN SYS SC:GOTO 150
149 GOSUB 92
150 GOSUB 101
151 PRINT "{HOME}{DOWN}CONGRATULATIONS-YOU
    'RE OUT IN";NM;"STEP!{LEFT}{INST}
    S"
152 REM DRAW PATH WALKED
153 POKE HX,H*HZ-HZ/2+1:POKE HY,V*VZ-VZ/2+
    1:SYS PL
154 FOR N=1 TO NM:IF N>100 THEN 158
155 F=WALK(N):V=V+(F=1)-(F=3):H=H+(F=4)-(F
    =2)
156 POKE HX,H*HZ-HZ/2+1:POKE HY,V*VZ-VZ/2+
    1:SYS LI
157 NEXT
158 PRINT:END
159 DATA 32, 33, 48, 173, 58, 3, 133
160 DATA 0, 173, 59, 3, 133, 1, 32
161 DATA 0, 49, 173, 62, 3, 205, 63
162 DATA 3, 16, 8, 240, 6, 32, 173
163 DATA 48, 76, 3, 48, 96, 169, 128
164 DATA 24, 109, 60, 3, 56, 237, 58
165 DATA 3, 141, 63, 3, 169, 128, 24
166 DATA 109, 61, 3, 56, 237, 59, 3
167 DATA 141, 64, 3, 162, 128, 142
168 DATA 66, 3, 142, 69, 3, 232, 142
169 DATA 67, 3, 142, 68, 3, 173, 63
170 DATA 3, 201, 128, 176, 11, 169
171 DATA 127, 141, 68, 3, 169, 0, 56
172 DATA 237, 63, 3, 41, 127, 141
173 DATA 63, 3, 173, 64, 3, 201, 128
174 DATA 176, 11, 169, 127, 141, 67
175 DATA 3, 169, 0, 56, 237, 64, 3
176 DATA 41, 127, 141, 64, 3, 173
177 DATA 63, 3, 205, 64, 3, 176, 32
178 DATA 174, 63, 3, 172, 64, 3, 142
179 DATA 64, 3, 140, 63, 3, 173, 68
180 DATA 3, 141, 66, 3, 173, 67, 3
181 DATA 141, 69, 3, 169, 128, 141
182 DATA 67, 3, 141, 68, 3, 173, 63
183 DATA 3, 74, 141, 65, 3, 169, 0
184 DATA 141, 62, 3, 96, 173, 68, 3
185 DATA 56, 233, 128, 24, 109, 58
186 DATA 3, 141, 58, 3, 173, 69, 3
187 DATA 56, 233, 128, 24, 109, 59
188 DATA 3, 141, 59, 3, 173, 65, 3
189 DATA 24, 109, 64, 3, 141, 65, 3
190 DATA 238, 62, 3, 173, 65, 3, 205
191 DATA 63, 3, 48, 35, 240, 33, 56
192 DATA 237, 63, 3, 141, 65, 3, 173
193 DATA 66, 3, 56, 233, 128, 24, 109
194 DATA 58, 3, 141, 58, 3, 173, 67
195 DATA 3, 56, 233, 128, 24, 109
196 DATA 59, 3, 141, 59, 3, 96, 169
197 DATA 0, 133, 148, 169, 32, 133
198 DATA 2, 165, 0, 201, 80, 176, 56
199 DATA 165, 1, 201, 50, 176, 50
200 DATA 234, 234, 234, 234, 70, 0
201 DATA 38, 148, 106, 38, 148, 133
202 DATA 1, 10, 10, 101, 1, 10, 10
203 DATA 38, 2, 10, 38, 2, 234, 234
204 DATA 234, 133, 1, 166, 148, 189
205 DATA 99, 49, 133, 148, 164, 0
206 DATA 177, 1, 162, 15, 221, 103
207 DATA 49, 240, 4, 202, 16, 248
208 DATA 96, 173, 156, 3, 240, 6, 138
209 DATA 5, 148, 170, 208, 8, 138
210 DATA 73, 255, 5, 148, 73, 255
211 DATA 170, 189, 103, 49, 164, 0
212 DATA 145, 1, 96, 1, 1, 2, 4, 8
213 DATA 32, 126, 123, 97, 124, 226
214 DATA 255, 236, 108, 127, 98, 252
215 DATA 225, 251, 254, 160, 234, 0
216 DATA 173, 60, 3, 141, 58, 3, 133
217 DATA 0, 173, 61, 3, 141, 59, 3
218 DATA 133, 1, 32, 0, 49, 96, 162
219 DATA 128, 160, 0, 134, 34, 132
220 DATA 33, 177, 33, 41, 127, 201
```

Software first, computer second.

Workhorse solutions for tough questions.

OK. You're ready to make the leap. You need a computer and you're ready to buy. But which one? There are so many. Every store you've visited has six different lines. At least two were always **right** for you. How can you make an intelligent business decision?

Decide on your software first. No computer is better than the software that runs the operation. No software is better than **Southern Solutions.** We have real business accounting and record keeping software that is right for today's business world. We sell only through professional computer dealers. That is where you are going to get the help you will need to make any computer become the productivity tool it should be.

Compare our software solutions with theirs:

FileGuard™ — Protects your precious data files. Never allows you to lose files, even if the electricity goes off while you are working. Most other software loses files after power loss.

SuperMath™ — Maybe you are small now, but you don't plan to stay that way. Our software with SuperMath™ will handle numbers up to $1 billion. Most micros stop at far less.

User-Defined Reports — You can decide what your Balance Sheet, P&L, Budget Analysis, etc., will look like.

Complete Systems or Individual Modules — General ledger, accounts receivable, billing, payroll, accounts payable, mailing list management, oil accounting, pharmacy management, encumbrance accounting, etc.

Printer Compatible - Our software uses practically any printer, so you can get what you will need for other uses.

Dealer Supported — Sold only through professional computer dealers.

Hardware Flexible — Software for almost every Commodore computer ever made, including the exciting new Commodore 64*.

Real business software for real business computers, with capability you need, at prices you can afford.

Call or write for the name of the dealer nearest you.

Dealer Hotline:
1-800-527-4548

OFFICIALLY APPROVED BY COMMODORE

Southern Solutions
P.O. Box P, McKinney, Texas 75069 - (214) 542-0278

*Commodore 64 is a registered trademark of Commodore

```
221 DATA 64, 48, 2, 169, 32, 145, 33
222 DATA 200, 208, 241, 232, 224, 132
223 DATA 208, 232, 169, 0, 170, 157
224 DATA 58, 3, 232, 224, 12, 208
225 DATA 248, 96, 169, 218, 133, 31
226 DATA 169, 49, 133, 32, 169, 0
227 DATA 133, 33, 169, 128, 133, 34
228 DATA 162, 4, 160, 0, 177, 31, 145
229 DATA 33, 136, 208, 249, 230, 32
230 DATA 230, 34, 202, 48, 2, 208
231 DATA 240, 96
```

Program 2:

RATS! 64 Version – Setup Program

Run this program before RUNning RATS! on the 64.

```
100 POKE16384,0:POKE16385,0
110 POKE56578,PEEK(56578)OR3
120 POKE56576,(PEEK(56576)AND252)OR1
130 POKE53272,4:POKE648,128
140 POKE53280,12:POKE53281,12
145 POKE641,0:POKE642,64
150 POKE43,1:POKE44,64:POKE55,0:POKE56,1
    28:POKE646,1:PRINT"{CLR}"
```

Program 3:

RATS! 64 Version – Adjustments To Program 1

Replace these lines in Program 1 if you are using the 64.

```
3 :PRINT CHR$(142):GX=49152:GOTO 38
46 CK=0:FOR L=12288 TO 12761:READ A:POKE
    L,A:CK=CK+A:NEXT:FORK=GXTOGX+23:READ
    GX
47 POKEK,GX:NEXT:IF CK<>50144 THEN PRINT
    "{DOWN}ERROR IN DATA STATEMENTS":STO
    P
56 PRINT "{CLR}{DOWN}GENERATING MAZE..."
    ;:GOSUB 2000
90 PRINT "{CLR}{DOWN}MAZE COMPLETED.":GO
    SUB 2000:GOTO 105
93 SYS IN:POKE 214,24:PRINT TAB(25);"
    {UP}{9 SPACES}{HOME}";
117 GOSUB 30:PRINT "{CLR}";:SYS49152:GOS
    UB 5
121 GOSUB2000:GOTO112
131 Z=M%(H,V):T=Z*2↑(F-1):T=(T/128) AND
    1:IF T=1 THEN GOSUB 2000:GOTO 119
132 NM=NM+1:POKE 214,24:PRINT TAB(25);"
    {UP}MOVE";NM;"{HOME}";
147 GOSUB2000:V=V1:H=H1:IF MS THEN POKE
    FL,218:POKE FH,49:POKE TL,0:POKE TH,
    128
```

Program 4:

Add these lines to Program 1 if you are using the 64.

```
2000 S0=54272:FORE=S0TOS0+28:POKEE,0:NEX
     T
2010 POKE54296, 15 :POKE54277, 51 :POKE5
     4278, 211
2020 POKE 54276, 33 :POKE 54273, 63 :POK
     E54272, 75
2030 FORT=1TO 200 :NEXT:POKE54276, 32:FO
     RT=1TO 100 :NEXT
2040 FORE=S0TOS0+28:POKEE,0:NEXT
2050 RETURN
```

Remove lines 32, 33, 34, 35, and 36 if you are using the 64.

RATS! For 64

Gregg Peele, Programming Assistant

The Commodore 64 version of "RATS!" utilizes the same machine language program that was used in the PET version. The program was changed significantly in only two ways. First, zero-page locations were altered because there is limited zero page space on the 64. Second, a routine to fill screen with color has been added to make the maze visible on the newer 64s. (Color RAM must be POKEd on newer 64s, or values POKEd to the screen are invisible.)

Whenever you run the 64 version, you must prepare the 64 by running Program 2 first. Program 2 sets screen memory at 32768 ($8000) and places BASIC at 16384 ($4000); this emulates the PET screen and provides a safe place for both BASIC and the machine language program. Since the screen normally resides at 1024 ($0400), be careful not to hit the RUN/STOP and RESTORE keys simultaneously while you are within the program. If you do this, then the 64 will "forget" where your BASIC program resides, and you will lose your program.

To transform Program 1 (the PET version) into a 64 version, type in Program 1 as is except replace, add, and delete lines as instructed below. Also, *all* DATA statement lines are different (see Program 5).

Program 5:

Use none of the DATA statements from Program 1. Instead, use these for the 64.

```
160 DATA 32, 33, 48, 173, 58, 3, 133, 2
170 DATA 173, 59, 3, 133, 195, 32, 0, 49
180 DATA 173, 62, 3, 205, 63, 3, 16, 8
190 DATA 240, 6, 32, 173, 48, 76, 3, 48
200 DATA 96, 169, 128, 24, 109, 60, 3, 56
210 DATA 237, 58, 3, 141, 63, 3, 169, 128
220 DATA 24, 109, 61, 3, 56, 237, 59, 3
230 DATA 141, 64, 3, 162, 128, 142, 66, 3
240 DATA 142, 69, 3, 232, 142, 67, 3, 142
250 DATA 68, 3, 173, 63, 3, 201, 128, 176
260 DATA 11, 169, 127, 141, 68, 3, 169, 0
270 DATA 56, 237, 63, 3, 41, 127, 141, 63
280 DATA 3, 173, 64, 3, 201, 128, 176, 11
290 DATA 169, 127, 141, 67, 3, 169, 0, 56
300 DATA 237, 64, 3, 41, 127, 141, 64, 3
310 DATA 173, 63, 3, 205, 64, 3, 176, 32
320 DATA 174, 63, 3, 172, 64, 3, 142, 64
330 DATA 3, 140, 63, 3, 173, 68, 3, 141
340 DATA 66, 3, 173, 67, 3, 141, 69, 3
350 DATA 169, 128, 141, 67, 3, 141, 68, 3
360 DATA 173, 63, 3, 74, 141, 65, 3, 169
370 DATA 0, 141, 62, 3, 96, 173, 68, 3
380 DATA 56, 233, 128, 24, 109, 58, 3, 1
    41
390 DATA 58, 3, 173, 69, 3, 56, 233, 128
400 DATA 24, 109, 59, 3, 141, 59, 3, 173
410 DATA 65, 3, 24, 109, 64, 3, 141, 65
```

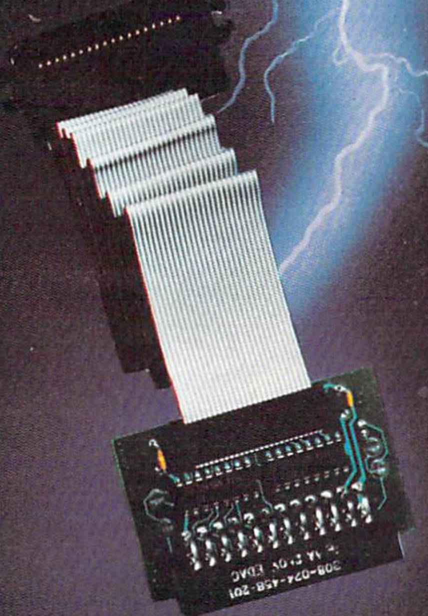
DYNAMIC PRINTER INTERFACES
for the VIC 20™ and the COMMODORE 64™
UNLIKE ANY OTHERS THAT HAVE COME BEFORE

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

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

BUFFERED PARALLEL CABLE & DRIVER — A parallel interface for the budget minded. This interface plugs into the USER-port and comes with an extensive manual with driver listings for the VIC 20™ and the Commodore 64™. It can be used with virtually any printer that has a standard Centronics type configuration and connection. Fully buffered for maximum protection of your computer ONLY $24.95 Add "CABLE BRAINS" cartridge for the VIC 20 (diskette for the 64) and get a full Graphic Emulation Driver. With this package you can print all of the Graphic Characters that your computer has plus EMULATE the Commodore® printer. This product is printer specific to take full advantage of your printer. Available for most graphic matrix printers . . . $29.95

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



MasterCard VISA

Dealer and Distributor Inquiries Invited

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

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

```
420 DATA 3, 238, 62, 3, 173, 65, 3, 205
430 DATA 63, 3, 48, 35, 240, 33, 56, 237
440 DATA 63, 3, 141, 65, 3, 173, 66, 3
450 DATA 56, 233, 128, 24, 109, 58, 3, 1
    41
460 DATA 58, 3, 173, 67, 3, 56, 233, 128
470 DATA 24, 109, 59, 3, 141, 59, 3, 96
480 DATA 169, 0, 133, 168, 169, 32, 133,
     196
490 DATA 165, 2, 201, 80, 176, 56, 165,
    195
500 DATA 201, 50, 176, 50, 234, 234, 234
    , 234
510 DATA 70, 2, 38, 168, 106, 38, 168, 1
    33
520 DATA 195, 10, 10, 101, 195, 10, 10,
    38
530 DATA 196, 10, 38, 196, 234, 234, 234
    , 133
540 DATA 195, 166, 168, 189, 99, 49, 133
    , 168
550 DATA 164, 2, 177, 195, 162, 15, 221,
     103
560 DATA 49, 240, 4, 202, 16, 248, 96, 1
    73
570 DATA 98, 49, 240, 6, 138, 5, 168, 17
    0
580 DATA 208, 8, 138, 73, 255, 5, 168, 7
    3
590 DATA 255, 170, 189, 103, 49, 164, 2,
     145
600 DATA 195, 96, 1, 1, 2, 4, 8, 32
610 DATA 126, 123, 97, 124, 226, 255, 23
    6, 108
620 DATA 127, 98, 252, 225, 251, 254, 16
    0, 234
630 DATA 0, 173, 60, 3, 141, 58, 3, 133
640 DATA 2, 173, 61, 3, 141, 59, 3, 133
650 DATA 195, 32, 0, 49, 96, 162, 128, 1
    60
660 DATA 0, 134, 254, 132, 253, 177, 253
    , 41
670 DATA 127, 201, 64, 48, 2, 169, 32, 1
    45
680 DATA 253, 200, 208, 241, 232, 224, 1
    32, 208
690 DATA 232, 169, 0, 170, 157, 58, 3, 2
    32
700 DATA 224, 12, 208, 248, 96, 169, 218
    , 133
710 DATA 251, 169, 49, 133, 252, 169, 0,
     133
720 DATA 253, 169, 128, 133, 254, 162, 4
    , 160
730 DATA 0, 177, 251, 145, 253, 136, 208
    , 249
740 DATA 230, 252, 230, 254, 202, 48, 2,
     208
750 DATA 240, 96
1000 DATA 162, 0, 169, 1, 157, 0, 216, 1
     57
1010 DATA 0, 217, 157, 0, 218, 157, 0, 2
     19
1020 DATA 232, 208, 241, 96, 234, 234, 2
     34, 0
```
©

Richvale Telecommunications

10610 BAYVIEW (Bayview Plaza)
RICHMOND HILL, ONTARIO, CANADA L4C 3N8
(416) 884-4165

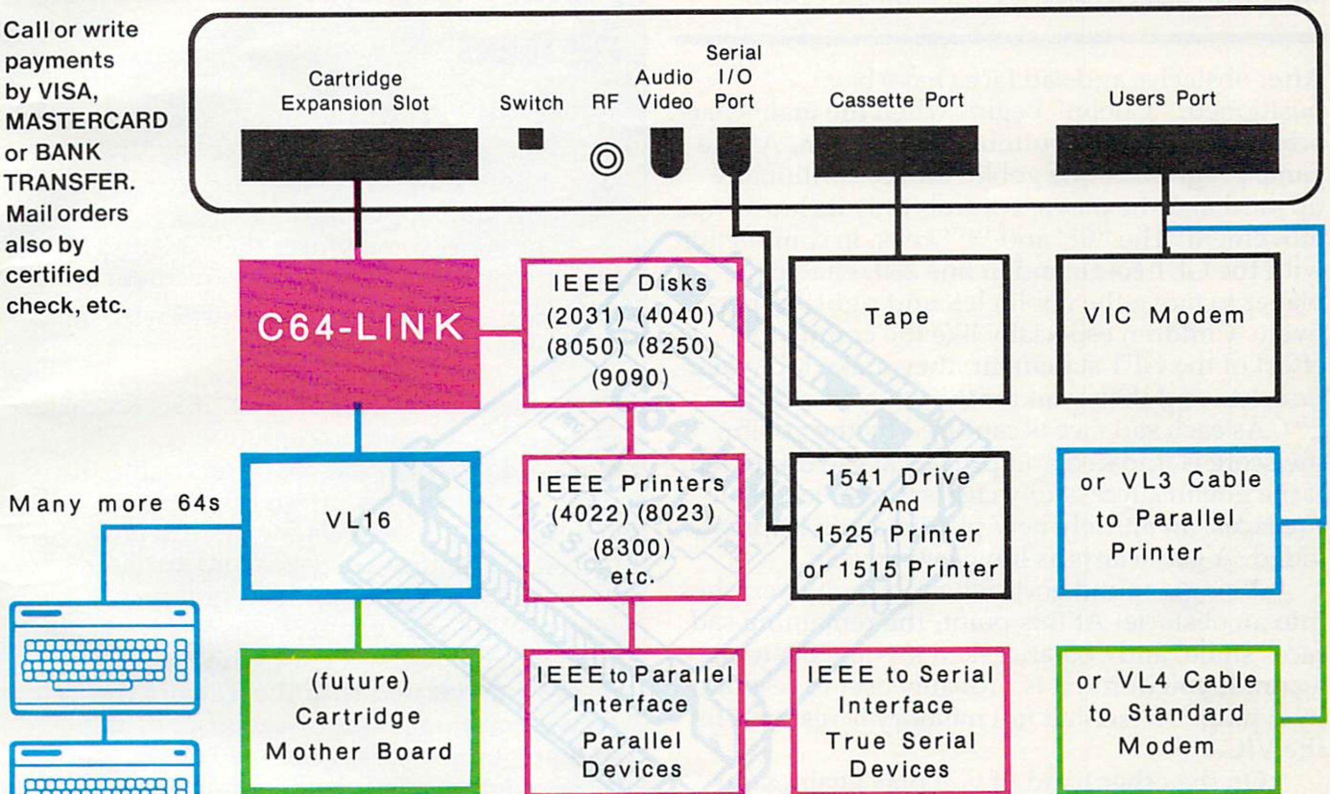
$185⁰⁰ Canadian
$149⁰⁰ U.S.
PLUS CUSTOMS BROKERAGE, HANDLING AND MAILING CHARGE.

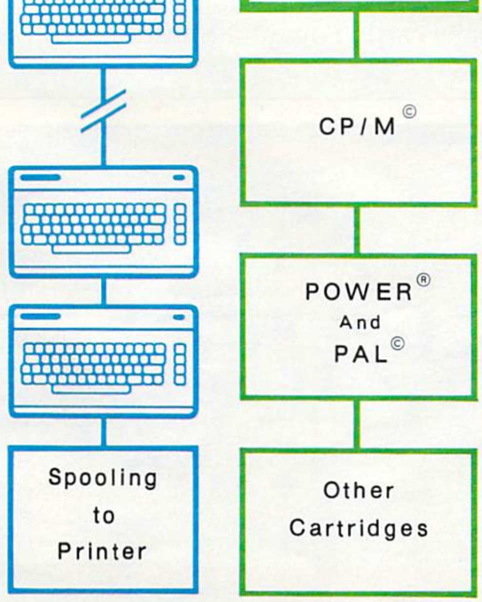
Also available for VIC 20

C64-LINK ©
The Smart 64

RTC — RTC

Call or write payments by VISA, MASTERCARD or BANK TRANSFER. Mail orders also by certified check, etc.





Give These Expanded Capabilities To Your 64 And VIC

- ★ The ability to transfer data from any type of device to another (IEEE, Serial, Parallel)
- ★ BASIC 4.0 which allows you to run more PET BASIC programs and gives you extended disk and I/O commands.
- ★ The ability to have several 64s on line together - sharing common IEEE devices such as disks or printers with Spooling Capability.
- ★ Built-in machine language monitor
- ★ A built-in terminal or modem program which allows the system to communicate through a modem to many bulletin board systems and other computer mainframes.
- ★ Compatibility with CP/M.

Contact your local Commodore Dealer or RTC.

GOBLIN

Dan Goff

In "Goblin" (for the unexpanded VIC, 64, Atari, TI, and Apple) custom characters are used to create a simple yet entertaining game. The object is to capture the scowling creatures with your goblin while avoiding the many block-shaped obstacles that lie in your path.

After obstacles and sad faces have been positioned, "Goblin" begins when the main character appears at the bottom of the screen. As the game progresses, the goblin moves continually upward and the player controls only its horizontal movement. The "O" and "P" keys, in conjunction with the GET command in line 260, enable the player to move the goblin left and right, respectively. Children especially like the cumulative effect of the GET statement; they make rapid key punches and then wait for the delayed effects.

As each sad face is captured by the goblin, the score is updated and printed at the upper left. If the goblin successfully clears the screen of all the faces, an entirely new playfield will be provided. A game lasts as long as you wish.

A single round ends when the goblin crashes into an obstacle. At this point, the remaining sad faces smile, and you are asked if you wish to play again. If you don't, it is probably best to respond by typing "N" so that full memory is restored to the VIC.

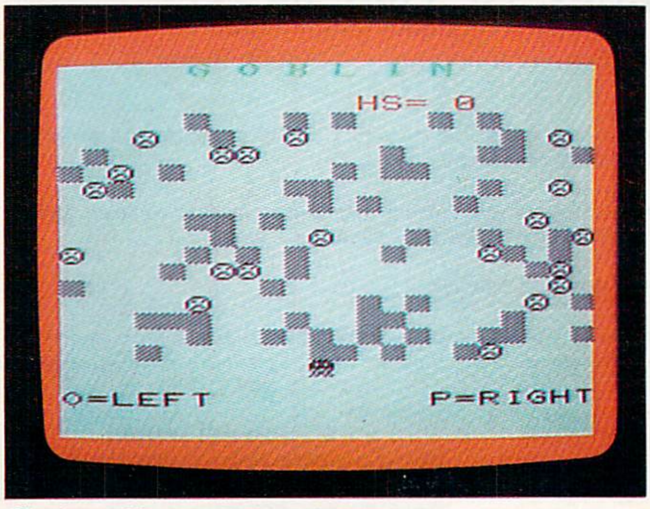
On the other hand, if you play again, your previous highest score will be posted as the new game begins. The incentive to exceed a record score makes any game more fun.

64, Atari, TI-99/4A And Apple Version Notes

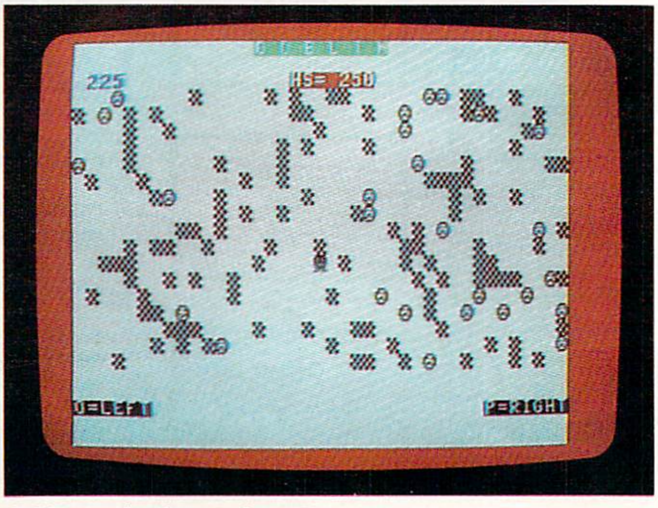
The 64, Atari, TI-99/4A, and Apple versions of Goblin are almost identical to the VIC version. Minor differences do exist, however, in the Atari and Apple versions.

The Atari version uses the "+" and "*" keys to control left and right movement of the goblin. The Apple uses the left and right arrow keys.

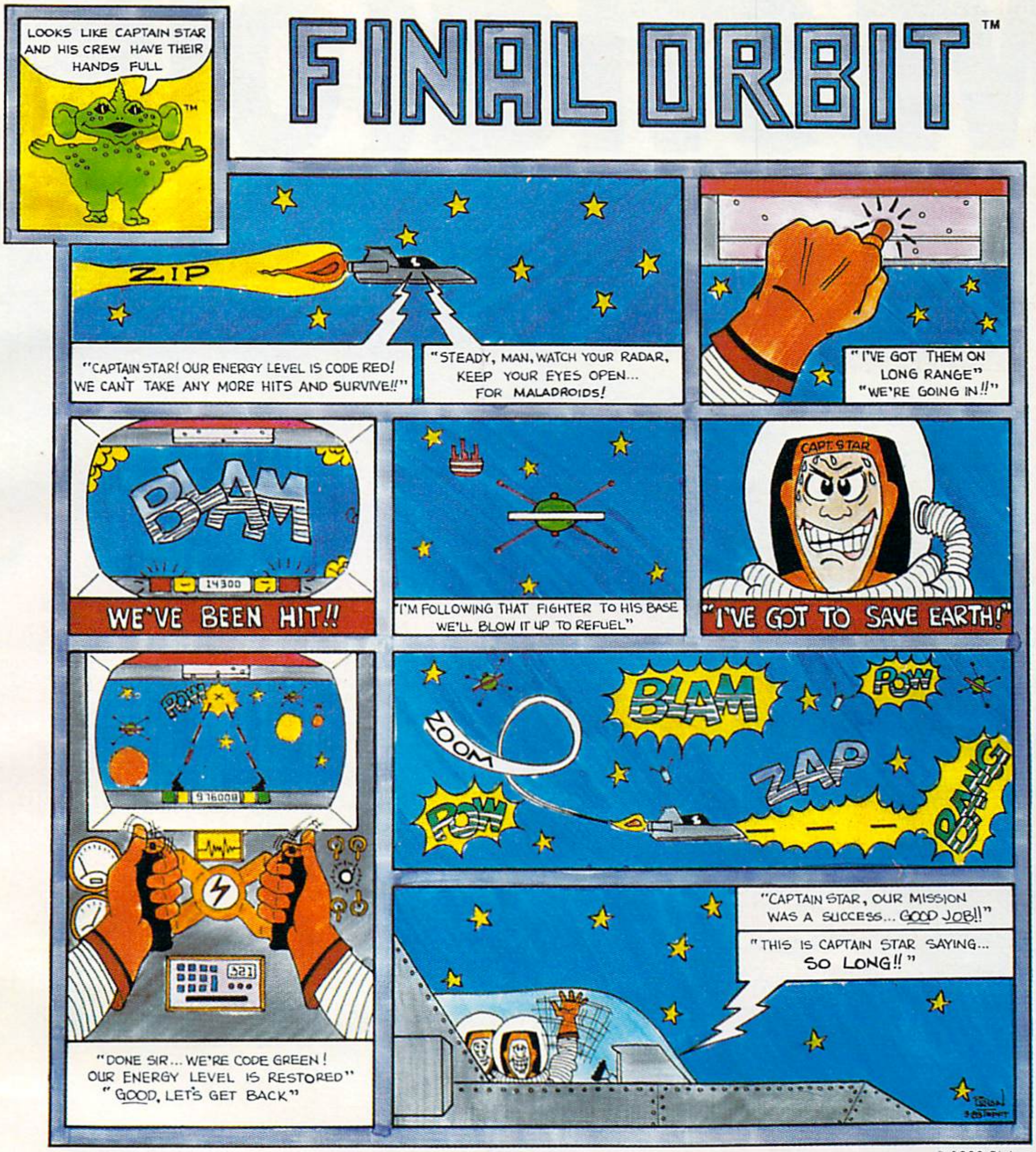
The Apple version requires that you have a disk drive with the DOS Tool Kit disk in the drive when the program is run. This version defines certain characters using the program "Animatrix" from this disk. As Goblin is run, these custom characters are placed in memory as shapes and are later drawn on the high-resolution graphics screen. When the game begins, they are simultaneously POKEd into the areas of memory associated with the text and the high-resolution graphics screens. So, although you see these redefined characters on the high-resolution page, collision detection is actually carried out by PEEKing text screen memory.



Chasing goblins on the VIC-20 version of Goblin.

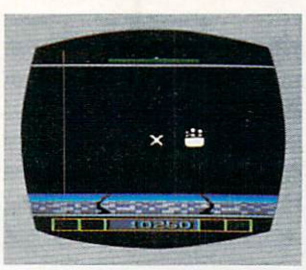


Goblin on the Commodore 64.



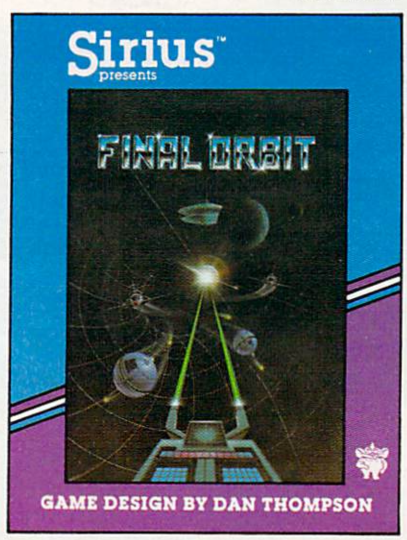
Sirius™

For more information contact your local Sirius dealer or contact Sirius directly at 10364 Rockingham Drive, Sacramento, CA 95827 (916) 366-1195.

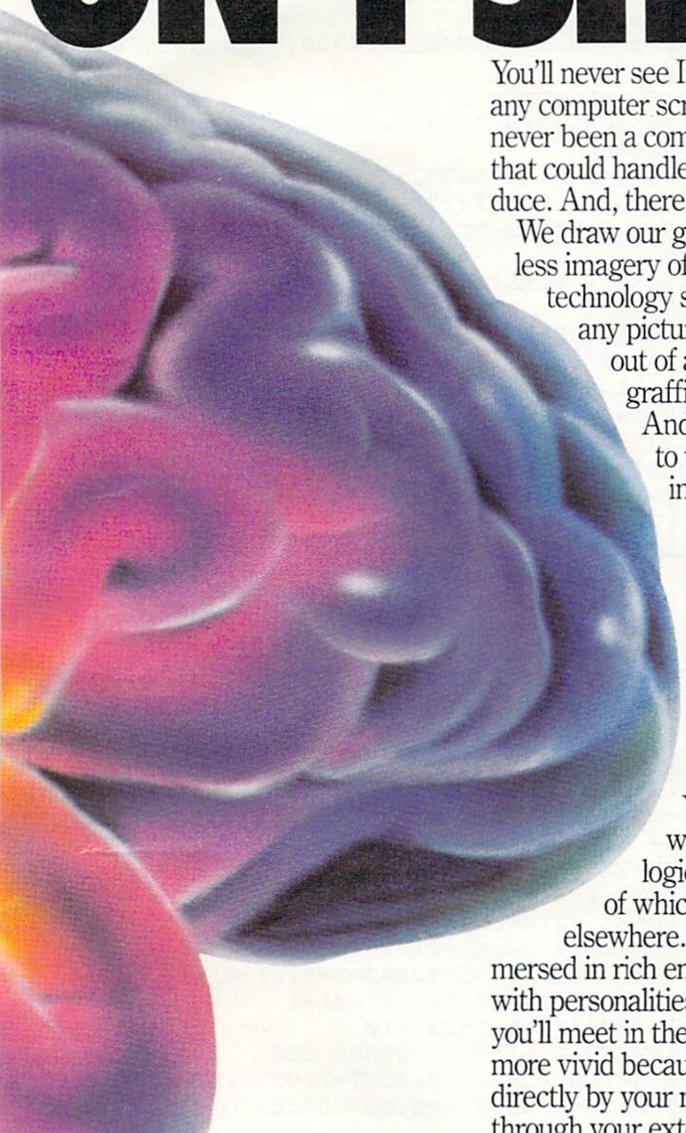


Atari 400, 800 & 1200 Cartridge

VIC-20 Cartridge



RAPHICS WHERE ON'T SHINE.



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

We draw our graphics from the limitless imagery of your imagination—a technology so powerful, it makes any picture that's ever come out of a screen look like graffiti by comparison.

And nobody knows how to unleash your imagination like Infocom.

Through our prose, your imagination makes you part of our stories, in control of what you do and where you go—yet unable to predict or control the course of events.

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

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

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

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

INFOCOM™

The next dimension.

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

For your: Apple II, Atari, Commodore 64, CP/M 8," DEC Rainbow, DEC RT-11, IBM, NEC APC, NEC PC-8000, Osborne 1, TI Professional, TRS-80 Model I, TRS-80 Model III.

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

Program 1: Goblin – VIC Version

```
100 PRINT"{CLR}":POKE 52,28:POKE 56,28:C
    LR:POKE 36869,255:POKE 36879,26
110 IFS>HSTHENHS=S
115 RESTORE:B=230:Z=8152:Z1=Z+30720:W=0:
    S=J:G=0
120 FOR X=1TO32:READ A:POKEX+7167,A:NEXT
    :FORX=1TO8:READA:POKEX+7423,A:NEXT
130 PRINT"{CLR}{RVS}{GRN}{5 RIGHT}G O B
    L I N"
140 PRINT"{HOME}{RED}{2 DOWN}"SPC(12)"
    {RVS}HS="HS:PRINT"{HOME}{RVS}{BLK}
    {20 DOWN}O=LEFT{9 RIGHT}P=RIGHT"
150 FOR I=1 TO 65
160 X=INT(RND(1)*330)+7746
170 IFPEEK(X)=BTHEN 160
180 POKEX,B:POKEX+30720,0:NEXTI
190 FORI=1TO20
200 X=INT(RND(1)*330)+7746
210 IF PEEK(X)=BORPEEK(X)=1ORPEEK(X)=3TH
    EN 200
220 IFPEEK(X+21)=BANDPEEK(X+22)=BANDPEEK
    (X+23)=BTHENPOKEX,3:POKEX+30720,0:G=
    G+1:GOTO240
230 POKEX,1:POKEX+30720,0
240 NEXT I
250 POKEZ,32:Z=Z-22:Z1=Z1-22:IF Z<7746 T
    HEN Z=Z+374:Z1=Z1+374
260 GET A$:IFA$="O"THENZ=Z-1:Z1=Z1-1
270 IFA$="P"THENZ=Z+1:Z1=Z1+1
280 IFPEEK(Z)=B THEN 410
290 IFPEEK(Z)=1 THEN GOSUB 330
300 POKEZ,0:POKEZ1,0:FORT=1TO220:NEXT
310 IFW=20-G THEN J=S:GOSUB350:GOTO110
320 GOTO 250
330 W=W+1:S=S+25:PRINT"{HOME}{BLU}
    {2 DOWN}{RVS}"S:POKE36878,15
340 FORT=235TO250:POKE36876,T:NEXT:POKE3
    6876,0:RETURN
350 PRINT"{HOME}{RED}{16 DOWN}{RVS}*****
    *ALL RIGHT!******"
355 FORI=1TO10:GETA$:NEXTI:REM COLLECT G
    ARBAGE
360 FORI=1TO25
370 X=INT(RND(1)*15)+233
380 POKE36878,15:POKE36875,X
390 FORT=1TO30:NEXTT:NEXTI
400 POKE36878,0:POKE36875,0:RETURN
410 POKE36877,200:FORV=15TO0STEP-1:POKE3
    6878,V:NEXT:POKE36877,0:POKEZ,2
420 FORX=7746TO8075:IF PEEK(X)<>1THEN NE
    XTX
430 IFPEEK(X)=1THEN POKEX,3:NEXTX
440 J=0
445 FORI=1TO10:GET C$:NEXTI
450 PRINT"{HOME}{BLU}{18 DOWN}{RIGHT}
    {RVS}PLAY AGAIN? (Y/N)"
465 GET C$:IF C$="" THEN 465
470 IFC$="Y"THEN 110
490 POKE 36869,240:POKE36879,27:POKE52,3
    0:POKE56,30:PRINT"{CLR}SEE YA!"
500 DATA126,219,219,255,165,90,90,165,60
    ,66,165,129,153,165,66,60
510 DATA 170,85,170,85,126,219,255,189,6
    0,66,165,129,165,153,66,60
520 DATA 0,0,0,0,0,0,0,0
```

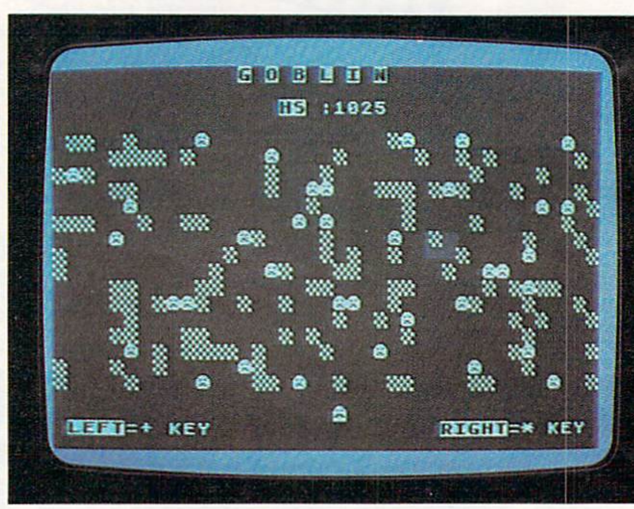
Program 2: Goblin – 64 Version

```
80 POKE 53280,2:POKE 53281,1
90 PRINT"{CLR}{7 DOWN}{4 RIGHT}PLEASE WA
   IT...DEFINING CHARACTERS";
100 POKE 52,48:POKE 56,48:CLR:POKE56334,
    PEEK(56334)AND254
105 POKE1,PEEK(1)AND251
108 FORN=0TO2047:POKEN+12288,PEEK(N+5324
    8):NEXTN
109 FOR N=0 TO 7:POKEN+12320,PEEK(N+5406
    4):NEXT N
110 IFS>HSTHENHS=S
112 RESTORE:B=4:Z=1964:Z1=Z+54272:W=0:S=
    J:G=0
115 VS=54296:AD=54277:SR=54278:WF=54276:
    LB=54272:HB=54273
120 FOR X=0TO31:READ A:POKEX+12288,A:NEX
    T
123 POKE 1,PEEK(1)OR4:POKE56334,PEEK(563
    34)OR1
125 POKE 53272,(PEEK(53272)AND240)+12
130 PRINT"{CLR}{GRN}{14 RIGHT}{RVS}G O B
     L I N"
140 PRINT"{HOME}{RED}{2 DOWN}{RVS}"SPC(1
    7)"HS="HS
145 PRINT"{HOME}{BLK}{22 DOWN}{RVS}O=LEF
    T";SPC(27);"P=RIGHT"
150 FOR I=1 TO 118
160 X=INT(RND(1)*680)+1144
170 IFPEEK(X)=BTHEN 160
180 POKEX,B:POKEX+54272,0:NEXTI
190 FORI=1TO36
195 G1=0
200 X=INT(RND(1)*680)+1144
210 IF PEEK(X)=BORPEEK(X)=1ORPEEK(X)=3TH
    EN 200
220 IFPEEK(X+39)=BANDPEEK(X+40)=BANDPEEK
    (X+41)=BTHENPOKEX,3:POKEX+54272,0:G1
    =1
225 IF G1=1 THEN G=G+1:GOTO 240
230 POKEX,1:POKEX+54272,0
240 NEXT I
250 POKEZ,32:Z=Z-40:Z1=Z1-40:IF Z<1144 T
    HEN Z=Z+760:Z1=Z1+760
260 GET A$:IFA$="O"THENZ=Z-1:Z1=Z1-1
270 IFA$="P"THENZ=Z+1:Z1=Z1+1
280 IFPEEK(Z)=B THEN 410
290 IFPEEK(Z)=1 THEN GOSUB 330
300 POKEZ,0:POKEZ1,0:FORT=1TO220:NEXT
310 IFW=36-G THEN J=S:GOSUB350:GOTO110
320 GOTO 250
330 W=W+1:S=S+25:PRINT"{HOME}{BLU}
    {2 DOWN}"S:POKE VS,15:POKE AD,30:POK
    E SR,200:POKE WF,17
340 POKEHB,71:POKELB,12:FORT=1TO90:NEXTT
    :POKEVS,0:POKEHB,0:POKELB,0:RETURN
350 PRINT"{HOME}{RED}{18 DOWN}{8 RIGHT}
    {RVS}******ALL RIGHT!******"
355 FORI=1TO10:GETC$:NEXTI:REM COLLECT G
    ARBAGE
360 POKE VS,15:POKE AD,30:POKE SR,200:PO
    KE WF,17:FOR I=1 TO 17
370 H=INT(RND(0)*10)+21:L=INT(RND(0)*45)
    +210:POKE HB,H:POKE LB,L
380 FOR T=1 TO 80:NEXT T:NEXTI:POKE VS,0
    :POKE HB,0:POKE LB,0
```

```
400 RETURN
410 POKEZ,2:POKEVS,15:POKEAD,3Ø:POKESR,2
    ØØ:POKEWF,129:POKE HB,2:POKE LB,125
415 FOR I=1 TO 4ØØ:NEXT I:POKE VS,15:POK
    E HB,Ø:POKE LB,Ø
42Ø FORX=1144TO1823:IF PEEK(X)<>1THEN NE
    XTX
43Ø IFPEEK(X)=1THEN POKEX,3:NEXTX
44Ø J=Ø
445 FORI=1TO1Ø:GET C$:NEXTI
45Ø PRINT"{HOME}{BLU}{2Ø DOWN}{RVS}PLAY
    AGAIN? (Y/N)":POKE 646,14
465 GET C$:IF C$="" THEN 465
47Ø IFC$="Y"THEN 11Ø
49Ø POKE53272,21:POKE5328Ø,14:POKE53281,
    6:POKE 52,5Ø:POKE56,5Ø:PRINT"{CLR}SE
    E YA!"
5ØØ DATA126,219,219,255,165,9Ø,9Ø,165,6Ø
    ,66,165,129,153,165,66,6Ø
51Ø DATA 17Ø,85,17Ø,85,126,219,255,189,6
    Ø,66,165,129,165,153,66,6Ø
52Ø DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
```

Program 3: Goblin – Atari Version

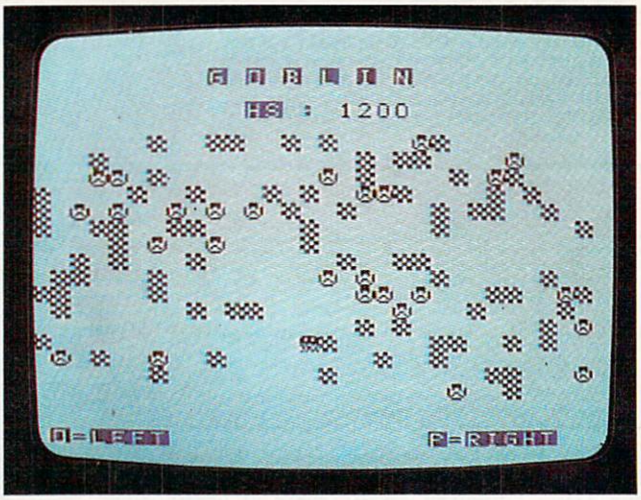
```
9Ø SCREEN=PEEK(88)+256*PEEK(89):DIM
   A$(3):OPEN #1,4,Ø,"K:"
1ØØ GRAPHICS 1+16:POSITION 1,1Ø:? #6
    ;"...PLEASE WAIT..."
1Ø5 GOSUB 2ØØØ
1Ø7 IF S>HS THEN HS=S
1Ø8 S=J1:W=Ø:Z=SCREEN+9ØØ:G=Ø
11Ø GRAPHICS Ø:POKE 752,1:SETCOLOR 4
    ,6,6:SETCOLOR 2,1Ø,1:POSITION 13
    ,Ø:PRINT "G O B L I N"
115 POKE 756,CHSET/256
12Ø POSITION 16,2:? "HS :";HS:POSITI
    ON 1,22:? "LEFT=+ KEY":POSITION
    28,22:? "RIGHT=* KEY";
15Ø FOR I=1 TO 12Ø
16Ø X=SCREEN+INT(RND(Ø)*64Ø)+16Ø
17Ø IF PEEK(X)=7 THEN 16Ø
18Ø POKE X,7:NEXT I
19Ø FOR I=1 TO 36
2ØØ X=SCREEN+INT(RND(Ø)*64Ø)+16Ø
21Ø IF PEEK(X)=7 OR PEEK(X)=1 OR PEE
    K(X)=32 THEN 2ØØ
22Ø IF PEEK(X+39)=7 AND PEEK(X+4Ø)=7
     AND PEEK(X+41)=7 THEN POKE X,1:
    G=G+1:GOTO 24Ø
23Ø POKE X,32
24Ø NEXT I
245 SOUND 1,5Ø,1Ø,12:FOR I=1 TO 5Ø:N
    EXT I:SOUND 1,Ø,Ø,Ø:FOR I=1 TO 2
    ØØ:NEXT I
25Ø POKE Z,Ø:Z=Z-4Ø:IF Z<SCREEN+12Ø
    THEN Z=Z+76Ø
26Ø A=PEEK(764):POKE 764,255:IF A=7
    THEN Z=Z+1
27Ø IF A=6 THEN Z=Z-1
28Ø IF PEEK(Z)=7 THEN 41Ø
29Ø IF PEEK(Z)=32 THEN GOSUB 33Ø
3ØØ POKE Z,5:FOR T=1 TO 1ØØ:NEXT T
31Ø IF W=36-G THEN J1=S:GOSUB 35Ø:GO
    TO 1Ø7
32Ø GOTO 25Ø
33Ø W=W+1:S=S+25:POSITION 3,2:? S
34Ø SOUND 2,2Ø,14,12:FOR I=1 TO 2Ø:S
    OUND 2,Ø,Ø,Ø
345 RETURN
35Ø FOR I=SCREEN+36Ø TO SCREEN+48Ø:P
    OKE I,Ø:NEXT I:POSITION 1Ø,1Ø:?
    "**** ALL RIGHT ***"
355 J1=S
36Ø FOR X=1 TO 2Ø:SOUND 1,3Ø-X,1Ø,12
    :FOR I=1 TO 4Ø:NEXT I:NEXT X:SOU
    ND 1,Ø,Ø,Ø
4ØØ RETURN
41Ø POKE Z,6
415 FOR V=12 TO Ø STEP -1:SOUND 1,4Ø
    ,2,V:SOUND 2,7Ø,12,V:SETCOLOR 4,
    V,6:FOR I=1 TO 4Ø:NEXT I:NEXT V
418 SETCOLOR 4,6,6:SOUND 1,Ø,Ø,Ø:SOU
    ND 2,Ø,Ø,Ø
42Ø FOR X=SCREEN+16Ø TO SCREEN+8ØØ:I
    F PEEK(X)<>32 THEN NEXT X
43Ø IF PEEK(X)=32 THEN POKE X,1:NEXT
     X
44Ø J1=Ø
45Ø POKE 764,255:POSITION 1Ø,21:? "P
    lay Again (Y/N)?";:GET #1,A
46Ø IF A=ASC("Y") THEN 1Ø7
47Ø GRAPHICS 1+16:POSITION 3,1Ø:? #6
    ;"...SEE YA...":FOR I=1 TO 8ØØ:N
    EXT I:END
2ØØØ CHSET=(PEEK(1Ø6)-8)*256:FOR I=Ø
      TO 1Ø23:POKE CHSET+I,PEEK(5734
     4+I):NEXT I
2ØØ1 RESTORE 2ØØ5
2ØØ2 READ A:IF A=-1 THEN RETURN
2ØØ3 FOR J=Ø TO 7:READ B:POKE CHSET+
     A*8+J,B:NEXT J
2ØØ4 GOTO 2ØØ2
2ØØ5 DATA 1,6Ø,126,219,255,189,195,1
     26,6Ø
2ØØ6 DATA 5,6Ø,126,219,255,195,153,2
     55,255
2ØØ7 DATA 6,2Ø4,2Ø4,51,51,2Ø4,126,21
     9,255
2ØØ8 DATA 7,2Ø4,2Ø4,51,51,2Ø4,2Ø4,51
     ,51
2ØØ9 DATA 32,6Ø,126,219,255,231,219,
     126,Ø
2Ø1Ø DATA -1
```



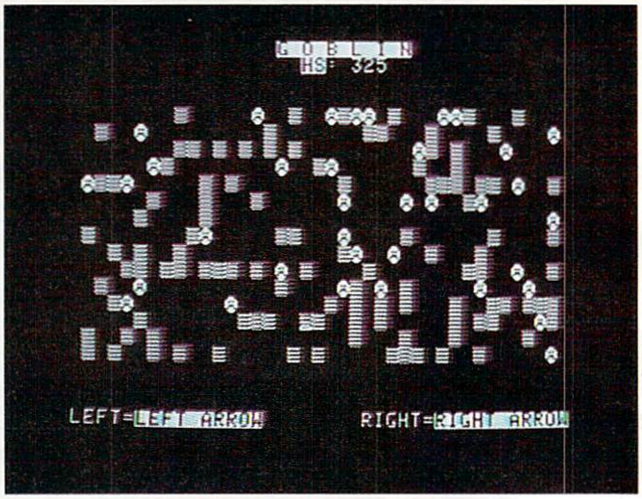
Atari version of Goblin.

Program 4: Goblin – TI-99/4A Version

```
100 RANDOMIZE
110 GOTO 170
120 FOR I=1 TO LEN(H$)
130 R=ASC(SEG$(H$,I,1))
140 CALL HCHAR(ROW,XCOL+I,R)
150 NEXT I
160 RETURN
170 A=96
180 B=97
190 C=104
200 D=105
210 Z=24
220 COL=16
230 W=0
240 G=0
250 S=J
260 CALL CLEAR
270 IF S>HS THEN 290
280 GOTO 300
290 HS=S
300 GOSUB 1270
310 CALL SCREEN(16)
320 PRINT "{8 SPACES}G O B L I N"
330 PRINT
340 PRINT "{10 SPACES}HS : "
350 FOR I=1 TO 19
360 PRINT
370 NEXT I
380 PRINT "O=LEFT{14 SPACES}P=RIGHT";
390 ROW=4
400 XCOL=17
410 H$=STR$(HS)
420 GOSUB 120
430 FOR I=1 TO 80
440 X=INT(RND*30)+2
450 Y=INT(RND*16)+6
460 CALL GCHAR(Y,X,L)
470 IF L=B THEN 440
480 CALL HCHAR(Y,X,B)
490 NEXT I
500 FOR I=1 TO 27
510 X=INT(RND*30)+2
520 Y=INT(RND*16)+6
530 CALL GCHAR(Y,X,L)
540 IF (L=B)+(L=C)+(L=D)THEN 510
550 CALL GCHAR(Y+1,X-1,L)
560 CALL GCHAR(Y+1,X,M)
570 CALL GCHAR(Y+1,X+1,N)
580 IF (L<>B)+(M<>B)+(N<>B)THEN 620
590 CALL HCHAR(Y,X,D)
600 G=G+1
610 GOTO 630
620 CALL HCHAR(Y,X,C)
630 NEXT I
640 CALL SOUND(100,500,6)
650 CALL HCHAR(Z,COL,32)
660 IF L<>C THEN 680
670 CALL SOUND(10,880,4)
680 Z=Z-1
690 IF Z>4 THEN 710
700 Z=23
710 CALL KEY(0,L,ST)
720 IF (L<>79)*(L<>80)THEN 770
730 IF L<>79 THEN 760
740 COL=COL-SGN(COL-2)
750 GOTO 770
760 COL=COL+SGN(30-COL)
770 CALL GCHAR(Z,COL,L)
780 IF L=B THEN 1060
790 IF L=C THEN 850
800 CALL HCHAR(Z,COL,A)
810 FOR I=1 TO 25
820 NEXT I
830 IF W=27-G THEN 920
840 GOTO 650
850 W=W+1
860 S=S+25
870 H$=STR$(S)
880 ROW=4
890 XCOL=3
900 GOSUB 120
910 GOTO 800
920 J=S
930 CALL HCHAR(10,1,32,31)
940 GOSUB 120
950 H$="**** ALL RIGHT! ****"
960 XCOL=6
970 ROW=10
980 GOSUB 120
990 FOR I=1 TO 15
1000 X=INT(RND*100)+300
1010 CALL SOUND(75,X,8)
1020 NEXT I
1030 FOR I=1 TO 100
1040 NEXT I
1050 GOTO 210
1060 REM WHOOPS! ...YOU CRASHED...
1070 CALL HCHAR(Z,COL,98)
1080 FOR I=3 TO 30 STEP 3
1090 CALL SOUND(50,-7,I)
1100 NEXT I
1110 CALL CHAR(104,"3C42A581A599423C")
1120 J=0
1130 HS=S
1140 H$="PLAY AGAIN (Y / N)?"
1150 ROW=22
1160 XCOL=2
1170 GOSUB 120
1180 CALL KEY(0,L,ST)
1190 IF ST=0 THEN 1180
1200 H$=CHR$(L)
1210 IF H$="Y" THEN 1250
1220 CALL CLEAR
1230 PRINT "SEE YA!"
1240 END
1250 CALL CHAR(104,"3C3CA58199A5423C")
1260 GOTO 210
1270 REM DEFINE CUSTOM CHARS
1280 REM CHAR 96 - GOBLIN
1290 CALL CHAR(96,"7EDBDBFFA55A5AA5")
1300 REM CHAR 97 - BARRIER
1310 CALL CHAR(97,"CCCC3333CCCC3333")
1320 REM CHAR 98 - CRUNCHED GOBLIN
1330 CALL CHAR(98,"CCCC33337EDBFFBD")
1340 REM CHAR - 104 - FROWN
1350 CALL CHAR(104,"3C3CA58199A5423C")
1360 REM CHAR - 105 - SMILE
1370 CALL CHAR(105,"3C42A581A599423C")
1380 CALL COLOR(10,7,1)
1390 FOR I=5 TO 8
1400 CALL COLOR(I,16,14)
1410 NEXT I
1420 RETURN
```



Goblin, TI-99/4A version.



The Apple version of Goblin.

Program 5: Goblin – Apple Version

```
10  REM *THIS PROGRAM REQUIRES A DISK D
    RIVE
20  REM  AND THE APPLE 'DOS TOOL KIT PR
    OGRAMMING
30  REM  UTILITIES DISK' TO RUN.
40  REM *PLACE THE ABOVE UTILITY DISK I
    N YOUR
50  REM  DRIVE BEFORE RUNNING THIS PROG
    RAM.
55  GOSUB 1000
60  DIM XL%(23): FOR I = 0 TO 7:XL%(I) =
    1024 + 128 * I:XL%(I + 8) = 1064 +
    128 * I:XL%(I + 16) = 1104 + 128 *
    I: NEXT I
85  HOME : HGR : POKE  - 16302,0: IF S >
    HS THEN HS = S
90 ZROW = 23:ZCOL = 19:W = 0:S = J1:G =
    0
100  VTAB 1: HTAB 17: PRINT  CHR$ (1);"
    0"; CHR$ (9);"G O B L I N";
110  VTAB 2: HTAB 19: PRINT "HS"; CHR$
    (14);": ";HS: VTAB 23: PRINT "LEFT
    ="; CHR$ (9);"LEFT ARROW"; CHR$ (1
    4);
120  HTAB 24: PRINT "RIGHT="; CHR$ (9);
    "RIGHT ARROW"; CHR$ (14);
125  PRINT  CHR$ (1);"1";
130  FOR I = 1 TO 120
135 ROW =  INT ( RND (1) * 15) + 5:COL =
     INT ( RND (1) * 38) + 2
140 X = XL%(ROW) + COL: IF  PEEK (X) =
    164 THEN 135
145  VTAB ROW: HTAB COL: PRINT  CHR$ (1
    64);: NEXT I
150  FOR I = 1 TO 36
160 ROW =  INT ( RND (1) * 15) + 5:COL =
     INT ( RND (1) * 38) + 2:X = XL%(R
    OW - 1) + COL - 1
170  IF  PEEK (X) = 164 OR  PEEK (X) =
    161 OR  PEEK (X) = 163 THEN 160
180  IF  PEEK (XL%(ROW) + COL - 2) = 16
    4 AND  PEEK (XL%(ROW) + COL - 1) =
    164 AND  PEEK (XL%(ROW) + COL) = 1
    64 THEN  HTAB COL: VTAB ROW: PRINT
     CHR$ (161);:G = G + 1: GOTO 200
190  HTAB COL: VTAB ROW: PRINT  CHR$ (163);
200  NEXT I
240  POKE 768,5: POKE 769,180: CALL 770
250 Z = XL%(ZROW) + ZCOL: HTAB ZCOL: VTAB
    ZROW: PRINT  CHR$ (167);:ZROW = ZR
    OW - 1: IF ZROW < 3 THEN ZROW = 21
260 A =  PEEK ( - 16384): POKE  - 16368
    ,0: IF A = 136 THEN ZCOL = ZCOL -
    1: IF ZCOL < 1 THEN ZCOL = 39
270  IF A = 149 THEN ZCOL = ZCOL + 1: IF
    ZCOL > 39 THEN ZCOL = 2
280  IF  PEEK (XL%(ZROW - 1) + ZCOL - 1
    ) = 164 THEN 410
290  IF  PEEK (XL%(ZROW - 1) + ZCOL - 1
    ) = 163 THEN  GOSUB 330
300  HTAB ZCOL: VTAB ZROW: PRINT  CHR$
    (165);: FOR T = 1 TO 100: NEXT T
310  IF W = 36 - G THEN J = S: GOSUB 35
    0: GOTO 85
320  GOTO 250
330 W = W + 1:S = S + 25: VTAB 2: HTAB
    3: PRINT  CHR$ (1);"0";S; CHR$ (14
    ); CHR$ (1);"1";
340  POKE 768,2: POKE 769,230: CALL 770
345  RETURN
350  FOR J = 10 TO 12: VTAB J: FOR I =
    0 TO 39: HTAB I: PRINT  CHR$ (167)
    ;: NEXT I: NEXT J: VTAB 17: HTAB 1
    0: PRINT  CHR$ (1);"0";"****** ALL
     RIGHT! ******"; CHR$ (1);"1";
360  FOR I = 1 TO 10
370  POKE 768, INT ( RND (1) * 3) + 1: POKE
    769, INT ( RND (1) * 15) + 130: CALL
    770
380  NEXT I
385 J1 = S
390  FOR J = 1 TO 500: NEXT J
400  RETURN
410  HTAB ZCOL: VTAB ZROW: PRINT  CHR$
    (166);:C = 0
415 X =  PEEK ( - 16336):C = C + 1: IF
    C < 15 THEN 415
420  FOR ROW = 0 TO 23: FOR COL = 1 TO
    38:X = XL%(ROW) + COL: IF  PEEK (X
    ) <  > 163 THEN  NEXT COL: NEXT ROW
430  IF  PEEK (X) = 163 THEN  VTAB ROW +
    1: HTAB COL + 1: PRINT  CHR$ (161)
    ;: NEXT COL: NEXT ROW
440 J1 = 0: VTAB 21: HTAB 13: PRINT  CHR$
    (1);"0";"PLAY AGAIN ("; CHR$ (9);"
```

```
     Y"; CHR$ (14);"/"; CHR$ (9);"N"; CHR$
     (14);") ?";
450  POKE  - 16368,0: GET C$: IF C$ = "
     Y" THEN 85
460  TEXT : HOME : VTAB 4: HTAB 2: PRINT
     "SEE YA!...HIT RESET...": FOR I =
     1 TO 1000: NEXT I: END
1000  REM  INIT SUBROUTINE
1020 ADRS = 0
1030  PRINT  CHR$ (4);"BLOAD RBOOT"
1040  CALL 520
1050 ADRS =  USR (0),"HRCG"
1051  IF ADRS < 0 THEN ADRS = ADRS + 65536
1060 CS = ADRS - 768
1061 CH =  INT (CS / 256):CL = CS - CH *256
1062  POKE ADRS + 7,CL: POKE ADRS + 8,CH
1070  HIMEM: CS
1080  READ A: IF A =  - 1 THEN 1100
1090  FOR J = 0 TO 7: READ B: POKE CS +
     A * 8 + J,B: NEXT : GOTO 1080
1100  CALL ADRS + 3
1110  PRINT  CHR$ (16)
1150  REM  SOUND ROUTINE
1160  FOR I = 770 TO 795: READ M: POKE
     I,M: NEXT
1170  RETURN
1500  DATA  1,28,62,127,107,127,93,34,28
1510  DATA  3,28,62,107,127,99,93,62,28
1520  DATA  4,85,42,85,42,85,42,85,42
1530  DATA  5,28,62,107,127,107,85,127,119
1540  DATA    6,85,42,85,42,85,62,107,127
1545  DATA  7,0,0,0,0,0,0,0,0
1550  DATA  -1
1560  DATA  172,1,3,174,1,3,169,4,32,16
     8,252,173,48,192,232,208,253,136,2
     08,239,206,0,3,208,231,96
```
©

SpeedSki

Dub Scroggin

SpeedSki takes VIC BASIC to its limits. Like most good action games, SpeedSki is easy to learn and hard to master. What's equally impressive, the program runs extremely fast, and creates an excellent, realistic physical challenge. It sounds and feels like skiing – complete with jumps, trees, fences, and an ever-changing pathway.

Also, if you're interested in programming games in VIC BASIC, the author provides a complete explanation of how the program works. He discusses the techniques which permit such amazing execution speed.

With five skill levels, for one to four players, on any unexpanded VIC. The world's champion SpeedSkier (the author himself) has managed to achieve a score of 168 during a five-run series. Do better than that and you'll be the new record holder.

"SpeedSki" is a fast, action VIC-20 game that fits in standard memory and makes full use of the VIC's color and sound capabilities. It is controlled from the keyboard and provides up to five rounds of play for one to four players, allowing each to select from any of five skill levels.

The game was designed around one central concept – speed. Every effort, short of machine language, has been used to make the game run as fast as possible without sacrificing too much realism. The result is an exciting game requiring concentration and practice. It's easy to learn the basics at skill level one, then step gradually up to level five, but mastery will take a lot of practice.

Avoid The Hazards

The object is to steer a skier through 10 gates, while avoiding the hazards posed by trees and fences. The optional jumps will improve your time. The best possible time, about 29 seconds, can be achieved at skill level five by avoiding every hazard, hitting every gate, and taking every jump. But getting this best time is not easy, even for an expert. I've played the game several hundred times and have made a perfect score only a handful of times. And I'm the greatest SpeedSki player in the world. The fact that as I write this there are only three players in the world could have something to do with this, of course. The other two are my daughter, who is second best in the world, and a friend's son, who has played only once. My best score for a five-run series is 168. Beat that score and you'll be the world's champion SpeedSki player.

You should take the jumps whenever you can – they not only move you ahead, they also take you over trees you might otherwise hit, and increase your speed. Every time you hit a tree, you move up one line on the screen (to a limit of ten), and you have more time to react to the slope coming up from the bottom. You are also a little farther from the finish line. Whenever you hit a jump, you move down a line (to a limit of three below the center), so you are closer to the finish line, but you must also react faster.

There are a number of REMarks in the program listing as an aid to understanding, but I recommend they not be typed in because of the memory they consume.

Defining Characters

Line 10 prints the title, and line 20 sets the memory limits that are necessary in a program employing user-defined characters. Moving the end of memory indicators hides a section of memory from BASIC, so this section can be used for storing the user-defined character values.

Try this: print FRE(X) and hit RETURN. Then type POKE 56,28: POKE 55,250: POKE 52,28: POKE 51,250 and hit RETURN. Now type FRE(X) and hit RETURN, again. You'll see the difference. BASIC has been fooled into thinking the end of its memory is closer than it really is, and you appear to have lost about 260 bytes of memory. Line 20 also sets the screen and border colors to white and white, like snow.

Line 30 reads X, a memory location in the protected area set up by line 20. If X is 0, then all data has been read, and control passes to the instructions starting in line 70. Otherwise, line 40 reads the values to be placed in X and the seven following bytes, and POKEs these values in. For instance, line 30 reads "7672". Line 40 then reads "16" and POKEs 7672 to 16. Then it reads "56" and POKEs 7673 to 56, then 7674 to 56, and so on.

Control then goes back to line 30 where the next value of X is read in and tested. The final data step contains a 0 for the value of X following the eight values of Y. So when all the data has been read in, line 30 ends this part of the program.

Players And Skill Levels

Lines 70-90 print the directions. Note that the symbol "T" in line 70 means to press the Commodore flag key, and then hit the "T" to underline the title. Line 100 is used for inputting the number

of players and also for rejecting bad input. A value outside the allowed range passes control back to line 70; the screen is cleared, the instructions are reprinted, and you are asked for the number of players again. Line 110 accepts the number of rounds desired and rejects bad input in the same manner as line 100.

Line 120 initializes the values of R (the number of the present round) and P (the number of the present player). Lines 130-140 prompt the player skill levels, and line 150 accepts the player choice as a string variable, A$. Lines 160-200 assign values to S$ based on the skill level input, and line 210 converts A$ to the numeric variable SK. It then uses SK to establish a value for RN, which will control the number of trees printed.

The number of trees is tied to the skill level, so that the higher the skill level, the more trees there will be. If you'd like more trees, change the "1" to a larger number, but no more than 5. If SK is not an integer, or is outside the range of 1 to 5, line 210 rejects it. Moving the cursor up ten spaces and passing control back to line 130 makes it appear that the program does nothing but sit there until a correct input is given.

Speed Versus Obstacles

Line 220 establishes a new value for SK to control the speed of the program – faster for higher skill levels. Line 230 POKEs 36869 to 255 and causes the user-defined character set to be used instead of the normal set. This may cause some problems with debugging.

If an error is present after this step, the program will stop, but all you'll see on the screen will be garbage with an occasional skier or tree thrown in. If this happens, hit the CTRL and RVS keys, then type POKE 36869, 240 and RETURN. All that garbage will suddenly make sense. Line 230 also clears the screen, sets the volume, and establishes S as the noise generator.

Line 240 prints the trees on the screen for the initial setup. Each time through this loop, a random value "L" between 1 and 19 is calculated. Then a fence section is printed on the left, a tree is printed at TAB (L), and a fence section is printed on the right.

The initial value of B is set to 7910 in line 250. This is the location of the skier in screen memory. C is the difference between the screen map position and the color code map position. F is the POKE value for the skier figure; the POKE value will be 55 when he's going to the left and 53 when he's going to the right. The last three statements of line 250 insure that the player is not faced with the no-escape situation of having trees directly in front of him at the start of a run.

Line 260 POKEs the flags of the first gate onto the screen, and line 270 prints the level that was determined in lines 160-200. Line 280 puts the line between the flags for the first gate, and line 290 sounds the warning tones to let you know it's time to start. Just after the last tone, line 300 sets the timer. Line 310 then waits for you to press a key. If you don't hit a key for a while, that's okay, but the timer is running. You should use the time that the warning tones give you to plan your course through the first gate and then take off as soon as the last tone sounds.

Line 320 starts the main program loop. If SK is not zero, then the computer counts to SK before proceeding. This time delay, remember, is tied to the skill level to start with, but it may be reduced by hitting the jumps.

Skier Movement

If F is 55 in line 330, the skier is going left, and a track is POKEd in behind him using a POKE value of 58. If not, he's going right and the track's POKE value is 59. The track is handled in line 340.

Lines 350 and 360 are the keyboard control steps. If PEEK (197) – which is the memory location that contains the current key pressed – is 29, then the key for going left has been pressed. D will later be used to produce movement to the left; F is set to the figure for going left; and S, which is the noise generator, is set to 245. If any other key is pressed, or even if no key is pressed, then the skier will be going to the right, and the values needed for D, F, and S are set by line 360. You'll notice this slight change in sound when you change directions; it should sound like wind.

Gates And The Finish Line

G is incremented in line 370. If it's less than 28, control passes to line 410, because no gate or finish line is required. Otherwise, G is reset to 0 in line 380, and E, which counts the gates, is incremented. If E is 10, a finish line is printed and control passes to 460. Line 390, which causes the program to end, is executed only if the skier is past the finish line. If E is less than 10, then a random value between 2 and 11, inclusive, is calculated. A gate is then printed starting at TAB(X), X being the random number just calculated. Control then passes to line 460.

If no gate or finish line needs printing, control passes from line 370 to line 410, skipping all the above to reduce the time required for a pass through the loop. If G is 10, then line 410 prints a jump at TAB(X), X now being a random number between 4 and 13, inclusive. Fence sections are also printed at the left and right sides of the screen.

Line 240 decides whether a tree will be printed using the value of RN that was established in line 210. For skill level five, RN will have a value of .6; if a random number is more than this, no tree is

VIC-20
OWNERS
CREEPY CORRIDORS
SIERRAVISION
CREEPY CORRIDORS™— Find the jewels before the creepy critters find you!
FLIP N MATCH
SIERRAVISION
FLIP-N-MATCH™. Find twin objects before time runs out.
Ultima
ESCAPE FROM MT. DRASH
SIERRAVENTURE
ULTIMA · Escape from Mt. Drash™. A real-time, fantastic adventure.
The sky's the limit! Now Vic 20 owners can get a bigger piece of the action with Sierra On-Line's new titles. We're starting with CREEPY CORRIDORS,™ FLIP-N-MATCH,™ and ULTIMA · Escape from Mt. Drash,™ and as weeks pass, we'll be bringing you more of our most popular titles.
AVAILABLE AT YOUR LOCAL COMPUTER STORE.
Sierra ON-LINE INC
SIERRA ON-LINE BUILDING COARSEGOLD CA 93614
209 683 6858
TM designates trademark of SIERRA ON-LINE, INC.

printed. This means a tree will be printed roughly 60 percent of the time. For the lower skill levels, this probability is reduced so that the lower the skill level, the fewer trees there will be. If no tree is to be printed, line 440 prints only the fence sections. Otherwise, line 430 prints a tree at TAB(L), L being a random value between 1 and 19, inclusive.

If PEEK (B) in line 450 is not 32 (a blank), then the skier has run into something and control passes to line 500 to find out what the skier has run into and what to do about it.

The Illusion Of Motion

Line 460 POKEs the skier's location blank, then calculates a new position by adding the value of D (determined in lines 350 and 360) to B, the skier's location. It then POKEs the appropriate figure into that location. Essentially, the skier is placed on a horizontal line on the screen and is allowed to move only back and forth on that line. However, the screen is scrolling upward beneath him, so the illusion of forward motion is created.

The movement taken care of, control passes back to line 320 for another pass through the main loop. This loop, lines 320-470, has been kept as small as possible in order to minimize the time required for each pass through it. I have tried to be very stingy with time in this section, figuring that even one instruction repeated a few hundred times adds a lot of potentially unnecessary time.

Flags And Fences

Line 500 is reached when line 450 detects that something has been struck. This entire section was originally a part of the main loop, but removing it from the loop and replacing it with the single statement in line 450 produced a significant increase in speed. Line 500 checks to see if a gate was hit. If so, it sounds a high tone to let you know you got credit for the gate, then increments H, the number of gates hit, and passes control back into the main loop.

Line 510 checks to see if a finish line was struck. If so, H is changed to the number of gates missed, the elapsed time is placed in TM, and control passes to line 640 to end the run.

If a flag was hit, line 520 sounds a low tone to let you know you were close but get no credit for the gate. Control then passes to line 570.

If a jump wasn't hit, line 530 transfers control to 570. Lines 540-560 handle the jumps. The skier is moved two spaces horizontally in the direction (D) that he was going, the value of G is stepped up to bring the next gate closer, the screen is skipped up ten spaces, and the value of SK is reduced, which results in a slight increase in speed. The skier is moved down one line on the screen unless he is already three lines below the center. Moving him further down makes seeing what is coming very difficult, but if you'd like to try it, one way is to put a larger negative value here in place of the -3. If, for instance, you put a -10, the skier will move down every time you hit a jump. Another way would be to start the skier at a lower position on the screen. This would require simply changing the initial value of B in line 250.

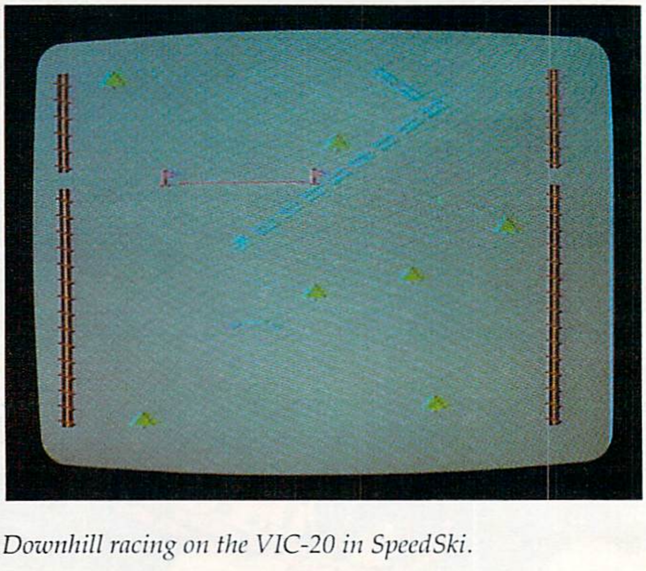
Line 570 checks to see if a fence section was hit. If so, it changes your direction and passes control to 610 for the sound effect. Getting out of the fence may take a couple of tries. If a tree was struck, then line 580 changes the figure to a cross and passes control to line 600. Line 590 POKEs S-3 to 0 in case it was set by hitting a flag in line 520, then passes you back to the main loop.

Shaking The Screen

Line 600 causes the screen to shake a bit when you hit a tree. The inner loop here counts from 3 to 7, then from 4 to 6, and stops at 5. POKEing these values into location 36864, which controls horizontal centering, shifts the screen rapidly back and forth around the normal value of 5. Line 610 increments OS, the number of objects that have been struck, and also controls both the sound effect and the changes in color of the cross in line 580. If a tree was struck, line 620 moves the skier up a line, adjusts the value of U, and checks to see if U has reached its limit of 10. If so, the run is aborted and you are given another chance. If not, line 630 passes control back to the main loop.

Line 640, the finish line sound effect, is reached only if the finish line was detected in line 510. Lines 650-660 print out the statistics on the run just completed and finish off the sound effect. Line 660 also POKEs 36869 back to its normal state so that the scores can be printed.

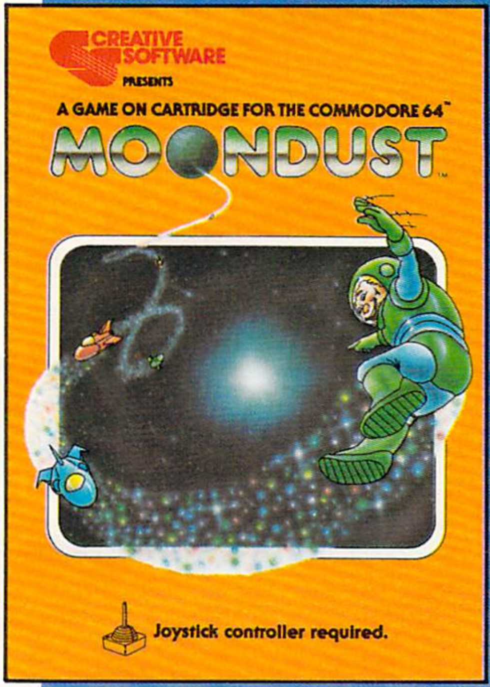
Line 670 computes the player's cumulative score, adding the score for the run just completed to his total from previous rounds, and also prints

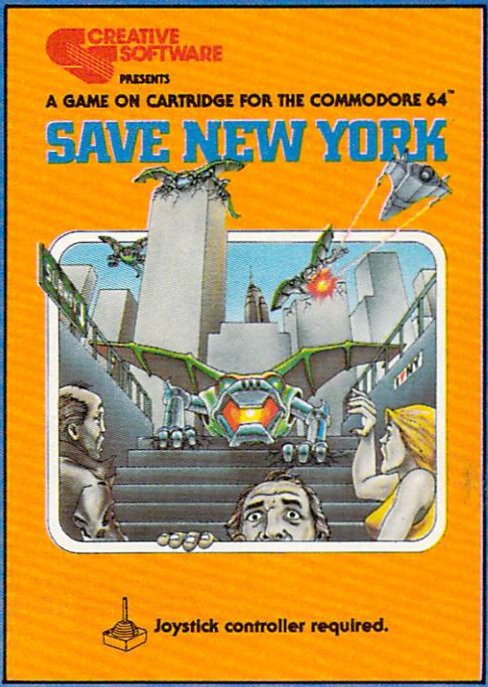


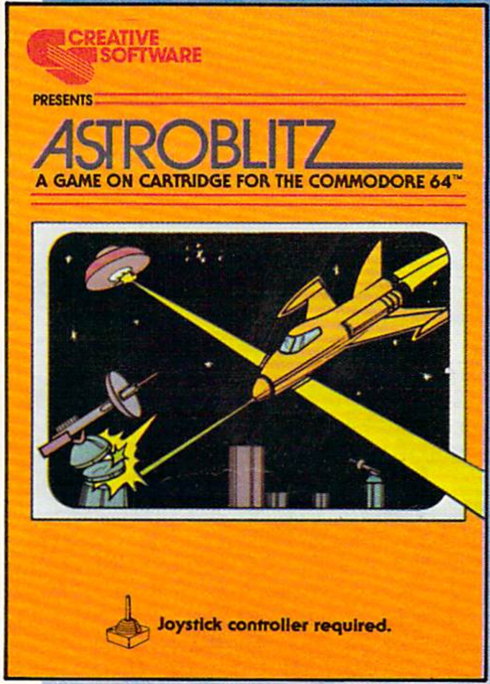
Downhill racing on the VIC-20 in SpeedSki.

CREATIVE SOFTWARE

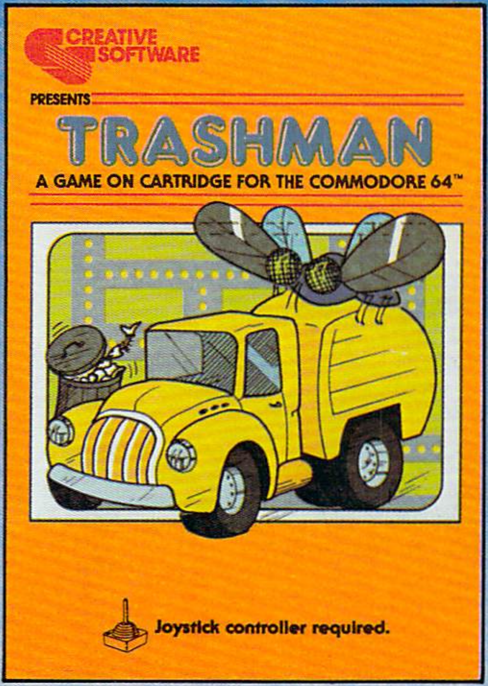
– the #1* independent VIC-20 full-line software publisher in the U.S. – is proud to announce 4 new Game Cartridges & 5 Home Applications for the COMMODORE 64.™





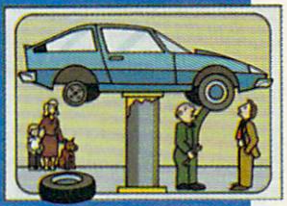


THIS GAME IS ALSO AVAILABLE FOR THE VIC-20.



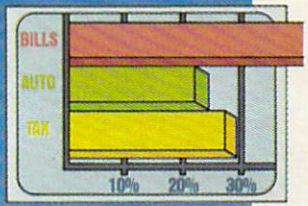
THIS GAME IS ALSO AVAILABLE FOR THE VIC-20.

CAR COSTS
A HOME APPLICATION PROGRAM

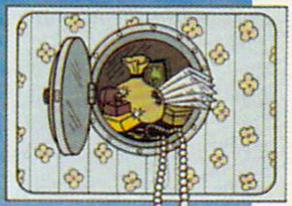


LOAN ANALYZER
A HOME APPLICATION PROGRAM

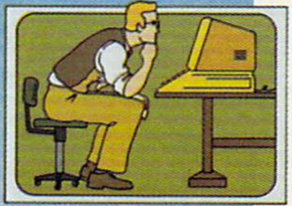
HOUSEHOLD FINANCE
A HOME APPLICATION PROGRAM



HOME INVENTORY
A HOME APPLICATION PROGRAM



DECISION MAKER
A HOME APPLICATION PROGRAM



These Home Application Programs are also available for the VIC-20.

CREATIVE SOFTWARE
A Division of ASCI, Inc.
230 East Caribbean Drive
Sunnyvale, CA 94086

*Based on survey of distributors and retailers.

the round number. Line 680 then prints the cumulative scores for all the players, and line 690 reinitializes for the next run.

Line 700 increments the player number; if the last player hasn't gone yet, control passes back to line 130 to start another run. If the last player has just gone, line 710 increments the round number and checks to see if the game is over. If not, the player number is changed to 1 and a new round is begun. Otherwise, line 720 lets you know the game is over. It then turns the cursor white.

To rerun the program, hit RETURN, then type RUN and hit RETURN again. The reason for this odd procedure: it isn't visible because it's white on a white background, but some garbage has been picked up during the run and lies on the same line as the cursor. During the program this garbage is disposed of by the loop that rejects bad input for the skill level. There is no such loop at the end of the program, though.

Okay, time to get the program typed in, then hit the slopes. There's a world record waiting to be broken. Good luck.

Variable Listing

NP	Number of players
NR	Number of rounds
R	Present round number
P	Present player number
S$	Slope title
SK	Time delay factor in main loop
RN	Controls probability of a tree being printed
S	Noise generator (36877)
L	Random variable used to position trees
B	Skier's location
C	Difference between screen map and color code map
F	Skier figure
TI$	System clock
D	Direction (1 or -1) to be added to skier's location
G	Counts spaces between gates and jumps
E	Counts gates
X	Random variable used to position gates and jumps
H	Counts gates hit
TM	Elapsed time for run
U	Controls vertical movement of skier on screen
OS	Counts number of trees and fence sections struck
SC	Player's score for a run
Z(P)	Player's cumulative score where P is the player number

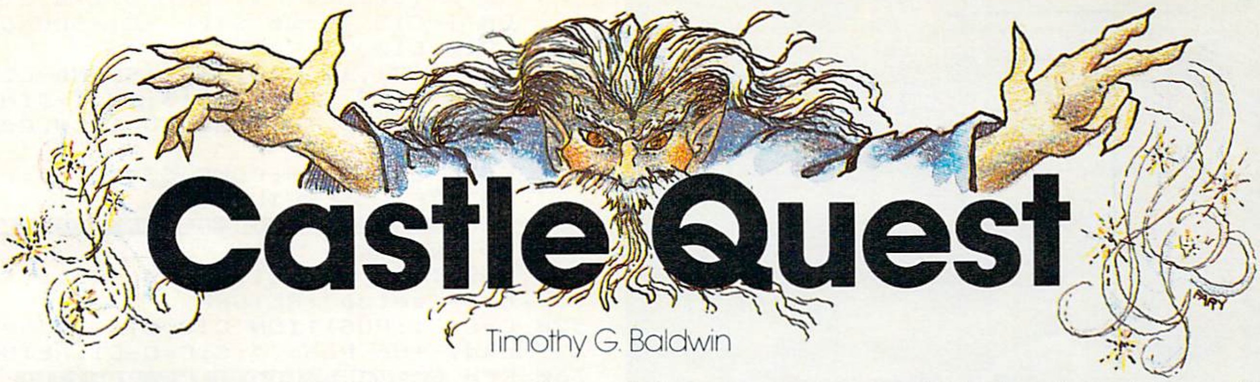
BEGINNING PROGRAMMERS

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

SpeedSki

```
10 PRINT"{CLR}{9 DOWN}{3 SPACES}SPEED-SK
   I":PRINT"{9 DOWN}"
20 POKE56,28:POKE55,250:POKE52,28:POKE51
   ,250:POKE36879,25
30 READX:IFX=0THEN70
40 FORI=XTOX+7:READY:POKEI,Y:NEXTI:GOTO3
   0
50 DATA7672,16,56,56,124,124,254,254,16
51 DATA7664,0,0,15,32,64,128,0,0
52 DATA7656,0,0,240,4,2,1,0,0
53 DATA7648,40,40,40,40,104,56,44,40
54 DATA7640,32,16,136,68,34,17,8,4
55 DATA7632,4,8,17,34,68,136,16,32
56 DATA7624,16,28,30,28,16,16,16,56
57 DATA7616,0,0,0,0,255,85,170,255
58 DATA7608,16,24,126,24,26,44,72,16
59 DATA7424,0,0,0,0,0,0,0,0
60 DATA7592,8,24,126,24,88,52,18,8
61 DATA7584,0,0,0,0,0,0,255,0
62 DATA7576,8,8,28,8,62,8,127,8
63 DATA7568,8,8,62,8,8,8,0,0,0
70 PRINT"{CLR}{BLK}{5 SPACES}SPEED-SKI":
   PRINT"{22 T}"
80 PRINT"{UP}{BLU}YOUR SCORE IS ELAPSED
   TIME + 5 FOR EACH GATEMISSED.
   {2 SPACES}LOWEST SCORE WINS."
90 PRINT"{DOWN}PRESS {RVS}<{OFF} TO GO L
   EFT{4 SPACES}AND {RVS}>{OFF}TO GO RIG
   HT."
100 INPUT"{DOWN}NO. PLAYERS (1-4)";NP:IF
    NP<1ORNP>4THEN70
110 INPUT"{DOWN}NO. ROUNDS{2 SPACES}(1-5
    )";NR:IFNR<1ORNR>5THEN70
120 R=1:P=1
130 PRINT"{DOWN}{RVS}{CYN}SKIER #";P:PRI
    NT"{DOWN}{BLU}SLOPE DESIRED":PRINT"1
    =BEGINNER":PRINT"2=INTERMEDIATE"
140 PRINT"3=ADVANCED":PRINT"4=OLYMPIC":P
    RINT"5=PROFESSIONAL"
150 A$="":GETA$:IFA$=""THEN150
160 IFA$="1"THENS$="{2 SPACES}BEGINNER"
170 IFA$="2"THENS$="INTERMEDIATE"
180 IFA$="3"THENS$="{2 SPACES}ADVANCED"
190 IFA$="4"THENS$="{2 SPACES}OLYMPIC"
200 IFA$="5"THENS$="PROFESSIONAL"
210 SK=VAL(A$):RN=(SK+1)/10:IFSK<1ORSK>5
    ORSK<>INT(SK)THENPRINT"{10 UP}":GOTO
    130
220 SK=35-5*SK
230 POKE36869,255:PRINT"{CLR}":POKE36878
    ,15:S=36877
240 FORI=1TO22:L=INT(RND(1)*19)+1:PRINT"
    {RED}<";TAB(L);"{GRN}?";TAB(20)"
    {RED}<":NEXTI
250 B=7910:C=30720:F=55:POKEB,F:POKEB+C,
    3:POKEB+22,32:POKEB+21,32:POKEB+23,3
    2
260 POKE8125,57:POKE8131,57:POKE8125+C,4
    :POKE8131+C,4
270 PRINT"{HOME}{8 DOWN}{4 SPACES}{RVS}"
    ;S$;"{13 DOWN}"
280 FORI=8126TO8130:POKEI,52:POKEI+C,4:N
    EXTI
290 FORI=1TO5:POKES-1,220+5*I:FORT=1TO10
    0:NEXTT:POKES-1,0:NEXTI
300 TI$="000000"
310 GETA$:IFA$=""THEN310
320 IFSKTHENFORT=1TOSK:NEXTT
330 IFF=55THENPOKEB-21,58:GOTO350
340 POKEB-23,59
350 IFPEEK(197)=29THEND=-1:F=55:POKES,24
    5:GOTO370
360 D=1:F=53:POKES,246
370 G=G+1:IFG<28THEN410
380 G=0:E=E+1:IFE=10THENPRINT"{PUR}98888
    8888888888888889":GOTO460
```

```
390 IFE>1ØTHENPOKEB,56:GOTO51Ø
400 X=INT(RND(1)*1Ø)+2:PRINTTAB(X)"{PUR}
    9444449":GOTO46Ø
410 IFG=1ØTHENX=INT(RND(1)*1Ø)+4:PRINT"
    {UP}{RED}<";TAB(X)"{CYN}>=";TAB(2Ø);
    "{RED}<"
420 IFRND(1)>RNTHEN44Ø
430 L=INT(RND(1)*19)+1:PRINT"{RED}<";TAB
    (L)"{GRN}?";TAB(2Ø)"{RED}<":GOTO45Ø
440 PRINT"{RED}<";TAB(2Ø);"<"
450 IFPEEK(B)<>32THEN5ØØ
460 POKEB,32:B=B+D:POKEB,F:POKEB+C,3
470 GOTO32Ø
480 END
500 IFPEEK(B)=52THENH=H+1:POKES-1,24Ø:FO
    RT=1TO3Ø:NEXTT:POKES-1,Ø:GOTO46Ø
510 IFPEEK(B)=56THENH=1Ø-H:TM=INT(TI/6Ø)
    :POKES-1,Ø:POKEB+D,F:GOTO64Ø
520 IFPEEK(B)=57THENPOKES-3,22Ø:GOTO57Ø
530 IFPEEK(B)<>62ANDPEEK(B)<>61THEN57Ø
540 POKES,253:D=D*2:G=G+1Ø:FORI=1TO1Ø:PR
    INT"{RED}<";TAB(2Ø)"{RED}<":NEXTI:IF
    SK>ØTHENSK=SK-2
550 IFU>-3THENB=B+22:U=U-1
560 GOTO46Ø
570 IFPEEK(B)=6ØTHENPOKEB,6Ø:D=D*-2:GOTO
    6ØØ
580 IFPEEK(B)=63THENPOKEB-22,5Ø:POKEB,51
    :GOTO6ØØ
590 POKES-3,Ø:GOTO46Ø
600 FORJ=2TOØSTEP-1:FORI=5-JTO5+J:POKE36
    864,I:NEXTI,J
610 OS=OS+1:FORT=ØTO127:POKES,255-T:POKE
    B-22+C,INT(T/22)+2:NEXTT:POKES-1,Ø
620 IFPEEK(B)=51THEND=-22:U=U+1:IFU=1ØTH
    ENPRINT"{RVS}{CLR}TRY AGAIN":POKE368
    69,24Ø:GOTO69Ø
630 GOTO46Ø
640 POKES,Ø:FORT=128TO255:POKES-3,T:NEXT
    T:POKES-3,Ø
650 U=Ø:PRINT"{CLR}{RVS}OBJECTS HIT=";OS
    :PRINT"{RVS}GATES MISSED=";H:PRINT"
    {RVS}TIME="TM:SC=TM+5*H
660 PRINT"{RVS}SCORE="SC:POKES-2,22Ø:FOR
    T=1TO1ØØ:NEXTT:POKES-2,Ø:POKE36869,2
    4Ø
670 Z(P)=Z(P)+SC:PRINT"{2 DOWN}
    {7 SPACES}{RVS}ROUND";R:PRINT" ":FOR
    I=1TONP
680 PRINT"{3 SPACES}SKIER #";I;Z(I):NEXT
    I
690 SC=Ø:G=Ø:E=Ø:OS=Ø:H=Ø:IFU=1ØTHENU=Ø:
    POKES,Ø:GOTO13Ø
700 P=P+1:IFP<NP+1THEN13Ø
710 R=R+1:IFR<NR+1THENP=1:GOTO13Ø
720 PRINT"{2 DOWN}{6 SPACES}{RVS}GAME OV
    ER":PRINT"{WHT}":END
```
©



Janice Fary

Castle Quest

Timothy G. Baldwin

This entrancing, well-designed game for any Atari offers you the best of both worlds. It has the drama, variety, and mystery of a good adventure game combined with the fast-paced excitement of an arcade game.

Your job is to rid the kingdom of the three evil wizards. All this would be easy if the wizards weren't so zealously guarded by servants whose names reflect their personalities: bat-wingers, blinkers, chokers, crushers, and stompers.

You are in love with the Princess Dilayna and have asked her father the King for her hand in marriage. Her father does not particularly like you. He challenges you to demonstrate your worthiness by capturing the three evil wizards that have been ravaging the kingdom for years. They each live in their own castle protected by their servants – the bat-wingers, the blinkers, the chokers, the stompers, and the crushers. The castle rooms are rumored to be deadly, with untouchable walls, fast-moving enemies, and no exits. You reluctantly accept the King's challenge.

Fortunately, a friendly magician gives you a cloak that makes its wearer invisible. But the cloak's power works only for a limited time in each room. Once the time is up, you are instantly destroyed. The magician also gives you a magic spell that temporarily freezes all servants in a room. But you must use this spell with care: it will consume a portion of the cloak's power each time it is used.

Armed with these aids, you leave on your quest. The King wishes you good luck – or did he say good riddance?

The Three Wizards

The object of "Castle Quest" is to capture the three wizards. To reach each wizard, you must pass through the ten rooms of his castle. The rooms are inhabited by the wizard's servants, who move about quickly in an unpredictable manner. The higher numbered rooms in each castle have more servants (up to 32). The servants move progressively faster as you complete more rooms.

You have three (3) lives to capture the first wizard. Capturing a wizard earns you three additional lives. Touching a servant or a room wall or failing to exit a room within the allotted time will cause loss of a life. You cannot exit a room until you capture both door keys in that room by touching them. One key is invisible until the other key is touched.

Once both keys are captured, the room's exit appears – unless you are in a castle's tenth room. In this case, the wizard appears, and you must capture him before you can escape. Also, once you capture the first key, your presence becomes known to the wizard, and he causes room wall segments to move to block your escape. You must move quickly to avoid destruction.

Secret Passages

A counter at the top of the screen signals the amount of "cloak time" remaining. Pressing the joystick fire button will temporarily freeze the action, permitting you to move safely past a tight corner, but you lose 50 units of cloak time each time you use the freeze option. The room number and the number of your remaining lives are displayed at the top left of the screen. Your score – a measure of your ability to elude the many dangers involved – is displayed at the top right of the screen.

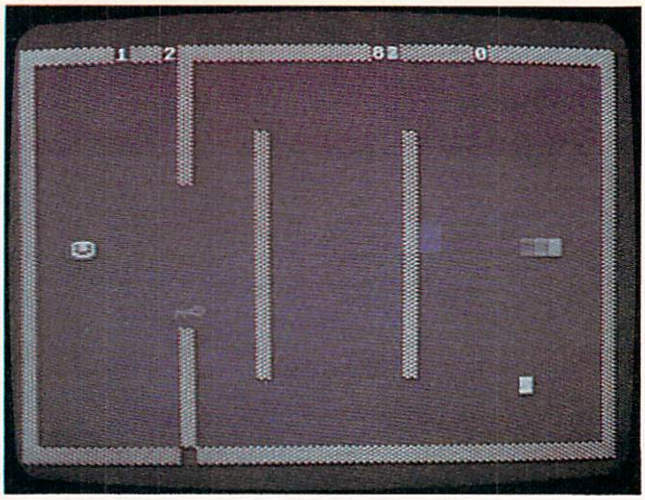
Room patterns, key locations, servant locations, and wizard placement are randomly generated, so be prepared to touch keys partially embedded in walls, move through weird mazes, etc. Sometimes a secret passageway is created at the screen bottom or in a room's right wall. You may use these passageways for a quick, easy escape.

I will make tape or disk copies for anyone who sends me a blank tape or disk, a stamped, self-addressed mailer, and $3.

T. G. Baldwin
Box 354, Route 2
Hayes, VA 23072

Castle Quest

```
10 REM {5 SPACES} MEMORY SAVER
   {14 SPACES}
20 C0=0:C1=1:C2=2:C3=3:C4=4:C5=5:C6=
   6:C7=7:C8=8:C9=9:C10=10:C15=15:C1
   6=16:C256=256:RAMTOP=PEEK(106):MI
   SSION=C1
30 REM  INITIALIZATION ROUTINE
   {7 SPACES}
```



Searching for the keys to the hidden door on Atari's Castle Quest.

```
40 GOSUB 1080:GOSUB 770:GRAPHICS C16
:? "{CLEAR}":POKE 752,C1:SETCOLOR
 C2,CØ,CØ:GOSUB 310
50 T1=C8:GOSUB 1150:T1=C16:GOSUB 115
Ø:G=CØ:L=C3:Q=CØ:C=CØ:X1=CØ:SCORE
=CØ
60 GOSUB 320
70 REM {4 SPACES}ROOM SETUP ROUTINE
{9 SPACES}
80 GOSUB 970:GOSUB 450:GOSUB 1340:GO
SUB 1500:POKE 1568,C1:POKE 77,Ø:P
OKE 53248,60:POKE 53249,W1
90 IF C=C10 THEN GOSUB 340
100 X=USR(1767):FOR I=CØ TO 100:NEXT
 I:POKE 1568,F
110 REM {6 SPACES}MAIN PROGRAM LOOP
{8 SPACES}
120 G=G-C1:IF (PEEK(1566)<>CØ) OR (G
<CØ) THEN 400
130 IF PEEK(203)>204 THEN 520
140 POSITION 23-(G>999)-(G>99)-(G>C9
),CØ:? CHR$(B);G;CHR$(B):IF G<10
Ø THEN SETCOLOR C2,C4,CØ
150 X=PEEK(53260):IF (X-X1)>=C2 THEN
 POKE 53250,W2:POKE 53249,CØ:IF
 PEEK(706)<>N THEN GOSUB 380:POKE
 706,N
160 IF X-X1>=C4 THEN POKE 53251,W3:P
OKE 53250,CØ
170 IF X>=C6 THEN GOSUB 260
180 IF STRIG(CØ)=CØ THEN POKE 1568,C
1:G=G-50:FOR I=Ø TO 250:NEXT I:P
OKE 1568,F
190 CHBASE=RAMTOP-C8-C8*(INT(G/2)=G/
2):POKE 756,CHBASE
200 IF PEEK(706)=N THEN IF RND(CØ)>Ø
.95 THEN PLOT INT(RND(CØ)*38),IN
T(RND(CØ)*22):GOSUB 240
210 IF STICK(CØ)<>15 THEN SOUND C2,1
ØØ,C6,C8:SOUND C2,CØ,CØ,CØ
220 GOTO 120
230 REM {3 SPACES}"SHOOTING" SOUND R
OUTINE{3 SPACES}
240 FOR I=CØ TO 30:SOUND CØ,I,CØ,C15
:NEXT I:SOUND CØ,CØ,CØ,CØ:RETURN
250 REM  ROOM EXIT OPENING ROUTINE
{4 SPACES}
260 IF C=C10 THEN IF X<>14 THEN RETU
RN
270 FOR I=CØ TO C5:POKE SC+C10*40+I*
40-C1,CØ:NEXT I:POKE 53278,255:F
OR I=C15 TO CØ STEP -C1:SOUND CØ
,C10,C10,I
280 SOUND C1,11,C10,I+C1:SOUND C2,12
,C10,I+C2:SOUND 3,13,10,I+3:NEXT
 I:FOR I=Ø TO 3:SOUND I,CØ,CØ,CØ
:NEXT I
290 POKE 53251,CØ:POKE 53250,CØ:POKE
 53278,255:RETURN
300 REM  USER INFORMATION ROUTINES
{3 SPACES}
310 POSITION C10+C1,C10:? "Wait for
game setup":RETURN
320 C=C+C1:POSITION C10,C10:? "Get r
eady for Room ";C:C=C-C1:RETURN
330 REM  WIZARD PLOTTING ROUTINE
{6 SPACES}
340 PL=(RAMTOP-9)*256:PL=PL+52+INT(R
ND(CØ)*151):RESTORE 350:FOR I=CØ
 TO 11:READ Z:POKE PL+I,Z:NEXT I
350 DATA 102,36,126,90,126,126,66,90
,60,60,36,102
360 W3=70+INT(RND(CØ)*130):POKE 707,
P:RETURN
370 REM  "KEY TOUCHING" SOUND ROUTIN
E
380 SOUND C2,20,C10,C10:SOUND C1,80,
C10,C10:FOR I=Ø TO 30:NEXT I:SOU
ND C1,CØ,CØ,CØ:SOUND C2,CØ,CØ,CØ
:RETURN
390 REM  USER FAILS TO ESCAPE ROOM
{3 SPACES}{8 SPACES}  ROUTINE
{21 SPACES}
400 FOR I=CØ TO C3:POKE 53248+I,C1:N
EXT I:POKE 1568,C1:? "{CLEAR}":S
ETCOLOR C2,CØ,CØ:IF Q THEN RETUR
N
410 POKE DL+C15,C7:POSITION C4,C10:I
F Q THEN RETURN
420 POKE 756,224:? "TOUGH LUCK!":FOR
 I=CØ TO 200:SOUND CØ,C6,100,C8:
NEXT I:SOUND CØ,CØ,CØ,CØ:T2=C1
430 POKE DL+C15,C2:L=L-C1:? "{CLEAR}
":C=C-1:GOSUB 320:C=C+1:GOTO 80+
500*(L<=CØ)
440 REM DETERMINE NEXT ROOM'S
{9 SPACES}{8 SPACES}CHARACTERIST
ICS ROUTINE{7 SPACES}
450 A=INT(C16*RND(CØ))*C16+C6:M=INT(
C16*RND(CØ))*C16+C2:N=INT(C16*RN
D(CØ))*C16+C4:P=INT(C16*RND(CØ))
*C16+C8
460 B=33+C-C6*(C>5):C=C+C1:D=C2+C2*(
C>C1)+C4*(C>C3)+C8*(C>C6)+C16*(C
>C9)
470 E=INT(RND(Ø)*5+7):POKE 1763,E
480 F=C2+(C>C9)+C2*(MISSION-C1)
490 G=100+C*50:COLOR B:POKE 1578,31:
POKE 1566,CØ:POKE 756,RAMTOP-C8:
POKE 53278,255:X1=CØ
500 SETCOLOR 2,C7*(C=7)+C2*(C=8)+C1*
(C=9)+C3*(C=10),CØ:RETURN
510 REM  USER ESCAPES FROM A ROOM
{5 SPACES}{8 SPACES}  ROUTINE
{22 SPACES}
520 Q=C1:GOSUB 400:GOSUB 410:POKE 75
6,224:? "{3 SPACES}ATTABOY!":Q=C
Ø
530 FOR I=CØ TO C5:SOUND CØ,C10,50,C
8:POKE 705,C10:POKE 706,C10:POKE
 710,C10:POKE 712,C10:FOR J=CØ T
O 50:NEXT J
540 SOUND CØ,C10,100,C8:POKE 705,CØ:
POKE 706,CØ:POKE 710,CØ:POKE 712
```

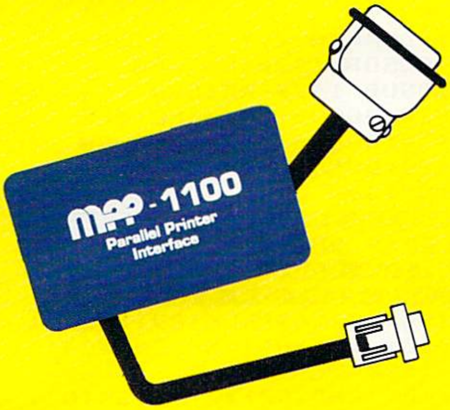
Stake a Claim on the most exciting New Game for your Atari Home Computer.
MINER 2049ER
STARRING BOUNTY BOB
BY BILL HOGUE
• From the programming team that brought you "Robot Attack", "Defense Command" and many other great Arcade games for your TRS-80™
• 100% machine language
• 16K ROM Cartridge, the largest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

```
,CØ:FOR J=CØ TO 5Ø:NEXT J:NEXT I
55Ø SOUND CØ,CØ,CØ,CØ:POKE DL+C15,C2
    :? "{CLEAR}":GOSUB 32Ø:SCORE=SCO
    RE+MISSION*INT((G*C)/C1Ø)
56Ø IF C=C1Ø THEN GOTO 58Ø+11Ø*(MISS
    ION=C3)
57Ø GOTO 8Ø
58Ø REM {3 SPACES}END A QUEST ROUTIN
    E{8 SPACES}
59Ø ? "{CLEAR}":POKE DL+C9,C6:POKE D
    L+11,C6:POKE DL+13,C6:POKE DL+15
    ,C6:POKE 7Ø7,CØ:IF L<=CØ THEN 66
    Ø
6ØØ POSITION C2,C4:? "congratulation
    s!":POSITION 26,C5:? "YOU HAVE":
    POSITION C3,C7:? "COMPLETED YOUR
    "
61Ø POSITION 27,C8:? "QUEST":C=CØ:L=
    L+C3
62Ø POSITION C5,15:? "Press START t
    o continue":POSITION C5,17:? "Pr
    ess SYSTEM RESET to quit"
63Ø POSITION C5,19:? "SCORE: ";SCORE
64Ø POKE 53279,C8:IF PEEK(53279)<>C6
     THEN 64Ø
65Ø ? "{CLEAR}":POKE DL+C9,C2:POKE D
    L+11,C2:POKE DL+13,C2:POKE DL+15
    ,C2:MISSION=MISSION+(L>CØ)*C1:GO
    TO 6Ø+62Ø*(L<=CØ)
66Ø POSITION C7,C4:? "SORRY!":POSITI
    ON 24,C5:? "you blew it.":POSITI
    ON C2,C7:? "quests completed ";M
    ISSION-C1
67Ø GOTO 62Ø
68Ø RUN
69Ø REM USER WINS THE GAME ROUTINE!
    {3 SPACES}
7ØØ GRAPHICS 2:SETCOLOR C2,CØ,CØ:POS
    ITION C6,C4:? #6;"YOU WON!":? "P
    ress SYSTEM RESET and then 'RUN'
     to";
71Ø POKE 752,1:? :? "begin a new gam
    e."
72Ø POSITION C1,C7:? #6;"final score
     ";SCORE
73Ø FOR I=255 TO CØ STEP -C1:SOUND C
    Ø,I,1Ø,1Ø:POKE 712,I:POKE 71Ø,I:
    NEXT I
74Ø GOTO 74Ø
75Ø POKE 1568,C1:RUN
76Ø REM PUT A VERTICAL BLANK INTERRU
    PT{8 SPACES} ROUTINE IN PAGE 6 O
    F MEMORY
77Ø RESTORE 79Ø:FOR I=1536 TO 1536+2
    47:READ A:POKE I,A:NEXT I
78Ø RETURN
79Ø DATA 173,4,2Ø8,2Ø1,4,24Ø,2,2Ø8,2
    2,173,99,228,141,36,2
8ØØ DATA 173,1ØØ,228,141,37,2,141,3Ø
    ,6,141,3Ø,2Ø8,76,98,228
81Ø DATA Ø,162,2,2Ø2,24Ø,42,138,72,1
    73,1Ø,21Ø,41,7,1Ø,17Ø
82Ø DATA 189,Ø,1,133,2Ø6,133,2Ø8,232
    ,189,Ø,1,133,2Ø7,133,2Ø9
83Ø DATA 32,148,6,165,2Ø7,157,Ø,1,2Ø
    2,165,2Ø6,157,Ø,1,1Ø4
84Ø DATA 17Ø,2Ø8,211,162,5,173,12Ø,2
    ,2Ø2,24Ø,197,24,1Ø6,176,249
85Ø DATA 72,224,2,24Ø,8,224,1,2Ø8,13
    ,23Ø,2Ø3,2Ø8,2,198,2Ø3
86Ø DATA 165,2Ø3,141,Ø,2Ø8,2Ø8,32,16
    9,Ø,224,4,24Ø,8,168,145
87Ø DATA 2Ø4,23Ø,2Ø4,76,134,6,16Ø,7,
    145,2Ø4,198,2Ø4,16Ø,Ø,185
88Ø DATA 24Ø,6,145,2Ø4,2ØØ,192,8,2Ø8
    ,246,1Ø4,76,83,6,16Ø,Ø
89Ø DATA 152,145,2Ø6,173,1Ø,21Ø,41,1
    ,2Ø8,15,169,56,141,2Ø1,6
9ØØ DATA 169,233,141,2Ø4,6,141,21Ø,6
    ,2Ø8,13,169,24,141,2Ø1,6
91Ø DATA 169,1Ø5,141,2Ø4,6,141,21Ø,6
    ,173,1Ø,21Ø,41,1,2Ø8,2
92Ø DATA 169,4Ø,141,2Ø5,6,216,Ø,165,
    2Ø6,Ø,Ø,133,2Ø6,165,2Ø7,Ø
93Ø DATA Ø,133,2Ø7,177,2Ø6,24Ø,8,165
    ,2Ø8,133,2Ø6,165,2Ø9,133,2Ø7
94Ø DATA 169,11,145,2Ø6,96,1Ø4,168,1
    62,6,169,7,76,92,228,6Ø
95Ø DATA 126,9Ø,126,9Ø,1Ø2,126,6Ø
96Ø REM SETUP PLAYER-MISSILE GRAPH-
    {9 SPACES}ICS ROUTINE
    {18 SPACES}
97Ø POKE 559,62:POKE 54279,RAMTOP-C1
    6:POKE 53248,C1:POKE 53277,C3
98Ø PL=RAMTOP-12:Y=PEEK(88):Z=PEEK(8
    9):POKE 88,CØ:POKE 89,PL:POKE 1Ø
    6,PL+C3:? "{CLEAR}":POKE 88,Y:PO
    KE 89,Z
99Ø POKE 1Ø6,PL+12:PL=PL*C256+12Ø:IF
     C=CØ OR C=C1Ø THEN Z=(RAMTOP-C9
    )*C256:FOR I=Z TO Z+255:POKE I,C
    Ø:NEXT I
1ØØØ FOR I=CØ TO C7:POKE PL+I,PEEK(1
     776+I):NEXT I
1Ø1Ø POKE 2Ø3,6Ø:POKE 2Ø4,PL-INT(PL/
     C256)*C256:POKE 2Ø5,INT(PL/C256
     )
1Ø2Ø PL=(RAMTOP-11)*C256:PL=PL+52+IN
     T(RND(CØ)*151):RESTORE 1Ø3Ø:FOR
      I=CØ TO C7:READ Z:POKE PL+I,Z:
     NEXT I
1Ø3Ø DATA Ø,6,15,249,255,166,16Ø,Ø
1Ø4Ø W1=7Ø+INT(RND(CØ)*13Ø):PL=(RAMT
     OP-C1Ø)*C256:PL=PL+52+INT(RND(C
     Ø)*151):RESTORE 1Ø3Ø:FOR I=CØ T
     O C7
1Ø5Ø READ Z:POKE PL+I,Z:NEXT I:W2=7Ø
     +INT(RND(CØ)*13Ø):POKE 7Ø5,M:IF
      T2=C1 THEN C=C-C1:T2=CØ
1Ø6Ø POKE 53249,CØ:POKE 5325Ø,CØ:RET
     URN
1Ø7Ø REM {4 SPACES}TITLE PAGE ROUTIN
     E{7 SPACES}
1Ø8Ø GRAPHICS 18:SETCOLOR C2,CØ,CØ:P
     OKE 7Ø8,2Ø2:POSITION C5,C2:? #C
     6;"CASTLE":POSITION C9,C4:? #C6
     ;"QUEST"
1Ø9Ø DL=PEEK(56Ø)+C256*PEEK(561):POK
     E DL+13,C2
11ØØ POSITION C3,C8:? #C6;"How many
     rooms can you survive?"
111Ø FOR I=CØ TO C3:POKE 7Ø8,CØ:SOUN
     D CØ,6Ø,C1Ø,C8:FOR J=CØ TO 1ØØ:
     NEXT J:SOUND CØ,16Ø,C1Ø,C8:POKE
      7Ø8,2Ø2
112Ø FOR J=CØ TO 1ØØ:NEXT J:NEXT I
113Ø SOUND CØ,CØ,CØ,CØ:RETURN
114Ø REM SETUP SPECIAL CHARACTER SET
     S {9 SPACES}ROUTINE{22 SPACES}
115Ø RESTORE 116Ø:CL=(RAMTOP-T1)*C25
     6:FOR I=CL+C8 TO CL+95:READ A:P
     OKE I,A:NEXT I
116Ø DATA 2Ø4,51,2Ø4,51,2Ø4,51,2Ø4,5
     1,1Ø2,153,1Ø2,153,1Ø2,153,1Ø2,1
     53
117Ø DATA 136,34,136,34,136,34,136,3
```

Let your Atari experience the

MPP CONNECTION!

MPP-1100 Parallel Printer Interface

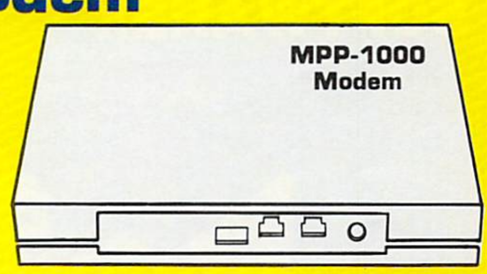


- No Atari 850™ Interface Module needed.
- Compatible with all software (including Visicalc™, Text Wizard™, and Filemanager 800™, etc.).
- 5 foot cable with Centronic plug (compatible with Epson, NEC, IDS, etc.)
- Faster data transfer.
- 8 bit data transfer.
- 2 year warranty.

only $99.95

MICROBITS PERIPHERAL PRODUCTS

MPP-1000 Modem



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

Smart Terminal Features:

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

only $199.00

MICROBITS PERIPHERAL PRODUCTS

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

```
     4,68,17,68,17,68,17,68,17
1180 DATA 36,146,73,36,146,73,36,146
     ,255,255,255,255,255,255,255,25
     5
1190 DATA 195,102,60,24,24,Ø,Ø,Ø
1200 DATA 255,255,195,195,195,195,25
     5,255
1210 DATA 255,255,Ø,Ø,Ø,Ø,255,255
1220 DATA 24,24,60,24,255,199,199,25
     5
1230 DATA 24,255,Ø,Ø,Ø,Ø,Ø,Ø
1240 FOR I=128 TO 224:POKE CL+I,PEEK
     (57344+I):NEXT I
1250 DL=PEEK(560)+C256*PEEK(561):IF
     T1=C16 THEN RESTORE 1260:FOR I=
     CL+56 TO CL+95:READ A:POKE I,A:
     NEXT I
1260 DATA Ø,Ø,Ø,24,24,60,102,195
1270 DATA Ø,Ø,60,60,60,60,Ø,Ø
1280 DATA Ø,Ø,255,255,255,255,Ø,Ø
1290 DATA 60,24,24,24,60,60,Ø,Ø
1300 DATA 24,24,24,24,24,24,24,255
1310 IF T1=C16 THEN FOR I=CL TO CL+C
     7:POKE I,CØ:NEXT I
1320 RETURN
1330 REM     RANDOM ROOM MAZE GENERATO
     R {9 SPACES}     ROUTINE
     {20 SPACES}
1340 ? "{CLEAR}":POKE 752,C1
1350 PLOT CØ,CØ:DRAWTO 39,CØ:DRAWTO
     39,23:DRAWTO CØ,23:DRAWTO CØ,CØ
1360 X=C1Ø:Y=CØ:Z=C7:GOSUB 1400:X=C1
     5:Y=C5:Z=13:GOSUB 1400:X=C1Ø:Y=
     C16:Z=C7:GOSUB 1400
1370 IF RND(CØ)<Ø.5 THEN PLOT RND(CØ
     )*31+C8,11:DRAWTO RND(CØ)*31+C8
     ,11
1380 POSITION C6,CØ:? C:POKE 704,A:P
     OKE 705,M
1390 POSITION C9,CØ:? L:POSITION 3Ø,
     CØ:? SCORE:RETURN
1400 ON INT(RND(CØ)*C8+C1) GOSUB 141
     Ø,1420,143Ø,1440,145Ø,1460,147Ø
     ,148Ø
1410 RETURN
1420 PLOT X,Y:DRAWTO X,Y+Z:RETURN
1430 X=X+C1Ø:GOSUB 1420:RETURN
1440 X=X+2Ø:GOSUB 1420:RETURN
1450 GOSUB 1420:GOSUB 143Ø:RETURN
1460 GOSUB 143Ø:GOSUB 143Ø:RETURN
1470 GOSUB 1420:GOSUB 1460:RETURN
1480 POP :GOTO 1360
1490 REM WIZARD'S SERVANTS PLOTTING
     {3 SPACES}{9 SPACES}AND ADDRESS
      CALCULATION ROUT-{9 SPACES}INE
     . ADDRESSES KEPT IN STACK
1500 SC=PEEK(88)+C256*PEEK(89):FOR I
     =CØ TO D-C1:IF INT(RND(CØ)*C4)>
     C2 THEN 152Ø
1510 H=SC+4Ø+INT(RND(CØ)*279):GOTO 1
     53Ø
1520 H=SC+68Ø+INT(RND(CØ)*239)
1530 HI=INT(H/C256):LO=H-HI*C256:POK
     E C256+I*C2,LO:POKE H,E
1540 POKE C256+I*C2+C1,HI:NEXT I:IF
     D=32 THEN RETURN
1550 FOR I=(D-C1) TO 31:POKE C256+I*
     C2+C1,254:NEXT I:RETURN
```
©

FOR THE WONDERFUL WORLD OF ATARI

THE LOVE AFFAIR BEGINS . . .

THE COMMANDER 2400 SERIES: 2400 with KEYPAD, 2400 STANDARD AND 2400 PROFESSIONAL NUMERIC KEYPAD.

WITH YOUR FIRST TOUCH

As your fingers caress its sculptured full stroke keys and . . . you realize you can install it **YOURSELF** . . . in just minutes . . . without any soldering!

. . . IT GROWS as you experience . . . the responsiveness of superbly crafted engineering under your fingertips . . . the convenience of your own **PERSONAL** and detachable system designed to allow use of **BOTH** keyboards . . . the warmth, beauty and elegance of solid black walnut woods . . . the softness, luxuriousness, and durability of fine textured furniture vinyl in a magnificent enclosure designed by NIGHTSTAR . . . the totally unique keypad circuit that allows rapid calculator function by automatically entering basic print statements for you.

. . . AND IT CULMINATES

In a lasting happy relationship with your own personal **COMMANDER 2400.** A design engineered to return the thrill of personal command to computing!

A SUPERB ACHIEVMENT . . and

AVAILABLE NOW AT INTRODUCTORY PRICING

The **ULTIMATE** STEP UP FOR THE WONDERFUL **ATARI!!!**
Special Introductory Pricing $116 to $199. Regular Suggested List $139 to $229*
CALL (503)479-4711 OR WRITE FOR FREE BROCHURE
TO ORDER: TOLL FREE 800-547-2492
*Depending on Options - 10 Day Money Back Guarantee

Ralston Clearwaters Electronics
536 N.E. 'E' Street
Grants Pass, Oregon 97526
VISA / MASTERCARD WELCOME

RCE

RCE **INVITES YOU TO SAVE UP TO 40%**

COMPARE : OUR PRICES • OUR SERVICE

RALSTON CLEARWATERS ELECTRONICS
536 N.E. 'E' STREET • GRANTS PASS, OR 97526
FOR CATALOG WRITE OR CALL: (503) 479-4711
FOR CUSTOMER SERVICE CALL: (503) 479-4711 or 479-4150

THEN: ORDER TOLL-FREE 800-547-2492
IN OREGON CALL: (503) 479-4711

WE ARE A WARRANTY STATION FOR ATARI, FOURTH DIMENSION, MICRO SCI, SANYO, PANASONIC, & FRANKLIN.

SPECIAL
70 Column Graphic Word Processor Package-64K Computer, Drive w/ Apple Dos, 12" Zenith Green Screen, & Gemini 10 Printer. Send for details - $1799 w/o Printer - $1399
128K Computer, Drive w/ Apple Dos, CPM, Parallel & Serial Port, & 80 Columns - $1849
Commodore 64 - $459 Drive - $359

NOW NEW & EVEN LOWER PRICES - PLUS TAKE 50% OFF SHIPPING CHARGE FOR PREPAYMENT

ATARI

		LIST PRICE	OUR PRICE
	400 (16K)	~~$ OPEN~~	$219
	800 (48K)	~~$679~~	$498
	1200 (64K)	~~$899~~	$679
EPSON with GRAFTRAX	MX 80	~~$645~~	$439
	FX 80	~~$695~~	$598
	MX 100	~~$995~~	$689

Daiseywriter with 48K Buffer
Letter Quality Printer $1249
Apple Compatible Disk Drives - 1 year warranty
Fourth Dimension $269
Controller w/Apple 3.3 DOS $ 89

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

TERMS:
SHIPPING: Add 6% of total transaction for UPS brown (ground) or 9% for UPS blue (air), Parcel Post, or any special arrangements. Minimum shipping charge - $6.00.
PAYMENT: Cashier's checks, certified checks, money orders, and bank wires honored immediately. Visa & Master Charge accepted. Allow 20 days for personal checks to clear.
REFUNDS: 10% restocking charge on all returns or exchanges. No refunds on opened software. Call first.
GUARANTEE: All products with full manufacturer's warranty. Sanyo and Apple warranty available. We have full repair and service facilities for all electronic repairs with HP, Dynascan, Pioneer, Sanyo and Apple trained and certified technicians. For any technical service call them for instant advice or questions right on their benches at (503)479-4150.
REPAIRS: Call for details on quality guaranteed discount repair and reconditioning service.
We have been repairing electronic equipment for 12 years and love it!

683

ATARI HARDWARE	
400 16K Computer	$219
800 48K Computer	$498
CX-853 16K Ram Cartidge	79
410 Progam Recorder	75
810 Disk Drive	429
850 Interface Module	165
830 Modem	149
1020-40 Col Color Printer	239
1025-80 Col Impact Printer	439
1010 Progam Recorder	75
CX-30 Game Paddles	19
CX-40 Joystick	10
CX-40-4 Joysticks Pair	19
Pointmaster Stick	14
Epson Printer Cable	29
Video & Audio Cable	27
LeStick RAF Style Joystick	36
Micro Tek 32K Ram Card	99
Full-View 80-Column Card	299
PERCOM	
Atari Single Density	$449
Atari Double Density	599
APPLE HARDWARE	
MICRO SCI	
A2	$299
A40	339
A70	459
Controller A2-3.2,3.3 & self check	79
A40 or A70	89
RANA	
Elite I	$309
Elite II	499
Controller: Four Drives	109
INTERFACE CARDS	
16K RAM CARDS	
ALS	$ 89
MPC AP-16	149
Microtek BAM-1	99
Microsoft	79
Promethius	139

80 COLUMN CARDS	
M & R Sup-r-term	$249
Videx	239
ALS Smart-term	259
View Max 80	169
MULTI-FUNCTION CARDS	
Mountain CPS	$139
Promethius	149
MISCELLANEOUS	
Alien Voice Box:	
with Rom	$149
without Rom	99
Cool Stack w/fan	69
Expand-A-Port TG	45
Versawriter	289
Gibson LPS II Light Pen	269
Joysticks	
Muse TG	45
Kensington System Saver	69
Keypad ABT 15 key	129
Paddles	
Muse TG	29
Sup-r-fan	36
Universal Modulator	59
MISCELLANEOUS CARDS	
CCS:	
Math Proc. 7811	$339
Ser/Int. 7710-A	129
Ser/Int. 7710-D	129
Timer 7740	109
Cent/Par 7728	119
MPC:	
Par/Prtr AP-80	79
32K Ram AP-32	179
Microsoft:	
Premium Pak	499
Z-80 Card	239
Mountain Computer	
A/D Converter	269
Clock/Calendar	239
Exp Chassis	569

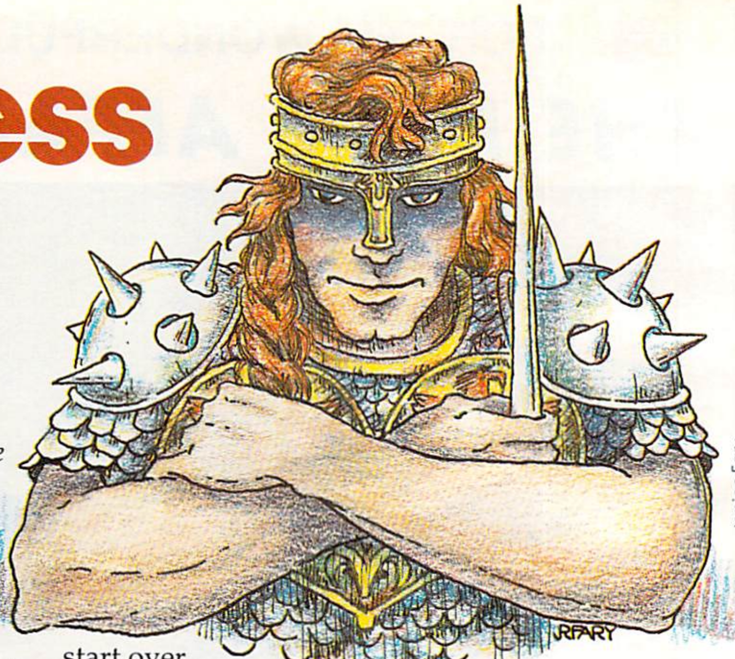
Introl/BSRX-10	179
Kbd Filter Rom	49
Music System	309
Romplus	159
Wild Card	115
Superclock	139
Romwriter	159
Supertalker	159
Videx	
Function Strip	69
Enhancer II	119
Soft Switch	27
MONITORS	
GREEN SCREEN	
BMC 12 inch	89
Zenith 12 inch	99
JC5 12 inch	145
Sanyo 12 inch	205
BLACK & WHITE	
Sanyo 9 inch	145
Sanyo12 inch	189
AMBER SCREEN	
JCS	$159
USI Pi3	169
Amdek	179
COLOR	
Amdek Color I	$349
Sanyo 13 inch	385
Taxan - RGB Vision I	319
Taxan - RGB Bision III	539
RGB - II Card for Apple	89
MODEMS	
Anchor Mark I	79
HAYES	
Chronograph	$195
Micromodem	269
Smartmodem	209
1200 Baud Smartmodem	529
NOVATION	
Apple Cat II	$289
Auto Cat	209

Cat	144
D Cat	159
212 Apple Cat	649
PRINTERS	
IDS	
Micro Prism	$599
Prism 80 w/out Color	849
With Color!!!	1329
Prism 132	1549
OKIDATA	
80	$330
82A	419
83A	699
84 Parallel	1069
92 Parallel	529
93 Parallel	949
STAR MICRONICS	
Gemini 10	$ CALL
Gemini 15	$ CALL
BROTHER	
Letter Quality Serial	$899
Letter Quality Parallel	799
SMITH CORONA	
Letter Quality	$599
INTERFACES	
Parl. & Cable	$ 69
Apple Dumpling	129
Grappler+	139
MicroBuffer 16K	209
MicroBuffer 32K	229

ALL BRAND NAMES ARE REGISTERED TRADE MARKS

NO SALES TAX IN OREGON! LIKE OUR PRICES?
SEND FOR OUR CATALOG CALL FOR COMPLETE LINE OF SOFTWARE

The Fortress Of Adnil



Janice Fary

George W. Miller

"The Fortress of Adnil" is an adventure game for the Timex/Sinclair 1000 using the graphics mode. The program is entirely in BASIC and includes several routines you'll find useful in other programs. It requires the 16K RAM expansion module.

Your objective in this game is to accumulate points by gathering energy pellets and recovering the treasure, while avoiding obstacles on the display. Each move costs you one unit of energy; using the laser sword costs more, depending on the range and the object you use it on.

Each move you make is accompanied by random placement of "NAWS" (defined as guards) on the screen. If this random placement puts a guard in the space you intend to occupy, you are captured. If your energy level is greater than 1000, the computer will allow you to pay a ransom, deduct the ransom from your score, and allow you to continue.

You can use your laser sword to cut a hole through any barricade and to oppose the guards. But be warned: the odds are even in any battle with the guards, and you may lose.

Since this game is written in BASIC, don't expect fast-paced action. The game began as a learning exercise in PEEKing and POKEing into the display file.

PEEK And POKE Programming

Enter lines 50 to 120 into your computer. They will print a border around the display and will provide a boundary limiting later POKE commands to the display file.

Now enter lines 220, 250 - 355, 510, and this line:

```
520 PRINT AT 21,0;PEEK (PEEK 16396 + PEEK
    16397*256 + S)
```

Then enter lines 530 and 610.

This will allow you to move a character around the screen and also find the CODE of the character stored in that address.

SAVE this before you try it, because if you POKE outside the display file, the program will crash. You'll have to turn off the power and start over.

Now add the missing lines to print a variety of characters on the screen, but don't change line 520 yet. Move around the screen and look at the codes returned from the different locations.

Lines 400 to 430 limit the movement to areas in the display file. Lines 450 to 500 check for the code at that address.

If this is new to you, just remember that POKEing is putting a value into an address, and PEEKing is looking at the value in an address.

Now change line 520 to the line as shown in the program listing. Enter the rest of the listing, and you'll have the complete game The Fortress of Adnil.

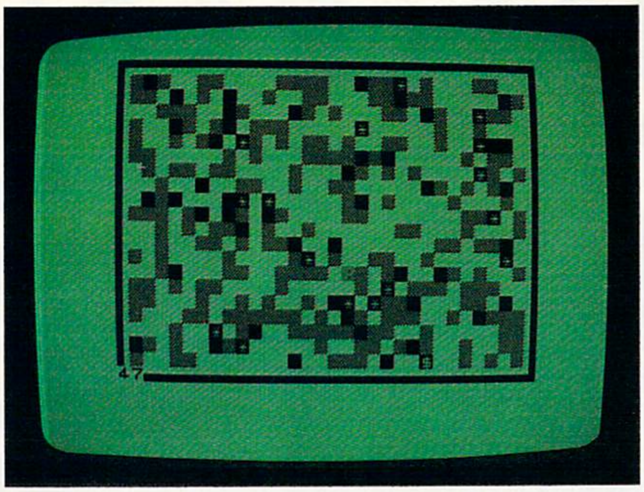
If you find the game too challenging, you can change the level of difficulty by changing the 2000 in line 525 to some lower value.

Line 9999 is a utility routine I use to keep track of the length of my program. Enter GOTO 9999, and the screen should say, "LENGTH OF PROGRAM 7456". Since I'm using a 16K RAM, with 16,384 bytes available, I'm well within the limits of memory. Note that line 9999 indicates all memory used, and includes the memory required for the variables, the display file, and the program.

Programming Hints

Now for some hints on making your programs look a little more professional.

The routine starting at line 9991 is self-starting. To SAVE the program, start your tape recorder and enter GOTO 9991. When you load the program again, you won't get the usual 0/0 display, but the program will begin to run, printing the title on the screen. To use this routine, change the program name in line 9995 to the name of your program, and the line number for the GOTO command to the first line in your program.



The Fortress of Adnil – an adventure game for the Timex/ Sinclair 1000

This is especially useful in working with files. You can store the data in variables, and when the program comes up it automatically begins, preserving your data, and going a long way towards making the program user friendly.

I have placed this function in the menu of my program "ZX-81/TS-1000 Data Management" (**COMPUTE!**, March 1983) and saved the data by a step in the program. This makes it a subroutine and it becomes very easy for even the most inexperienced user to save and run the program correctly.

A further step in making other programs user friendly is using INKEY$ instead of INPUT whenever possible. This keeps control of the program in the computer, and the computer will wait for the command it wants to see. (See lines 8020 to 8040.) INPUT permits any number of possible incorrect (or program-stopping) entries.

Attractive Displays

In some versions of BASIC, the command FLASH will cause the display to print normal and inverse characters. The Sinclair computer doesn't have this function, but you can get the same result by a routine similar to lines 8203 to 8205 in the listing. This makes your display a little more attractive, and adds a professional touch to your programs.

When building a display, make use of the graphic mode and the various commands for printing, such as TAB and PRINT AT.

Check each line by entering a GOTO command in the immediate mode after entering the line. If you don't like what you see on the display, press EDIT.

The upper portion of the display will remain unchanged, but the bottom part of the screen will now display your last line entered. Use your edit functions to move the cursor about the line and make any necessary changes. Hit enter, and execute another GOTO command in the immediate mode to recheck your work.

BEGINNING PROGRAMMERS

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

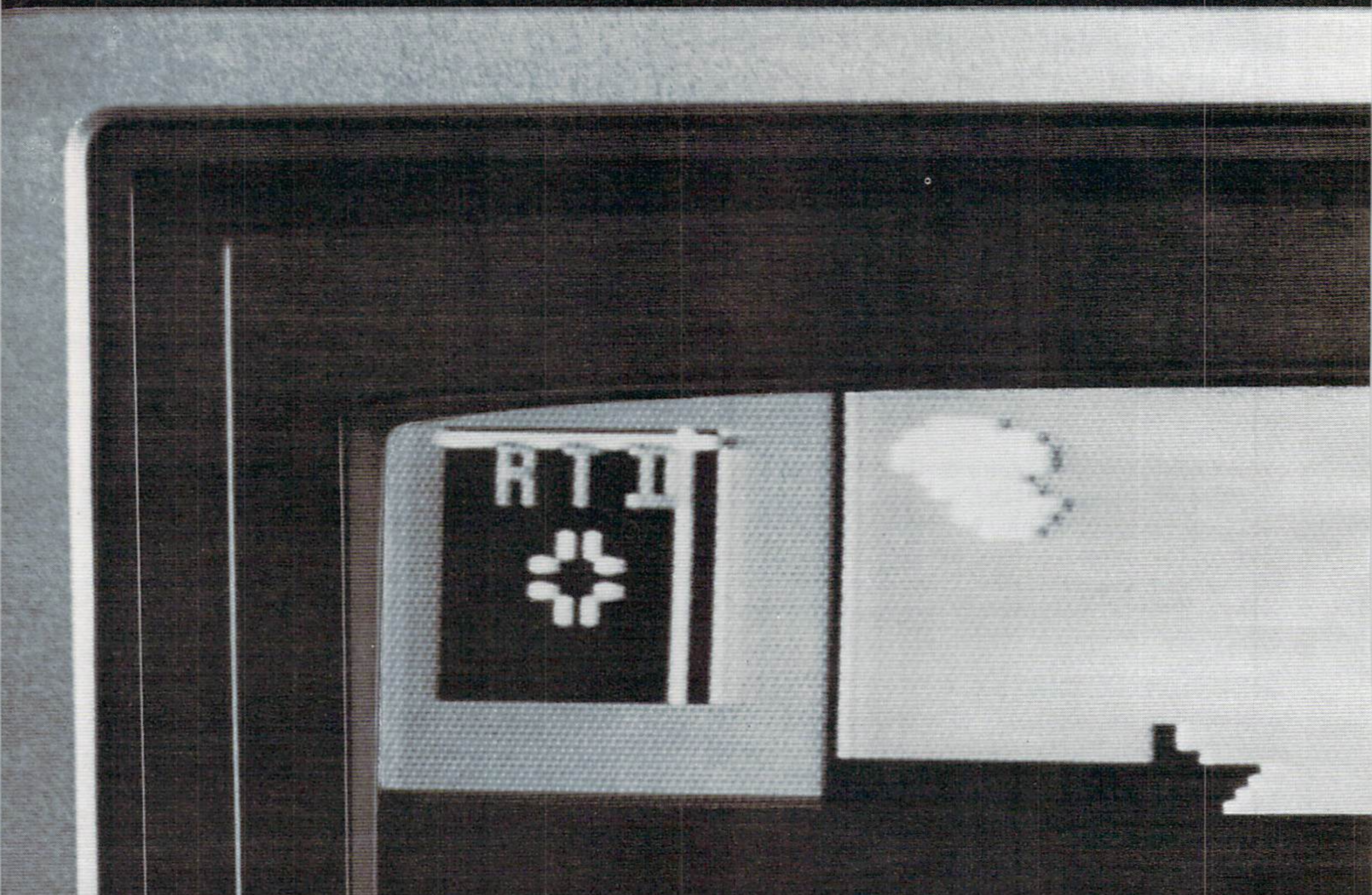
Fortress Of Adnil

Note: *All underlined characters in the program listing should be typed in* graphics mode. *The graphics characters in lines 530 and 7507 are produced by typing graphic shifted 6.*

```
5 GOTO 81ØØ
1Ø LET C=Ø
2Ø LET Z=Ø
3Ø LET G=Ø
35 LET T=5Ø
4Ø FAST
5Ø FOR N=1 TO 63
6Ø PLOT N,Ø
7Ø PLOT N,43
8Ø NEXT N
9Ø FOR M=Ø TO 43
1ØØ PLOT Ø,M
11Ø PLOT 63,M
12Ø NEXT M
13Ø FOR A=1 TO 3ØØ
14Ø GOSUB 1ØØØ
15Ø PRINT AT X,Y;"H"
16Ø NEXT A
17Ø FOR B=1 TO 2Ø
18Ø GOSUB 1ØØØ
19Ø PRINT AT X,Y;"*"
2ØØ NEXT B
2Ø2 IF G=1 THEN GOTO 25Ø
2Ø5 PRINT AT 21,Ø;T
21Ø SLOW
22Ø LET S=347
23Ø GOSUB 1ØØØ
24Ø PRINT AT X,Y;"$"
25Ø POKE PEEK 16396+PEEK 16397*256+S,149
26Ø POKE PEEK 16396+PEEK 16397*256+S,149
27Ø POKE PEEK 16396+PEEK 16397*256+S,21
28Ø IF INKEY$="" THEN GOTO 25Ø
29Ø LET P=S
3ØØ LET A$=INKEY$
31Ø LET S=S-(1 AND A$="5")
32Ø LET S=S+(33 AND A$="6")
33Ø LET S=S-(33 AND A$="7")
34Ø LET S=S+(1 AND A$="8")
35Ø IF A$="9" THEN GOSUB 2ØØØ
355 IF A$<>"5" AND A$<>"6" AND A$<>"7" A
    ND A$<>"8" AND A$<>"9" THEN GOTO 25Ø
36Ø FOR N=1 TO 2
37Ø GOSUB 1ØØØ
38Ø PRINT AT X,Y;"■"
39Ø NEXT N
4ØØ IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=5 THEN LET S=P
41Ø IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=3 THEN LET S=P
42Ø IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=131 THEN LET S=P
43Ø IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=133 THEN LET S=P
44Ø IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=136 THEN LET S=P
```

THE WORLD W
YOUR COMP

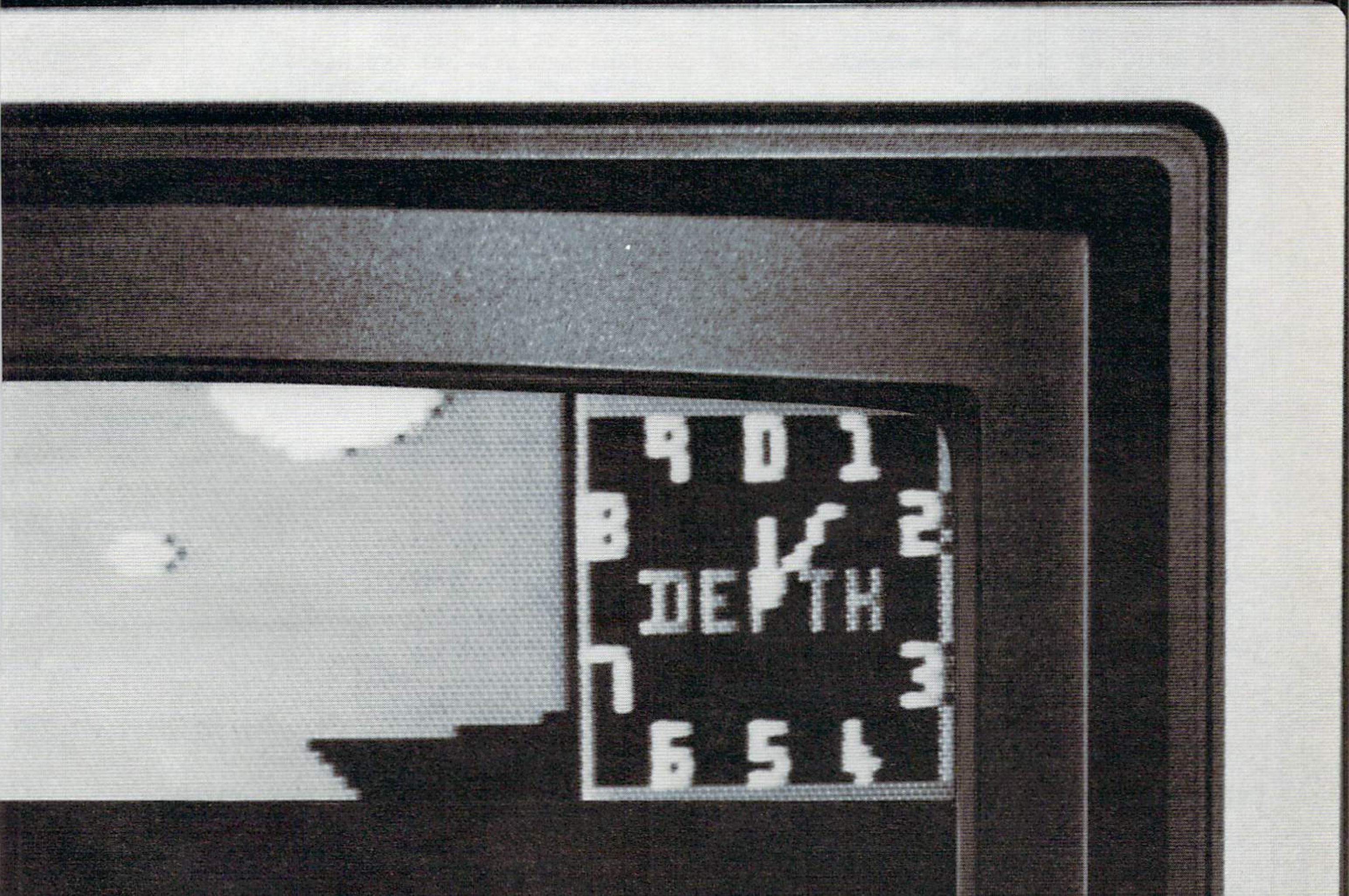
Send us your game and THO



Got a great home computer game you've programmed? Working on one? We'd like to hear about it. Play it. And if we like it, we want to help you sell it to the world.

We're THORN EMI Video, the worldwide entertainment/electronics company. One of the giants in home computer software. We market a full line of computer games to the world. So, if your computer game meets our own high standards, it means people all over the world could be buying it, playing it.

ANTS TO PLAY
UTER GAME!
RN EMI will give you the world.



Think you've got a great computer game? Then write us to find out how to submit your game to THORN EMI.

Just think. The whole world could be playing your computer game soon.

THORN EMI, HOME COMPUTER SOFTWARE DEVELOPMENT, 1370 AVENUE OF THE AMERICAS, NEW YORK, NY 10019.

THORN EMI
VIDEO

```
450 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=141 THEN LET T=T+200
460 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=141 THEN GOSUB 1000
465 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=141 THEN PRINT AT X,Y;"$"
470 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=151 THEN LET T=T+10
480 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=151 THEN LET C=C+1
485 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=151 THEN LET M=1
490 IF C=15 THEN GOTO 170
500 IF PEEK (PEEK 16396+PEEK 16397*256+S
    )=128 THEN GOTO 7500
510 POKE PEEK 16396+PEEK 16397*256+P,0
520 LET T=T-1
525 IF T>2000 THEN GOTO 8500
530 PRINT AT 21,0;"[redacted]"
540 PRINT AT 21,0;T
550 IF T<=0 THEN GOTO 7000
560 LET Z=Z+1
570 IF Z=40 THEN GOSUB 1000
580 IF Z=40 AND RND>.3 THEN PRINT AT X,Y
    ;"$"
590 IF Z=40 THEN LET Z=0
600 LET G=1
610 GOTO 250
1000 LET X=INT(RND*20)+1
1010 LET Y=INT(RND*30)+1
1020 RETURN
2000 POKE PEEK16396+PEEK16397*256+P,149
2002 LET B=P
2005 IF INKEY$="" THEN GOTO 2000
2010 IF INKEY$="9"THEN GOTO 2000
2015 LET B$=INKEY$
2020 IF B$<>"5" AND B$<>"6" AND B$<>"7"
     AND B$<>"8" THEN GOTO 2000
2025 FOR N=1 TO 5
2030 LET B=B-(1 AND B$="5")
2040 LET B=B+(33 AND B$="6")
2050 LET B=B-(33 AND B$="7")
2060 LET B=B+(1 AND B$="8")
2065 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=128 THEN GOSUB 4500
2070 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=128 THEN LET T=T+100
2075 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=128 THEN GOTO 4000
2080 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=128 THEN LET T=T-3*N
2090 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=136 THEN GOTO 4000
3000 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=5 THEN RETURN
3010 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=3 THEN RETURN
3020 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=133 THEN RETURN
3030 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=131 THEN RETURN
3040 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=151 THEN LET T=T-5*N
3050 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=151 THEN RETURN
3060 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=5 THEN RETURN
3080 POKE PEEK 16396+PEEK 16397*256+B,22
3085 POKE PEEK 16396+PEEK 16397*256+B,0
3090 NEXT N
4000 POKE PEEK 16396+PEEK 16397*256+B,0
4005 LET P=S
4010 RETURN
4500 IF RND>.5 THEN RETURN
4505 IF B$="5" THEN GOTO 4525
4510 IF B$="8" THEN GOTO 4527
4515 IF B$="6" THEN GOTO 4531
4520 IF B$="7" THEN GOTO 4533
4525 LET B$="8"
4526 GOTO 4540
4527 LET B$="5"
4528 GOTO 4540
4531 LET B$="7"
4532 GOTO 4540
4533 LET B$="6"
4540 FOR X=1 TO N
4541 LET B=B-(1 AND B$="5")
4542 LET B=B+(33 AND B$="6")
4543 LET B=B-(33 AND B$="7")
4545 LET B=B+(1 AND B$="8")
4555 POKE PEEK 16396+PEEK 16397*256+B,12
     8
4560 POKE PEEK 16396+PEEK 16397*256+B,22
4570 POKE PEEK 16396+PEEK 16397*256+B,0
4575 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=149 THEN GOTO 8000
4577 IF PEEK (PEEK 16396+PEEK 16397*256+
     B)=21 THEN GOTO 8000
4580 NEXT X
4590 GOTO 8000
7000 PRINT AT 21,0;" YOU HAVE USED ALL Y
     OUR POWER "
7010 GOTO 8010
7500 PRINT AT 21,0;" YOU HAVE BEEN CAPTU
     RED BY ADNIL "
7501 IF T-1000<=0 THEN GOTO 8010
7502 PAUSE 200
7503 PRINT AT 21,0;" YOU HAVE PAID RANSO
     M TO ADNIL "
7504 LET S=P
7505 LET T=T-1000
7506 PAUSE 200
7507 PRINT AT 21,0;"[redacted]"
7508 PRINT AT 21,0;T
7509 GOTO 250
8000 PRINT AT 21,0;"{5 SPACES}YOU HAVE B
     EEN DESTROYED{4 SPACES}"
8010 PAUSE 200
8012 PRINT AT 20,0;"{32 SPACES}"
8015 PRINT AT 20,0;"YOUR SCORE :";T
8020 PRINT AT 0,0;" PRESS ANY KEY TO PLA
     Y AGAIN{5 SPACES}"
8030 PRINT AT 0,0;" PRESS ANY KEY TO PLA
     Y AGAIN{5 SPACES}"
8040 IF INKEY$="" THEN 8020
8050 CLS
8060 GOTO 8245
8100 PRINT"{8 SPACES}THE FORTRESS
     {2 SPACES}"
8101 PRINT
8102 PRINT"{13 SPACES}OF"
8103 PRINT
8104 PRINT"{11 SPACES}ADNIL"
8140 PAUSE 600
8145 CLS
8150 PRINT AT 10,0;"DO YOU NEED INSTRUCT
     IONS?"
8160 PRINT AT 12,8;"Y OR N"
8170 IF INKEY$="" THEN GOTO 8170
```

```
8175 LET Y$=INKEY$
8180 CLS
8190 IF CODE Y$=51 THEN GOTO 8245
8200 PRINT "YOU ARE ABOUT TO ENTER THE",
     "FORTRESS OF ADNIL, A POWERFUL"
8201 PRINT "MAGICIAN-WARRIOR IN THE KING
     DOM","OF ANNEP."
8202 PRINT "YOUR GOAL IS TO FIND AS MUCH
     ","OF THE TREASURE ($) ADNIL HAS"
8203 PRINT "PLACED IN HIS FORTRESS AS YO
     U","CAN."
8204 PRINT "YOU MUST INCREASE YOUR ENERG
     Y","BY COLLECTING ENERGY PELLETS (*
     )"
8205 PRINT "WHICH ARE SCATTERED IN THE",
     "FORTRESS.{2 SPACES}YOUR POWER WILL
      ALSO"
8206 PRINT "INCREASE IF YOU DEFEAT THE",
     "NAWS (■) ADNIL USES AS GUARDS."
8207 PRINT AT 21,0;" PRESS ANY KEY TO CO
     NTINUE{7 SPACES}"
8208 PRINT AT 21,0;" PRESS ANY KEY TO CO
     NTINUE{7 SPACES}"
8209 IF INKEY$="" THEN 8207
8210 CLS
8211 PRINT "BE CAREFUL.{2 SPACES}IF ADNI
     L IS MORE","POWERFUL THAN YOU, THEN
      THE NAWS"
8212 PRINT "WILL DESTROY YOU."
8213 PRINT "IF YOU ARE CAPTURED, YOU WIL
     L","HAVE A CHANCE TO PAY A RANSOM"
8214 PRINT "FOR YOUR RELEASE, BUT THE","
     PRICE IS HIGH AND ADNIL MAY"
8215 PRINT "NOT ACCEPT YOUR OFFER."
8216 PRINT "YOUR ONLY WEAPON IS YOUR LAS
     ER","SWORD WHICH YOU USE BY PRESSIN
     G"
8217 PRINT "THE ""9"" KEY AND CHOOSING T
     HE","DIRECTION TO ATTACK."
8218 PRINT AT 21,0;" PRESS ANY KEY TO CO
     NTINUE{6 SPACES}"
8219 PRINT AT 21,0;" PRESS ANY KEY TO CO
     NTINUE{6 SPACES}"
8220 IF INKEY$="" THEN GOTO 8218
8221 CLS
8222 PRINT "TO MOVE :"
8223 PRINT TAB 5;"LEFT PRESS 5"
8224 PRINT TAB 5;"DOWN PRESS 6"
8225 PRINT TAB 5;"UP PRESS 7"
8226 PRINT TAB 5;"RIGHT PRESS 8"
8227 PRINT TAB 5;"YOUR LASER SWORD IS 9"
8228 PRINT
8229 PRINT "USE THE KEYS TO CHOOSE YOUR"
     ,"DIRECTION OF ATTACK WITH YOUR"
8230 PRINT "LASER SWORD."
8231 PRINT "YOU WILL START WITH AN ENERG
     Y","LEVEL OF 50 UNITS.{2 SPACES}EAC
     H MOVE"
8232 PRINT "WILL COST 1 UNIT AND USE","O
     F THE LASER COSTS MORE."
8233 PRINT "CAPTURE THE TREASURE ($) AND
     ","GAIN ENERGY (*) BY MOVING TO"
8234 PRINT "THOSE SPACES."
8235 PRINT AT 21,0;" PRESS ANY KEY TO CO
     NTINUE{6 SPACES}"
8236 PRINT AT 21,0;" PRESS ANY KEY TO CO
     NTINUE{6 SPACES}"
8237 IF INKEY$="" THEN GOTO 8235
8238 CLS
8239 PRINT "IF YOU ARE TRAPPED IN THE MA
     ZE","YOU MAY USE YOUR LASER SWORD"
8240 PRINT "TO BLAST THROUGH THE WALL."
8241 PRINT AT 10,0;"GOOD LUCK...YOU WILL
      NEED IT."
8242 PRINT AT 21,3;"PRESS ANY KEY WHEN R
     EADY"
8243 PRINT AT 21,3;"PRESS ANY KEY WHEN R
     EADY"
8244 IF INKEY$=""THEN GOTO 8242
8245 CLS
8246 PRINT AT 10,0;"THE SCREEN WILL BE B
     LANK FOR"
8247 PRINT "ABOUT 15 SECONDS WHILE YOU A
     RE{2 SPACES}TRANSPORTED TO THE FORT
     RESS OF"
8248 PRINT "{13 SPACES}ADNIL"
8249 PAUSE 400
8250 CLS
8251 GOTO 10
8500 CLS
8501 PRINT AT 10,0;"YOU HAVE DEFEATED AD
     NIL"
8502 PRINT AT 12,5;"YOUR SCORE :":T
8503 GOTO 8010
9990 REM SAVE
9991 PRINT AT 10,5;" START TAPE "
9992 PAUSE 200
9993 POKE 16437,255
9994 CLS
9995 SAVE "ADNIL"
9996 GOTO 5
9999 PRINT"LENGTH OF PROGRAM ";PEEK 1639
     6+256*PEEK 16397-16583
```
©

TECHNIQUES FOR WRITING YOUR OWN ADVENTURE GAME

Charles Perkins

Adventure games are as intriguing to write as they are to play. Here are a few techniques to help you create an intricate drama without running out of memory. These suggestions are useful for any computer, but the specific examples concern Commodore computers.

Remember, you have other tools at your disposal beside standard PEEKs, POKEs, and IF...THENs when programming games. One-byte pointers and ragged tables, for example, can sometimes come in handy as techniques to save memory and help with complicated game logic.

Using these techniques, I developed an adventure game entirely in BASIC for my 8K Commodore PET 2001 (actually 7167 bytes of free memory). It includes an adventure with 48 rooms, 576 vocabulary words, 12 objects (trolls, witches, etc.), and many descriptors and interactive responses. The game is table driven, and the entire adventure, including vocabulary, is stored as data. Many different adventures can be developed using this same program without change.

Computer game programs often use numbers which do not exceed the range of 0 to 255. Array indices and loop variables are common examples. The typical personal computer running BASIC does not permit one-byte variables (value range 0-255). A variable (either floating point or integer) on my PET is always seven bytes long. If your game program needs a good amount of memory and you store lots of variables with values in the range of 0-255, then this unneeded overhead is a problem.

BASIC (which causes the problem) also offers a solution. String manipulation functions permit the program to address a single character, and a character is stored in a single byte (plus some overhead which will be discussed later). With these string manipulation functions and simple algorithms to convert characters to numbers and vice versa, it is possible to efficiently store numbers in one byte.

This approach is particularly useful when a game program makes extensive use of pointers. Pointers are stored variables which "point" to specific pieces of data (i.e., the indices of a table entry). The approach is easily extended to the creation and use of "ragged" tables. A ragged table is one in which the number of columns varies with each row.

One-Byte Pointers

In its simplest form, a one-byte pointer is a value between 0 and 255 stored as a corresponding character in a string variable. Given the character (C$), its value (C) is determined by the equation C=ASC(C$). Given the value, the appropriate character is determined by the function C$=CHR$(C). Storing individual characters as individual strings is not efficient (it uses up eight bytes in the PET), so multiple variables must be stored together in a string (the overhead is constant, and each character adds only one additional byte of memory). To retrieve the Nth character from the storage string (A$), the equation is C$=MID$(A$,N,1). To store a new value in the string is a bit more trouble, but it's still just string manipulation.

Storing The Variables

The simple code number approach described above works if the one-byte variables are always kept internally in the computer. If you want to store the variables on tape or examine them on the screen, a problem arises: the internal character codes include special characters which cannot be saved or printed. In fact, only 128 characters (seven bits) can be saved or printed, and one of these (the quote mark) has special meaning to the PET and cannot be used. The usable character set in the PET has code numbers between 32 and 95 and between 160 and 223. The quote mark is character 34.

In my adventure game application, the storage strings are input from tape as data. I also chose to reserve seven characters as special flags and to eliminate the quote mark from the allowed character set for positive numbers. As a result, I was forced to use slightly more complex encoding and

Educational Software With A Competitive Edge

ATARI™

Pre-School

- Sammy The Sea Serpent . . (C) $13, (D) $19
- Oswald and the Golden Key . . . . (C) $13, (D) $19
- Pre-School I.Q. Builder . . . . (C) $13, (D) $24
- Hodge Podge (D) $16
- My First Alphabet (D) $26
- Ten Little Robots . . . . (C) $13, (D) $15
- Basic Math (+, −, *, /) (D) $19
- Basic Math (Add., Sub.) or Mult., Div.) (C) $10
- Alien Counter/Face Flash (C, D) $26
- Jar Game/Chaos (C, D) $26
- Pre-School Fun (Color, Shape, etc.) (C) $16
- Hickory Dickory/ Baa Baa Black Sheep (C) $25
- Humpty Dumpty/Jack and Jill (C) $25
- Counters (C, D) $19
- Facemaker (D) $23
- I'm Different (D) $19

Math

- Monkey Up a Tree (C, D) $19
- Video Math Flash Cards (C, D) $13
- Math-Tic-Tac-Toe (C, D) $13
- Calculus Demon (C, D) $19
- Cubbyholes (C, D) $19
- Metric and Problem Solving (D) $26
- Algicalc (C, D) $19
- Polycalc (C, D) $19
- Counters (Ages 3-6) (C, D) $26
- Basic Math (Add., Sub.) (C) $10
- Basic Mach (Mult., Div.) (C) $10
- Basic Math (+, −, *, /) (D) $19
- Ten Little Robots . . . . (C) $13, (D) $15
- Compumath-Fractions . . . . (C) $23, (D) $29
- Compumath-Decimals . . . . (C) $23, (D) $29
- Alien Numbers (C, D) $23
- Math Pak 1 (C, D) $23
- Alien Counter/Face Flash (C, D) $26
- Golf Classic/Compubar (Angles) (C, D) $26
- Jar Games/Chaos (Ages 6-10) (C, D) . . . $26
- Gulp and Arrow Graphics (7-12) (C, D) . $26
- Battling Bugs/Concentration (C, D) . . . $26
- Addition With Carrying . . . (C) $13, (D) $19
- Cash Register . . . . (C) $13, (D) $19
- Number Series . . . . (C) $13, (D) $19
- Quantitative Comparisons (C) $15, (D) $19
- Sky Rescue . . . . (C) $15, (D) $19
- Big Math Attack . . . . (C) $17, (D) $22
- Math Facts Level II Grade 1-3 . . . . (C) $13, (D) $15
- Com*putation/ Concentration . . . . (C) $13, (D) $15
- Ship's Ahoy (D) $20
- The Market Place (D) $26

Reading and Language Arts

- Hey Diddle Diddle $20
- Letterman (C, D) $19
- My First Alphabet (D) $26
- Wordmaker (C, D) $19
- Spelling Genie (C, D) $19
- Word Search Generator (D) $19
- Compuread . . . . (C) $17, (D) $23
- Astroquotes . . . . (C) $13, (D) $19
- Memory Builder/ Concentration . . . . (C) $13, (D) $19
- Let's Spell (C) $13
- Spelling Builder . . . . (C) $16, (D) $20
- Do-It-Yourself Spelling (C) $16
- S.A.T. College Board Prep. (C) $89
- Story Builder/ Word Master . . . . (C) $13, (D) $19
- What's Different . . . . (C) $13, (D) $19
- Analogies . . . . (C) $13, (D) $19
- Prefixes (D) $26
- Vocabulary Builder 1 . . . . . (C) $13, (D) $19
- Vocabulary Builder 2 . . . . . (C) $13, (D) $19
- Mini-Crosswords . . . . (C) $13, (D) $19
- Word Scramble Grades 1-4 (C) $13
- Fishing For Homonyms (C) $13
- Hidden Words 4 Levels (C) $16
- Snooper Troops #1 (D) $32
- Snooper Troops #2 (D) $32
- Story Machine (D) $23
- Word Race (D) $17
- Claim to Fame/Sports Derby $15
- Crossword Magic (D) $34
- Alphabet Arcade . . . . (C) $15, (D) $19
- Funbunch (D)
  - Elem. $25
  - Intermediate $25
  - High School (SAT) $25
- Time Bomb . . . . (C) $13, (D) $19
- Snake-O-Nyms $25
- Skywriter & Pop'r Spell $25

Music

- Rhyme & Pitch $26
- Player Piano (C, D) $19
- Keyboard Organ (C, D) $19
- Musical Computer—Music Tutor (D) . . $13
- Music 1—Terms and Notation (D) $26
- Advanced Music System (D) $25
- Music Composer . . . . (CT) $25
- Jerry White's Music Lessons (C) $20
- Magic Melody Box $14

Telling Time

- Hickory Dickory (C, D) $13

Social Studies and Geography

- Earth Science (D) $26
- Flags of Europe (D) $19
- Presidents of the U.S. (C, D) $13
- Astro Word Search . . . . . . (C) $13, (D) $19
- States and Capitals (C) $12
- European Countries & Capitals (C) . . . $12
- Computer Stocks and Bonds . . . . (C) $12, (D) $15
- Elementary Biology (D) $26
- Frogmaster (D) $19
- Starware (D) $19
- Mapware (D) $19
- British Heritage Jigsaw Puzzles $22
- European Scene Jigsaw Puzzles (C) . . $22
- Geography (D) $26

Programming Techniques

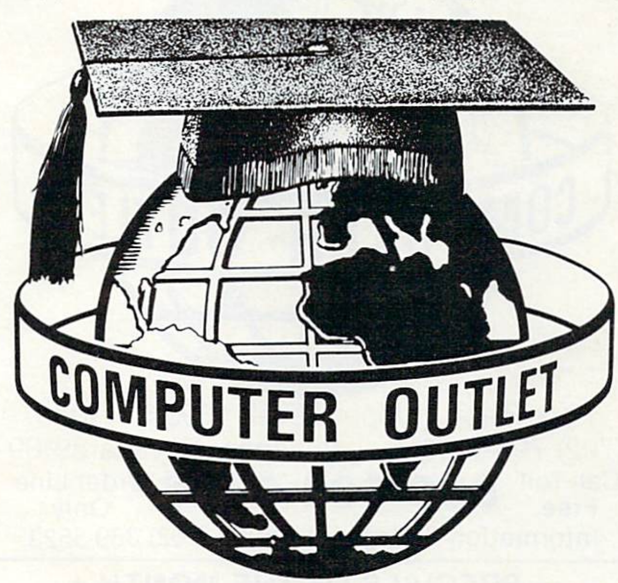
- Pilot (Cons. or Educator) . . (C) $59, (D) $99
- Invitation to Prog. #2 (C) $22
- Invitation to Prog. #3 (C) $22
- Tricky Tutorials—Santa Cruz
  - TT #1 Display Lists (C, D) $17
  - TT #2 Horiz/Vert. Scrolling (C, D) . . . $17
  - TT #3 Page Flipping (C, D) $17
  - TT #4 Basics of Animation (C, D) . . . $17
  - TT #5 Player Missile Graphics (C, D) $24
  - TT #6 Sound of Music (C,D) $24
  - TT #7 DOS Utilities (D) $24
- Page 6 . . . . $20
- The Next Step $27

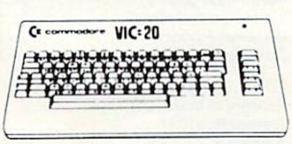
Typing

- Master Type (D) $27
- Touch Typing (C) $19
- Type Attack (C, D) $26

Foreign Languages

- Atari Conversational Languages French, Spanish, German, Italian (C) $45
- Astro Word Search (Specify Spanish or French) . . . . . (C) $13, (D) $19





commodore

Pre-School

- The Sky Is Falling (CT) $23
- Mole Attack (CT) $23
- Home Babysitter $23
- Facemaker 64 . . . . $23
- Kindercomp 64 . . . . $20

Math

- Sky Math (C) $12
- Space Division $12
- Bingo Speed Math (CT) $23
- Number Crunch (CT) $27
- Number Chaser $17
- Number Gulper $17

Music

- VIC Music Composer (CT) $29
- HES Synthesound (CT) $49

Language Arts

- Super Hangman (C) $14
- Simon/Hess (C) $13
- Concentration (C) $13
- Home Babysitting $23
- Hey Diddle Diddle 64 . . . . $20

Social Studies/Science

- Visible Solar System $23
- Reaganomics (CT) $27

Programming Techniques

- Intro to Basic Prog. I $22
- Intro to Basic Prog. II $22
- Programmers and Cart. $45
- Turtle Graphics/Hess (CT) $29
- Groteck & Microchip $23

*** BOOKS ***

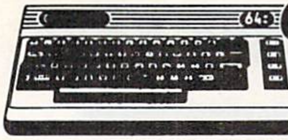
- KIDS AND THE ATARI $18
- KIDS AND THE VIC $18
- PROGRAMMERS REF. GUIDE (VIC) $14
- ELEMENTARY COMMODORE $14
- COMPUTERS FOR PEOPLE . . . . $ 8
- GAMES FOR THE ATARI . . . . $ 8
- DE RE ATARI $19
- ADVENTURE HINT BOOKS . . . . $ 8
- 6502 ASSEM. LG. PROG. $16
- SOME COMMON BASIC BASIC PROGRAMS $14
- YOUR ATARI COMPUTER $16
- ATARI ASSEMBLER — INMAN $12
- ATARI GAMES AND RECREATION $14
- ATARI PILOT FOR BEGINNERS $12
- VISICALC BOOK — ATARI EDITION $14
- ATARI BASIC — R. L. ALBRECHT . . . . $ 8

Computer Outlet

Park Place — Upper Level 1095 E. Twain — (702) 796-0296 Las Vegas, Nevada 89109

Call Toll Free **800-634-6766** Order Line Only

We accept Major Credit Cards Mon.-Fri. 8 A.M.-6 P.M.



Commodore Puts Excitement In Your Life

commodore

COMMODORE 64 $389

COMMODORE 64	$389
1530 DATTASETTE	$ 59
1541 DISK DRIVE	$329
1525 GRAPHIC PRINTER	$329
1600 MODEM	$ 89

COMMODORE 64 SOFTWARE

Avalon Hill Game Company

180-701	B-1 Nuclear Bomber (C)	$12
180-702	Midway Campaign (C)	$12
180-703	North Atlantic Convoy Raider (C)	$12
180-704	Nukewar (C)	$12
180-706	Planet Miners (C)	$12
180-712	Computer Stocks & Bonds (C)	$15
180-719	Andromeda Conquest (C)	$14
181-721	Computer Football Strategy (C)	$12
181-732	Telengard (C)	$16

EPYX/Automated Simulations

14E-036	Jump Man (D)	$27

Human Engineered Software (HES)

HEE-307	6502 Professional Dev. System (C)	$23
HEE-400	Retro Ball (Crt)	$27
HEE-401	Hesmon (Crt)	$27
HEE-402	Turtle Graphics II (Crt)	$45
HEE-404	Heswriter 64 (Crt)	$35
HEE-412	Gridrunner (Crt)	$27

Infocom

63E-001	Zork I (D)	$27
63E-002	Zork II (D)	$27
63E-003	Deadline (D)	$35
63E-004	Starcross (D)	$27
63E-005	Zork III (D)	$27

Sierra On-Line

54E-048	Frogger (D)	$23

Sirius Software Co-op!

70E-036	Blade of Blackpoole (D)	$27
70E-037	Type Attack (Crt)	$27
70E-043	Repton (D)	$27
70E-046	Critical Mass (D)	$27
70E-424	Snake Byte (Crt)	$23
70E-445	Spider City (Crt)	$27
70E-447	Squish 'em (Crt)	$23
70E-448	Final Orbit (Crt)	$23

Spinnaker

SKE-001	Snooper Troops #1 (D)	$30
SKE-004	Facemaker (D)	$23
SKE-006	Kindercomp (D)	$20
SKE-008	Hey Diddle Diddle (D)	$20
SKE-009	In Search of the Most Amazing Thing (D)	$27

Snyapse Software

SSE-011	Ft. Apocalyse (D)	$23
SSE-016	Drelbs (D)	$23
SSE-019	Survivor (D)	$23
SSE-020	Pharoh's Curse (D)	$23
SSE-311	Ft. Apocalypse (C)	$23
SSE-316	Drelbs (C)	$23
SSE-319	Survivor (C)	$23
SSE-320	Pharoh's Curse (C)	$23

United Microwave Industries (UMI)

92E-302	Renaissance (C)	$27
92E-331	Motor Mania (C)	$20

Creative Software

Black Hole (CT)	$36
Trashman (CT)	$36
Astroblitz (CT)	$36
City Bomber & Minefield (CT)	$20
Apple Panic (CT)	$36
Choplifter (CT)	$36
Serpentine (CT)	$36
Videomania (CT)	$36
Terraguard (CT)	$36

Thorn EMI

River Rescue (CT)	$29
VIC Music Composer (CT)	$29
Mutant Herd (CT)	$29

Automated Simulations

Rescue at Rigel (C)	$20
Ricochet (C)	$15
Monster Maze (CT)	$27
Sword of Fargoal	$27

Spectravision

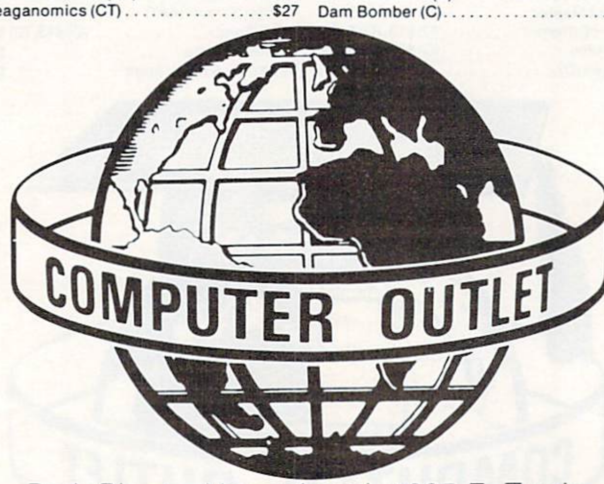
Cave In (CT)	$27
Number Crunch (CT)	$27
Reaganomics (CT)	$27

Tronix

Galactic Blitz (C)	$17
Swarm (C)	$20
Sidewinder (C)	$20

HES Software

VIC Forth (CT)	$45
HES Mon (CT)	$29
Turtle Graphics (CT)	$29
HES Writer (CT)	$29
Aggressor (CT)	$29
Shamus (CT)	$29
Protector (CT)	$33
Synthesound (Music Synthesizer) (CT)	$49
Skier (C)	$15
Maze of Mikor (C)	$15
Tank Wars (C)	$15
Victrek (C)	$15
Pinball (C)	$13
Simon (C)	$13
Fuel Pirates (C)	$13
Pak Bomber (C)	$13
Laser Blitz (C)	$15
Tank Trap (C)	$15
Concentration (C)	$13
Dam Bomber (C)	$13



Park Place - Upper Level, 1095 E. Twain
(702) 796-0296 — Las Vegas, Nevada 89109
Call Toll Free **800-634-6766** Order Line Only
Information & Order Inquiries (702) 369-5523

★ SPECIALS OF THE MONTH ★

AMDEK COLOR I MONITOR	$299
WICO JOYSTICK	$ 23
WICO REDBALL JOYSTICK	$ 24
WICO TRACKBALL	$ 49
KIDS & THE VIC	$ 18
PROGRAMMER'S REFERENCE GUIDE (VIC)	$ 14
PROGRAMMER'S REFERENCE GUIDE (64)	$ 18
SLAGH 24K MEMORY BOARD	$145
CARDCO 6 SLOT EXPANSION MOTHERBOARD	$ 79
CARDCO 3 SLOT EXPANSION MOTHERBOARD	$ 39
CARDWRITER LIGHT PEN	$ 29
CARDETTE PARALLEL INTERFACE	$ 59
DATA 20 40/80 COLUMN 16K BOARD	$219
DATA 20 40/80 COLUMN 64K BOARD	$289

ORDERING INFORMATION AND TERMS:

For Fast Delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. C.O.D. orders ($3.00 minimum) and 1% of all orders over $300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change. Please enclose your phone number with any orders. Shipping — Software ($2.50 minimum). Shipping — Hardware (please call). Foreign orders, APO & FPO orders — $10 minimum and 15% of all orders over $100. Nevada residents add 5¾% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must have a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement. All returned merchandise is subject to a restocking fee and must come with their original packaging in order to be accepted.

NO returns permitted after 21 days from shipping date.

commodore

VIC 20 $139

VIC 1530 DATASETTE	$ 59
VIC 1541 DISK DRIVE	$329
VIC 1525 GRAPHICS PRINTER	$329
VIC 1210 3K Memory Expander	$ 34
VIC 1110 8K Memory Expander	$ 52
VIC 1111 16K Memory Expander	$ 89
VIC 1011 RS 232 Terminal Interface	$ 43
VIC 1211 Super Expander	$ 59
VIC 1212 Programmers Aid Cartridge	$ 45
VIC 1213 Vicmon Machine Language Monitor	$ 45
VL 102 Introduction to Basic Programming	$ 21
VT 106A Recreation Pack	$ 45
VT 107 A Home Calculation Pack	$ 45
VT 164 Programmable Character Set	$ 12
VIC 1600 Vicmodem	$ 89
VIC 1311 Joystick	$ 8
VIC 1312 Game Paddles	$ 16
VM Programmers Reference Guide	$ 14

VIC Software

Avenger	$ 23
Superslot	$ 23
Super Alien	$ 23
Jupiter Lander	$ 23
Draw Poker	$ 23
Midnight Drive	$ 23
Radar Rat Race	$ 23
Raid on Fort Knox	$ 23
Sargon II Chess	$ 29
Super Smash	$ 23
Cosmic Cruncher	$ 23
Gorf	$ 29
Omega Race	$ 29
Money Wars	$ 23
Menagerie	$ 23
Cosmic Jailbreak	$ 23
Clowns	$ 23
Garden Wars	$ 23
Sea Wolf	$ 23
Adventureland	$ 29
Pirate Cove	$ 29
Mission Impossible	$ 29
The Count	$ 29
Voodoo Castle	$ 29
The Sky is Falling	$ 23
Mole Attack	$ 23
Bingo Speed Math	$ 23
Home Babysitter	$ 23
Visible Solar System	$ 23
Personal Finance	$ 29
Quick Brown Fox	$ 65

United Microware

Spiders of Mars (CT)	$ 34
Meteor Run (CT)	$ 34
Amok (C)	$ 17
Alien Blitz (C)	$ 17
Skymath (C)	$ 12
Space Division (C)	$ 12
Super Hangman (C)	$ 14
The Alien (C)	$ 17
3D Maze (C)	$ 12
Kosmic Kamikaze (C)	$ 17
Sub Chase (C)	$ 17
Amok (CT)	$ 27
Renaissance (CT)	$ 34
Alien Blitz (CT)	$ 27
Cloud Burst (CT)	$ 27
Satellites and Meteorites (CT)	$ 34
Outworld (CT)	$ 34
Wordcraft	$270

The Computer Outlet is an associate of The Computer Learning Center For Children. "...We are experts in educational technology featuring our own Computer Learning Center educational software. Dealers inquiries invited."

decoding subroutines:

Given a character C$, then the value C is computed by:

```
10 C=ASC(C$):C=C-40+(C>159)*64:RETURN
where (C>159)= -1 if C>159 and 0 if C=<159
```

Given a number D, then the character D$ is determined by:

```
20 IF D<56 THEN D$=CHR$(D+40):RETURN
30 D$=CHR$(D+104):RETURN
```

These routines yield a range from -8 to 119 with the quote mark at -6. The negative values were used internally as the special flags in my adventure game. In these routines, an open parenthesis is a zero; a close parenthesis is 1. A shifted back arrow is 119; a blank is -8. The encoding and decoding subroutines may have to be revised for other computers, depending on the code number schemes used.

Passages And Exits

To understand how one-byte pointers can be used to save memory in your game programs, consider the simple adventure map in Figure 1. You start at a crossroad (state 1). Movement to the north, south, and west places you in a forest or in houses of various colors. There is a secret, one-way passage from the red house to the blue house. Going east from the crossroad puts you in a cave from which there is no escape.

This adventure map can be expressed as a *state table,* as shown in Table 1. The rows of the table correspond to states (locations) in the map. The columns correspond to the possible movement directions (in this case north, south, east, and west). If you are in state 1 (the crossroad) and wish to move south, you end up in state 5 (the red house). (This state transition is shown in Table 1.) A further attempt to move south (while in state 5) has no path ("no exit"), indicated by the zero pointer. All exits from state 4 (the cave) put you back in state 4. This would appear as an endless cave to the person playing the game.

The state table can be programmed into the adventure game using the subroutines described above. The result is shown in Table 2. This encoded state table requires only 42 bytes of memory in the PET (including all overhead, as discussed below). Storing the table as a matrix of integer numbers would require 59 bytes on the PET. While the memory saved is not dramatic for this small example, when large tables are used, the memory saved can be quite substantial.

Ragged Tables

Suppose that we wish to add descriptions of each state to our game program. These would be printed on the screen each time a state was entered. A list for our simple adventure game map is shown in Table 3.

These could be stored in strings, but they would consume 118 bytes of memory (plus some overhead). Alternatively, these descriptions can be broken into phrases which are used in various combinations to make up the descriptions. These phrases are shown in Table 4. These phrases require only 53 bytes (plus some overhead), but we must also define the rules for combining phrases back into descriptions for each state. Once again, we use one-byte pointers. These new pointers can be simply added to the encoded state table, as shown in Table 5.

The procedure for creating a description when a state is entered is shown graphically in Figure 2. The BASIC code necessary to print the description of state 1 is as follows:

```
40 L=LEN(A$(I))   find length of string
50 FOR J=5 TO L   skip first four characters and scan
60 C$=MID$(A$(I),J,1)   select next character
70 GOSUB 10   convert to number C (see above)
80 PRINT B$(C)" ";   print phrase and blank
90 NEXT J
100 PRINT   end print line
```

Note that the number of characters in each state table entry shown in Table 5 is different. It is therefore a "ragged" table. It requires 18 additional bytes to store the pointers for all five state descriptions, a net savings of 47 bytes compared to storing the full descriptions (not including overhead).

Storage Methods

The techniques described above can be applied to computing problems other than games. The bigger the pointer tables are, the more advantages one-byte pointers offer. However, the tradeoff between one-byte pointers and simple integers is tricky because of the overhead required to set up strings or arrays of strings, and because extra programming is required to isolate and decode the stored character.

The storage technique used in my PET 2001 (original ROMs) requires seven bytes plus the number of characters for string variables. Thus, a single character pointer should never be used. When arrays are used, the tradeoff is dependent upon the number of rows and columns involved. For a ten by ten two-dimensional array, the memory used for a floating point array is 509 bytes. This is 500 bytes for the numbers and nine bytes for an array header (overhead). An integer array requires 209 bytes (200 bytes for the numbers and nine bytes for the header).

Using the one-byte variables reduces this array to a one-dimensional array of ten strings. Each string is ten characters long. The total memory requirement is 137 bytes. This is 100 bytes for the numbers, seven bytes for the header, three bytes for each string, for a total of 37 bytes overhead. As the arrays get larger, the one-byte approach uses approximately one-half the memory required by integer arrays and one-fifth of the memory required by floating point arrays. A more detailed explanation of the storage structures of Commodore computers can be found in *Programming The PET* (**COMPUTE! Books**, 1982).

The one-byte storage technique can be especially useful when: memory is at a premium, when large tables of pointers are needed, and when ragged tables provide a programming advantage.

When you're programming games into computers with limited memory (such as the unexpanded VIC-20), these techniques can be very advantageous.

Figure 1: Adventure Map

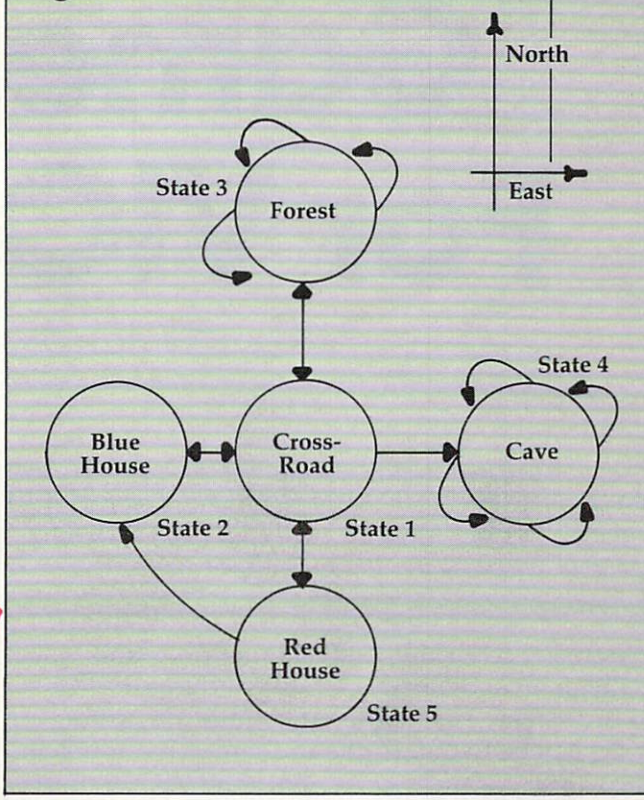


Table 1: State Table For The Adventure Map

	State	North	South	East	West	
		Movement Direction				
	1	3	5	4	2	59 bytes
	2	0	0	1	0	required
	3	3	1	3	3	in PET
State	4	4	4	4	4	to store
Transition	5	1	0	0	2	as integers

Table 2: Encoded State Table

A$(1) = + − , *
A$(2) = (( ) (
A$(3) = +) + +
A$(4) = , , , ,
A$(5) =) ((*

42 bytes required in PET

Table 3: State Descriptions

State 1 "YOU ARE AT A CROSSROAD"
State 2 "YOU ARE IN A BLUE HOUSE"
State 3 "YOU ARE IN A FOREST"
State 4 "YOU ARE IN A CAVE, YOU ARE LOST"
State 5 "YOU ARE IN A RED HOUSE"

118 bytes required plus overhead

Table 4: Phrase Table

B$(1) = "YOU ARE"
B$(2) = "IN A"
B$(3) = "AT A CROSSROAD"
B$(4) = "BLUE"
B$(5) = "RED"
B$(6) = "HOUSE"
B$(7) = "FOREST"
B$(8) = "CAVE,"
B$(9) = "LOST"

53 bytes required plus overhead

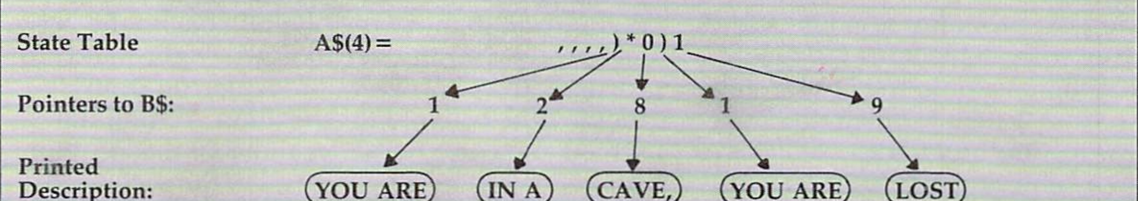
Table 5: Ragged State Table With Descriptor Pointers

A$(1) = + − , *) +
A$(2) = (( ) ( ) * , .
A$(3) = +) + +) * /
A$(4) = , , , ,) * 0) 1
A$(5) =) ((*) * − .

pointers to phrase table

18 added bytes for a total of 60 bytes required in PET

Figure 2: Decoding Example



©

Game POKEr For VIC And 64

Dan Carmichael, Assistant Editor

With one touch of the finger and the "Screen-plot," you can easily determine the screen locations of your PEEKs and POKEs. This can be of great help when designing games or graphics. For the VIC and 64.

When you're writing or designing programs, especially games, that use a lot of POKEs and PEEKs to the screen, one of the most time-consuming tasks can be to determine the screen locations of those POKEs and PEEKs. With the VIC-20 or the Commodore 64, you can use the charts supplied with either the instruction book or the *Programmer's Reference Guide,* or you can take a guess and do a number of POKEs until you "hit" the position you desire. But both methods can be time-consuming.

To solve this problem, you can use this useful "Screen-plot" utility program. The program will, with the touch of one finger, move a blinking ball ("●"-CHR$113) to any position on the screen while continuously displaying both the screen and color POKE locations of the blinking ball.

This is a machine language program written to run in the cassette buffer (but you can use it even if you don't understand machine language). It will require only one BASIC statement; otherwise, it will leave your available BASIC programming memory untouched.

First, type in the program. If you're going to use Screen-plot in conjunction with the program you are currently working on, either append the screen-plotter to it or load your program and then type in the screen-plotter after it. The line numbers, starting at 59995, should insure that it will always remain at the end of your program.

After entering the program, SAVE it before running. As is true with all machine language programs, even a slight error in the DATA statements can cause your system to crash, forcing you to turn off your computer to recover. Then run the program by entering "RUN 59995", and after a pause of about two seconds, the "READY" will be displayed. The Screen-plot program is now POKEd into memory and ready to run.

To run the Screen-plot utility, enter "RUN 60000". If you entered the program correctly, a blinking ball will be displayed on your screen, along with two numbers in the upper left-hand corner. The first number is the screen position of the blinking ball; the second is the color location. As you move the ball around the screen, these numbers will change, reflecting the changes in the screen and color locations.

Controlling The Program

Movement of the ball is accomplished via the F-keys. The following table shows which F-key controls which direction of movement.

F-Key	Blinking Ball Direction
F-1	→
F-3	←
F-5	↓
F-7	↑

Screen-plot has a built-in safety feature that prevents you from leaving the screen with the blinking ball and thereby altering other important memory locations in your computer.

This utility program runs in the cassette buffer, so you cannot use the cassette tape while this program is running. For you machine language programmers, the screen-plotter uses the zero-page locations hex $FB and $FC; so they are unavailable to you.

Also, because of Commodore's automatic scrolling feature, the screen display would scroll if you were to move the blinking ball to the very last position on the screen (lower right-hand corner). So the program prevents you from moving the ball into this position. To find the screen and color POKE locations of this position, simply move the blinking ball to the second to last position and add 1.

Hints And Tips

After the program has been successfully POKEd into memory and tested, you may delete lines 59994-59999. The only line necessary to support the running of Screen-plot is line 60000. Also, the screen-plotter will not clear the screen upon initialization, so you may use it successfully with whatever screen display your program generates. To stop the screen-plotter, simply press the STOP key.

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

Program 1:
Screen-plot For The Unexpanded VIC

```
59994 FORA=828TO921:READB:POKEA,B:NEXT:EN
      D
59995 DATA165,197,166,251,164,252,201,39,
      2Ø8,6,224,21,24Ø,2,23Ø,251,2Ø1,47,
      2Ø8,6,224,Ø
59996 DATA24Ø,2,198,251,2Ø1,55,2Ø8,6,192,
      22,24Ø,2,23Ø,252,2Ø1,63,2Ø8,6,192,
      Ø,24Ø,2,198
59997 DATA252,166,252,164,251,224,22,2Ø8,
      7,192,21,2Ø8,3,2Ø2,198,251,24,32,2
      4Ø,255,169
59998 DATA113,32,21Ø,255,162,Ø,16Ø,Ø,232,
      2Ø8,253,2ØØ,192,32,2Ø8,248,169,157
      ,32,21Ø,255
59999 DATA169,32,32,21Ø,255,96,234
6ØØØØ SYS828:A=PEEK(251)+PEEK(252)*22+768
      Ø:PRINT"{RED}{HOME}"A;A+3Ø72Ø"
      {BLU}":GOTO6ØØØØ
```

Program 2:
Screen-plot For The Expanded (8K Or More) VIC

```
59994 FORA=828TO921:READB:POKEA,B:NEXT:EN
      D
59995 DATA165,197,166,251,164,252,201,39,
      2Ø8,6,224,21,24Ø,2,23Ø,251,2Ø1,47,
      2Ø8,6,224,Ø
59996 DATA24Ø,2,198,251,2Ø1,55,2Ø8,6,192,
      22,24Ø,2,23Ø,252,2Ø1,63,2Ø8,6,192,
      Ø,24Ø,2,198
59997 DATA252,166,252,164,251,224,22,2Ø8,
      7,192,21,2Ø8,3,2Ø2,198,251,24,32,2
      4Ø,255,169
59998 DATA113,32,21Ø,255,162,Ø,16Ø,Ø,232,
      2Ø8,253,2ØØ,192,32,2Ø8,248,169,157
      ,32,21Ø,255
59999 DATA169,32,32,21Ø,255,96,234
6ØØØØ SYS828:A=PEEK(251)+PEEK(252)*22+4Ø9
      6:PRINT"{RED}{HOME}"A;A+33792"
      {BLU}":GOTO6ØØØØ
```

Program 3: Screen-plot For The 64

```
59994 FORA=828TO921:READB:POKEA,B:NEXT:EN
      D
59995 DATA165,197,166,251,164,252,2Ø1,4,2
      Ø8,6,224,39,24Ø,2,23Ø,251,2Ø1,5,2Ø
      8
59996 DATA6,224,Ø,24Ø,2,198,251,2Ø1,6,2Ø8
      ,6,192,24,24Ø,2,23Ø,252,2Ø1,3,2Ø8,
      6
59997 DATA192,Ø,24Ø,2,198,252,166,252,164
      ,251,224,24,2Ø8,7,192,39,2Ø8,3,2Ø2
      ,198
59998 DATA251,24,32,24Ø,255,169,113,32,21
      Ø,255,162,Ø,16Ø,Ø,232,2Ø8,253,2ØØ,
      192
59999 DATA32,2Ø8,248,169,157,32,21Ø,255,1
      69,32,32,21Ø,255,96,234
6ØØØØ SYS828:A=PEEK(251)+PEEK(252)*4Ø+1Ø2
      4:PRINT"{HOME}"A;A+54272:GOTO6ØØØØ
```

©

ROADBLOCK

Brian Holness

There's a bit of typing here, but it's worth it. This game, written entirely in machine language, is fast and flexible. You have a choice of five speeds, up to four players simultaneously, or you can compete directly against the computer. You try to control an ever-growing line without running into a boundary, another player, or yourself. For the Atari.

In **COMPUTE!** (August 1981) there was an action game called "Blockade," written entirely in BASIC. The idea is simple. Each player controls a line which continually grows in an enclosed box. The first player who crashes into anything (himself or herself included) loses a point. Players start with nine points, and when they reach zero they're out of the game.

The use of BASIC prevented the possibility of allowing increased speeds, multiple players, or computer play options. I wrote this version of Blockade – called "Roadblock" in machine language to add these options. If you don't know machine language you can still type it in and use it; the program contains all the DATA statements required to run the program via a USR statement.

One of the major stumbling blocks I had in writing this program was the use of graphics in machine language. Fortunately, Bill Wilkinson's "Insight: Atari" article in **COMPUTE!** (February 1982) came to my rescue; those familiar with his article will recognize his code.

When the main menu comes up, you are instructed to use the select, option, or start button. The option button controls the speed, from 1 to 5, where 1 is the slowest speed. The select button controls both the number of players (2, 3, or 4) and the computer play option. When the computer plays, it always plays as player number 2 and is included in the total number of players. Thus, if three players are indicated and the computer is playing, then player numbers 1 and 3 are the humans, and player number 2 is the computer.

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

Roadblock

```
10 FOR I=13824 TO 15010:READ B:POKE
   I,B:NEXT I
15 A=USR(14788)
1010 DATA 16,83,58,0,2,96,64
1020 DATA 37,21,16,21,1,0,46
1030 DATA 27,40,40,29,23,5,43
1040 DATA 1,255,0,0,0,0,1
1050 DATA 255,1,2,3,1,7,0
1060 DATA 9,9,8,15,37,64,4
1070 DATA 8,12,16,2,48,16,176
1080 DATA 48,0,0,7,11,14,13
1090 DATA 32,32,112,114,101,115,115
1100 DATA 32,239,240,244,233,239,238
1110 DATA 32,102,111,114,155,32,32
1120 DATA 112,114,101,115,115,32,243
1130 DATA 229,236,229,227,244,32,102
1140 DATA 111,114,155,112,114,101,11
5
1150 DATA 115,32,243,244,225,242,244
1160 DATA 32,116,111,32,98,101,103
1170 DATA 105,110,155,32,32,32,32
1180 DATA 32,82,111,193,228,32,66
1190 DATA 108,207,227,75,155,32,32
1200 DATA 66,121,32,194,114,73,225
1210 DATA 206,32,104,79,236,206,101
1220 DATA 83,243,155,155,32,83,80
1230 DATA 69,69,68,58,32,180,155
1240 DATA 32,80,76,65,89,69,82
1250 DATA 83,58,32,178,155,32,67
1260 DATA 79,77,80,85,84,69,82
1270 DATA 32,80,76,65,89,83,58
1280 DATA 32,206,155,78,89,231,225
1290 DATA 237,229,160,239,246,229,24
2
1300 DATA 155,13,66,40,40,23,23
1310 DATA 5,43,1,255,0,0,0
1320 DATA 0,1,255,169,9,162,3
1330 DATA 157,33,54,202,16,250,174
1340 DATA 11,54,202,189,5,54,141
1350 DATA 10,54,169,21,32,150,56
1360 DATA 173,48,2,133,203,173,49
1370 DATA 2,133,204,160,3,169,71
1380 DATA 145,203,160,6,152,145,203
1390 DATA 169,3,160,3,32,195,56
1400 DATA 169,0,162,2,160,3,32
```

EXPLORE A NEW DIMENSION IN SOFTWARE

INTRODUCING ACTION! — Now the fastest 8-bit language

Another first from OSS! ACTION! is a brand new language designed to run on 6502-based computers, including Atari, Apple II, and Commodore 64. A powerful, structured language, ACTION! can draw out a new, higher dimension of performance from these machines, with speeds never seen before. ACTION! combines some of the best features of such languages as Pascal, C, and Algol, and offers speeds over 100 times faster than BASIC interpreters.

ACTION! is ideal for games, music processing, real-time control, and many other applications. But if what you're really looking for is raw speed in compiled code, ACTION! is just for you. There's more . . . ACTION! comes with a 128-column screen editor which rivals word processing programs, as well as a monitor mode which allows you to choose between on-line activities. ACTION!'s unique one-pass compiler will accept code from memory, disk, or cassette, and ACTION! has the ability to include source library files.

ACTION! is provided in cartridge form only. Introductory price for ATARI Version $99.00

Call or write for availability of Apple II and Commodore 64 Versions.

A Strong Software Family

Other major systems software products from OSS include:

BASIC A+	the only logical upgrade to Atari BASIC with extra features for games and business programs....$80.00
C/65	the first native mode "small c" compiler for Atari and Apple computers....$80.00
MAC/65	the finest and fastest complete 6502 macro assembler/editor package you can buy....$80.00
BUG/65	a powerful, self-relocatable debugger. FREE with MAC/65....$34.95

And More...

OS/A+, the first and finest operating system for BOTH Atari and Apple II computers, is **NOW** included **FREE** as a part of every OSS systems software package. OS/A+ features a keyboard-driven, easy-to-use command processor, several simple resident commands, and logical and readable requests for even the most sophisticated utility commands. Versions of OS/A+ for some higher capacity drives are available at extra cost.

NOTE: Unless otherwise noted, all **OSS** products require 48K and at least one disk drive.

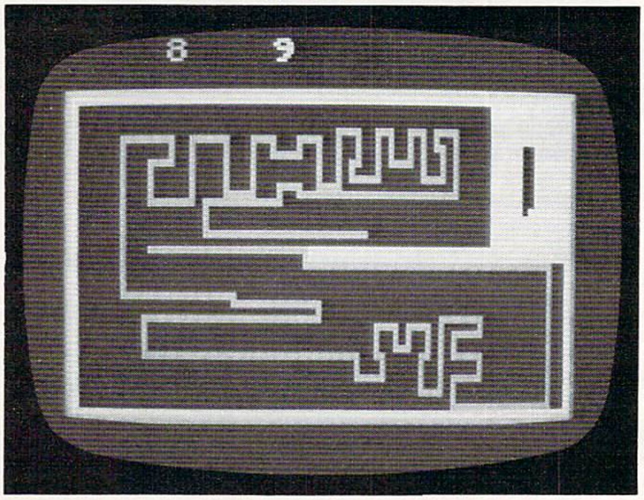
ASK YOUR DEALER, or call or write for our brochure.

ATARI, APPLE II, and TINY C are trademarks of Atari, Inc., Apple Computer, Inc., and Tiny C Associates, respectively. MAC/65, C/65, BASIC A+, BUG/65, and OS/A+ are trademarks of Optimized Systems Software, Inc.

OSS

Optimized Systems Software, Inc. 10379 Lansdale Avenue • Cupertino • California • 95014 • (408) 446-3099

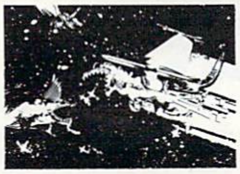
```
1410 DATA 232,56,169,0,162,2,160
1420 DATA 46,32,23,57,169,0,162
1430 DATA 79,160,46,32,23,57,169
1440 DATA 0,162,79,160,3,32,23
1450 DATA 57,169,0,162,2,160,3
1460 DATA 32,23,57,162,15,189,204
1470 DATA 54,157,13,54,202,16,247
1480 DATA 32,81,57,166,205,208,82
1490 DATA 134,206,174,31,208,224,7
1500 DATA 240,3,76,11,58,174,10
1510 DATA 54,32,53,57,166,207,240
1520 DATA 60,174,4,54,172,4,54
1530 DATA 202,189,33,54,208,39,136
1540 DATA 192,1,208,34,32,81,57
1550 DATA 169,2,133,87,162,6,160
1560 DATA 1,169,0,32,225,56,169
1570 DATA 194,160,54,32,199,56,173
1580 DATA 31,208,205,31,208,240,248
1590 DATA 76,11,58,224,0,208,207
1600 DATA 134,207,76,240,54,166,205
1610 DATA 189,33,54,208,3,76,133
1620 DATA 56,166,205,189,120,2,168
1630 DATA 166,205,192,14,208,10,169
1640 DATA 255,157,25,54,169,0,157
1650 DATA 21,54,192,7,208,10,169
1660 DATA 1,157,21,54,169,0,157
1670 DATA 25,54,192,11,208,10,169
1680 DATA 255,157,21,54,169,0,157
1690 DATA 25,54,192,13,208,10,169
1700 DATA 1,157,25,54,169,0,157
1710 DATA 21,54,24,189,13,54,125
1720 DATA 21,54,157,13,54,24,189
1730 DATA 17,54,125,25,54,157,17
1740 DATA 54,168,189,13,54,170,169
1750 DATA 0,32,1,57,201,0,240
1760 DATA 99,166,205,224,1,208,85
1770 DATA 173,50,54,240,80,230,206
1780 DATA 165,206,201,3,240,72,201
1790 DATA 2,240,19,173,10,210,41
1800 DATA 1,174,26,54,208,3,24
1810 DATA 105,2,141,51,54,76,61
1820 DATA 56,173,51,54,73,1,170
1830 DATA 188,52,54,173,26,54,240
1840 DATA 16,201,1,208,6,206,18
1850 DATA 54,76,178,55,238,18,54
1860 DATA 76,178,55,173,22,54,201
1870 DATA 1,208,6,206,14,54,76
1880 DATA 178,55,238,14,54,76,178
1890 DATA 55,166,205,32,129,57,76
1900 DATA 133,56,166,205,189,29,54
1910 DATA 32,195,56,188,17,54,189
1920 DATA 13,54,170,169,0,32,232
1930 DATA 56,166,205,232,138,56,237
1940 DATA 4,54,48,2,162,0,134
1950 DATA 205,76,76,55,72,162,96
1960 DATA 169,12,157,66,3,32,86
1970 DATA 228,162,96,169,3,157,66
1980 DATA 3,169,1,157,68,3,169
1990 DATA 54,157,69,3,104,157,75
2000 DATA 3,41,240,73,16,9,12
2010 DATA 157,74,3,32,86,228,96
2020 DATA 141,0,54,96,162,96,157
2030 DATA 68,3,152,157,69,3,169
2040 DATA 255,157,72,3,157,73,3
2050 DATA 169,9,157,66,3,32,86
2060 DATA 228,96,134,85,133,86,132
2070 DATA 84,96,32,225,56,162,96
2080 DATA 169,11,157,66,3,169,0
2090 DATA 157,72,3,157,73,3,173
2100 DATA 0,54,32,86,228,96,32
2110 DATA 225,56,162,96,169,7,157
2120 DATA 66,3,169,0,157,72,3
2130 DATA 157,73,3,32,86,228,96
2140 DATA 32,225,56,173,0,54,141
2150 DATA 251,2,162,96,169,17,157
2160 DATA 66,3,169,12,157,74,3
2170 DATA 169,0,157,75,3,32,86
2180 DATA 228,96,224,255,240,5,160
2190 DATA 168,140,1,210,160,255,142
2200 DATA 0,210,136,208,253,202,208
2210 DATA 245,162,255,160,0,140,1
2220 DATA 210,96,162,0,142,45,54
2230 DATA 169,2,133,87,174,45,54
2240 DATA 24,189,33,54,125,46,54
2250 DATA 32,195,56,189,41,54,170
2260 DATA 160,0,152,32,232,56,238
2270 DATA 45,54,174,45,54,236,4
2280 DATA 54,208,222,169,5,133,87
2290 DATA 96,166,205,189,37,54,141
2300 DATA 2,210,169,104,141,3,210
2310 DATA 189,196,2,72,169,54,157
2320 DATA 196,2,162,255,32,53,57
2330 DATA 32,53,57,32,53,57,166
2340 DATA 205,104,157,196,2,162,255
2350 DATA 32,53,57,166,205,222,33
2360 DATA 54,169,0,141,3,210,169
2370 DATA 1,133,207,96,169,150,160
2380 DATA 54,32,199,56,96,169,18
2390 DATA 32,150,56,169,32,141,12
2400 DATA 54,169,194,141,200,2,169
2410 DATA 0,170,160,4,32,225,56
2420 DATA 169,115,160,54,32,199,56
2430 DATA 32,188,57,169,131,160,54
2440 DATA 32,199,56,162,176,32,53
2450 DATA 57,173,196,2,72,162,1
2460 DATA 189,196,2,157,195,2,232
2470 DATA 224,4,208,245,104,141,199
2480 DATA 2,206,12,54,208,225,24
2490 DATA 173,11,54,105,176,141,159
2500 DATA 54,173,4,54,105,176,141
2510 DATA 171,54,174,50,54,189,192
2520 DATA 54,105,128,141,190,54,169
2530 DATA 2,32,150,56,162,0,169
2540 DATA 0,168,32,225,56,169,56
2550 DATA 160,54,32,199,56,169,151
2560 DATA 160,54,32,199,56,32,188
2570 DATA 57,169,75,160,54,32,199
2580 DATA 56,169,161,160,54,32,199
2590 DATA 56,169,173,160,54,32,199
2600 DATA 56,32,188,57,169,94,160
2610 DATA 54,32,199,56,172,31,208
2620 DATA 192,7,240,249,204,31,208
2630 DATA 240,251,192,5,208,24,174
2640 DATA 4,54,232,224,5,208,10
2650 DATA 162,2,173,50,54,73,1
2660 DATA 141,50,54,142,4,54,76
2670 DATA 11,58,192,3,208,16,174
2680 DATA 11,54,232,224,6,208,2
2690 DATA 162,1,142,11,54,76,11
2700 DATA 58,76,220,54
```



Maneuvering around the roadblocks. ©

Extra Atari Enjoyment NOW

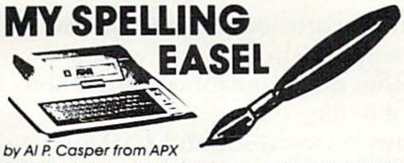
THE PROGRAM STORE™



DEMON ATTACK
by Dave Johnson from Imagic
Marooned on the ice planet Krybor, watch eerie creatures stream overhead and hover ominously. Attack and destroy them—or be destroyed! For 1-2 players with ever increasing dangers. Blast 'em and survive!
41656 ROM Cartridge ~~$39.95~~
$31.96 20% Off Til July 10

BURIED BUCKS
from Analog
Fast action "daredevil" game! Equipping your helicopter with explosives, you intend to blast the gold. But your arch-enemy plans to keep the treasure and you buried by dropping loads of dirt from his World War II bomber. Incredible 99 levels of joystick controlled action for 1-2 players. It's a race for the bucks as you avoid the falling dirt!
32364 16K Tape or 35873 16K Disk ~~$29.95~~
$23.96 20% Off Til July 10

JOURNEY TO THE PLANETS
from JV Software, Inc.
Space, adventure and arcade action game. Disembark on inviting planets — a different adventure awaits on each planet. Designed for your puzzle solving intellect enhanced by arcade action for excitement.
43568 32K Tape or 43579 32K Disk $29.95



MY SPELLING EASEL
by Al P. Casper from APX
Paint landscapes by typing letters and spelling words. Non-readers can make pictures by touching any key. Young readers are challenged to type a whole word and display its picture. 4 levels, ages 3-10. Requires Atari Basic; joystick optional.
16K Tape or 24K Disk $29.95

JUMPMAN
by Randy Glover from Epyx
New science fiction game! Jumpman must save all 30 levels of Jupiter Headquarters. Scale ladders, girders and perilous ropes while fighting off demonic destroyers. Joystick controlled, 5 game variations, 8 speed, music, sound effects and graphics. The ultimate test of reflexes for 1-4 players.
41713 32K Disk ~~$39.95~~
$31.96 20% Off Til July 10

SURVIVOR
by Richard Carr from Synapse Software
Fly your Starwedge Cruiser solo, with one or two gunners and/or a propulsion engineer, across a scrolling galaxy of danger and excitement. Only your skill can make you the Survivor. Requires joystick.
27739 16K Tape or 34872 16K Disk ~~$34.95~~ $27.96
33523 ROM Cartridge ~~$44.95~~ $35.96
20% Off Til July 10

CLIPPER
from Program Design, Inc.
Around the Horn in 1850! Choose the vessel, cargo, crew and course. Use your skill to overcome storms, icebergs, illness, delays, doldrums, mutiny and more. Voice-narrated, high adventure requires joystick.
32689 24K Tape $24.95
32690 32K Disk $29.95

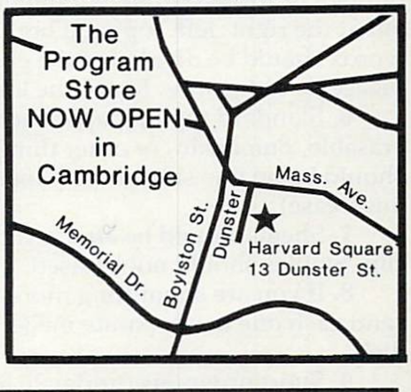
MATHEMATIC-TAC-TOE
by Nadav Caine from APX
Offbeat way for kids 8-16 years old to practice computational skills. It provides addition, subtraction, multiplication and division drills on 15 difficulty levels with time limit ranges from 2 to 23 seconds. Two players compete to fill a tic-tac-toe row with correct solutions.
21452 16K Tape or 21463 24K Disk $15.95

POKERSAM
by Jerry White from Don't Ask
This poker player can talk! POKERSAM narrates every hand, naming upturned cards, announcing bets and wise-cracking. He's a real character with a gift for gab!
28336 Disk $24.95

FINANCIAL WIZARD
by Bill McLachlin from Computari
Superb personal finance program with 26 expense categories. Simple check entry and search by name, category, number or tax deduction. Features check balancer, printing and audit. Instructions on screen eliminates need for constant referral to manual.
15118 24K Disk $59.95

INSIDE ATARI BASIC
by Bill Carris from Reston
A fast, friendly approach by the training director of Atari Home Computers. This workbook is an introduction to key concepts of BASIC.
37213 Softcover book $12.95

COMPUTE'S FIRST BOOK OF ATARI GRAPHICS
from Compute!
Extensive capabilities are available with Atari graphics. Games, tutorials, programs, and more, for the beginner to the most advanced ready to type right in and use!
23746 Softcover Book $12.95



Over 2500 Programs for TRS-80, ATARI 400/800, APPLE, IBM & VIC 20.

For Information Call
202-363-9797
Visit our other stores:
829 Bethel Rd., Columbus, OH
Seven Corners Center, Fall Church, VA
W. Bell Plaza, 6600 Security Blvd., Baltimore, MD
White Flint Mall, Rockville Pike, Rockville, MD
Harvard Square, 13 Dunster St., Cambridge, MA
Westmoreland Mall, Rte. 30 East, Greensburg, PA

THE PROGRAM STORE T.M.

Franchise Openings Available in selected cities.
Coming soon to Philadelphia

To Order Call Toll-Free
800-424-2738
MAIL ORDERS: Send check or M.O. for total purchase price, plus $2.00 postage & handling. D.C., MD & VA: add sales tax. Charge cards. Include all embossed information.

MasterCard VISA

© 1983 The Program Store Inc.

THE PROGRAM STORE • Dept. 10-07-3 • Box 9582 • 4200 Wisconsin Avenue, N.W. • Washington, D.C. 20016

Item # Title Tape/Disk/Rom/Book Price

Postage $2.00
Total
☐ CHECK ☐ VISA
☐ MASTERCARD
Computer ____ For Atari 400/800 unless otherwise specified.

Name
Address
City ____ State ____ Zip ____
Card # ____ Exp. ____

COMPUTE's Author Guide

Most of the following suggestions serve to improve the speed and accuracy of publication. **COMPUTE!** is primarily interested in new and timely articles on VIC, Apple, PET/CBM, Commodore 64, Atari, Timex/Sinclair, TI/99-4A, and Radio Shack Color Computer. We are much more concerned with the content of an article than with its style. Above all, articles should be clear and well-explained.

The guidelines below will permit your good ideas and programs to be more easily edited and published:

1. The upper left corner of the first page should contain your name, address, telephone number, and the date of submission.

2. The following information should appear in the upper right corner of the first page. If your article is specifically directed to one make of computer, please state the brand name and, if applicable, the BASIC or ROM or DOS version(s) involved. In addition, *please indicate the memory requirements of programs.*

3. The underlined title of the article should start about 2/3 of the way down the first page.

4. Following pages should be typed normally, except that in the upper right corner there should be an abbreviation of the title, your last name, and the page number. For example: Memory Map/Smith/2.

5. All lines within the text of the article must be double- or triple-spaced. A one-inch margin should be left at the right, left, top, and bottom of each page. No words should be divided at the ends of lines. And please do not justify. Leave the lines ragged.

6. Standard typing paper should be used (no erasable, onionskin, or other thin paper) and typing should be on one side of the paper only (upper- and lowercase).

7. Sheets should be attached together with a paper clip. Staples should not be used.

8. If you are submitting more than one article, send each one in a separate mailer with its own tape or disk.

9. Short programs (under 20 lines) can easily be included within the text. Longer programs should be separate listings. *It is essential that we have a copy of the program, recorded twice, on a tape or disk.* Please use high quality 10 or 30 minute tapes with the program recorded on both sides. The tape or disk should be labeled with the author's name, the title of the article, and, if applicable, the BASIC/ROM/DOS version(s). Atari tapes should specify whether they are to be LOADed or ENTERed. We prefer to receive Apple programs on disk rather than tape. On the other hand, tapes are preferred for the Radio Shack computer. Tapes are fairly sturdy, but disks need to be enclosed within plastic or cardboard mailers (available at photography, stationery, or computer supply stores).

It is far easier for others to type in your program if you use CHR$(X) values and TAB(X) or SPC(X) instead of cursor manipulations to format your output. For five carriage returns, FOR I = 1 TO 5:PRINT:NEXT is far more "portable" to other computers with other BASICs and also easier to type in. And, instead of a dozen right-cursor symbols, why not simply use PRINT SPC(12)? A quick check through your program – making these substitutions – would be greatly appreciated by your editors and by your readers.

10. A good general rule is to spell out the numbers zero through ten in your article and write higher numbers as numerals (1024). The exceptions to this are: Figure 5, Table 3, TAB(4), etc. Within ordinary text, however, the zero through ten should appear as words, not numbers. Also, symbols and abbreviations should not be used within text: use "and" (not &), "reference" (not ref.), "through" (not thru).

11. For greater clarity, use all capitals when referring to keys (RETURN, TAB, ESC, SHIFT), BASIC words (LIST, RND, GOTO), and three languages (BASIC, APL, PILOT). Headlines and subheads should, however, be initial caps only, and emphasized words are not capitalized. If you wish to emphasize, underline the word and it will be italicized during typesetting.

12. Articles can be of any length – from a single-line routine to a multi-issue series. The average article is about four to eight double-spaced, typed pages.

13. If you want to include photographs, they should be either 5x7, black and white glossies or color slides.

14. We do not consider articles which are submitted simultaneously to other publishers. If you wish to send an article to another magazine for consideration, please do not submit it to us.

15. **COMPUTE!** pays between $50 and $600 for published articles. In general, the rate reflects the length of the article. Payment is made upon acceptance of an article. Following submission (Editorial Department, **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403) it will take from four to eight weeks for us to reply. If your work is accepted, you will be notified by a letter which will include a contract for you to sign and return. *Rejected manuscripts are returned to authors who enclose an SASE.*

16. If your article is accepted and you have since made improvements to the program, please submit an entirely new tape or disk and a new copy of the article reflecting the update. We cannot easily make revisions to programs and articles. It is necessary that you send the revised version as if it were a new submission entirely, but be sure to indicate that your submission is a revised version by writing "Revision" on the envelope and the article.

17. **COMPUTE!** does not accept unsolicited product reviews. If you are interested in serving on our panel of reviewers, contact the Review Coordinator for details.

PRODUCTS FOR ATARI* 400/800
FROM ELCOMP

HOFACKER

Books + Software for ATARI, VIC-20, OSI, SINCLAIR, TIMEX

Payment: check, money order, VISA, MASTER-CHARGE, Euroscheck.
Orders from outside USA: add 15% shipping. CA residents add 6.5% tax
*ATARI is a registered trademark of ATARI Inc.
*VIC-20 is a registered trademark of Commodore

ELCOMP PUBLISHING, INC
53 Redrock Lane
Pomona, CA 91766
Phone: (714) 623 8314

BOOKS for ATARI Computers

ATARI BASIC – Learning by Using
An excellent book for the beginner. Many short programs and learning exercises. All important features of the ATARI computers are described (screen drawings, special sounds, keys, paddles, joysticks, specialized screen routines, graphics, sound applications, peeks, pokes, and special stuff). Also suggestions are made that challenge you to change and write program routines.
Order #164 — $7.95

Games for the ATARI Computer
This book describes advanced programming techniques like player-missile-graphics and use of the hardware-registers. Contains many ready to run programs in BASIC and one called GUNFIGHT in machine language.
Order #162 — $7.95


How to program your ATARI in 6502 Mach.Lang.
Introduction to machine language for the BASIC programmer
Order #169 — $9.95

FORTH on the ATARI – Learning by Using
Introduction, programs, applications, learning exercises.
Order #170 — $7.95


A Look into the Future – ASTROLOGY
on your ATARI 800.
How to calculate your own horoscope.
Order #171 — $9.95

Our catalog is free with every order. Send $1.00 and SASE for catalog only.

SUPERMAIL
(500 addr. on 1 disk)
Completely written in FORTH. Comes on autoboot disk. No cartridge, no DOS, no FORTH Language required!
Order #7312 — $49.00

SUPERINVENTORY
(1000 terms per disk)
Completely written in FORTH. Same as above. (Disk only)
Order #7320 — $49.00

BUSIPACK-1
(written in FORTH). Complete order entry, inventory, mailing and invoicing. (Disk only)
Order #7313 — $98.00

ATAMEMO
Datablock to keep track of your appointments. (D+C)
Order #7310 — $29.95

ATCASH
Convert your ATARI 800 into a powerful cash register. (Disk only)
Order #7307 — $49.95

Invoicing progr. i. BASIC
Order #7201 (C) $29.95
Order #7200 (D) $39.95

Mailing List in BASIC
Order #7212 (C) $19.95
Order #7213 (D) $24.95

Inventory control in BASIC
Order #7214 (C) $19.95
Order #7215 (D) $24.95

Microcomputer Hardware Handbook (845 pages)
Descriptions, pinouts and specifications of the most popular microprocessors and support chips.
A MUST for the hardware buff.
Order-No. 29
$14.95

SOFTWARE IN MACHINE LANGUAGE for ATARI

ATMONA-1
This is a machine language monitor that provides you with the most important commands for programming in machine-language. Disassemble, dump (hex and ASCII), change memory location, block transfer, fill memory block, save and load machine-language programs, start programs. Printer option via three different interfaces.

Order #7022	cassette version	$19.95
Order #7023	disk version	$24.95
Order #7024	cartridge version	$59,00

ATMONA-2
This is a tracer (debugger) that lets you explore the ATARI RAM/ROM area. You can stop at previously selected address, opcode, or operand. Also very valuable in understanding the microprocessor. At each stop, all registers of the CPU may be changed. Includes ATMONA-1.

Order #7049	cassette version	$49.95
Order #7050	disk version	$54.00

ATMAS
Macro-Assembler for ATARI-800/48k. One of the most powerful editor assemblers on the market. Versatile editor with scrolling. Up to 17k of source-Code. Very fast, translates 5k source-code in about 5 seconds. Source code can be saved on disk or cassette. (Includes ATMONA-1)

Order #7099	disk version	$89.00
Order #7999	cartridge version	$129.00

ATAS
Same as ATMAS but without macro-capability. Cassette-based.

Order #7098	32k RAM	$49.95
Order #7998	48k RAM	$49.95

ATEXT-1
This wordprocessor is an excellent buy for your money. It features screen oriented editing, scrolling, string search (even nested), left and right margin justification. Over 30 commands. Text can be saved on disk or cassette.

Order #7210	cassette version	$29.95
Order #7216	disk version	$34.95
Order #7217	cartridge version	$69.00

GUNFIGHT
This game (8k machine-language) needs two joysticks. Animation and sound. Two cowboys fight against each other. Comes on a bootable cassette.
Order #7207 — $19.95

FORTH for the ATARI

FORTH from Elcomp Publishing, Inc. is an extended Fig-Forth-version, Editor and I/O package included. Utility package includes decompiler, sector copy, Hexdump (ASCII), ATARI Filehandling, total graphic and sound, joystick program and player missile. Extremely powerful!
Order #7055 — disk — $39.95

Floating point package with trigonometric functions (0 – 90°).
Order #7230 — disk — $29.95

Learn-FORTH from Elcomp Publishing, Inc.
A subset of Fig-Forth for the beginner. On disk (32k RAM) or on cassette (16k RAM).
Order #7053 — $19.95

Expansion boards for the APPLE II

The Custom Apple + Other Mysteries
A complete guide to customizing the Apple Software und Hardware
Order-No. 680 — $24.95

We also stock the boards which are used in the book "The Custom Apple" (barebords)

6522 I/O Board No. 605	$39.00
EPROM Burner No. 607	$49.00
8K EPROM/RAM Board No. 609	$29.00
Prototyping board for the Apple II No. 604	$29.00
Slot repeater board for the Apple II No. 606	$49.00

Order two boards and get the book free!

Care and Feeding of the Commodore PET
Eight chapters exploring PET hardware. Includes repair and interfacing information. Programming tricks and schematics.
Order #150 — $9.95

Hardware – ADD–ONS for ATARI

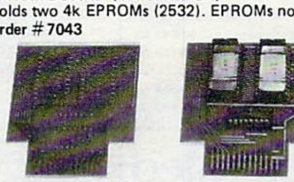
PRINTER INTERFACE
This construction article comes with printed circuit board and software. You can use the EPSON printer without the ATARI printer interface. (Works with gameports 3 and 4).
Order #7211 — $19.95

RS-232 Interface for your ATARI 400/800
Software with connector and construction article.
Order #7291 — $19.95

EPROM BURNER for ATARI 400/800
Works with gameports. No additional power supply needed. Comes compl. assembled with software (2716, 2732, 2532).
Order #7042 — $179.00

EPROM BURNER for ATARI 400/800 KIT
Printed circuit board incl. Software and extensive construction article.
Order #7292 — $49.00

EPROM BOARD (CARTRIDGE)
Holds two 4k EPROMs (2532). EPROMs not included.
Order #7043 — $29.95



EPROM BOARD KIT
Same as above but bare board only with description.
Order #7224 — $14.95

ATARI, VIC-20, Sinclair, Timex and OSI

NEW – for your ATARI 400/800
Astrology and Biorhythm for ATARI (cass. or disk).
Order-No. 7223 — $29.95
Birth control with the ATARI (Knaus Ogino)
Order-No. 7222 — cass. or disk — $29.95

Books + Software for VIC-20 (requires 3KRAM Exp.)
No. 4870 Wordprocessor for VIC-20, 8KRAM — $19.95
No. 4883 Mailing List for VIC-20, 16K RAM — $14.95
No. 176 Tricks for VICs (book, 115 pages) — $ 9.95

INPUT/OUTPUT Programming with your VIC, No. 4886 — $9.95
Miniassembler for VIC-20 No. 4896 — $19.95
Tennis, Squash, Break, No. 4881 — $9.95
Runfill for VIC, No. 4894 — $9.95
TIC TAC VIC, No. 4880 — $9.95
GAMEPACK I (3 Games) No. 4881 — $14.95
Dual Joystick Instr. No.4885 $9.95

Progr. in 6502 Machine Language on your PET+CBM
2 complete Editor/Assemblers (Source code 3 hexdump + description plus a powerful machine language monitor (Hexdump). Order-No. 166 — $19.95

Universal Experimenter Board for the VIC-20
(Save money with this great board). This board plugs right into the expansion slot of the VIC-20.
Order #4844 — $18.95

Software for SINCLAIR ZX-81 and TIMEX 1000
#2399 Machine Language Monitor — $9.95
#2398 Mailing List — $19.95

Programming in BASIC and machine language with the ZX-81 (82) or TIMEX 1000.
Order-No. 174 (book) $ 9.95

BOOKS FOR OSI

No. 157 1. Book of Ohio	$7.95
No. 158 2. Book of Ohio	$7.95
No. 159 3. Book of Ohio	$7.95
No. 160 4. Book of Ohio	$7.95
No. 161 5. Book of Ohio	$7.95

#151 8K Microsoft BASIC Ref. Man.	$9.95
#152 Expansion Handbook for 6502 and 6802	$9.95
#153 Microcomputer Appl. Notes	$9.95

Complex Sound Generation
New revised applications manual for the Texas Instruments SN 76477 Complex Sound Generator.
Order #154 — $6.95

Small Business Programs Order #156
Complete listings for the business user. Inventory, Invoice Writing, Mailing List and much more. Introduction to Business Applications. — $14.90

computer mail order

computer mail order west

PRINTERS
SMITH CORONA
Call on $50.00 Factory Rebate!
- TP-1 — $499.00
- Tractor Feed — $129.00
- Ultrasonic I Typewriter — $439.00

C.ITOH(TEC)
- GX-100 — $209.00
- Prowriter 8510P — $379.00
- Prowriter 8510S — $579.00
- Prowriter 1550P — $699.00
- Prowriter 1550S — $749.00
- Starwriter F10-40 — $1259.00
- Printmaster F10-55 — $1649.00

OKIDATA
- 82A — CALL
- 83A — CALL
- 84 (Parallel) — CALL
- 84 (Serial) — CALL
- 92 — CALL
- 93 — CALL

IDS
- MicroPrism — $649.00
- 132 (Fully Configured) — $1599.00
- 80 (Fully Configured) — $1399.00

Call for other configurations.

STAR
- Gemini 10 — $349.00
- Gemini 15 — $489.00
- Serial Board — $75.00

DAISYWRITER
- 2000 (Letter Quality) — $1049.00
- Tractor Feed — $109.00

DIABLO
- 620 — $999.00
- 630 — $1769.00

TeleVideo

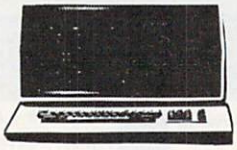

TERMINALS
- 910 — $559.00
- 912C — $689.00
- 920C — $739.00
- 925C — $719.00
- 950 — $929.00

COMPUTERS
- 800A — $1259.00
- 802 — $2649.00
- 802H — $4695.00
- 806/20 — $4949.00
- 816/40 — $8999.00
- 803 — $1949.00
- 1602G — $3399.00
- 1603 — CALL

NEC
COMPUTERS
- 8000 — CALL
- 8001A — $719.00
- 8031 — $719.00
- 8012 — $549.00

PRINTERS
- 8023 — $419.00
- 7710/7730 — $2249.00
- 3510/3530 — $1549.00

MONITORS
- JB-1260 — $119.00
- JB-1201 — $149.00
- JC-1212 — $299.00
- JC-12-202 — $299.00
- JC-1203 — $569.00

EAGLE


Call on Eagle 8 Bit & 16 Bit Computers & Software

MODEMS
HAYES
- Smart — $219.00
- Smart 1200 (1200 Baud) — $549.00
- Chronograph — $199.00
- Micromodem 100 — $309.00
- Micromodem II — $279.00
- Micromodem II(with term) — $299.00
- Smart Com II — $99.00

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

ANCHOR
- Mark I (RS-232) — $79.00
- Mark II (Atari) — $79.00
- Mark III (T.I.-99) — $109.00
- Mark IV (CBM-PET) — $125.00
- Mark V (Osborne) — $95.00
- Mark VI (IBM-PC) — $179.00
- Mark VII (Auto Ans./Auto Dial) — $119.00
- TRS-80 Color Computer — $99.00
- 9 Volt Power Supply — $9.00
- Mark VIII — CALL

MONITORS
AMDEK
- 300G — $159.00
- 300A — $169.00
- 310G — $179.00
- 310A — $169.00
- Color I — $299.00
- Color II — $599.00
- Color IIA — $799.00
- Color III — $349.00

BMC
- 12AU 12" Green — $79.99
- 1401 13" Color(Mid-Res) — $369.00
- 9191 U 13" Composite — $329.00

TAXAN
- RGB 1 (Hi-Res) — $299.00
- RGB III — $499.00
- 12 N Green — $119.00
- 12 A Amber — $125.00

ZENITH
- ZVM 121 — $95.00
- ZT-1 Terminal — $369.00

U.S.I.
- Pi 1, 9" Green — $99.00
- Pi 2, 12" Green — $119.00
- Pi 3, 12" Amber — $159.00
- Pi 4, 9" Amber — $139.00
- 1400C 14" Color — $299.00

PANASONIC
- JR200U 32K Pers. Computer — $309.00

MONITORS
- TR-120 12" Hi-Res Green — $159.00
- CT-160 10" Dual Mode Color — $299.00
- DT-D1000 10" RGB — $349.00
- DT-D1300 13" RGB/Compos. — $429.00

IBM
NEC
3550 PRINTER...$1549

PERCOM/TANDOM DRIVES
- 5¼" 160K Disk Drive — $249.00
- 5¼" 320K Disk Drive — $299.00

AMDEK
- 310A Amber Monitor — $169.00
- 310G Green Monitor — $179.00
- Amdisk (3¼" Drive) — $679.00
- XY Plotter — $649.00
- Color II — $599.00

AST
- Combo Card 64K — $429.00

QUBIE
- PC Keyboard — $219.00

SOFTWARE
- MicroPro WordStar/MailMerge — $349.00
- I.U.S. Easywriter II — $249.00
- I.U.S. Easyspeller — $129.00
- Peach Package (GL/AP/AR) — $419.00

PROFESSIONAL SOFTWARE
- IBM-PC Word Processing — $319.00

CONTINENTAL SOFTWARE
- 1st Class Mail/Form Letter — $89.00
- The Home Accountant Plus — $109.00

SYNAPSE
- File Manager — $119.00

IMMS
- Privecode Phone — $239.00

READY FORMS
- 1"or2" Address Labls(Tract.Feed) — $9.95
- 15" Report Paper(Tract.Feed) — $24.95
- 8½" Blnk Wht Paper(Tract.Feed) — $19.95
- 8½" Blnk Env(Tract.Feed) — $14.95

SANYO
- MB1000 Computer — $1599.00

INCLUDES FREE SOFTWARE! MicroPro, WordStar, Calcstar, Mail Merge & Report Star.

- MB 160 Add on Drive — $539.00
- 5500 Letter Quality Printer — $699.00

TIMEX SINCLAIR 1000


$39.95

- 16K Memory Module — $44.95
- Vu-Calc — $17.95
- Check Book Manager — $13.95
- The Organizer — $14.95
- The Budgeter — $13.95
- Stock Option — $14.95
- Loan & Mortgage Amortizer — $12.95
- Mindware Printer — $109.00

ORBYTE SOFTWARE
- Graphics — $12.99
- Home Budgeter — $12.99
- Home Inventory — $12.99
- Income Tax — $14.99
- Mega Mind — $15.99
- Salvo — $12.99
- The Quiz — $12.99
- Words — $12.99

commodore
CALL ON CBM 64 $100.00 REBATE!

- 1520 Color Printer/Plotter — $169.00
- 1525 80 Column Printer — $339.00
- 1530 Datasette — $69.00
- 1541 Single Disk Drive — $339.00
- 1600 VIC Modem — $95.00
- 1610 VIC Term 40 — $49.00
- 1650 AD/AA Modem — $159.00
- 1701 14" Color Monitor — $269.00
- 1311 Joysticks (each) — $5.99
- 1312 Paddles — $11.99
- 1110 VIC 8K — $42.00
- 1111 VIC 16K — $69.00
- 1011 RS232 Interface — $42.00
- 1211 Super Expander — $53.00
- 1906 Super Alien — $23.00
- 1910 Radar Rat Race — $23.00
- 1917 VooDoo Castle — $29.00
- 1922 Cosmic Cruncher — $35.00
- 1923 Gorf — $29.00
- 1924 Omega Race — $30.00
- 110 VIC 20 Reference Guide — $15.00
- CBM 64 Reference Guide — $18.00

EASY BUSINESS SERIES 64
- Easy File — $79.00
- Easy Finance — $39.00
- Easy Mail — $39.00
- Easy Script — $79.00
- Word Machine/Name Machine — $23.00

PROGRAMMER SERIES 64
- Assembler — $39.00
- Logo — $79.00
- Pilot — $79.00
- Pet Emulator — $25.00
- Screen Editor — $25.00
- Video/Music Support — $39.00

ART AND MUSIC SERIES 64
- Music Machine — $25.00
- Music Composer — $25.00
- Meta Music I — $79.00

COMMERCIAL DATA
- Motor Mouse/20 — $23.00
- Centipode/20 — $23.00
- Froggee/20 — $23.00
- Froggee/64 — $23.00

CREATIVE SOFTWARE
- Astro Blitz — $32.00
- Black Hole — $32.00
- Trashman — $32.00
- Home Finance — $27.00
- Home Inventory — $13.00

UMI
- Amok — $30.00
- Meteor Run — $40.00
- Alien Blitz — $30.00
- Viterm A — $16.00
- The Alien — $16.00

CBM 64
$399

VIC 20
$99.

- 4064 — $569.00
- 12840 — $649.00
- 12880 — $749.00
- 8032 — $1039.00
- 4032 — $739.00
- 8096 Upgrade Kit — $369.00
- 9000 — $1499.00
- 2031 — $449.00
- 4040 — $949.00
- 8050 — $1279.00
- 8250 — $1639.00
- 9060 (5 Meg. HD) — $1999.00
- 9090 (7.5 Meg. HD) — $2199.00
- 4022 — $389.00
- 8023 — $589.00
- 6400 Letter Quality Printer — CALL
- Spell Master — $99.00
- Z Ram adds CP/M® & 64K — $549.00
- Silicon Office — $749.00
- Calc Result — $159.00
- The Manager — $209.00
- The Soft Rom — $129.00
- Jinsam — CALL
- ADA 1600 CBM to Parl. Int. — $89.00
- ADA 1450 CBM to Serial Int. — $99.00

PROFESSIONAL SOFTWARE
- Power — $79.00
- Info Pro — $219.00
- Word Pro 2 Plus — $159.00
- Word Pro 3 Plus — $199.00
- Word Pro 4 Plus — $299.00
- Word Pro 5 Plus — $299.00
- Administrator — $379.00
- Word Pro 64 — $79.95

CARDCO VIC 20/CBM 64
- Light Pen — $32.00
- Cassette Interface — $29.00
- Parallel Printer Interface — $64.00
- 3Slot Expans.Interface(20 only) — $32.00
- 6Slot Expans.Interface(20only) — $79.00

HES
- Shamus (ROM) — $29.00
- Protector (ROM) — $32.00
- Robot Panic (ROM) — $29.00
- Pirates (ROM) — $29.00
- HES Writer (ROM) — $29.00

Call on our Large Selection of VIC 20 & CBM 64 Software, such as: **EPYX, Microspec and Kansas City Software.**

SHARP


PC-1500 POCKET COMPUTER $169.

PC1250.... $89.00

- CE150 Printer, Plotter and Cassette Interface Unit — $172.00
- CE152 Cassette Recorder — $62.00
- CE155 8K Ram Expans. Mod. — $94.00
- CE125 Printer/Micro Cass. — $129.00
- Statistics Pack — $49.00

VISA

computer mail order west
800-648-3311
IN PA. CALL (717)327-9575

DEPT. 706, 477 E. THIRD ST., WILLIAMSPORT, PA 17701

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

Time Bomb

Doug Smoak

This program is deceptively short – it is easily one of the best games we've ever seen for the VIC. You're in a maze, larger than your screen will show. You must move through it, trying to defuse a ticking bomb hidden somewhere at the top of the puzzle. As you move, the screen will move, but you must learn from your mistakes or the ticking will grow more shrill until all is lost. For the unexpanded VIC.

You play "Time Bomb" against the clock. You start at the bottom of a maze, which is about three times the size of the VIC's screen. At the top of the maze is a time bomb ticking away. The closer it gets to blowing up, the higher pitched the ticking becomes. If you reach the bomb, you must steer the pointer into it to defuse it. If you are successful, you have a go at the same maze, but with the bomb in a different place and with a shorter fuse. This continues until you run out of time. If you fail to defuse it, you get a new maze and a new bomb with a longer fuse.

Friends I've played this with usually don't consider it a game for competition. Instead, they become back-seat drivers, telling the player where to go and pulling for him or her at every turn.

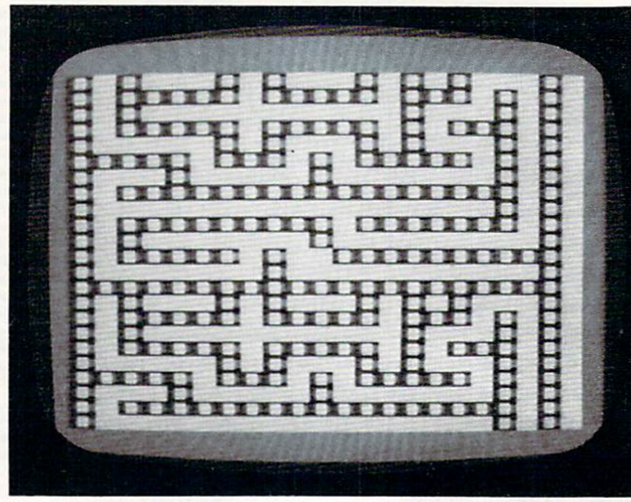
Time Bomb is quite challenging to a player's memory of spatial relationships. People who are at first intimidated by seeing only a portion of the maze quickly become accustomed to thinking ahead and remembering the dead ends and clear paths through the maze. An ability to recall the good and bad moves is crucial to getting into the later rounds.

I started thinking about this game when I saw Kenneth Szajda's "Mastermaze" (**COMPUTE!**, February 1983). I wanted to create something more challenging than a single screen maze, but I didn't want to duplicate his game and I also had to consider the VIC's smaller memory. I then hit onto the idea that makes this game so entertaining: to make the maze larger than the screen and bring it on and off the display by scrolling it out of a much larger block of memory.

How The Idea Came

It sounded great, but how would I do it? The secret lies in a short machine code routine that is "called" to update the display whenever the player goes up or down in the maze. It does this so quickly that I used the BASIC joystick routine from *COMPUTE!'s First Book of VIC* just to keep things at a reasonable pace.

There are actually three separate machine language routines that are represented by the DATA statements. One fills the maze area with the proper character, another fills the screen's "color RAM" with the proper color, and the third one scrolls the maze. I could have used BASIC POKEs to do all these things, but the time consumed would be too great. It would be impossible to use POKEs to do the scrolling of the maze with enough speed to be any fun at all.



Searching for a time bomb.

When typing in the program, be sure to SAVE it before you RUN it, since a typo in the DATA statements could cause you to lose the whole program. Be very careful as you enter the DATA statements. If you have a bug in the program, it is most likely in the DATA statements, so look there first.

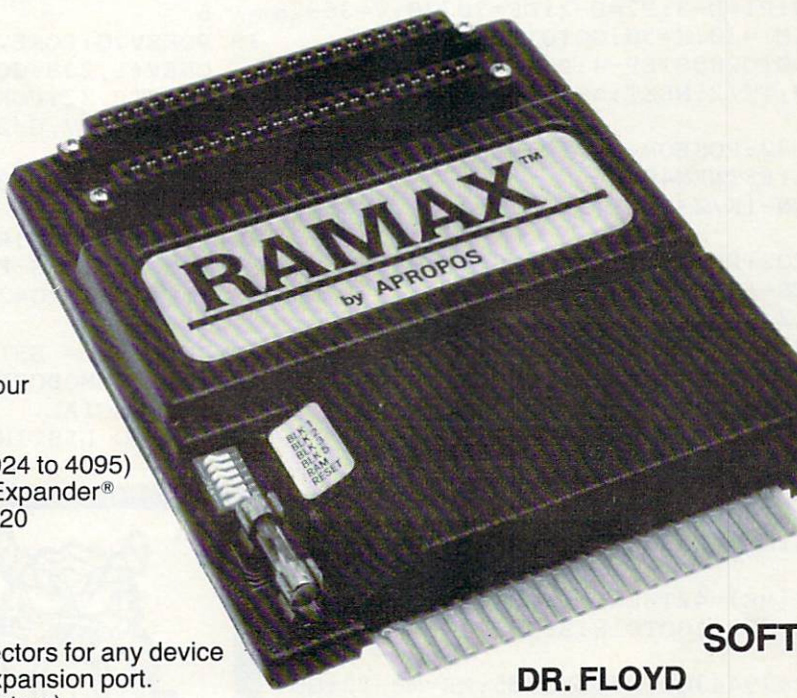
When you do RUN it, there will be a slight pause while the machine language parts are POKEd into the cassette buffer. Then the screen should clear, and the words "Making Maze" should appear. Because of the size of the maze, the VIC needs almost a minute to draw it, so be patient. When the maze is complete, a little musical announcement alerts you to begin playing. Don't give up if you are eliminated on the first round; it takes a while to get used to looking ahead in the maze and planning your route.

If you don't want to type in the program, I will make copies for the usual $3, a cassette, and

RAMAX™

by APROPOS

The **ONLY** MEMORY your **VIC-20®** will need



FEATURES

- A full 27k bytes of RAM (added to VICs 5k equals 32k.)
- Fully switchable in sections:
  - BLK 1 switches 8k (Adr. 8192 to 16383)
  - BLK 2 switches 8k (Adr. 16384 to 24575)
  - BLK 3 switches 8k (Adr. 24576 to 32767)
  - BLK 5 allows/disallows your 8k ROM (games) (Adr. 40960 to 49152)
  - RAM switches 3k (Adr. 1024 to 4095)
- May be used with Super Expander® games or **ANY** other VIC-20 compatible cartridge.
- Built in RESET switch.
- Fuse protected.
- Totally self-contained.
- 2 duplicate extension connectors for any device normally plugged into the expansion port. (BLK 5 is switched to connectors)
- Very low power usage. (.150 amp max.)
- High reliability gold plated connectors.
- 6 month parts and labor warranty.
- Factory service. - Extended service always available.

THIS SUPERB PLUG-IN GIVES YOUR VIC-20 REAL POWER AND EXPANDABILITY

FOR ONLY $149.00 Shipping included

10 DAY SATISFACTION OR YOUR MONEY BACK GUARANTEE

WE ARE NOW OFFERING "RAMAX Jr." (19k), which is identical to RAMAX in EVERY way, except the top 8k (BLK 3) is not incorporated. Our introduction price is **$129.00**, shipping included.

WE SERVICE WHAT WE SELL

TO ORDER:

Send Check or Money Order For the Total
Calif. residents add 6% tax.
Phone orders: CALL **(805) 482-3604 24 HRS.**
For credit card orders, include all information on card.
or **contact your local dealer.**

Foreign orders, add $15.00.
All items shipped from stock.
DEALER INQUIRIES WELCOME

SOFTWARE

DR. FLOYD

Psychoanalysis by computer? — well, not quite, but Dr. Floyd will carry on a conversation with you using psychoanalytic techniques giving the appearance of artificial intelligence. Requires 16k RAM or more.
$14.95 shipping included.

WORD PLAY

"WORDPLAY" is a collection of programs which allow the user to make original stories, write a form of Japanese poetry, play the fun game of Animal (children love this one), and create jargon. A bonus secret message (cypher) program is also included. In a word, "WORDPLAY" is a bargain.
Requires 16k RAM or more.
$14.95 shipping included.

TYPE FOR YOUR LIFE

With more challenge than an arcade game, learn to type up to 75+ words/min. (User selectable, but no FOOLING AROUND allowed). TEXT IS WIDELY VARIED SINCE IT COMES FROM THE PROGRAM TAPE. Action color graphics with sound fix your eyes to the screen (away from your fingers - clever!) Your man rows your boat up stream as fast as you can type. Maintain speed and destroy the Sea Monster; slow down and he will get you. Runs on the unexpanded VIC.
$14.95 shipping included.

All software is on high quality cassettes and is replacement guaranteed.

VIC-20 & SUPER EXPANDER are registered trademarks of Commodore Business Machines, Inc.

APROPOS TECHNOLOGY

350 N. Lantana Ave., Suite 821
Camarillo, CA 93010

a postpaid mailer:

Doug Smoak
303 Heyward St.
Columbia, SC 29201

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

Time Bomb

```
2 POKE56,24:POKE55,103:GOSUB29
3 D=37154:P1=D-3:P2=D-2:DF=3Ø72Ø:V=36878:S=V-4:M1=3Ø:X=5Ø:GOTO19
4 FORT=24ØTO2Ø8STEP-4:POKES,T:FORTT=ØTO3Ø:POKEV,TT/2:NEXT:NEXTT:POKES,Ø:ME=7932
5 POKEOM,32:POKEOM+DF,1Ø:POKEME,M1:POKEME+DF,7:IFFTHEN4Ø
6 K=K+1:ON-(K/2<>INT(K/2))GOTO8:IFK>6ØØTHEN37
7 FORT=1TO2:POKEV,T*4:POKES+1,128+K/5:NEXT:POKES+1,Ø
8 POKED,127:P=PEEK(P2)AND128:JØ=-(P=Ø)
9 POKED,255:P=PEEK(P1):J1=-((PAND8)=Ø):J2=-((PAND16)=Ø):J3=-((PAND4)=Ø)
1Ø IFJØTHENC=1:M1=62:GOTO14
11 IFJ1THENC=22:M1=22:GOTO14
12 IFJ2THENC=-1:M1=6Ø:GOTO14
13 IFJ3THENC=-22:M1=3Ø
14 OM=ME:ME=ME+C:C=Ø
15 IFPEEK(ME)<>32ANDPEEK(ME)<>42THENME=OM
16 IFPEEK(ME)=42THENF=1:GOTO5
17 ON-(ME>7921)GOTO18:SYS887:ME=ME+22:GOTO5
18 ON-(ME<7944)GOTO5:SYS9Ø5:ME=ME-22:GOTO5
19 DIMA(3):A(Ø)= 2:A(1)=-44:A(2)=-2:A(3)=44:WL=2Ø9:HL=32:SC=6228:A9=6943
2Ø SYS861:PRINT"{CLR} {DOWN}MAKING MAZE"
21 FORT=SC+21TO7679STEP22:POKET,32:NEXT:FORT=SCTOSC+21:POKET,32:NEXT
22 J=INT(RND(1)*4):X3=J
23 B=A9+A(J)
24 IFPEEK(B)=WLTHENPOKEB,J:POKEA9+A(J)/2,HL:A9=B:GOTO22
25 J=(J+1)*-(J<3):IFJ<>X3THEN23
26 J=PEEK(A9):POKEA9,HL:IFJ<4THENA9=A9-A(J):GOTO22
27 TB=SC+INT(RND(Ø)*2Ø)+22Ø:ON-(PEEK(TB)<>32)GOTO27:POKETB,42
28 SYS83Ø:POKE828,2Ø4:POKE829,28:SYS923:GOTO4
29 FORI=83ØTO974:READA:POKEI,A:NEXT:RETURN
3Ø DATA169,238,141,15,144,169,Ø,133,251,169,15Ø,133,252,16Ø,Ø,169,1Ø,145,251,2ØØ,2Ø8
31 DATA251,23Ø,252,165,252,2Ø1,152,2Ø8,241,96,169,84,133,251,169,24,133,252,16Ø,Ø,169
32 DATA2Ø9,145,251,2ØØ,2Ø8,251,23Ø,252,165,252,2Ø1,3Ø,2Ø8
33 DATA241,96,173,6Ø,3,56,233,22,176,3,2Ø6,61,3,141,6Ø,3,56,176,19,234,173,6Ø,3,24,1Ø5
34 DATA22,144,3,238,61,3,141,6Ø,3,24,144,1,234,169,Ø,133,Ø,169,3Ø,133,1,173,6Ø,3,133
35 DATA254,173,61,3,133,255,169,Ø,133,253,16Ø,Ø,177,254,164,253,145,Ø,132,253,23Ø,253
36 DATA234,2Ø8,2,23Ø,1,23Ø,254,2Ø8,2,23Ø,255,169,32,197,1,2Ø8,227,96
37 POKEV,15:FORT=255TO127STEP-2:POKES,T:POKEV-9,255:FORG=1TO1Ø:NEXT
38 POKEV-9,242:FORG=1TO1Ø:NEXT:POKEV-9,24Ø:NEXT:POKEV-1,22Ø:FORG=15TOØSTEP-.Ø5
39 POKEV,G:POKEV+1,G*1Ø:NEXT:POKEV-1,Ø:POKEV+1,238:GOSUB42:RUN
4Ø POKETB,32:POKEV-1,253:FORG=3ØTOØSTEP-.15:POKEV,G/2:NEXT:X=X+5Ø:IFX>449THENX=45Ø
41 POKEV-1,Ø:F=Ø:K=X:R=R+1:GOSUB42:GOTO27
42 PRINT"{HOME}ROUND"R"{LEFT} ":PRINT"{DOWN}PRESS F7 ":A$="":GETA$:ON-(A$<>"{F7}")GOTO42:RETURN
```

UNDERLINE = SHIFT,
[ ] = COMMODORE KEY,
{ }= SPECIAL.
REFER TO LISTING CONVENTIONS ©

Copy-Writer Word Processor

Louis F. Sander

Copy-Writer is a full-featured program that merits the close attention of any Apple, Commodore 64, PET, or CBM owner needing a word processor. It has most of the useful features found in other word processing systems, plus several that are unique. *Copy-Writer* is easy to use, clearly documented, and comes with a guarantee that future enhancements, no matter how extensive, will be offered to registered users for a nominal disk copying charge.

The developers of *Copy-Writer*, the IDPC Co. of Philadelphia, originally wrote it in 1979. Since then, it has been used by professional programmers and technical writers, and extensively revised. It seems to be a solid program with good features and few bugs.

Since a detailed discussion of software features can be confusing to those who haven't used similar programs, let's start our review with something easy to comprehend. *Copy-Writer* is available for the PET/CBM with 2040, 4040, 8050, or PEDISK II drives; it supports all ROM variations and virtually any printer from any manufacturer. The program is also available for the Apple II with 3.2 or 3.3 disks, for the Apple III, and for the Commodore 64 with 1541 or PEDISK III drives. The version I have worked with is for the PET/CBM with PEDISK II drive, but the other versions are identical to it in all important respects.

A Special, Tailored Program

The software consists of one diskette and a small, but thorough, manual. There are no ROMs or other plug-in devices. The diskette cannot be copied, but that is not a problem – you use it to create a machine language program configured especially for your own ROMs, screen size, keyboard, and printer, and *that* program can be saved and copied without limit. If you change printers or upgrade your computer, you load the master diskette, answer eight simple questions, and within a few seconds you have a reconfigured and copyable program in memory.

The 44-page instruction manual is remarkable for its clarity and usefulness, as well as for its brevity. In spite of never having learned to use a commercial word processor before, I was able to sit down with it and quickly master most of its features. The manual contains a useful table of contents and a well-thought-out index, both of which are quite helpful in using the program itself. It is written for the reader who is familiar with elementary computer operation, and who knows what he wants his word processor to accomplish.

Using *Copy-Writer* is exceptionally easy and straightforward. There is no need for sheets of stick-on key labels, or for a two-pound reference manual. When the system comes up, a "paper scale" appears at the bottom of the screen; tab stops are marked on it in reverse field. The number of the text line at the top of the screen and the number of lines still available in memory also appear down here, as does a line for special commands and error messages.

Editing Features

Routine typing and text editing is done in the Edit Mode, in which the cursor moves freely about the screen. The PET's familiar cursor control keys are used to move, insert, and delete characters. The up arrow, left arrow, HOME and RVS keys are used for opening up lines, moving words around, etc., and it is very easy to remember which key does what.

The STOP key puts the system in the "Command Mode." In that mode, the cursor jumps to a special area at the bottom of the screen and waits for your instructions. There are about 30 of these, most having to do with disk file handling, searching and replacing text, and printing. *Copy-Writer*'s authors have made the commands *very* easy to remember: A means append a file, D means down scroll, S means save a file on disk, etc. For those who haven't used the commands enough to have memorized them, they are listed in a table in the index of the instruction manual, which also notes the page where the command is described in detail.

The process of entering text and moving it around is similar to that in most good word processors. *Copy-Writer* seems to have all the necessary features in this area, and most of the typical frills.

Copy-Writer has two separate buffers for handling changes and text movement. Buffer #1 is used for moving entire para-

graphs from place to place and is activated from Command Mode. Buffer #2 is used in Edit Mode and is ideally suited for moving words and short phrases, although it has a 1000-character capacity. To use it, you place the cursor on the first letter to be saved, and press the shifted left arrow key. Letter-by-letter, text is "sucked" from the screen into the buffer.

When you've picked up everything you want to move, you put the cursor wherever you want it, and press the unshifted left arrow key, which automatically inserts the buffered text at that point. The text remains in the buffer, so you can insert it as many places as you'd like. This feature can save time and keystrokes whenever the same phrase is used repeatedly in the text (as are the words "*Copy-Writer*" in this review). You can put such a phrase in the buffer and use one key to print it out every time it is used.

Another feature worthy of note is the ability to input repeated characters, such as a series of dashes, just by entering: a special character, the character to be repeated, and the number of repeats desired. There is also a graphics mode which allows dot-by-dot control over printers having that capability. Neither of these features is a necessity, but their presence is an indication of the authors' attention to detail in making the program useful.

Copy-Writer is extremely powerful for formatting the printed page. Format control is done by special commands embedded in the text, and there are many to choose from. Once again, the commands are easily understood by themselves, alphabetically listed in the index, and well-described in the manual. AP means append a file, LM sets the left margin, HD defines a page heading, and so on for over two dozen commands. The power here is really impressive – you can print things in double columns (like this magazine is printed), customize page breaks (based on a variety of conditions), and on and on.

By using a special format command, you can send individual hex characters to your printer, for control of character size, impact, or whatever features the printer happens to have. The capability is completely general, so if you know what character code switches your printer into Martian Hieroglyphic mode, you can put it there whenever you want. This is a very desirable feature and it's one of many desirable features available on this most impressive product.

Copy-Writer
CGRS Microtech
P.O. Box 102
Langhorne, PA 19047
$145

©

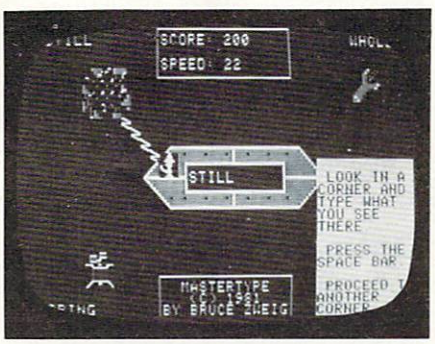
Mastertype

Tina Halcomb

Mastertype, by Bruce Zweig (Atari version by Aric Wilmunder), makes learning touch-typing fun.

As an educational program, *Mastertype* is impressive. It is menu-driven and the lesson plan begins with basic keyboard and finger placement presentations. In the manual supplied with *Mastertype* are illustrations and diagrams which clearly show proper finger placement. Your skill builds from this point. You begin to practice typing single letters or simple three- to four-letter words. Once you are comfortable with these, you move on to longer words, numbers, and symbols.

Each lesson can run in either of two modes. The Beginner mode displays single letters only, and the Normal mode asks you to type the complete word and press the space bar.



You can even create your own word lists to practice with words that are related to your occupation. After first booting *Mastertype*, you will see this option offered, and you respond by typing an "M" (make your own lesson). Each word list consists of 40 words ranging from one to nine characters. You *must* enter 40 words – there's no way around it. If you make any errors when entering the words, you may edit them after you complete the 40th word. Once satisfied with your customized lesson, you can name it and save it on your disk.

In each lesson you control the mode, the speed, and any upper- and lowercase variations.

Battle Of Words

But what makes this a truly effective, pleasant learning experience is the *game* it becomes. You, the Command Ship, are hovering out in space. Look out! Four enemy words have just appeared in the corners of your computer screen. They're sending satellites, missiles, and atomic meteors to destroy you. You are not helpless, though. If you can type the enemy words correctly, you can eliminate them. You won't destroy the enemy word unless you fire your laser before or just as the enemy word releases its weapon.

Even when you need not be particularly concerned with the exact path of your laser, you must type the word correctly before the laser is released.

As soon as you successfully defend your ship by destroying all enemy words, you can see your game score and typing speed. You may get so involved in playing the game that you won't even realize you're acquiring a very useful skill.

Mastertype is available on disk for a 32K Atari or 48K Apple.

Mastertype
Lightning Software
P.O. Box 11725
Palo Alto, CA 94306
$39.95

©

Claim Jumper For Atari

Fred Pinho

Synapse Software has produced a number of high-quality game packages in the past. In *Claim Jumper,* the company has done nothing to damage that reputation. *Claim Jumper* is basically a combination shoot-'em-up and strategy game for one or two players. I found it fascinating.

Two cowboys (brown and pink) are controlled by the players using joysticks. The cowboys act out their lives on a playfield consisting of: (a) two banks (one for each player); (b) an assay office; (c) two hospitals; (d) assorted other houses and cacti. At intervals during the game, a gold nugget will appear. The object is to pick up the nugget, take it to the assay office, and exchange it for cash. The cash must then be taken and deposited in your bank "to buy a house" (ten bills are needed). House buying is completely immaterial to the game. The object is solely to collect the ten bills.

While this all sounds easy, it can be frustratingly difficult. While performing these functions, you must also dodge your opponent's bullets while avoiding numerous obstacles. Although diagonal movement is the fastest, the cowboys can shoot only when moving horizontally or vertically.

The animation of the cowboys is relatively crude, but this in no way detracts from the game. If you shoot your opponent, his hat comes off and he drops whatever he is carrying (gold or money). He is then transported to one of the two hospitals (chosen by moving the joystick left or right). While in the hospital, he cannot shoot. After a very brief stay, however, he recovers completely and can re-enter the fray.

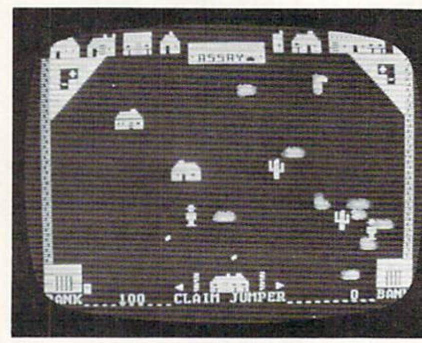
How They Get Your Treasure

The real fun in the game, and much of the strategy, involves the obstacles. Specifically, watch out for the notorious snakes and tumbleweeds. Shortly after the game starts, these objects start to appear. The pink snakes chase the brown cowboy while the brown tumbleweeds stalk the pink cowpoke. If you touch the opposite creature, you will be paralyzed for two seconds and drop whatever you are carrying. This allows your opponent to steal your treasure. After the two seconds are up, you're fit and ready to go as before. Mercifully, you will have a brief period of immunity which will allow you to move away from your pursuing tormentors. The creature graphics are very well done.

How can you fight off these unpleasant intrusions? One way is to shoot things. Plugging them with your trusty "shootin' ahrn" will turn the creature into the opposite type, which then promptly goes off after your opponent. However, there is a second tactic which adds considerably to the game. This involves dropping seeds or eggs. To do so, you must stop and then press your joystick button. The brown cowboy drops tumbleweed seeds. If a snake eats a seed, it turns into a tumbleweed. Conversely, the pink cowboy can drop snake eggs. If a tumbleweed hits the egg, it turns into a snake. A maximum of six eggs and six seeds can be on the screen at any one time. If you drop a seventh seed or egg, the oldest one disappears.

As you can imagine, *Claim Jumper* gets quite hectic. In addition to pursuing the gold, shooting creatures and your opponent, and dropping eggs/seeds, the cowboy must also avoid other obstacles. If the brown cowboy touches anything pink or a cactus, he experiences the two-second freeze. Pink obstacles include a pink house, the pink bank, one of the playfield borders and, of course, snake eggs. The opposite is true for the pink cowboy.

It's more difficult to explain this game than to play it. The nuances of the game are easily learned, and built-in prompts help during play. When you pick up the gold, a flashing arrow indicates where to deposit it in



Cowboys, cacti, snakes, Western buildings, and drifting tumbleweeds set the scene for Claim Jumper.

the assay office. Similarly, once you get the money, another arrow indicates the correct bank.

Option Menus

There are also two option menus for game variations. The first features the normal game and two options: Buy Bullets and Head Start. In Buy Bullets, you no longer have an unlimited supply of bullets. You start with ten. When you run out, you must take money to the bullet store to buy ten more. Head Start allows you to start with five bills already in the bank.

The second menu allows you to select either the normal game or a single-player game with two levels of difficulty. In the single-player game, you must destroy all the snakes and tumbleweeds before you are paralyzed for the third (and last) time. The problem is that you start with no bullets and thus must buy some with your gold. Again, you can buy only ten bullets at a time.

Although I have high praise for this game, I do have one gripe involving the scoring system. The winner is the first player to reach 25,000 points. You score 100 points for each snake or tumbleweed that you convert. But the first player to reach ten bills then gets 20,000 points! Somehow it doesn't seem fair. Here you are in a close battle with each player at about 6000 points and nine bills. Then your opponent gets one more bill, and you lose 26,000 to 6000! The final score does not reflect the intensity and closeness of such a contest. There *is* an option to continue the game until 50,000 which does help somewhat. A better way might be to receive a given number of points for each bill deposited until the winner reaches the target score.

Claim Jumper
Synapse Software
5327 Jacuzzi St.
Suite 1
Richmond, CA 94804
Requires 16K RAM
Disk/Cassette
$34.95

©

Courseware Report Card And *Educational Software Directory*

Sheila Cory

Just a couple of years ago, the greatest concern of parents and educators interested in the educational use of microcomputers was which computer to buy from the great variety available. Hardware selection was a major topic of discussion whenever the subject of computers came up. More and more, however, the questions posed these days relate to software selection. A number of schools and homes already have their computers and are trying to determine the best use of their machines.

Fortunately, some excellent educational software is now on the market. But parents and educators need to sift through an enormous amount of software in order to find what's best for their application. Educational software directories and evaluation journals have recently been developed to cope with this problem. This review looks at two of them.

Courseware Report Card

Courseware Report Card provides in-depth reviews and evaluations of both elementary and secondary software. Unlike many software review journals, it reviews software for more than one computer: Apple, Atari, PET/CBM, and TRS-80.

Selection of software to review is based primarily on software publishers' response to requests for review copies. A secondary source is software made available by teachers, software dealers, or educational media centers. To be of value to all people interested in educational computing, the journal covers a cross-section of subject area and grade level.

Most *Courseware Report Card* reviews are prepared by members of the editorial staff, all of whom are former teachers with experience in curriculum evaluation and design. A few of the reviews are prepared by nonstaff members. These reviews are signed, and the qualifications of the reviewer are listed in the introduction.

Graded In Six Categories

The standard format of the reviews makes it easy to find information. A box at the top of the first page of each review highlights subject area, grade level, type of program (drill and practice, tutorial, or game), system requirements, price, and publisher's name and address. A box at the bottom of the page gives a letter grade (A through F) for performance, ease of use, error handling, appropriateness, documentation, and educational value. These two boxes, plus a short summary of the program, provide all the information necessary to decide whether or not to read the entire review.

The reviews proper begin with a description of the program, explaining exactly what the student sees as the program progresses. Screen representations and photographs make it easy to visualize what the text is describing. The "performance" section of the evaluation explores the overall quality of the program. Errors of punctuation in the text, problems with speed of operation, and sound that can't be turned off are examples of comments made in this section.

Ease Of Use And Error Handling

The "ease-of-use" comments focus on standardization of commands, use of menus in the program, and other programming possibilities that make the program as easy as possible for the user. How well a program accepts input from the keyboard is among the criteria evaluated under "error handling."

The value of the computer over other modes of instruction is addressed under "appropriateness." The editors take a firm position on the appropriateness of drill and practice software by having a policy of never awarding a grade higher than C to any software designed for drill and practice unless it is enhanced by additional features. (This view is not universally shared, but it is constantly discussed.)

Documentation And Educational Value

The paragraph of each review covering documentation looks at the books, pamphlets, and other hard copy provided to supplement the software. "Educational value," perhaps the most important of all of the evaluation components, examines whether the particular area covered by the software has any real place in the curriculum.

The evaluations included in *Courseware Report Card* are well written and complete. However, you must keep in mind (as the introduction to the journal states) that much software evaluation is subjective. There is room for disagreement, and you should make the decision of whether to use software with your students or your own children only after looking at the software from beginning to end yourself.

Apple, Atari, PET/CBM, And TRS-80

This review of *Courseware Report Card* is based on the first issue, dated September 1982. *Course-*

COM-STAR F/T

Tractor Friction Printer

only $279

• Lowest price quality tractor friction printer in the U.S.A. • Fast 80 characters per second • 40, 46, 66, 80, 96, or 132 characters per line spacing • Prints labels, letters, graphs, and tables • List your programs • Print out data from modem services

Deluxe COMSTAR F/T PRINTER — $279.00

The Comstar is an excellent addition to any micro-computer system. (Interfaces are available for Apple, VIC-20, Commodore-64, Pet, Atari 400 and 800, and Hewlett Packard) At only $279, the Comstar gives you print quality and features found only on printers costing twice as much. Compare these features.

• **BI-DIRECTIONAL PRINTING** with a LOGIC SEEKING CARRIAGE CONTROL for higher through-put in actual text printing. 80 characters per second.

• **PRINTING VERSATILITY:** standard 96 ASCII character set plus block graphics and international scripts. An EPROM character generator includes up to 224 characters.

• **INTERFACE FLEXIBILITY:** Centronics is standard. Options include EIA RS232C, 20mA Current Loop. (Add $20.00 for RS232)

• **LONG LIFE PRINT HEAD:** 100 million character life expectancy.

• **THREE SELECTABLE CHARACTER PITCHES:** • 10, 12 or 16.5 characters per inch. 132 columns maximum. Double-width font also is standard for each character pitch.

• **THREE SELECTABLE LINE SPACINGS:** 6, 8 or 12 lines per inch.

• **PROGRAMMABLE LINE FEED:** programmable length from 1/144 to 255/144 inches.

• **VERTICAL FORMAT CONTROL:** programmable form length up to 127 lines, useful for short or over-sized preprinted forms.

• **FRICTION AND TRACTOR FEED:** will accept single sheet paper.

• **224 TOTAL CHARACTERS**

• **USES STANDARD SIZE PAPER**

If you want more try —

Premium Quality COMSTAR F/T SUPER-10" PRINTER — $329.00

More Features Than MX-80
For $250 Less

For $329.00 you get all of the features of the Comstar plus 10" carriage, 100 cps, 9 x 9 dot matrix with double strike capability for 18 x 18 dotmatrix. High resolution bit image (120 x 144 dot matrix), underlining, backspacing, 2.3K buffer, left and right margin settings, true lower descenders, with super and subscripts, and prints standard, Italic, Block Graphics, special characters, plus 2K of user definable characters. For the ultimate in price performance the Comstar F/T Super 10" leads the pack!

80 COLUMN PRINTER $199

Super silent operation, 60 CPS, prints Hi-resolution graphics and block graphics, expanded character set, exceptionally clear characters, fantastic print quality, uses inexpensive thermal roll paper!

Double Immediate Replacement Warranty

We have doubled the normal 90 day warranty to 180 days. Therefore if your printer fails within "180 days" from the date of purchase you simply send your printer to us via United Parcel Service, prepaid. We will IMMEDIATELY send you a replacement printer at no charge via United Parcel Service, prepaid. This warranty, once again, proves that WE LOVE OUR CUSTOMERS!

15 DAY FREE TRIAL

OTHER OPTIONS

Extra Ribbons	$ 5.95
Roll Paper Holder	32.95
Roll Paper	4.95
5000 Labels	19.95
1100 Sheets Fan Fold Paper	13.95

Add $20.00 shipping, handling and insurance. Illinois residents please add 6% tax. Add $40.00 for CANADA, PUERTO RICO, HAWAII, ALASKA orders. WE DO NOT EXPORT TO OTHER COUNTRIES. Enclose cashiers check, money order or personal check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail available!! Canada orders must be in U.S. dollars.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COMSTAR F/T

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk
lmn opqrstuvwxyz 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890

SUPER-10" ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

ware Report Card/Elementary evaluated 22 programs, including 16 for the Apple, 11 for the Atari, seven for the PET/CBM, and seven for the TRS-80 (many programs are designed to run on more than one computer). *Courseware Report Card/Secondary* also evaluated 22 programs – 18 for the Apple, eight for the Atari, seven for the PET/CBM, and ten for the TRS-80. Future editions of the *Courseware Report Card* promise to be quite interesting: software publishers will have opportunity to respond to reviews, and teachers and administrators will have a chance to hear corroborating or dissenting opinions. A forum for such a dialogue is a welcome addition for people excited about possibilities in educational microcomputing.

Courseware Report Card
(five issues per year)
Educational Insights, Inc.
150 W. Carob St.
Compton, CA 90220
Elementary Edition $49.50
Secondary Edition $49.50
both editions $95
single copies $12.50

Educational Software Directory

The *Educational Software Directory* is designed to help educators determine exactly what software is available in their subject area. It can answer such questions as "How can I use the computer when teaching a poetry class?" or "Is there any software available for the PET that teaches grammar?" It tells what software is available, but makes no attempt to evaluate it.

The directory covers programs for grades kindergarten through 12 and includes all categories of educational software (except programs intended primarily for administrative purposes). Software selected for inclusion in the directory met a set of criteria: the software had to be usable for the grade level for which it was intended, the subject matter had to be appropriate to the learning environment and to the computer medium itself, and the listing of the software in the catalog had to be clear and complete. No software was actually examined in the process of compiling the directory; descriptions given in software catalogs were used instead.

Software listed in the directory includes general software (encompassing more than one subject), basic living skills, business education, computer literacy, courseware development (teacher utilities), fine arts, foreign language, language arts, library skills, math, science, and social studies. Each entry in the directory contains the program name, publisher's name, availability (which suppliers sell it), release date, grade level, hardware configuration required, storage medium (diskette or cassette), the computer language it's written in, price, availability of the source code (original program code), and a description of the program.

The value of this book results from the ease with which information can be found. *Educational Software Directory* is excellent. It has both a subject and a title index and cover markings to allow the user to locate a specific subject quickly. Addresses and the policies of publishers and distributors of educational software are also listed, making purchase of desired software easy.

Educational Software Directory
Libraries Unlimited, Inc.
P.O. Box 263
Littleton, CO 80160
$22.50

©

Legionnaire For Atari

E. P. McMahon

Chris Crawford has created a playable, fast, and enjoyable war game called *Legionnaire*. This game is sure to be compared with his magnum opus, *Eastern Front*, and, indeed, there are some similarities. He has retained the attractive features of fine-scrolling across a detailed map and the simple joystick input command concept from *Eastern Front*. But there are significant differences. The most striking difference is that *Legionnaire* is realtime. That is, once START is pressed, the enemy launches its attack and does not stop until the game is over.

Legionnaire is a simulation of Roman-barbarian conflict during Julius Caesar's campaigns in Gaul. You define the scenario by selecting one to ten legions to command. Of the ten, two are cavalry, Crassus and Labienus, and the rest are infantry. Caesar's legion, the Tenth, is the strongest and steadiest.

After choosing the number of legions you wish to command, you must select the tribes of barbarians to be the enemy. The barbarians come as infantry and cavalry, and range from the inept Aedui and sword-fodder Auscii up to the very challenging Helvetii and Huns. Once the order of battle is defined (by joystick), each group of combatants is placed on the map in (almost) random locations.

The Barbarians Are On The March

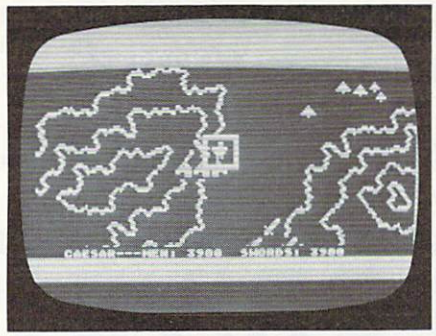
Before pressing START, you

move the hollow square cursor over the map to locate the units and inspect the terrain so you can plan your strategy. Roman units appear in an orange-pink color, and the barbarians are blue. Infantry is symbolized by swords, cavalry by horse heads, and Caesar's unit by an eagle. As you move the cursor to the edge of the screen, the map will fine-scroll under the cursor to show the entire 2¼ by 3½ screen map. Details on the map include effectively visual elevation contour lines and various forest symbols.

Now push START. A drumbeat signals that the barbarians are on the march. They continue to march and attack until they are all eliminated or until Caesar is destroyed. They march whether or not you give orders to your troops. It is in this sense that the game is played in realtime.

Let's examine the differences from *Eastern Front* for a moment. *Legionnaire*'s continuous action and ten units make it a reasonably fast game (it takes roughly between 2 to 15 minutes to play). It is fast enough when battle is joined to keep the interest of an arcade-game aficionado, but it also rewards good tactics enough to give those of us with slower reflexes a chance to win. Good tactics lead to fewer command corrections or panic moves.

While commanding your units, you should be aware of the effects of fatigue, slope, forests, and the differences in direct and flank attacks. Some units tire easily when marching or fighting and must rest to recover strength. Some units break up easily and should be backed up and given a chance to reorganize. Some are better at defense than offense. All these characteristics are spelled out in the 20-page booklet that accompanies the game. The booklet also has short sections on getting started, Caesar's campaigns, and helpful tactics.



Legionnaire

The Legion That Has Trouble Standing Up

Crawford points out that the traits of each tribe are fictitious and are not meant to be historically precise, but do offer you a wide selection of game scenarios. By the way, as you choose more and more legions to command, the added legions are, generally, less and less capable. On your tenth pick, you get Sabinus, whose legion has trouble standing up, let alone fighting. Oh yes. For every legion you pick, the enemy gets two units: one infantry and one cavalry. That can make things interesting.

You might want to play your first game against the Aedui and Auscii to become familiar with the mechanics of the game. Count the loss of any of your units against these tribes as a devastating defeat, and aim for a score in the 30s.

On the other hand, choose the Huns as opponents only when you want the ultimate challenge, feel lucky, and want to play for the least negative score. It doesn't matter which tribe you select for the enemy infantry. The Huns will get to you first and the game will be over before the infantry arrives. When I can reduce the Huns from five to three units before losing, I consider it a success.

The middle choices are fun. One of the most enjoyable games I played was against the Senones ("average troops... neither aggressive nor steady... unreliable when attacked from the flanks or the rear") and the Nervii ("most circumspect...generals value preparation...do not recover from combat shock easily"). The random placement was favorable, and allowed me to deploy my five units in good order at the top of a hill and then rest before the Nervii cavalry arrived.

I counterattacked their uphill charge and hit their flanks with Crassus and Labienus. They broke, and I eventually conquered them with the loss of only one unit, but with permanent reduction in strength to my remaining units. By this time I was on low ground, so I fell back to the forests and allowed the Senones to tire from marching.

They did not immediately attack when they got close, but stopped to rest to rebuild their strength, so I had to attack before they recovered too much. Since the enemy was tired, I was able to break their units away from each other one by one and use the speed of Caesar and the cavalry to surround and then reduce each unit. Without too much fight left in any of my units, I finally won.

Legionnaire is not the historical simulation that *Eastern Front* is, but I think it will appeal to a much broader audience because the game is faster-paced, has fewer units to control, and is, therefore, a faster game. The choice of scenarios makes the game rich enough to hold your interest and offers a variety of skill levels. *Legionnaire* is an entertaining, attractive game in which thinking is more important than fast reflexes.

Legionnaire comes on cassette tape for the Atari 400 and 800, and requires at least 16K RAM.

Legionnaire
Avalon Hill Game Co.
Microcomputer Games Division
4517 Harford Road
Baltimore, MD 21214
$35

©

How To Type COMPUTE!'s Programs

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

Atari 400/800

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

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

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

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

Commodore PET/CBM/VIC

> *Please refer to "A Beginner's Guide To Typing In Programs" for an explanation of the changes in Commodore listing conventions.*

Generally, any PET/CBM/VIC program listings will contain bracketed words which spell out any special characters: {DOWN} would mean to press the cursor-down key; {3DOWN} would mean to press the cursor-down key three times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listing. For example, S would mean to type the S key while holding the shift key. This would result in the "heart" graphics symbol appearing on your screen. Some graphics characters are inaccessible from the keyboard on CBM Business models (32N, 8032).

Sometimes in a program listing, especially within quoted text when a line runs over into the next line, it is difficult to tell where the first line ends. How many times should you type the SPACE bar? In our convention, when a line breaks in this way, the ~ symbol shows exactly where it broke.

All Commodore Machines

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

VIC/CBM 64 Conventions

Set Color To Black	{BLK}	Function Two	{F2}
Set Color To White	{WHT}	Function Three	{F3}
Set Color To Red	{RED}	Function Four	{F4}
Set Color To Cyan	{CYN}	Function Five	{F5}
Set Color To Purple	{PUR}	Function Six	{F6}
Set Color To Green	{GRN}	Function Seven	{F7}
Set Color To Blue	{BLU}	Function Eight	{F8}
Set Color To Yellow	{YEL}	Any Non-implemented Function	{NIM}
Function One	{F1}		

To enter any color code, hold down CTRL and press the appropriate color key. Use CTRL-9 for RVS on and CTRL-0 for RVS off.

8032/Fat 40 Conventions

Set Window Top	{SET TOP}	Erase To Beginning	{ERASE BEG}
Set Window Bottom	{SET BOT}	Erase To End	{ERASE END}
Scroll Up	{SCR UP}	Toggle Tab	{TGL TAB}
Scroll Down	{SCR DOWN}	Tab	{TAB}
Insert Line	{INST LINE}	Escape Key	{ESC}
Delete Line	{DEL LINE}		

When you see an underlined character in a PET/CBM/VIC program listing, you need to hold down SHIFT as you enter it. Since the VIC-20 and Commodore 64 have fewer keys than the PET/CBM, some graphics are grouped with other keys and have to be entered by holding down the Commodore key. If you see any of the symbols in the left column underlined in a listing, hold down the Commodore key and enter the symbol in the right column. Just use SHIFT to enter all other underlined characters.

!	K	←	*	1	E
"	I	↑	PI	2	R
#	T	.	S	3	W
$	@	–	Z	4	H
%	G	=	X	5	J
'	M	<	C	6	L
&	+	>	V	7	Y
\	–	,	D	8	U
;	F	/	P	9	I
?	B	*	N	@	SHIFT*
(	£	+	Q	[	SHIFT+
)	SHIFT-£	0	A	]	SHIFT–

Apple II / Apple II Plus

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

TRS-80 Color Computer

No special characters are used, other than lowercase. When you see letters printed in inverse video (white on black), press SHIFT-0 to enter the characters, and then press SHIFT-0 again to return to normal uppercase typing.

Texas Instruments 99/4

No special control characters are used. Enter all programs with the ALPHA lock on (in the down position). Release the ALPHA lock to enter lowercase text.

Timex TS-1000, Sinclair ZX-81

Study your computer manual carefully to see how to enter programs. Do not type in the letters for each command, since your machine features single-keystroke entry of BASIC commands. You may want to switch to the FAST mode (where the screen blanks) while entering programs, since there will be less delay between lines. (If the blanking screen bothers you, switch to the SLOW mode.)

AARDVARK — THE ADVENTURE PLACE

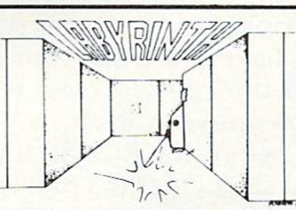
TRS-80 COLOR COMMODORE 64 VIC-20 SINCLAIR/TIMEX TI99

WE CARRY MORE THAN ADVENTURES!!

MAXI-PROS WORD PROCESSING NEW

The easiest to use word processor that I know of. Has all the features of a major word processor (right and left margin justification, page numbering, global and line editing, single, double, triple spacing, text centering, etc.) at a very cheap price because we wrote it in BASIC. Includes 40 page manual and learning guide. Easily modified to handle almost any printer combination. Available on disk or tape for VIC20, COMMODORE64, and TRS-80 COLOR computer. Requires 13k RAM on Vic, 16k EXTENDED on TRS-80 COLOR. $19.95 on tape $24.95 on disk. NEW

GENERAL LEDGER — Complete bookkeeping for a small business. Disk required. For Vic20 (13k), Commodore64, TRS-80 COLOR (16k EXTENDED). $69.95 (Send $1.00 for manual before ordering.)



LABYRINTH — 16K EXTENDED COLOR BASIC — With amazing 3D graphics, you fight your way through a maze facing real time monsters. The graphics are real enough to cause claustrophobia.
Similar game for Timex/Sinclair 16k - hunting treasure instead of monsters $14.95.

ADVENTURE WRITING/DEATHSHIP by Rodger Olsen — This is a data sheet showing how we do it. It is about 14 pages of detailed instructions how to write your own adventures. It contains the entire text of Deathship. Data sheet - $3.95. NOTE: Owners of TI99, TRS-80, TRS-80 Color, and Vic 20 computers can also get Deathship on tape for an additional $5.00.

Dealers—We have the best deal going for you. Good discounts, exchange programs, and factory support. Send for Dealer Information.

Authors—Aardvark pays the highest commissions in the industry and gives programs the widest possible advertising coverage. Send a Self Addressed Stamped Envelope for our Authors Information Package.

ADVENTURES — Adventures are a unique form of computer game. They let you spend 30 to 70 hours exploring and conquering a world you have never seen before. There is little or no luck in Adventuring. The rewards are for creative thinking, courage, and wise gambling — not fast reflexes.

In Adventuring, the computer speaks and listens to plain English. No prior knowledge of computers, special controls, or games is required so everyone enjoys them—even people who do not like computers.

Except for Quest, itself unique among Adventure games, Adventures are non-graphic. Adventures are more like a novel than a comic book or arcade game. It is like reading a particular exciting book where you are the main character.

All of the Adventures in this ad are in Basic. They are full featured, fully plotted adventures that will take a minimum of thirty hours (in several sittings) to play.

Adventuring requires 16k on Sinclair, TRS-80, and TRS-80 Color. They require 8k on OSI and 13k on VIC-20. Sinclair requires extended BASIC. Now available for TI99.

TREK ADVENTURE by Bob Retelle — This one takes place aboard a familiar starship and is a must for trekkies. The problem is a familiar one — The ship is in a "decaying orbit" (the Captain never could learn to park!) and the engines are out (You would think that in all those years, they would have learned to build some that didn't die once a week). Your options are to start the engine, save the ship, get off the ship, or die. Good Luck.

Authors note to players — I wrote this one with a concordance in hand. It is very accurate — and a lot of fun. It was nice to wander around the ship instead of watching it on T.V.

DERELICT by Rodger Olsen and Bob Anderson — For Wealth and Glory, you have to ransack a thousand year old space ship. You'll have to learn to speak their language and operate the machinery they left behind. The hardest problem of all is to live through it.

Authors note to players — This adventure is the new winner in the "Toughest Adventure at Aardvark Sweepstakes". Our most difficult problem in writing the adventure was to keep it logical and realistic. There are no irrational traps and sudden senseless deaths in Derelict. This ship was designed to be perfectly safe for its' builders. It just happens to be deadly to alien invaders like you.

Dungeons of Death — Just for the 16k TRS-80 COLOR, this is the first D&D type game good enough to qualify at Aardvark. This is serious D&D that allows 1 to 6 players to go on a Dragon Hunting, Monster Killing, Dungeon Exploring Quest. Played on an on-screen map, you get a choice of race and character (Human, Dwarf, Soldier, Wizard, etc.), a chance to grow from game to game, and a 15 page manual. At the normal price for an Adventure ($14.95 tape, $19.95 disk), this is a giveaway.

PYRAMID by Rodger Olsen — This is one of our toughest Adventures. Average time through the Pyramid is 50 to 70 hours. The old boys who built this Pyramid did not mean for it to be ransacked by people like you.

Authors note to players — This is a very entertaining and very tough adventure. I left clues everywhere but came up with some ingenous problems. This one has captivated people so much that I get calls daily from as far away as New Zealand and France from bleary eyed people who are stuck in the Pyramid and desperate for more clues.

MARS by Rodger Olsen — Your ship crashed on the Red Planet and you have to get home. You will have to explore a Martian city, repair your ship and deal with possibly hostile aliens to get home again.

Authors note to players — This is highly recommended as a first adventure. It is in no way simple—playing time normally runs from 30 to 50 hours — but it is constructed in a more "open" manner to let you try out adventuring and get used to the game before you hit the really tough problems.

QUEST by Bob Retelle and Rodger Olsen — THIS IS DIFFERENT FROM ALL THE OTHER GAMES OF ADVENTURE!!!! It is played on a computer generated map of Alesia. You lead a small band of adventurers on a mission to conquer the Citadel of Moorlock. You have to build an army and then arm and feed them by combat, bargaining, exploration of ruins and temples, and outright banditry. The game takes 2 to 5 hours to play and is different each time. The TRS-80 Color version has nice visual effects and sound. Not available on OSI. This is the most popular game we have ever published.

32K TRS 80 COLOR Version $24.95.

Adds a second level with dungeons and more Questing.

PRICE AND AVAILABILITY:

All adventures are $14.95 on tape. Disk versions are available on VIC/COMMODORE and TRS-80 Color for $2.00 additional. $2.00 shipping charge on each order.

Please specify system on all orders

ALSO FROM AARDVARK — This is only a partial list of what we carry. We have a lot of other games (particularly for the TRS-80 Color and OSI), business programs, blank tapes and disks and hardware. Send $1.00 for our complete catalog.

AARDVARK

2352 S. Commerce, Walled Lake, MI 48088 / (313) 669-3110

Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.

TRS-80 COLOR TIMEX/SINCLAIR COMMODORE 64 VIC-20

$2.00 shipping on each order

A Beginner's Guide To Typing In Programs

A Change In Commodore Listing Conventions

Commodore owners may notice some slightly unfamiliar characters in a few of the programs this month. We're making a transition to new listing conventions for Commodore machines which should make typing in listings easier.

By next month, all listings will conform to the new conventions. Most of the changes should be fairly easily understood. Brackets still indicate special characters, although a few labels have been changed to make them more nearly match their equivalent keys. For example, {CLEAR} has been replaced with {CLR}. In the old conventions, underlining was used to indicate both shifted characters and (for the VIC and 64) graphics characters accessed with the Commodore logo key. In the new conventions, underlining is used *only* to indicate characters which should be typed while holding down the SHIFT key.

A new set of brackets has been introduced to indicate characters accessed with the Commodore logo key. Whenever you see a character surrounded by ⁅⁆, you should hold down the Commodore logo key and type the indicated key. For example, the graphics ball character is represented by ⁅Q⁆. As with the other brackets, a character preceded by a number indicates how many times you should type the specified character. For example, ⁅22 T⁆ means to hold down the Commodore key and type T twenty-two times.

BASIC Programs

Computers can be picky. Unlike the English language, which is full of ambiguities, BASIC usually has only one "right way" of stating something. Every letter, character, or number is significant. Also, you must enter all punctuation such as colons and commas just as they appear in the magazine. Spacing can be important. To be safe, type in the listings *exactly* as they appear.

Brackets And Special Characters

The exception to this typing rule is when you see the curved bracket, such as "{DOWN}". Anything within a set of brackets is a special character or characters that cannot easily be listed on a printer. When you come across such a special statement, refer to the appropriate key for your computer. For example, if you have an Atari, refer to the "Atari" section in "How to Type COMPUTE!'s Programs."

About DATA Statements

Some programs contain a section or sections of DATA statements. These lines provide information needed by the program. Some DATA statements contain actual programs (called machine language); others contain graphics codes. These lines are especially sensitive to errors.

If a single number in any one DATA statement is mistyped, your machine could "lock up," or "crash." The keyboard, break key, and RESET (or STOP) keys may all seem "dead," and the screen may go blank. Don't panic – no damage is done. To regain control, you have to turn off your computer, then turn it back on. This will erase whatever program was in memory, so always SAVE a copy of your program before you RUN it. If your computer crashes, you can LOAD the program and look for your mistake.

Sometimes a mistyped DATA statement will cause an error message when the program is RUN. The error message may refer to the program line that READs the data. *The error is still in the DATA statements, though.*

Get To Know Your Machine

You should familiarize yourself with your computer before attempting to type in a program. Learn the statements you use to store and retrieve programs from tape or disk. You'll want to save a copy of your program, so that you won't have to type it in every time you want to use it. Learn to use your machine's editing functions. How do you change a line if you made a mistake? You can always retype the line, but you at least need to know how to backspace. Do you know how to enter inverse video, lowercase, and control characters? It's all explained in your computer's manuals.

A Quick Review

1) Type in the program a line at a time, in order. Press RETURN or ENTER at the end of each line. Use backspace or the back arrow to correct mistakes.

2) Check the line you've typed against the line in the magazine. You can check the entire program again if you get an error when you RUN the program.

3) Make sure you've entered statements in brackets as the appropriate control key (see "How To Type COMPUTE!'s Programs" elsewhere in the magazine.)

Sesame Street And Interactive TV

It was like Super TV. I was sitting in a folding chair in the Grand Ballroom at the Hyatt Regency Hotel in Tampa, Florida. In front of me was a giant TV screen. Behind me was an audience numbering in the hundreds. Nearby were all sorts of mysterious high-technology devices. Writhing across the floor, like rainbow-colored pythons from a tropical rainforest, were dozens of cables.

The room darkened. The screen grew bright.

A big, blue, scruffy-looking creature appeared on the screen. It was Cookie Monster. He was wearing a chef's hat and munching a chocolate chip cookie. Crumbs flew in all directions.

It wasn't TV after all. It was a new computer game from the Children's *Computer* Workshop (CCW). CCW is a new division of Children's Television Workshop (CTW), the producers of *Sesame Street*, *Electric Company*, *3-2-1 Contact* and other children's educational programs and materials.

Last year CCW released its first four electron learning disk packages:

Ernie's Quiz (For children 4 to 7)*
Instant Zoo (Ages 7 to 10)*
Spotlight (Ages 9 to 13)*
Mix and Match (For the whole family)

Each package contains four programs that run on a 48K Apple. The starred packages (*) require Integer BASIC. The unstarred package (*Mix and Match*) requires Applesoft BASIC. *Ernie's Quiz* and *Spotlight* require paddles. All packages are more effective if you have a color TV. The packages each cost $49.95. For more information, contact your local Apple dealer, or write:

Apple Computer Company
20525 Mariani Avenue
Cupertino, CA 95014
408/996-1010

Cookie Monster Munch

Barbara Stewart, a project manager from CCW, had brought Cookie Monster to the Hyatt Regency Hotel in Tampa. The occasion was the third annual Florida Instructional Computing Conference, one of the largest regional educational computing conferences in the country, held from March 28-30.

Barbara was the conference's keynote speakér. In her speech, she announced that CCW was producing a new line of educational programs for the Radio Shack Color Computer (16K) and for the Atari 2600 VCS computer and game system. CCW plans to develop each cluster of programs on a particular machine and have the computer manufacturer distribute them through its standard outlets. Eventually, at least one set of CCW packages will be available for many of the bestselling computers. In 1983, CCW will be producing 24 children's learning games. Half of the games will be for classroom use, half for home use.

Cookie Monster Munch is typical of the new Atari games. The game is a numerical maze game for kids ages three to seven. It comes with a colorful booklet explaining how the game works. The Table of Contents and other sections are all hand printed, as if by Cookie himself. I like the "Note to Parents" at the beginning of the booklet. Also, a symbol of a parent with his or her arm around the shoulders of a child is used throughout the booklet. The symbols are accompanied by suggestions to increase and enrich parent-child interactions with the computer and *with each other*.

And how do the kids and their parents interact with the computer? They use the new Atari Kid's Controller. CCW worked with Atari to develop the Controller. It's a large numerical keypad with big buttons and is very sturdy. It plugs into the left controller jack at the back of the Atari 2600 VCS and is an easy-to-use keyboard or joystick for game play. Each CCW package contains a colorful plastic overlay that fits atop the Kid's Controller.

Cookie Monster's Munch *for the Atari.*

Cookie Monster Munch is a maze game, so the child has to make characters in the game move up and down, left and right, through the maze. Accordingly, the overlay has a big picture of Cookie Monster and designates buttons (hidden underneath the overlay) as movement buttons with big arrows for all four directions. It's so easy to use that even toddlers with small hands and adults with keyboard phobia will be able to play.

Another nice feature of the games is the Read Aloud Story in the beginning of each booklet. With personal computer graphics (especially VCS graphics) still at a relatively primitive level, the images of the Sesame Street characters, like Cookie Monster, are nowhere near as nice looking as they are on TV. But the story helps remedy this problem. *It engages the child's and the parent's imagination, and it gives the simple looking game on the TV display meaning and depth.*

Cookie Monster discovers a chocolate chip cookie garden. He begins running around the garden picking up cookies. He takes the cookies and puts them in his cookie jar. Cookie's intentions are sensible, but he can't resist eating the cookies before he makes it to the jar. A little kid appears – the Cookie Kid. Cookie Kid tries to collect the cookies and put them in the jar before Cookie can eat them.

The paths in the cookie garden are like a maze. There are ten different game levels and mazes. The easier games are one-person games. The harder games are one- and two-person games.

Like the *Sesame Street* TV program, the games are designed as entertaining ways to teach kids prereading skills. The kids get to move Cookie Monster or the Cookie Kid through the mazelike cookie garden. Tracing the maze pattern while remaining within its borders helps kids practice the hand-eye coordination they'll need for beginning reading and writing. Also they learn to follow directional arrows and become familiar with the relational concepts of up, down, left, and right.

Peanut Butter Panic

Here are some other new CCW games:

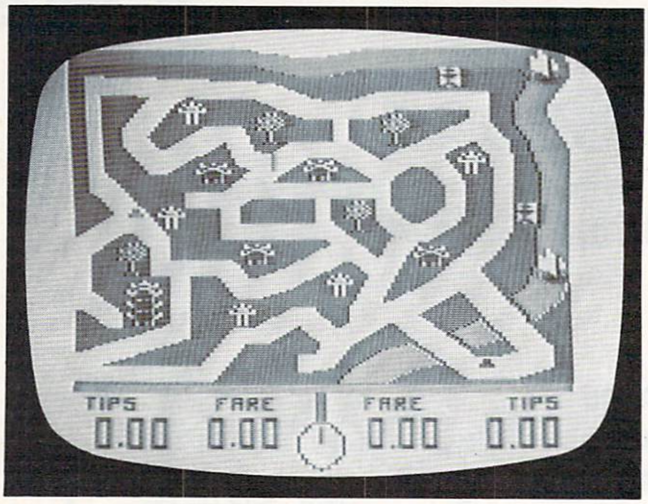
- *Ernie's Magic Shapes.* This is a home game for kids ages three to six that runs on the TRS-80 (16K) Color Computer. Kids help Ernie zap geometric shapes and use them to build colorful figures. The games help kids develop classification skills including matching shapes, recognizing embedded figures, structuring parts of an object into a meaningful whole, and discriminating between similar and different shapes.

Ernie's Magic Shapes *on TRS/80 Color Computer 16K.*

- *Grover's Number Rover.* This is a home game for kids ages three to six that runs on the (16K) Color Computer. Grover floats across the top of the screen in his Number Rover. The child helps Grover find the answer to his arithmetic problem. When the child discovers the number that solves Grover's problem, Grover picks up that number of Twiddlebugs. This is a humorous part of the game. The Twiddlebugs are upside down.

Grover's Number Rover *on TRS/80 Color Computer 16K.*

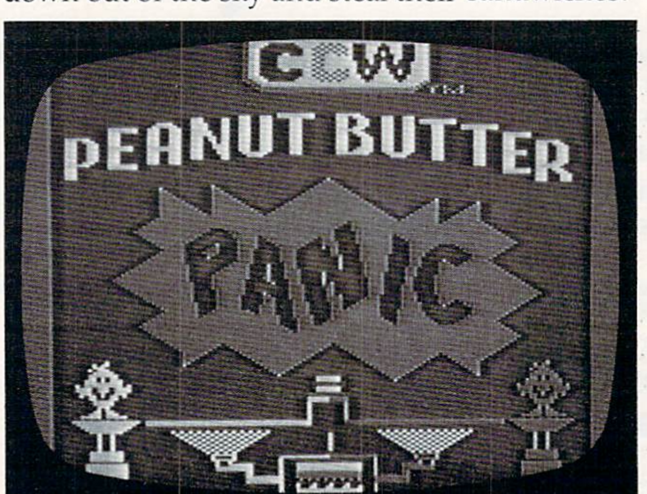
• *Taxi.* This is a home game for kids ages seven and up that runs on the 16K Color Computer. This is a junior adventure game. Kids get to operate a two-cab company in any one of six cities around the world. They pick up passengers, deliver them to their destinations, and earn fares and tips. The game encourages visual problem-solving in a cooperative environment.



Screen from Taxi *game on TRS/80 Color Computer 16K.*

• *Peanut Butter Panic.* This is another funny game. It is a home game for kids seven and up that runs on the 16K Color Computer. Two little nutniks try to catch stars that zip by above them in the sky. Kids control the nutniks and launch them from a platform that resembles a giant seesaw. The nutniks can jump up and down on their own, or two kids can launch them from the seesaw.

When the nutniks jump up and down they lose weight and get real skinny. When they get skinny, they can't jump as high. To get fat again they have to eat peanut butter sandwiches. They build a peanut butter sandwich by catching stars. They have to watch out for mean snarfs who swoop down out of the sky and steal their sandwiches.



Screen from Peanut Butter Panic *on TRS/80 Color Computer 16K.*

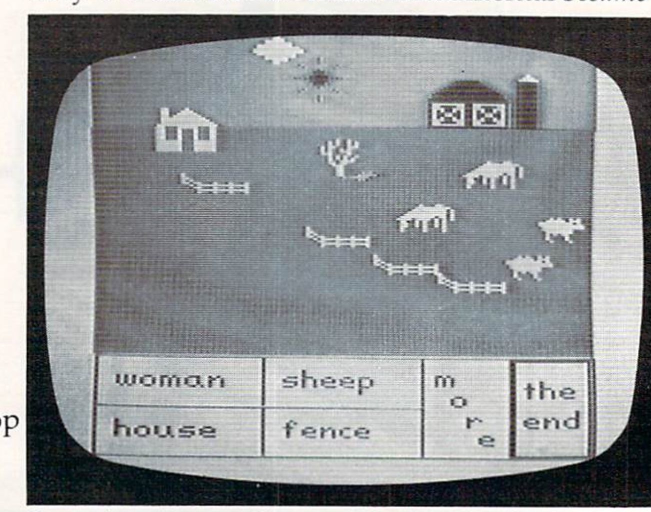
The primary objective of this delightful game is teamwork and cooperation.

• *Picture Place.* This *school* game is for kids ages five and up. (I think that it is a good game for preschoolers, too.)

Kids get to choose a picture from a library of six background scenes, including a city and a countryside. At the bottom of the screen are word boxes with words inside like dragon, car, bicycle, family, and castle. Kids choose a word by moving a joystick and positioning a big "cursor box" so that it overlaps with one of the word boxes. They pick up the word box and move it up the screen and position it on the background scene. Then, when they press the RETURN button, the word transforms into a picture. For example, the word "dragon" becomes a picture of a dragon, set in the world pictured in the background scene.

CCW's Values And Goals

Barbara Stewart thinks that personal computers will evolve into "interactive TV." She wants to create programs for TV that will accomplish the same goals as the *Sesame Street* programs on regular TV. The programs will focus primarily on developing math and reading readiness skills. But they will also stress certain fundamental *Sesame Street* values, including teamwork, cooperation, and nonsexist, nonviolent, pro-social play.

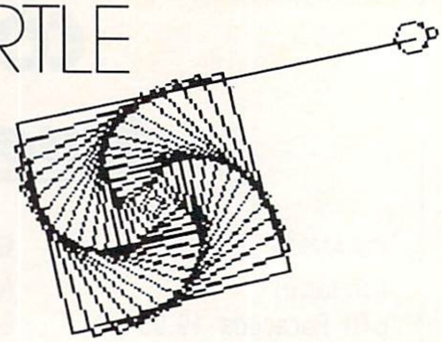


Screen from Picture Place *on TRS/80 Color Computer 32K.*

The programs are to be appropriate to their target age group and appealing to both girls and boys. They should meet educational goals for children of each age group and development level. They should be easy to understand, easy to play, and nonjudgmental. They should not frustrate children. Instead, they should encourage a child to grow and improve his or her self-image.

If these games prove to be as thoughtfully and as creatively executed as *Sesame Street* itself, children (and parents) everywhere can look forward to exceptionally rewarding educational experiences via "interactive television." ©

FRIENDS OF THE TURTLE



David D. Thornburg, Associate Editor

PILOT And Logo – A Tale Of Two Languages

PILOT and Logo are two of the most popular user-friendly computer languages available for personal computers. Because Atari PILOT and Apple SuperPILOT both contain a powerful turtle graphics environment, many people wonder if PILOT might not be a substitute for Logo.

As I will show, Logo and PILOT are quite different languages. Although they can be used for many of the same applications, each language has special features that make it more appropriate for some applications than for others. The goal of this article is to provide enough information about both languages to aid someone who is trying to decide which to use. I will assume that you are already familiar with turtle graphics.

PILOT

PILOT stands for Programmed Inquiry, Learning Or Teaching. It was so named by its developer, John Starkweather, because he wanted to create a programming language that easily allowed teachers to generate computer-aided instructional materials. Research in the late 1960s by Dean Brown showed that PILOT was also a good programming language for children.

The key to PILOT's appeal is its simple command structure and powerful ability to manipulate text-oriented material. At its core, PILOT has only eight commands, yet these eight commands allow the creation of quite sophisticated programs. The core commands for PILOT are shown below:

PILOT Command	Function
T:	*T*ypes text and variables on the screen.
A:	*A*ccepts input from the keyboard.
M:	*M*atches words or phrases against the result of the most recent accept command.
J:	*J*umps execution to a label.
U:	*U*ses a labeled procedure.
C:	*C*omputes the value of a variable.
R:	allows *R*emarks to be added to a procedure.
E:	*E*nds a program or procedure.

Notice that none of these commands has anything to do with graphics. The incorporation of turtle graphics in PILOT is a fairly recent event. Also, most versions of PILOT have additional text manipulation commands that add significantly to its power.

Core PILOT's most powerful command is M:, the match command. To see why this command is so powerful, consider the following PILOT procedure:

```
*QUESTION1
  T: WHAT GROWS ON TREES?
  A:
  M: MOSS, LEAVES, BUGS, INSECTS, NEEDLES
     TY: YOU ARE CORRECT
     TN: ARE YOU SURE? LET'S TRY AGAIN.
     JN: *QUESTION1
  E:
```

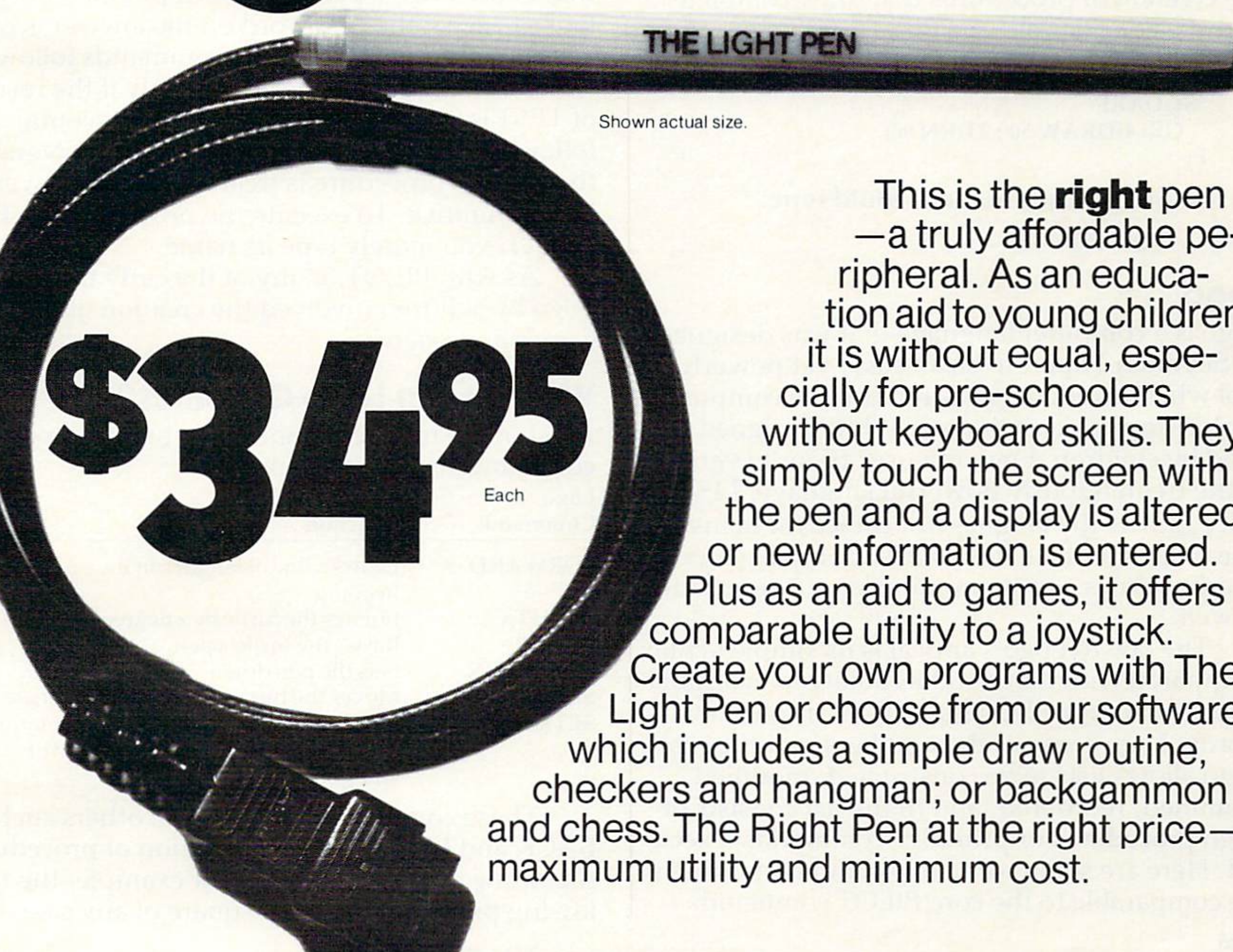
This PILOT procedure works in the following way. First, a question is typed on the screen. The user then types a response that is saved in the "accept buffer." The match command then checks to see if any of the words, MOSS, LEAVES, etc., appear anywhere in this buffer. If there is a match, a "yes flag" (Y) is set to be true and a "no flag" (N) is set to be false. The execution of any PILOT command can be made conditional on the status of these flags by entering Y or N after the command name. For example, the command TY: will print on the screen only if the yes flag is true. The JN: command causes the procedure to be used over again if the user's response is *not* matched.

As a result of PILOT's ability to manipulate words and phrases, many of the early uses of PILOT by children involved the creation of word games and "poetry generators."

What About PILOT Graphics?

As mentioned, graphics is a recent addition to PILOT. Turtle graphics is incorporated through the use of special commands. In Atari PILOT, for example, this command is GR: followed by specific graphics instructions. The fundamental graphics commands allow the turtle to be moved in its present heading or to have its heading changed.

The Light Pen at the Right Price:



Shown actual size.

This is the **right** pen —a truly affordable peripheral. As an education aid to young children it is without equal, especially for pre-schoolers without keyboard skills. They simply touch the screen with the pen and a display is altered or new information is entered. Plus as an aid to games, it offers comparable utility to a joystick. Create your own programs with The Light Pen or choose from our software which includes a simple draw routine, checkers and hangman; or backgammon and chess. The Right Pen at the right price—maximum utility and minimum cost.

Prices subject to change without notice. See your local dealer or order direct. New catalog available.
Add $3.00 for postage and handling. Credit card orders call toll free:

1-800-334-SOFT

DEALER INQUIRIES INVITED

programmer's institute

a division of **future**house

p.o. box 3470, dept. C, chapel hill, north carolina 27514, 919-967-0861

Here's a list of the more commonly used Atari PILOT graphics commands:

PILOT Command	Function
GR: DRAW x	Draws a line of length x in the present heading.
GR: TURN x	Rotates the turtle by x degrees.
GR: PEN UP	Raises the turtle's pen.
GR: PEN YELLOW	Sets the pen color to yellow and sets the pen down.
GR: GOTO x, y	Moves the turtle to absolute coordinates x,y.
GR: TURNTO x	Rotates the turtle to absolute orientation of x degrees measured to the right of straight up.

These commands (and several others) allow the creation of procedures that draw complete figures. For example, the PILOT procedure shown below draws a square 50 units on a side:

```
*SQUARE
  GR: 4(DRAW 50 ; TURN 90)
E:
```

To use this procedure, one would type:

```
U: *SQUARE
```

Logo

Logo is a computer language that was designed by Seymour Papert to be an easy, yet powerful tool which would let children use the computer to explore topics on their own. While designed to be used by children, Logo is a user-friendly version of the tremendously powerful language, LISP. Since LISP is the language of choice for many researchers in the field of artificial intelligence, clearly Logo is a programming language for adults as well.

The key to Logo's appeal is its simple syntax (compared with LISP) and its ability to manipulate data structures called *lists*. A list is a collection of words, Logo commands, numbers, or other lists. Logo allows lists to be constructed, modified, examined, reordered, and (if the list consists of Logo procedures or primitive commands) executed. Here are some core Logo commands which are comparable to the core PILOT commands:

Logo Command	Function
PRINT	Prints a list of text on the screen.
READLIST	Reads a list from the keyboard.
MEMBERP	A predicate that matches a word against the elements of a list.
MAKE	Assigns (or "binds") a number, word, or list to a variable named by a word.
END	Ends a procedure.
FIRST	Returns the first element of a list.
BUTFIRST	Returns all but the first element of a list.
LAST	Returns the last element of a list.
BUTLAST	Returns all but the last element of a list.

Notice that none of these commands has anything to do with graphics. Turtle graphics was incorporated into Logo after the language had been in use for a while. The list of Logo primitives shown above is quite incomplete, but it allows us to build a procedure comparable to the QUESTION1 procedure we wrote in PILOT:

```
TO QUESTION1
   PRINT [WHAT GROWS ON TREES?]
   MAKE "ANSWER READLIST
   TEST MEMBERP FIRST :ANSWER [MOSS
      LEAVES BUGS INSECTS NEEDLES]
      IFTRUE [PRINT [YOU ARE CORRECT]]
      IFFALSE [PRINT [ARE YOU SURE? LET's
      TRY AGAIN]
         QUESTION1]
END
```

This procedure performs a function similar to that of the PILOT procedure except that it only looks to see if the first word on the answer is contained in the answer list. The commands following the words IFTRUE are executed only if the result of TEST is true. If the result is false, the commands following IFFALSE are executed instead. Notice that a Logo procedure is treated just as if it were a Logo primitive. To execute the procedure QUESTION1, you merely type its name.

As with PILOT, many of the early uses of Logo by children involved the creation of word games and poetry.

What About Logo Graphics?

A list of the more common Logo turtle graphics commands is shown below:

Logo Command	Function
FORWARD x	Draws a line of length x in the present heading.
RIGHT x	Rotates the turtle by x degrees.
PENUP	Raises the turtle's pen.
PENDOWN	Sets the pen down.
SETPOS x y	Moves the turtle to absolute coordinates x,y.
SETHEADING x	Rotates the turtle to absolute orientation of x degrees measured to the right of straight up.

These commands (and several others such as BACK and LEFT) allow the creation of procedures that draw complete figures. For example, the following procedure draws a square of any size:

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

To use this procedure to draw a square 50 units on a side, one would enter:

```
SQUARE 50
```

Differences Between Logo And PILOT

The previous sections have suggested that PILOT and Logo are similar in application areas and syntax. In fact, there are some major differences between the languages that may cause one to be clearly the language of choice for a particular task.

For example, PILOT makes it very easy to create programs in which the contents of variables are printed along with text. Also, the match command will compare each element of a list with the entire response. In Logo, you would have to write a procedure to do this.

Another important feature of PILOT is its compactness. Most Logo implementations require large amounts of RAM. Most (but not all) versions of PILOT will operate in 16K of RAM with plenty of space left for the user's program.

In terms of overall symbol manipulation, Logo is the more powerful of the two languages. The ability to write programs that generate other programs is of great utility when constructing environments that "learn from experience." The fact that user-defined procedures are treated exactly as if they were Logo primitives gives Logo a feature called *extensibility*. This means that you can add new words to Logo's vocabulary (as we did with QUESTION1 and SQUARE). There is no need in Logo for the *jump* or *use* commands. To execute a procedure, you just type its name.

Logo also supports *local variables*. This means that the value associated with a variable is assigned to the specific procedure (and level) in which it is used. This allows you to write procedures that use themselves recursively. For more information on this topic, you might want to read the "Friends of the Turtle" columns on recursion that appeared a few months back.

Logo's turtle graphics commands are, perhaps, easier to grasp than PILOT's, but there are indications that this will not always be the case as new versions of PILOT are likely to become more "Logo-like."

Apart from these differences, Logo and PILOT both encourage a procedure-oriented programming style that makes complex programs easy to read and correct.

I use both languages regularly and find that I would be reluctant to abandon either one. Your application areas might indicate that one of these languages has a clear advantage over the other. No matter which you choose, you will be using a language that allows the creation of very sophisticated and powerful programs.

Notes From All Over

I have just heard from my Argentinian friend, Horacio Reginni, who has just started the Asociacion Amigos de Logo (Logo Friends Association) to promote the development of Logo centers, sponsor meetings, and spread information about Logo all over the world. The association can be reached at 2969 Salguero St., Buenos Aires, 1425 Argentina. True Logophiles will be interested in attending their first International Logo Conference in Buenos Aires on September 16-18. Registration is only $25. As for the air fare ....

©

Learning With Computers

Glenn M. Kleiman

A Library At Your Fingertips

The ability to use computers to efficiently access, organize, and analyze information is becoming a critically important skill. In fact, knowing how to use computerized information bases is rapidly becoming as important as knowing how to use a library. People in many occupations – travel agents, bank tellers, librarians, stockbrokers, and insurance agents – already use computerized information bases every day. Doctors, lawyers, scientists, teachers, and many others will be added to the list in the next few years.

There are many computerized information bases. In this column, I discuss my favorite one, which is called *DIALOG*. DIALOG is the world's largest computer storehouse of information available to the public. It contains over 170 data bases with a total of more than 75 million records of references, abstracts, and statistical data on a great diversity of topics. A simple set of commands lets you locate information quickly and easily. Widely used by libraries and businesses, DIALOG and its new cousin, *Knowledge Index*, can also be used by schools and individuals.

To use DIALOG, you need a terminal or a computer with the hardware (a modem and interface) and software to make it function as a terminal. You also need an account number on the DIALOG system and a telephone. Like other large data base systems, DIALOG uses special networks (Telenet and Tymnet) so you can access it with a local telephone call from most places in the United States.

An Example Information Search

I've recently used DIALOG to search for information about one of my main professional interests, the use of computers by children who have learning disabilities. There has not been a great deal of research in this area, and reports of the research that has been done are scattered in many different journals and books. A data base on the DIALOG system, called ERIC, lets me search an enormous body of literature for relevant references, and to do so in a few minutes.

ERIC is an acronym for Educational Resources Information Center. It is an index to the contents of more than 700 journals in education, as well as a large number of books, technical reports, conference papers, government agency reports, and other documents. It contains approximately 500,000 references, dating back to 1966. The index is kept up-to-date and about 3,000 references are added each month.

All the information about each journal article or document is grouped together into what is called a *record*. Each record contains the title, author, journal and date of publication (or other information needed to locate the actual document), the language in which it is written, a set of descriptive (subject indexing) terms and an abstract (short summary). The descriptive terms are keywords which characterize the contents of the document. There is also a *Thesaurus of ERIC Descriptors* which enables you to find the best descriptor terms for each topic.

The many volumes of printed ERIC indexes are familiar to many educators and researchers. For some of my articles and research projects in years past, I've spent hours scanning through many pages of small print, hunting for relevant references. I can now accomplish the same work in a few minutes via the computer on my desk.

After using a modem and telephone to connect my computer to the DIALOG computer, I enter my account number and password. My search for references about computers and learning disabled children then proceeds as shown below. (In some cases, I have slightly altered the computer's response, leaving out code numbers and other extraneous information and spelling out abbreviations for clarity.)

First, I tell the system I want to use the ERIC data base (which happens to be number 1). I enter:

<u>BEGIN 1</u>

(My commands will be underlined throughout this column.) The computer responds:

NEW MULTI-USER SOFTWARE LETS THE WHOLE FAMILY SHARE IN THE JOY OF LEARNING.

Is the personal computer doing all it can to help our children learn?

To some degree, no, although it's not fair to blame it entirely on the computer. After all, computers are only as good as their software.

How can we improve this situation?

A solution already exists. But first, some background.

Where personal computers fail.

For years, studies have shown that children learn more efficiently in group situations. Peer groups, for example, motivate slower learners to persevere. Groups of older and younger children encourage divergent thinking. Even the simple "group" of a parent and child promotes faster acceptance of new ideas by combining education with trust and confidence.

But personal computers and their programs are designed to be personal. One computer, one child. It's hard for anyone else to be part of the learning experience, even you.

At least not until today.

A simple solution.

When two educational researchers, Dr. Matilda Butler and Dr. William Paisley, observed this problem they proposed an interesting, yet simple, solution. Instead of writing programs that shut out brothers, sisters, friends, and parents, why not give everyone the opportunity to share learning simultaneously. This one idea sparked an entire line of unique educational programs and gave birth to a new company, Edupro.

Software that shares.

With Edupro's Microgroup™ computer programs, up to eight players work at solving math, language, social studies, or science problems which are presented as contests, races, and puzzles. The players work together, either competitively or cooperatively, as they race against time, each other, or both.

The *Math-Race* program, for example, converts your computer into an electronic race track where children compete to answer math problems and advance toward the finish line. *Picture-Play* encourages everyone to create pictures together, teaching both spatial relationships and the value of cooperation. And *Team-Work* combines both cooperation and competition by pitting two teams (of up to four players) against each other in a race to solve word and number puzzles.

For the first time, your personal computer can bring all the benefits of group learning into your home. With a little assist from Edupro.

Designed for the simplest computers.

These unique programs run on the Atari 400 or 800, two of the world's most popular home computers. Remember, these aren't game cartridges, they're full *computer programs*, designed by educators. All are available on floppy disk or cassette, and each one requires the minimum amount of computer memory (16K for cassette, 24K for disk). That means the simplest Atari computer can let your children share the learning experience with up to seven additional friends. Joysticks required for *Word-Draw*, *Math-Hunt*, and *Picture-Play*; paddles required for *Word-Race*, *Math-Race*, and *Team-Work*.

Trust your own experience.

At the fall 1982 Computer-Using Educators Conference hundreds of educators witnessed hands-on demonstrations of our programs. Many of them said that this was a most effective way to judge their potential. But we want to offer you an even better opportunity. One those educators missed.

We want you and your children to experience this new way to learn. So choose one or more programs on either disk or cassette. Try them yourself. Watch your children get more excited about learning. Enjoy the thrill of sharing the experience with them. We know of no other software that can turn a personal computer into a tool for sharing the joy of learning.

Fill out the order form and see the results for yourself.

Atari® and 400/800™ are trademarks of ATARI Inc.

I want to share the joy of learning with my children. Please send me the programs I've indicated below. I understand that each program is available on either disk or cassette (my choice) and comes with a complete set of instructions and catalog listing over 50 programs. Plus a coupon good for a 10% discount on my next order.

Quantity	Program Description	# of Disk	# of Cassette
	STORYBOOK FRIENDS: Ages 5–9		
____	WORD-DRAW: Storybook People and Places	____	____
____	MATH-HUNT: Number Relationships	____	____
	AMERICAN THEMES: Ages 8–13		
____	TEAM-WORK: Social Studies	____	____
____	MATH-HUNT: American Years: Multiplication and Division	____	____
	THE WORLD AROUND US: Ages 12–Adult		
____	WORD-DRAW: Science	____	____
____	MATH-RACE: Powers and Roots	____	____
	JUST FOR FUN: All Ages		
____	PICTURE-PLAY	____	____

Total #		Total Amount $
____	programs on disk @ $24.95 each	____
____	programs on cassette @ $19.95 each	____
____	Picture-Play, disk @ $19.95	____
____	Picture-Play, cassette @ $14.95	____
	CA residents add sales tax	____
	Postage and handling	$2.50
	Total	____

My check or money order is enclosed for $ ____

Please bill ____ MasterCard ____ Visa

____ (card no.) ____ (exp. date)

Name ____

Address ____

City ____ State ____ Zip ____

Signature ____

Allow 3 weeks for delivery. Satisfaction guaranteed.

Send to: Edupro, Dept. C01, P.O. Box 51346, Palo Alto, CA 94303.
Write to above address for brochure/catalog listing or phone inquiries: (415) 494-2790.

Edupro™

Edupro, P.O. Box 51346 Palo Alto, CA 94303, (415) 494-2790

27 Feb 83 12:59:37 File:ERIC

This search, and the other examples in this column, were all done on February 27, 1983.

Next, I give the computer the words for which I want it to search. It searches through all the information in each record, including the abstract. (You can limit the search to the descriptor terms or title if you prefer.)

SELECT LEARNING DISABILITIES

DIALOG responds:

1 4734 LEARNING DISABILITIES

I've told the computer to select all references about learning disabilities. It gives this set the number 1, so I can refer to it later. The number following the set number shows how many relevant records have been found. I then enter:

SELECT COMPUTER

DIALOG responds:

2 16684 COMPUTER

So now I know that there are 4734 references about learning disabilities and 16684 about computers in the ERIC data base. But what I really want to know is how many are about both computers and learning disabilities. The appropriate command is:

COMBINE 1 AND 2

DIALOG responds:

3 70 1 AND 2

This tells me that 70 references appear in both set 1 and set 2 (i.e., the learning disabled set and the computer set).

DIALOG also allows more complex combinations using OR and NOT. This provides tremendously powerful searching capabilities. I could, for example, further restrict the search to references that are about reading disabilities or language disabilities, while excluding references about hyperactivity. I could also restrict the search to particular years, journals, authors, types of publications, languages, or any combination of these. Since you work on-line with DIALOG, you can expand or restrict the search as you go. For example, if I find more references than I want on a topic, I usually restrict the search to articles published in the last year or two.

Next, I want to see the titles of some of the references:

DISPLAY 3/6/1-5

This command tells the computer to display the references in set 3. The 6 is a code number telling it that I only want to see the titles, not the other information in the record. The 1-5 tells it to display references number 1 through 5. The computer responds with:

1. Remediating Spelling Problems of Learning Handicapped Students Through the Use of Microcomputers.
2. Microcomputers: Powerful Learning Tools with Proper Programming.
3. Microcomputers: An Available Technology for Special Education.
4. How Can Microcomputers Help?
5. Instructional Technology for Special Needs.

Item 3 sounds interesting, and I haven't seen it before. I therefore tell the computer to print the full record:

DISPLAY 3/7/3

This command says display from set 3 the full record (code 7) of item 3. The computer responds with:

Microcomputers: An Available Technology for Special Education.

Joiner, Lee Marvin; and Others

Journal of Special Education Technology, Vol. 3, number 2, pages 37-47. Winter, 1980.

Language: English

Document Type: Journal Article; Teaching Guide

Abstract: The article describes the capabilities of features of basic microcomputer systems and describes special education applications: computer assisted instruction, testing communication, and enhancing personal relations. Problems such as the availability of authoring languages, high quality educational software, and computer safety are described.

My entire search took less than five minutes, most of which I spent examining the titles of articles. I next instructed DIALOG to print all 70 records about computers and learning disabilities, with the citation and abstract for each. To save time and expense, I had this done off-line by high-speed printers at DIALOG and mailed to me. The 25 pages of materials arrived a few days later. I then used DIALOG to order complete copies of several of the articles.

Other Data Bases

ERIC is just one of over 170 data bases available on DIALOG. There are data bases covering the sciences, business, law, current affairs, humanities, books, book reviews, foundations, biographies, patents, dissertations – an incredible array of information. Some of the data bases likely to be of interest to readers of this column are described below.

The *Magazine Index* covers 435 of the most popular magazines in North America, including all those indexed by the *Readers' Guide to Periodical Literature*. It contains over one million records, dating back to 1969. Approximately 12,000 records are added each month. There is also a *National Newspaper Index*.

I was curious about whether magazines have reflected the increase in interest about computers in education during the last few years. I therefore checked the number of articles in the Magazine Index on computers and education for each year from 1976 to 1982. In about two minutes I obtained the following answer:

Year	Computers & Education Articles
1976	2
1977	19
1978	9
1979	27
1980	39
1981	59
1982	145

Clearly, the number of articles has been growing rapidly.

Newsearch is an index of current news stories, information articles, and book reviews from over 1,400 newspapers, magazines, and periodicals. Newsearch is updated daily, so most items are added the day after they are published. At the end of each month, the information is transferred to the Magazine Index, the National Newspaper Index, and other relevant indexes.

The *Books in Print* index contains records on virtually all books published in the United States, including books that have gone out of print in the last few years and books that are to be published in the next few months. A quick check found 6,450 books on computers, 46,478 on education, and 168 about computers and education. There is also a *Book Reviews* index.

The *Microcomputer Index* is a new one which contains citations about the use of microcomputers in business, education, and the home. Magazine articles, as well as software and hardware reviews, new product announcements, and book reviews are included. Over 25 microcomputer periodicals are currently indexed, along with selected articles from other publications. A quick check showed 1,294 articles on education.

The International Software Database is another new one. It contains over 10,000 records on all types of software, classified by application, machine, operating system, vendor and price.

Classroom Instruction

The cost of using the indexes I have described ranges from $25 per hour for ERIC to $95 per hour for Newsearch. The cost of off-line printing is typically 20 cents for each full record. Since DIALOG makes finding information so efficient, I regard it as an excellent value for professional use. DIALOG has also introduced lower-cost special arrangements for schools that want to teach students to use it and for individuals who want to use the system during evenings, nights, and weekends.

The Classroom Instruction Program provides access to most of the DIALOG data bases at a special rate of $15 per hour. This rate is available only to academic institutions for instructional purposes. A special students' workbook is also available.

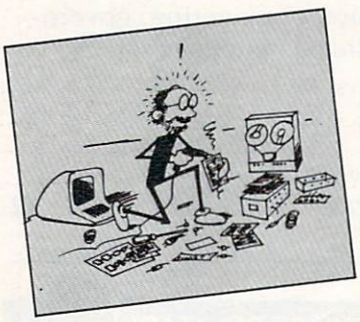
Knowledge Index is a new service which provides access to the most popular data bases at the reduced price of $24 per hour. It is not available during business hours, so this service is designed mostly for individuals. All the data bases I have described are available, except for Books in Print and Book Reviews. In addition, Knowledge Index includes data bases covering business, agriculture, computers and electronics, engineering, government publications, medicine, and psychology.

DIALOG Information Services, Inc.
3460 Hillview Ave.
Palo Alto, CA 94304
(800) 227-1960 (except California)
(800) 982-5838 (from California)

©

Superbaby Meets The Computer

Fred D'Ignazio, Associate Editor



If you haven't seen it already, you should go to a library and find the March 28, 1983, issue of *Newsweek* magazine. Turn to page 62 and read the cover story, "Bringing Up Superbaby." The story is about how parents are pushing their kids to learn earlier and earlier. Kids who are only a few months old are studying art books, gazing at flash cards, doing toddler gymnastics, going to dance class, putting together puzzles, taking swimming lessons, and *learning how to compute*. In the article there's a picture of a little kid who is pounding away on the keyboard of an IBM Personal Computer.

Just a few years ago, Elizabeth Wall (a media specialist in Sarasota, Florida, and author of *The Computer Alphabet*, Avon, 1983) sat down next to one of the pioneers of personal computing. He asked her what she was up to. "Teaching elementary school kids how to use computers," she told him. He was shocked. "There's no future in teaching little kids computers," he said. "They will never get the hang of it."

Since that expert made his remark, use of computers has dribbled downward, from college to high school kids; from high school kids to middle schoolers; from middle schoolers to kids in elementary school – and beyond.

In Bruce and Diane Mitchell's Small World preschool and kindergarten, in Durham, North Carolina, four-year-olds and five-year-olds are playing educational games on Atari computers and Timex Sinclairs. They are programming a Turtle robot by tapping on the keyboard of an Atari 800.

But preschoolers and kindergartners are old. They're almost over the hill! The *Newsweek* article mentioned a school called Tiny Bytes where kids can begin computing before they've celebrated their first birthday.

Computer Literacy Or Else

Some toddlers are going to be victimized by pushy parents trying to fill their offsprings' "little sponges" with computer facts even before they've learned to walk or talk. I can imagine an "enlightened" household where the parents are trying to give their three-month-old an early start on her way to a high-tech future. The baby, blithely unaware of her parents' designs, is reaching for a rubber ducky. The mother pushes the duck away. "Too easy," she says. She whips out a stack of big white flash cards. "Let's practice these first, then you can see the duck on your lunch break." As the baby gazes sweetly at her mother, the mother runs through the flash cards. "RAM!" she calls out. "RAM .. R .. A .. M .. RAM! BIT! .. B .. I .. T .. BIT! CHIP! .. C .. H .."

One wonders what a kid who gets computer flash cards at three months is going to be like when she gets to the ripe old age of five years, or ten, or fifteen. She may have a lot of computer facts under her belt, but how well adjusted will she be? What will be the result of all this parental prodding?

This is not to say that computers shouldn't be introduced to kids who are still wandering around the house in dirty diapers. Because they should be!

The question is *how*.

Parents who are pushing their babies and toddlers into computer literacy are missing the point – at least as far as computer literacy is defined. We are presently in the Age of Computer Literacy. But we are quickly moving beyond it.

Pretty soon it will not be productive for us to study such arcane terms as *bit, byte,* and *CPU*. We won't have to know how a computer works, just how to work a computer. We will be leaving the age of computer literacy and entering the Age of Computer *Intimacy*.

Take the TV or the car. These are high-tech machines that are part of almost every little kid's environment, right from birth. Do parents go around with flash cards with words like CHANNEL SELECTOR or PHOSPHOR SCREEN? Or with words like CARBURETOR or PISTON? Of course not. Nevertheless, the smallest children learn how to operate TVs, almost before they can walk. And little kids play with model cars, toy cars and trucks, all through their childhood. And when their magic birthday arrives and they can get their driving license, they quickly learn to drive and operate an automobile.

How many kids suffer from automobile anxiety or TV phobia? Very few.

Even more important, how many kids can expect to find a job when they grow up as an automobile mechanic or an expert in TV repair? Again, very few.

Yet TVs and cars are far more common than personal computers.

The point is that we have moved beyond "TV literacy" and "automobile literacy" to a new age of intimacy with both these machines. The technologies have matured. They are black boxes, idiot boxes that almost anyone can learn how to use. They're everywhere. We're comfortable with them in our garages, our living rooms and bedrooms.

This is where computers are headed, too. They've just started, but, at the speed they're going, it won't take long. By the end of the 1980s, computers will be black boxes, just like cars and TVs. They will be in most people's homes. They will become so common that they will cease being an eye-catching phenomenon. In fact, they will almost be invisible. Like electric motors, they will slip into other appliances and disappear from view.

Kids who are less than one year old in 1983 will be less than seven in 1990. So why are parents teaching them computer literacy terms and concepts, preparing them for a job market that exists in 1983, but will change radically even before the kids have made it through elementary school?

Parents are pushing because they are panicking. The swift pace of computer technology has them running scared.

And they are pushing their kids because of the status of having them say "floppy disk" as their first word.

What they don't realize is that they are training their kids in what will soon be an obsolete technology and, worse, an obsolete approach to technology. They are being trained to become the automobile mechanics and TV repairpersons of the 21st century. These are honorable professions. But is this what the parents intend?

Computer Osmosis Vs. Computer Bullying

Millions of personal computers are going into people's homes. Millions and millions of little children are waking up each morning and walking or being carried past computers on their way to their bottle, their Boo Berries, or baby cereal. For them, computers are no more wondrous or rare than the floor lamp, vacuum cleaner, or telephone. They're just one of the many things that "belong" in their lives. They have a place, along with everything else.

This is exactly as it should be. Computers *are* a big deal to us. And our kids will see that. When we spend all night in front of a keyboard trying to debug a program or escape from the wizard's castle in an adventure game, they'll notice. If we shout and point at the new computer and say "Gee whiz!" and "Oh, gosh!" enough times, they'll notice. And if we get frustrated with the computer and begin saying unkind things to it or give it a good bop, they'll notice that, too. Whether positive or negative, our kids will pick up on the attention we give to computers and the amount of emotional involvement we have with them. Kids are very sensitive about this sort of thing.

Growing Up Together

You and I are already grown. We're big people. But computers and kids haven't stopped growing. In fact, they've just begun. Both are going to change rapidly over the next 20 years.

At the end of that 20 years, what will they be like?

We imagine that our kids will end up pretty much like us. But how about computers? When kids enter the job market in the late 1990s or early 21st century, what will computers be like?

According to experts, we are quickly entering a new era of personal computers. I call this era the Age of Computer Intimacy. Others call it: The Age of User Friendliness. The Age of Forgiving Systems. The Age of Easy Computing. The Age of Humanlike Machines.

As anyone who has struggled with a cranky program recorder, or with a cryptic BASIC error message, or with computer cables, plugs, and connections knows, we have not reached computer heaven yet. Far from it!

But we are moving closer. While at the West Coast Computer Faire in San Francisco, I attended a seminar on "Second Generation PC Software." It was mind-boggling.

I learned that if you have enough money, you can buy computer programs and computers that are really, truly friendly. They hold your hand. They speak English (most of the time). They help you out of tight spots. They remind you of what you are supposed to be doing when you get lost.

And, boy, are they powerful! With just one package, one electronic mouse, and 45 windows, you can figure out your income tax, send electronic mail, draw pie charts and bar charts, do word processing, and file, sort, and retrieve records. All with the same set of commands.

At present, these systems are extremely expensive. Only the folks who carry around Pierre Cardin calculators can afford them: But computers, in general, used to be this way, too. Only wealthy, technically sophisticated organizations (universities, large corporations, the government, and the military) could afford them. But computers have come a long way. Now you can buy a programmable computer for under 60 dollars. Pretty soon the price will be even lower, and the computer will be more powerful and easier to use.

The new generation of "easy" computers and "friendly" computer software is coming. And it will include machines and programs that we can all afford.

What Do We Tell Our Kids?

If we're not supposed to tell our children (and babies) about bits and bytes, then what do we tell them?

Nothing is okay. Unless they ask. Or unless you're so excited about something neat that you just feel like babbling.

Just have a computer around the house. That's enough. Treat it like you'd treat a typewriter, a telephone, or a calculator. *But let your kids touch it.* That's the best way for them to learn. For example, my four-year-old son, Eric, drives me crazy when he uses a computer. He has grimy, dirty fingers. He presses buttons in such a way as to make a computer act like an amnesiac. But he loves to play on the computers because he is allowed to play freely. And (with quiet wincing and cringing) I let him. One of his favorite games is filling up the picture screen with graphics symbols, multicolored bars (using color keys and the reverse-video button), and random letters, numbers, and punctuation symbols.

Another of his games is to use the computer as a Gobbledygook Processor (that's "GP"). He types all sorts of strange looking words like

IXCCY##559 ISK ERIC !!!!! AAAAAAAAAAAAA

then sends them to the computer's printer. He rips off the printer paper (in the same lavish, boisterous way he handles toilet paper) and tapes it up around the house as a sign of who knows what. Or he stuffs a wad of it in an envelope, and it becomes a letter. Or he gives it, as a gift, to me, to his mother, his sister, his kitty, or his toy robot, Denby.

A Tool Or A Crutch?

Actually, there's more to computer education than this. Our responsibility as parents (and teachers) extends beyond just making computers available to our children. Much further, in fact.

When our youngest children start entering the job market, in another 15 to 20 years, *all* computers will be "easy" computers; *all* programs will be "friendly." Computers and programs will also be a lot more intelligent than they are now. There will be a tremendous temptation to let computers take over many of the thinking chores that we humans find bothersome, tiresome, boring, or too difficult. At some point, for many people, the computer will cease to be a support and start to be a crutch.

Our responsibility, as parents and teachers, is to teach our children the value of using computers in the proper way: to help them do their *own* thinking.

What Do You Think?

What do *you* think? How early should kids begin learning about computers? What should be the role of parents (and teachers)? What should kids learn? How should they learn it?

Please send your ideas and comments to me:

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

I'll return to this subject in a future issue of **COMPUTE!**, and I'll reprint a number of your letters.

New Resources

A book has just been published for parents of older children (ages nine and up) who are interested in computers. I recommend the book because it is a practical guide to the technology *as it exists today*. If you want to launch yourself and your family into computing today (and you should), then you need a survival manual. The best survival manual of all is this magazine (**COMPUTE!**), with all its tutorials, articles for beginners, practical programming tips, and actual programs for you to copy into your machine. But, if you're a parent, you should also take a look at:

Eugene Galanter, *Kids and Computers: The Parent's Microcomputer Handbook* (Perigee Books, The Putnam Publishing Group, 200 Madison Avenue, New York, NY 10016; $7.95; Paperback; 7-page index; 190 pages)

Sample chapters: Microcomputers and Your Child; What Is a Microcomputer?; The Microcomputer's Parts; Programming by, for, and with Children; Running the Machine; Kids Can Write Programs; Evaluating Computer Education.

The author, Eugene Galanter, has been teaching kids about computers for several years. You can write his school for additional information or to ask him specific questions about kids and computers:

Eugene Galanter
The Children's Computer School
21 West 86th Street
New York, NY 10024

Looking For Good Software?

Of course you are. So you should get in touch with an organization that evaluates the newest educational software:

Educational Products Information Exchange (EPIE)
Columbia University Teachers College
P.O. Box 27
New York, NY 10027

EPIE has recently entered into an agreement with the Consumer's Union to test and evaluate hundreds of consumer-oriented and educational computer products. The results of their research, reviews, and laboratory tests are just becoming available. You can read about these results in their MICROgram, published monthly as part of:

The Computing Teacher
The International Council for Computers in Education (ICCE)
135 Education
University of Oregon
Eugene, OR 97403
Monthly/Subscription: $16.50/year

Teachers (and interested parents) can write EPIE & Consumer's Union and obtain special publications, including the Micro Courseware PRO/FILES for K-8 Educators ($84), and the Courseware Resource List (as a bonus for ordering the PRO/FILES):

EPIE & Consumer's Union
P.O. Box 839
Watermill, NY 11976

©

How To Create A Data Filing System

Part I. Choosing The Right File Type

Jim Fowler

It's always a good idea to analyze your data storage problems and plan your solution carefully before you start programming. This article begins a four-part series on writing a data file/retrieval system for any computer.

Remember how your disk drive was going to solve your data storage problems? All those address cards, recipe files, inventories, and accounts were somehow going to become organized and never frustrate you again. It *can* happen, but you will have to do some thinking about the problem before you solve it satisfactorily.

Of course, the commercial data base systems can serve you very well, but you ought to know something about such systems before you spend money for features you may never use. It is not impossibly hard to write a program that does everything you want. I have lived through a couple of such projects so maybe I can point out areas to think about and where you might take a wrong turn.

Planning is the name of the game. I can recommend writing your own system. If you like programming, you can easily develop a system that fits your needs, and you will know it well enough to alter it when you need to make changes. One thing to keep in mind as you plan is that once you begin to put the data on a disk you are, in a sense, a hostage to your own work. The more time you have spent typing in the data, the more reluctant you will be to start over. So *plan ahead*.

Another bit of advice – automation is not automatically a good thing. If you have a recipe card file with the cards filed under a few headings ("salads," "desserts," "meats," etc.) and if there are only 30 or so cards in each section, you can probably find the one you want faster by flipping through the cards by hand. I remind you of that eternal verity: "If it works, don't fix it."

Pick Your Goals

The first step is to draw up a list of what you want. Actually write down what you hope a session with the file would be like: you turn on the computer, insert the disk, sit down to the keyboard, then what? Do you want a long list printed out (address labels?) or are you going to look for a needle in a haystack, such as the one record with exactly the right data to match your needs? It is well worth writing such scenarios several times on different days.

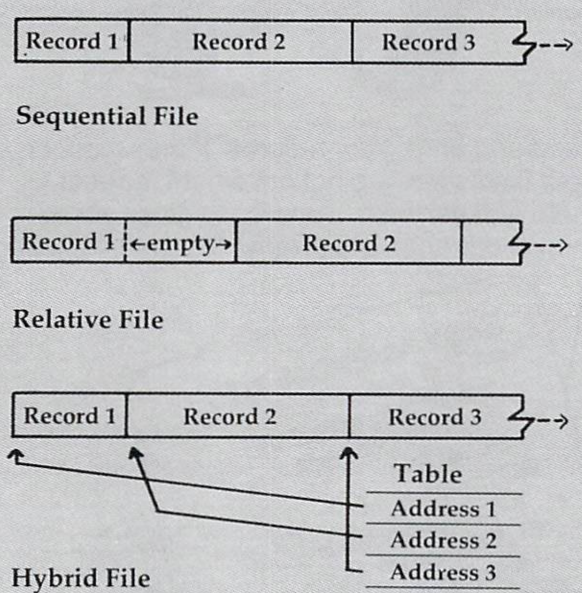
Another important consideration is flexibility. Whenever you are faced with a choice, always pick the one that gives you the greatest future flexibility. Of course, most of your choices will be made for you by the necessities of your data, your hardware, its operating system, etc. But keep flexibility in mind. This applies to every feature of your system – the number of records you expect to store, the amount of information in each, the "keys" you might use to retrieve records, and so on.

The *key* for an address file which is organized alphabetically by last names would be the last names of each entry. The key allows for quick searches and for sorting and entering new items into the proper order.

Finally, go to great lengths to make your system easy to use. It is so tempting to short-cut some tedious programming by saying to yourself, "Oh well, I can always remember that hitting RETURN without any input will drop me out of the program. After all, I've been running this machine for awhile, and I don't make that mistake any more."

The important thing about data file systems is that you enter and retrieve records hundreds of times. A small stone in your shoe is no big deal if you are sitting down, but walk a few miles and see how important it gets! A small annoyance in a program is tolerable if you only encounter it once

in a while, but in a data entry or retrieval operation it can doom the whole system. Many a card file has been restored to active duty because, for reasons like these, its owner got fed up with automation. So, be prepared to go to great lengths to make life easy for the user.



The three types of files. Record 1 is shown as if it is only half as long as Record 2.

The Three Kinds Of Files

There are three kinds of disk files. The first is one you probably already know, a *Sequential File*. All the data is strung together head to tail and put on the disk that way. Your programs are recorded on tape or disk in a sequential file. If you use a sequential file, you will need to put separators (called *delimiters*) of some kind between items of data so that you know where one ends and the next begins.

One problem with sequential files arises when you want to change a record and the new one is of a different length. It is like putting books on a shelf: take out a thin one and put a fat one in its place – you'll have to move all the rest to make room. If you rarely make any changes, it might be worthwhile just erasing the old record by filling it with blanks and adding the new version at the end.

The second kind is a *Relative File*. This is like a series of pigeon holes. One may be filled, another partially empty, but you do not have to move them to make room when you enlarge a record. As long as each hole is big enough to take the biggest record, you have no problem. This is the kind I use for my most complex data file.

The third kind is a sequential file, but with a "Table of Contents" like the directory on a disk. Call it a *Hybrid File*. To use this kind takes a lot of programming. I cannot recommend it unless the saving in space is much greater than the space taken by the extra programming and the table. Only big professional systems are likely to go this route.

The figure diagrams the three file types. If your disk operating system supports relative files (also called *random-access files*), you will probably want to use that kind unless you are going to be very short of space on the disk. If your system doesn't automatically support relative files, you can make your program do it. Keep a table or use a formula which turns a record number into its "address" on the disk – its track and sector. Then you read or write a record directly by track and sector. This is a bit complicated, but worth doing.

Next month, we will look at methods of retrieval and how they can affect the way you keep records. ©


JINSAM™
EXECUTIVE™
space
age
micro
software
Used at NASA,
Kennedy Space Center
With Multiple Applications Related
to the Columbia Space Shuttle Project
including rescue operations, statistical
reports, inventory and vehicle tracking.
JINSAM EXECUTIVE™
has broken the 10,000 record limit. You may now have up to 65,000 records in one database.
We also have included a free form report generator for data entry, eliminating the need for WordPro™ and have included automatic mathematical relations eliminating the need for VisiCalc™. However, you still have these superb interfaces available.
Executive™ will be available for CBM and IBM personal computers.

How To Make Backup Disks For VIC And 64

Harvey B. Herman, Associate Editor

LOAD, switch disks, SAVE, LOAD, switch, SAVE – it can be cumbersome and tedious to make backups of disks when you don't have a dual disk drive. What's worse, you need to go through special extra steps to transfer machine language programs. This utility, for any 64 or expanded VIC, makes creating safe backups on single disk drives nearly automatic.

I recently purchased a 1541 disk drive for my expanded VIC. The diskette that came with it included a few sample programs. Conspicuous by its absence, however, was a program to make duplicate copies of diskettes for *backup* purposes. I have learned the hard way that diskettes do not last forever and it is foolish to have only one copy of important programs.

What do to? Well, I was lucky to have acquired an excellent backup program for the Commodore 2031 single disk drive (written by Jim Law and Keith Hope and distributed by the Toronto PET User's Group). I adapted this program to work on the Commodore 64 and expanded VIC-20 computers. One program works for both. The modifications in the original program were quite modest – a few PEEKs and POKEs were changed, and the machine language portion was relocated to the cassette buffer and POKEd in from DATA statements.

The program is quite easy to use; no knowledge of machine language is necessary. First, the destination diskette is formatted, a good idea if you will be using it later on the same drive. Please be careful to format only blank diskettes, or ones that are no longer needed. Next, the diskettes are swapped and the source diskette is read to determine how much to copy. Successive blocks are then read from the source into the available computer memory. (I can read 124 blocks on the Commodore 64 and proportionately less on the expanded VIC, which has less memory.) The diskettes are swapped again, and identical blocks on the destination disk are written from data saved in memory. The swapping of source and destination diskette continues, until the entire diskette has been copied.

Of course, it would be easier (but not much faster) if a second drive were available. However, this program is the next best thing. It surely beats loading and saving BASIC programs, one at a time, or finding the loading address of machine language files. Try *that* sometime if you doubt it.

One caution – the program will not work on an unexpanded VIC. I have added 24K of RAM, by means of the Cardboard, and this minimizes swapping. Much less than 16K may not be practical, as too few blocks are copied in one swap. Obviously, the Commodore 64 does not have this problem.

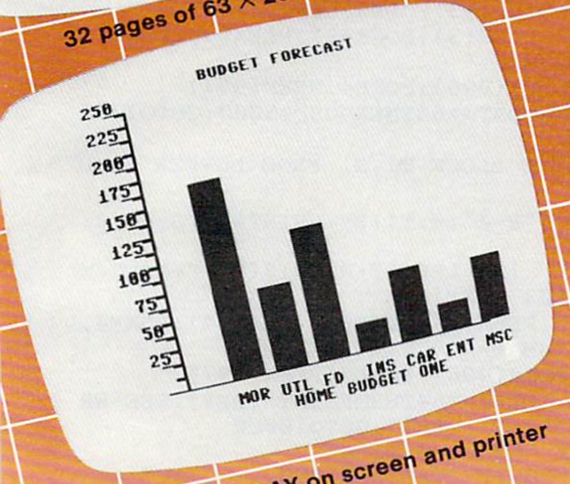
If you want to save the trouble of typing this in, I will make a copy for you on cassette or diskette (1540/1541 format) for $3. Just send me the medium, a self-addressed mailer, and proper postage. If you have any questions please enclose an SASE. My address is:

Harvey B. Herman
Chemistry Department
UNC-Greensboro
Greensboro, NC 27412

VIC/64 Disk Backup

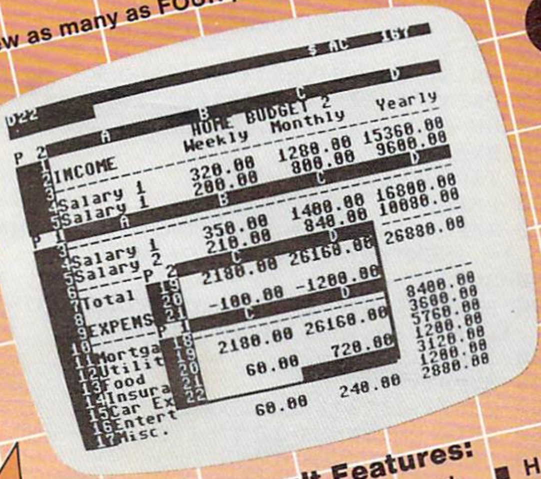
```
1 FORI=828TO883:READA:POKEI,A:NEXTI
10 REM"D=DSAVE"@BACK2",D0:?DS$:CATALOGD0
20 BB=PEEK(44)+27:POKE995,BB
30 POKE998,PEEK(55):POKE999,PEEK(56):POKE
   55,0:POKE56,BB:CLR
40 BB=PEEK(995)
50 N=PEEK(999)-BB-1:BA=BB*256:MA=828
60 DIMBM%(35,24)
70 FORJ=0TO7:TA(J)=2↑J:NEXT
80 PRINT"{CLEAR}{03 RIGHT}{REV}BACKUP 154
   1{OFF}"
```

32 pages of 63 × 254 cells



GRAPHIC DISPLAY on screen and printer

View as many as FOUR pages at one time



the foxiest

PLANNING and FORECASTING TOOL AVAILABLE

Calc Result™

For the COMMODORE 64™

with disk drive

We started with the best that standard spreadsheet programs could offer.

Then we added features designed to create a planning tool more useful than ever before.

The Result?

CALC RESULT! The most powerful, understandable and economical spreadsheet program on the market today.

Thirty-two pages, graphics and the ability to view up to four pages (spreadsheets) at once make CALC RESULT the spreadsheet program that outfoxes them all!

Calc Result Features:

- A three dimensional spreadsheet with 32 pages of 63 × 254 cells, offering unrivaled flexibility
- Graphic display on screen and printer
- The ability to view as many as four pages at once through a window and split screen. This allows you to compare spreadsheets!
- Help function on-line to make Calc Result's features easy to use
- Color coordinated cells that speed calculations
- IF-THEN-ELSE with AND, OR and NOT-ELSE functions in each cell give you unlimited possibilities for decision making
- Timesaving full function editing
- Flexible column width printouts for formatting reports
- Utilization of memory only in cells that are active
- Replicate, copy and move commands that save time
- Consolidation of spreadsheets to get the "bottom line"
- Protection of cells containing formulas
- Ability to load VisiCalc™ files

Distributed by:

COMPUTER MARKETING SERVICES INC.

300 W. Marlton Pike • Cherry Hill, N.J. 08002 • 609-795-9480

"Our products outfox them all!"

```
90 PRINT"{DOWN}'GOTO10000' IF PROGRAM QUI
   TS ABNORMALLY"
100 PRINT"{DOWN}"N"BUFFERS AVAILABLE"
110 OPEN1,8,15
200 REM *** MAIN FUNCTIONS ****
210 GOSUB1000
220 D$="S":GOSUB3200:I2$=IR$
230 IFDR$<>"2A"THENPRINT"{REV}ILLEGAL DOS ~
   1.0 DISK{OFF}":GOTO10000
240 IFI2$=I1$THENPRINT"{REV}SOURCE AND DES
   TINATION HAVE SAME ID CODE{OFF}":
   GOTO10000
250 GOSUB2500
260 T=TS:S=0:NU=1:T1=T:S1=S
270 PRINT#1,"I0":OPEN3,8,3,"#"
280 PRINT"READING BLOCK #";
290 IFBM%(T1,S1)=0THENGOSUB2000:NU=NU+1:IF
   NU>NTHEN320
300 S1=S1+1:IFS1>20THENS1=0:T1=T1+1
310 IFT1<TF+1THEN290
320 PRINT"{DOWN}"
330 CLOSE3
340 D$="D":GOSUB3200:IFIR$<>I1$THENGOTO340
350 PRINT#1,"I0":OPEN3,8,3,"#"
360 PRINT"WRITING BUFFER #";
370 NU=1:T1=T:S1=S
380 IFBM%(T1,S1)=0THENGOSUB2200:NU=NU+1:IF
   NU>NTHEN410
390 S1=S1+1:IFS1>20THENS1=0:T1=T1+1
400 IFT1<TF+1THEN380
410 PRINT"{DOWN}"
420 CLOSE3
430 S=S1+1:IFS>20THENS=0:T1=T1+1
440 T=T1:IFT>TFTHEN500
450 D$="S":GOSUB3200:IFIR$<>I2$THEN450
460 NU=1:T1=T:S1=S:GOTO270
500 REM FINISHED XFERS
510 CLOSE1
520 POKE55,PEEK(998):POKE56,PEEK(999):CLR
530 PRINT"{02 DOWN}BACKUP COMPLETE"
540 OPEN1,8,0,"$0"
550 GET#1,A$:IFA$<>"{REV}"THEN550
560 PRINTA$;:GOTO610
570 GET#1,A$:SS=ST:A=LEN(A$):IFATHENA=ASC(
   A$)
580 GET#1,B$:SS=ST:B=LEN(B$):IFBTHENA=ASC(
   B$)
590 IFSSTHEN660
600 IFA=1ANDB=1THENGOSUB630
610 GET#1,A$:IFA$=""THENPRINT:GOTO570
620 PRINTA$;:GOTO610
630 GET#1,A$:SS=ST:A=LEN(A$):IFATHENA=ASC(
   A$)
640 GET#1,B$:SS=ST:B=LEN(B$):IFBTHENB=ASC(
   B$)
650 N=B*256+A:PRINTN;:RETURN
660 CLOSE1
670 END
1000 REM HEADER DEST DISK
1010 PRINT"{DOWN}INSERT DESTINATION DISK TO
   BE FORMATTED"
1020 INPUT"{02 DOWN}DISK NAME{03 RIGHT}_    ~
              {19 LEFT}";DN$
1030 IFDN$="_"THENPRINT"{03 UP}";:GOTO1020
1040 IFLEN(DN$)>16THENCLR:GOTO40
1050 F=0:FORJ=1TOLEN(DN$):S1$=MID$(DN$,J,1)
1060 IFS1$="_"ORS1$=CHR$(34)THENF=1
1070 NEXTJ:IFFTHENPRINT"{03 UP}";:GOTO1020
1080 INPUT"{DOWN}UNIQUE DISK ID{03 RIGHT}_  ~
              {23 LEFT}";I1$
1090 IFI1$=" "THENPRINT"{02 UP}";:GOTO1080
1100 IFLEN(I1$)<>2THENPRINT"{02 UP}";:GOTO1
 080
1110 PRINT#1,"N0:"+DN$+","+I1$
1120 GOSUB3000
1130 IFERTHENPRINTER$:GOTO10000
1140 RETURN
2000 REM READ BLOCK T1,S1 TO BUFFER # NU
2010 C=.
2020 PRINT#1,"U1";3;0;T1;S1
2030 GOSUB3000:IFNOTERTHEN2060
2040 C=C+1:IFC<3GOTO2020
2050 PRINTER$:FORJ=(BB+NU)*256TO(BB+NU)*256
   +255:POKEJ,.:NEXTJ:GOTO2100
2060 PRINT#1,"B-P";3;0
2070 IFNU<>0THENPRINT"   {03 LEFT}";RIGHT$(
   "  "+STR$(NU),3);"{03 LEFT}";
2080 POKE996,PEEK(3):POKE997,PEEK(4):POKE4,
   BB+NU:SYSMA
2085 POKE3,PEEK(996):POKE4,PEEK(997)
2090 IFST<>.ANDST<>64THENGOSUB3000:GOTO2050
2100 RETURN
2200 REM WRITE BLOCK T1,S1 FROM BUFFER # NU
2210 C=.
2220 PRINT#1,"B-A";0;T1;S1:PRINT#1,"B-P";3;
   0
2230 PRINT"   {03 LEFT}";RIGHT$("  "+STR$(N
   U),3);"{03 LEFT}";
2240 POKE996,PEEK(3):POKE997,PEEK(4):POKE4,
   BB+NU:SYSMA+3
2245 POKE3,PEEK(996):POKE4,PEEK(997)
2250 IFST<>.ANDST<>64THENPRINT"{REV}IEEE WR
   ITE ERROR"ST"{OFF}":GOTO10000
2260 PRINT#1,"U2";3;0;T1;S1
2270 GOSUB3000:IFNOTERTHEN2300
2280 C=C+1:IFC<3THEN2260
2290 PRINT"{REV}UNRECOVERABLE WRITE ERROR"E
   R$:GOTO10000
2300 RETURN
2500 REM GET BAM TO BM%(T,S)
2510 TS=1:TF=.
2520 PRINT#1,"I0":OPEN3,8,3,"#"
2530 S9=0
2540 PRINT"{DOWN}TRACK #   BLOCKS TO XFER"
2550 PRINT"########################"
2560 NU=0:T1=18:S1=0:C0$=CHR$(.):GOSUB2000
2570 BY=4
2580 T%=(BY-4)/4+1
2590 PRINT"  ";T%;
2600 IFPEEK(BA+BY)=.THENFORJ=.TO20:BM%(T%,J
   )=.:NEXT:BY=BY+4:GOTO2650
2610 S=0
2620 BY=BY+1:A0=PEEK(BA+BY):FORJ=.TO7:BM%(T
   %,S)=A0ANDTA(J):S=S+1:NEXT
2630 IFS<22THEN2620
2640 BY=BY+1
2650 ES=21:IFT%>17THENES=19
2660 IFT%>24THENES=18
2670 IFT%>30THENES=17
2680 FORJ=ESTO24:BM%(T%,J)=-1:NEXT
2690 SM=.:FORJ=.TO20:IFBM%(T%,J)=.THENSM=SM
   +1
2700 NEXT:PRINT TAB(12);SM:S9=S9+SM
2710 IFSM=.ANDTS=T%THENTS=TS+1:GOTO2730
2720 IFSM<>.THENTF=T%
2730 IFBY<143THEN2580
2740 CLOSE3
2750 PRINT"START =";TS;" FINISH =";TF
2760 PRINT"{DOWN}A TOTAL OF";S9;"BLOCKS TO ~
   XFER"
2770 S8=90+25+(.650+.980)*S9
2780 S7=INT(S8/60):PRINT"APPROX";S7":"INT(S
   8-S7*60);"FOR COPY"
2790 RETURN
3000 REM READ ERR CH TO ER,ER$
3010 INPUT#1,E0$,E1$,E2$,E3$:ER$=E0$+","+E1
```

the *foxiest*

"REALLY FOXY
IS BEING LETTER PERFECT"

WORDPROCESSOR FOR THE COMMODORE 64™ ALSO CHECKS YOUR SPELLING!

SCRIPT 64™

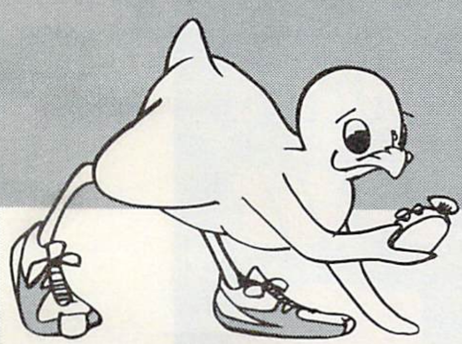
Suggested Retail: $139.95

Contact Your Nearest Commodore Dealer Today . . .
You'll Be So Glad You Did!

Distributed By:
COMPUTER MARKETING SERVICES INC.

300 W. Marlton Pike
Cherry Hill, New Jersey 08002
(609) 795-9480

Commodore 64 is a trademark of Commodore Electronics Limited   Script 64 is a trademark of Richvale Telecommunications



It's Time for TOTL SOFTWARE!

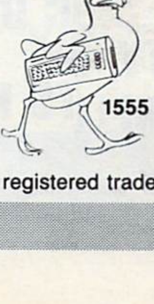
for the VIC 20™ and COMMODORE 64™
WORD PROCESSING AND MAILING LIST & LABEL
now available with ***CHICKSPEED***

FAST PRINTING • LIGHTNING LOADS • SIMPLE COMMANDS

TOTL.TEXT 2.0 + CS VIC + 8K expansion	$25.00
TOTL.TEXT 2.5 + CS VIC + 16K expansion	$35.00
TOTL.TEXT 2.6 + CS Commodore 64	$40.00
TOTL.LABEL 2.1 + CS VIC + 16K expansion	$20.00
TOTL.LABEL 2.6 + CS Commodore 64	$20.00
TOTL TIME MANAGER 2.1 VIC + 8K expansion	$30.00
TOTL TIME MANAGER 2.6 Commodore 64	$35.00
time management, scheduling, reports	
RESEARCH ASSISTANT 2.0 VIC + 8K expansion	$30.00
RESEARCH ASSISTANT 2.0 Commodore 64	$35.00
key word cross-reference research tool	
TOTL.BUSINESS 3.0 VIC + 16K expansion	$85.00
TOTL.BUSINESS 3.6 Commodore 64	$95.00
business programs require disk and are shipped on disk	
One Megabyte Fuzzy Diskette	$25.00
computer novelty pillow	

All programs work with 40/80 column (VIC) and 80 column (64) adapters— compatible with tape or disk systems —shipped on cassette tape— available on disk $4.00 extra.

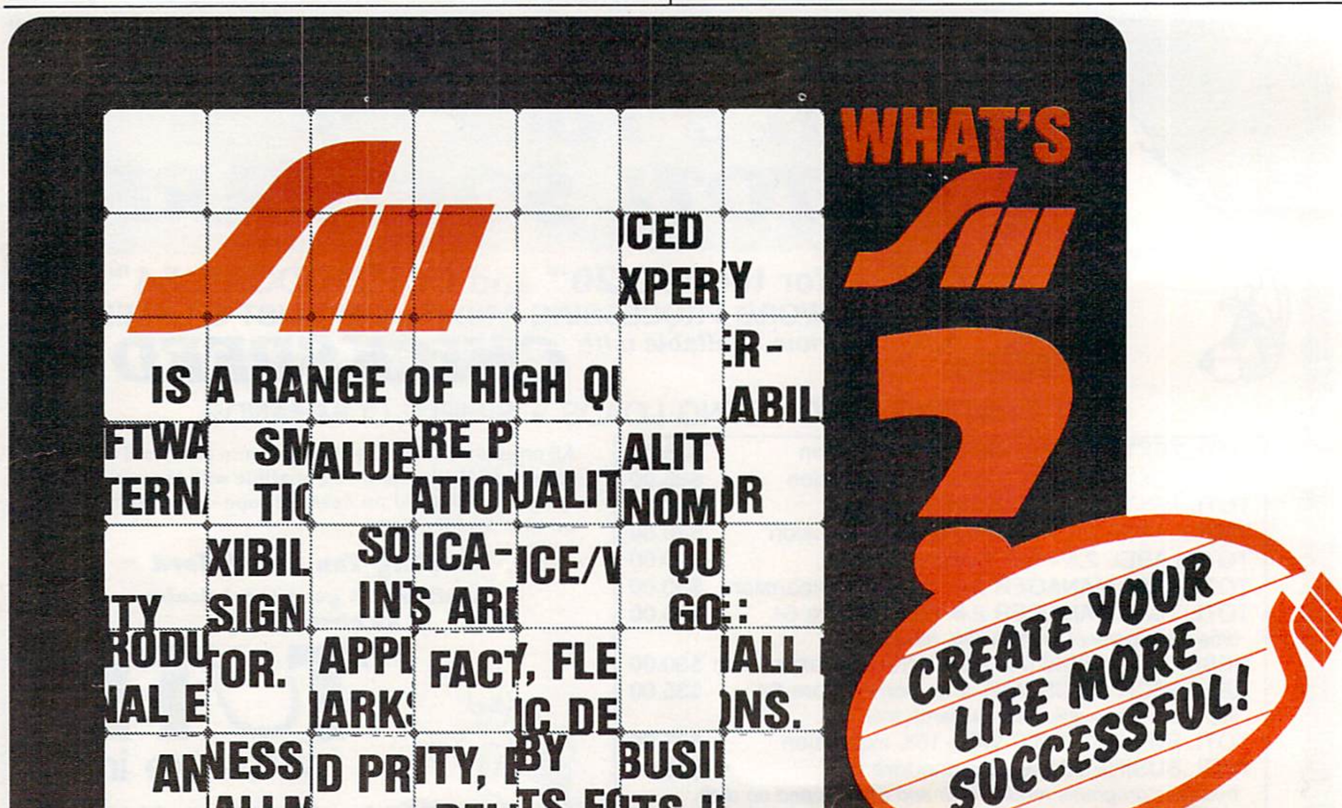
Quality You Can Afford
Available at your local dealer or by phone order



TOTL software inc.
1555 Third Ave., Walnut Creek, CA 94596
MasterCard VISA **Call (415) 943-7877**

Commodore 64 and VIC 20 are registered trademarks of Commodore Electronics, Ltd.

```
    $+","+E2$+","+E3$
3020 ER=LEN(E0$):IFERTHENER=VAL(E0$)
3030 RETURN
3200 REM INSTRUCT TO SWAP TO DISK GIVEN IN ~
    D$
3210 IFD$="D"THENS1$="DESTINATION":GOTO3230
3220 S1$="SOURCE"
3230 PRINT"{DOWN}INSERT ";S1$;" DISK, PRESS
    {REV}SPACE{OFF}"
3240 GETA$:IFA$<>" "THEN3240
3250 OPEN2,8,0,"$0"
3260 GOSUB3000:IFER>0THEN10000
3270 FORJ=1TO26:GET#2,A$:NEXTJ
3280 GET#2,A$:GET#2,B$:IR$=A$+B$
3290 GET#2,A$:GET#2,A$:GET#2,B$:DR$=A$+B$
3300 CLOSE2:RETURN
10000 REM DROP OUT
10010 POKE55,PEEK(998):POKE56,PEEK(999):
    CLR:STOP
15000 DATA 76,66,3,76,91,3,162,3,32,198,255,
    160,0,132,3,32,207,255,145
15010 DATA 3,165,144,208,3,200,208,244,32,
    204,255,96,162,3,32,201,255,160
15020 DATA 0,132,3,177,3,32,210,255,165,144,
    208,3,200,208,244,32,204,255,96
```
©



CIRCLES

Jeffrey S. McArthur

Every Atari graphics programmer needs to draw circles. This tutorial will show you how to draw a circle – and draw one fast – without jumping through hoops. There are several drawing utilities here, from an elementary BASIC routine which takes 60 seconds to a machine language version that finishes in a fraction of a second. Even if you're not interested in the methodology, you can still use these subroutines in your graphics and games.

Program 1 draws circles, but takes more than a minute to draw a circle, no matter how big or small it is.

Reflections

A circle is symmetrical, so why don't we take advantage of its symmetry? If we know the value of one point, we can reflect it across the X-axis or across the Y-axis. That is, if we know (X,Y) is a point on the circle, then so is (X,-Y). The same is true for (-X,Y) and (-X,-Y). So we have to do only a quarter of the work. Circles are also symmetrical along the X=Y line. If we know (X,Y) is on the circle, then so is (Y,X). Now we have to find only an eighth of the points. Program 2 uses that method.

Unfortunately, even doing only one-eighth of the work, we still need more than ten seconds to draw the circle. Perhaps there is a better way. Instead of using sines and cosines, use the equation:

X*X + Y*Y = R*R

That isn't very useful, but we can rearrange the equation and get:

Y = SQRT (R*R - X*X)

So all we have to do is find Y for X = -R to R. However, since the square root function returns only the positive square root, we also have to plot the negative square root. Program 3 is an example of how to do that. This method is faster than using sines or cosines, but it still takes more than 16 seconds. So using Program 4, we reflect it, like we did in Program 2.

Now we have a method that takes only five seconds on a large circle and is a lot faster on the smaller ones. If you take a close look at how Program 4 draws the circle, you see it draws lines of different lengths. This method works fine on a screen, but on a plotter the circle has flat spots.

A Faster Circle

The screen is made up of an array of points. Each point is addressed by two coordinates (X,Y). However, X and Y are *always* integers. In Atari BASIC you can PLOT 0.5,0.5, but the points are rounded to integers. So if you are at one point on the circle and are trying to figure where the next point is, you can go in eight directions.

If you divide the circle into quarters, then only three of those directions are valid. If you divide the circle into eight parts, you can go in only two directions. For example, if you are on the circle at (R,0), the next point is either (R-1,0) or (R-1,1). This method is called a *potential function*. Since the screen cannot plot points except with integers, there is a small error that is not always equal to zero.

We want to keep the error as small as possible. We also reflect it eight ways as before. That takes only three seconds, and we never have to draw any long lines. Program 5 uses this method.

Notice also that you can achieve the entire result using only addition and subtraction. Such programs can be easily converted to machine language since we don't have to multiply or divide. Program 7 is a machine language program to draw a circle. Program 6 calls the machine language and takes less than two-tenths of a second to draw a circle.

The machine language is called by a USR function. The parameters that are passed to it are, in order: the address of the code, the X coordinate of the center of the circle, the Y coordinate of the center of the circle, the radius, and the mode of drawing. The mode of drawing means

0: turn point off
1: turn point on
2: invert point

The program can be converted to any 6502 machine. The only things that need to be changed are where the variables are stored and how to plot the points.

The only problem with the machine language program is that it does no checking to see if the circle goes off screen. And no clipping is done. Therefore, if your circle goes off screen, you will write over other memory.

Program 1: Sines And Cosines

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #1
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 61 SECONDS
200 DEG
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
300 FOR ALPHA=0 TO 360
310 X1=INT(R*COS(ALPHA)+0.5)
320 Y1=INT(R*SIN(ALPHA)+0.5)
330 PLOT A+X1,B+Y1
340 NEXT ALPHA
```

Program 2: Sines And Cosines Reflected

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #2
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 11 SECONDS
200 DEG
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 PLOT A+R,B
300 FOR ALPHA=0 TO 45
310 X1=INT(R*COS(ALPHA)+0.5)
320 Y1=INT(R*SIN(ALPHA)+0.5)
330 PLOT A+X1,B+Y1
340 PLOT A-X1,B+Y1
350 PLOT A+X1,B-Y1
360 PLOT A-X1,B-Y1
370 PLOT A+Y1,B+X1
380 PLOT A-Y1,B+X1
390 PLOT A+Y1,B-X1
400 PLOT A-Y1,B-X1
410 NEXT ALPHA
```

Program 3: Square Root

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #3
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 17 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 X0=-R:Y0=0
300 FOR X1=-R TO R
310 Y1=INT(0.5+SQR(R*R-X1*X1))
330 PLOT A+X0,B+Y0:DRAWTO A+X1,B+Y1
335 PLOT A+X0,B-Y0:DRAWTO A+X1,B-Y1
336 X0=X1:Y0=Y1
340 NEXT X1
```

Program 4: Square Root Reflected

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #4
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 5 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 X0=-R:Y0=0
280 X1=-R
290 Y1=INT(0.5+SQR(R*R-X1*X1))
300 PLOT A+X0,B+Y0:DRAWTO A+X1,B+Y1
310 PLOT A-X0,B+Y0:DRAWTO A-X1,B+Y1
320 PLOT A+X0,B-Y0:DRAWTO A+X1,B-Y1
330 PLOT A-X0,B-Y0:DRAWTO A-X1,B-Y1
340 PLOT A+Y0,B+X0:DRAWTO A+Y1,B+X1
350 PLOT A-Y0,B+X0:DRAWTO A-Y1,B+X1
360 PLOT A+Y0,B-X0:DRAWTO A+Y1,B-X1
370 PLOT A-Y0,B-X0:DRAWTO A-Y1,B-X1
380 X0=X1:Y0=Y1
390 IF -X1>=Y1 THEN X1=X1+1:GOTO 290
```

Program 5: Potential

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #5
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 3 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 PHI=0
280 Y1=0
290 X1=R
300 PHIY=PHI+Y1+Y1+1
310 PHIXY=PHIY-X1-X1+1
400 PLOT A+X1,B+Y1
410 PLOT A-X1,B+Y1
420 PLOT A+X1,B-Y1
430 PLOT A-X1,B-Y1
440 PLOT A+Y1,B+X1
450 PLOT A-Y1,B+X1
460 PLOT A+Y1,B-X1
470 PLOT A-Y1,B-X1
500 PHI=PHIY
510 Y1=Y1+1
520 IF ABS(PHIXY)<ABS(PHIY) THEN PHI
    =PHIXY:X1=X1-1
530 IF X1>=Y1 THEN 300
```

Program 6: BASIC Call To Machine Language

```
100 REM CIRCLE DEMONSTRATION
110 REM PROGRAM #6
140 REM THIS METHOD TAKES APPROXIMAT
    ELY 0.1833 SECONDS
210 GRAPHICS 8
220 COLOR 1
230 SETCOLOR 2,0,0
240 A=160
250 B=80
260 R=50
270 P=7*16*16*16
300 I=USR(P,A,B,R,1)
```

Program 7:
Machine Language Circle Drawing Subroutine

```
10 REM 28000- IS SUBROUTINE
20 GOSUB 28000
30 END
28000 FOR I=0 TO 758:READ A:POKE 286
   72+I,A:NEXT I
28010 RETURN
28672 DATA 104,104,141,5,6,104
28678 DATA 141,4,6,104,141,7
28684 DATA 6,104,141,6,6,104
28690 DATA 141,9,6,141,12,6
28696 DATA 104,141,8,6,141,11
28702 DATA 6,104,104,141,10,6
28708 DATA 201,3,144,1,96,169
28714 DATA 0,141,13,6,141,14
28720 DATA 6,141,15,6,141,16
28726 DATA 6,24,173,4,6,109
28732 DATA 11,6,141,25,6,173
28738 DATA 5,6,109,12,6,141
28744 DATA 26,6,24,173,4,6
28750 DATA 109,13,6,141,29,6
28756 DATA 173,5,6,109,14,6
28762 DATA 141,30,6,56,173,4
28768 DATA 6,237,11,6,141,27
28774 DATA 6,173,5,6,237,12
28780 DATA 6,141,28,6,56,173
28786 DATA 4,6,237,13,6,141
28792 DATA 31,6,173,5,6,141
28798 DATA 14,6,141,32,6,24
28804 DATA 173,6,6,109,11,6
28810 DATA 141,33,6,173,7,6
28816 DATA 109,12,6,141,34,6
28822 DATA 24,173,6,6,109,13
28828 DATA 6,141,37,6,173,7
28834 DATA 6,109,14,6,141,38
28840 DATA 6,56,173,6,6,237
28846 DATA 11,6,141,35,6,173
28852 DATA 7,6,237,12,6,141
28858 DATA 36,6,56,173,6,6
28864 DATA 237,13,6,141,39,6
28870 DATA 173,7,6,237,14,6
28876 DATA 141,40,6,173,25,6
28882 DATA 141,0,6,173,26,6
28888 DATA 141,1,6,173,37,6
28894 DATA 141,2,6,173,38,6
28900 DATA 141,3,6,32,106,114
28906 DATA 173,27,6,141,0,6
28912 DATA 173,28,6,141,1,6
28918 DATA 32,106,114,173,25,6
28924 DATA 141,0,6,173,26,6
28930 DATA 141,1,6,173,39,6
28936 DATA 141,2,6,173,40,6
28942 DATA 141,3,6,32,106,114
28948 DATA 173,27,6,141,0,6
28954 DATA 173,28,6,141,1,6
28960 DATA 32,106,114,173,29,6
28966 DATA 141,0,6,173,30,6
28972 DATA 141,1,6,173,33,6
28978 DATA 141,2,6,173,34,6
28984 DATA 141,3,6,32,106,114
28990 DATA 173,31,6,141,0,6
28996 DATA 173,32,6,141,1,6
29002 DATA 32,106,114,173,29,6
29008 DATA 141,0,6,173,30,6
29014 DATA 141,1,6,173,35,6
29020 DATA 141,2,6,173,36,6
29026 DATA 141,3,6,32,106,114
29032 DATA 173,31,6,141,0,6
29038 DATA 173,32,6,141,1,6
29044 DATA 32,106,114,173,14,6
29050 DATA 205,12,6,240,3,144
29056 DATA 10,96,173,13,6,205
29062 DATA 11,6,144,1,96,173
29068 DATA 11,6,133,4,173,12
29074 DATA 6,133,5,173,13,6
29080 DATA 133,205,173,14,6,133
29086 DATA 206,6,4,38,5,6
29092 DATA 205,38,206,56,165,205
29098 DATA 109,15,6,141,17,6
29104 DATA 165,206,109,16,6,141
29110 DATA 18,6,24,173,17,6
29116 DATA 229,4,141,19,6,173
29122 DATA 18,6,229,5,141,20
29128 DATA 6,173,18,6,16,27
29134 DATA 73,255,141,22,6,173
29140 DATA 17,6,73,255,24,105
29146 DATA 1,141,21,6,173,22
29152 DATA 6,105,0,141,22,6
29158 DATA 24,144,9,141,22,6
29164 DATA 173,17,6,141,21,6
29170 DATA 173,20,6,16,27,73
29176 DATA 255,141,24,6,173,19
29182 DATA 6,73,255,24,105,1
29188 DATA 141,23,6,173,24,6
29194 DATA 105,0,141,24,6,24
29200 DATA 144,9,141,24,6,173
29206 DATA 19,6,141,23,6,173
29212 DATA 17,6,141,15,6,173
29218 DATA 18,6,141,16,6,24
29224 DATA 173,13,6,105,1,141
29230 DATA 13,6,173,14,6,105
29236 DATA 0,141,14,6,173,22
29242 DATA 6,205,24,6,144,39
29248 DATA 208,8,173,21,6,205
29254 DATA 23,6,144,29,173,19
29260 DATA 6,141,15,6,173,20
29266 DATA 6,141,16,6,56,173
29272 DATA 11,6,233,1,141,11
29278 DATA 6,173,12,6,233,0
29284 DATA 141,12,6,76,55,112
29290 DATA 173,2,6,133,205,169
29296 DATA 0,133,206,6,205,38
29302 DATA 206,6,205,38,206,6
29308 DATA 205,38,206,165,205,133
29314 DATA 4,165,206,133,5,6
29320 DATA 205,38,206,6,205,38
29326 DATA 206,24,165,205,101,4
29332 DATA 133,205,165,206,101,5
29338 DATA 133,206,173,0,6,133
29344 DATA 4,173,1,6,133,5
29350 DATA 70,5,102,4,70,5
29356 DATA 102,4,70,5,102,4
29362 DATA 24,165,205,101,4,133
29368 DATA 205,165,206,101,5,133
29374 DATA 206,24,165,205,101,88
29380 DATA 133,205,165,206,101,89
29386 DATA 133,206,173,0,6,41
29392 DATA 7,170,160,0,173,10
29398 DATA 6,208,10,189,41,6
29404 DATA 73,255,49,205,145,205
29410 DATA 96,201,1,208,8,189
29416 DATA 41,6,17,205,145,205
29422 DATA 96,189,41,6,81,205
29428 DATA 145,205,96,0,0,0
```

©

COMPUTE!'s Second Book Of Atari

After only three years on the market, the Atari 400/800 microcomputers have become among the most popular personal computers ever made. So it was no surprise when *COMPUTE!'s First Book of Atari*, a collection of the best Atari articles published during 1980-81 in **COMPUTE!** Magazine, also became a "bestseller" with Atari enthusiasts. The first printing sold out in just a few months.

That's why we've followed up with *COMPUTE!'s Second Book of Atari*. Available immediately, the *Second Book of Atari* continues **COMPUTE!**'s tradition for personal computer users.

But the *Second Book of Atari* differs from the *First Book* in one important respect – all the articles are totally new and previously unpublished. The *Second Book of Atari* includes such interesting articles as "Page Flipping," "Fun With Scrolling," "Perfect Pitch," "Player-Missile Drawing Editor," and "TextPlot Makes a Game." Whole chapters are devoted to subjects such as "Advanced Graphics and Game Utilities," "Programming Techniques," and "Beyond BASIC." With 250 pages – more than 25 percent thicker than the *First Book* at the same price – the *Second Book of Atari* is crammed with information and ready-to-type program listings. And the book is spiral-bound to lie flat and is fully indexed for quick reference.

Best of all, *COMPUTE!'s Second Book of Atari*, like **COMPUTE!** Magazine itself, is written and edited to appeal to all computer enthusiasts – beginners and experts alike. Priced at only $12.95.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2.00 shipping and handling. Outside the U.S., add $5.00 for air mail, $2.00 for surface mail. All orders prepaid, U.S. funds only.

PET Uncompactor

David L. Evans

The PET Compactor program (July 1982) was a popular, very fast way to squeeze a BASIC program into the smallest amount of memory possible. It created "metalines," some far longer than 80 characters, using a new line number only when the program's logic demanded it. Here's the companion utility. Also written entirely in machine language (and requiring Upgrade or 4.0 BASIC, with disk), the Uncompactor stretches a compact BASIC program out into many small lines. This makes modifications and program analysis easier. Often, a compacted program cannot be changed at all without being uncompacted first. The program is provided as a hex dump, with instructions on how to enter it into your PET.

This machine language routine uncompacts fast. In fact, it represents a 3300% increase in speed over an uncompactor written in BASIC.

Unlike my "Compactor" program published last year, it requires *no* changes to run on either Upgrade or 4.0 PET BASIC. It achieves this by making heavy use of the "kernal" (the jump table located at the top of memory in all PET/CBMs). The kernal is used to PRINT, OPEN and CLOSE files, GET and INPUT bytes, and to restore the original environment (default parameters) of the PET.

The routines to GET, INPUT, and RESTORE are all straightforward; all the user does is execute a subroutine call to the desired routine.

For example, to use the routine RESTORE, the user types:

```
JSR $FFCC
```

Both of the routines to GET and INPUT return the value that was input into the accumulator.

The PRINT routine (JSR $FFD2) requires that the accumulator be loaded with the byte that the user wishes to be printed. The routine to set the OUTPUT or INPUT device also requires the user to set up some parameters before calling them. The user must first open the file to be accessed, then load the X-register with the file number, and finally execute a subroutine call to the routine desired.

Example: To print a colon to file number two, do the following (this assumes that file number two has been opened):

```
LDX #$02
JSR $FFC9   ; set current output device
LDA #$3A
JSR $FFD2   ; print a colon
JSR $FFCC   ; restore default devices
```

All the routines discussed above are widely used. The routines to OPEN and CLOSE files, however, are not as well-known. Each routine requires that you have, somewhere in memory, a string of characters containing the OPEN/CLOSE command. BASIC is informed where the command string is, by setting locations $77 and $78 to point to it.

Example: To open file number 15,8,15, type the following:

```
LDA #<COMAND
STA $77
LDA #>COMAND
STA $78
JSR $FFC0 ;open the file
COMAND .BYTE '15,8,15'
```

Note: My assembler uses "<" to load the LSB of a label and ">" to load the MSB of a label. To CLOSE a file, the same procedure is used.

The program is provided in the form of a hex dump of memory. To enter this into your computer, invoke the built-in monitor by typing SYS 4. Next, display the first block of memory by typing m 0400 047f. Type over the numbers already in memory with the new values in the program, hitting RETURN after each line of eight bytes. Repeat this procedure for the following blocks of memory until all changes have been made. Then save the program to disk by typing:

```
S "UNCOMPACTOR", 08, 0400, 08E7
```

Since the program occupies the normal BASIC program area, and since the first 13 bytes constitute a short "self-calling" routine, the program can be loaded and run as if it were in BASIC.

It is not necessary to initialize the drives used; the program will automatically do it for you. If the output file name exists on the destination diskette, the program will overlay it. Follow the directions printed on the screen and your program will then be uncompacted. When the program is finished, LOAD the new version of your program and type the CLR command. This is necessary to relink the BASIC program. Be sure to reSAVE your CLRed program or else you will lose it.

For those who do not want to type this in, send $3 and a tape or disk along with a SASE mailer to the address below. If you send a disk, I have DOS 2.0 so all disks will be written in DOS 2.0.

I have source code available in CBM assembler format. If you would like a copy of the source code, be sure to make a note of it when you send for a copy of my program.

David L. Evans
2202 Ellis Avenue
Caldwell, ID 83605

PET Machine Language Uncompactor.

```
0400 00 0B 04 FF FF 9E 31 30
0408 33 37 00 00 00 A9 E2 85
0410 01 A9 08 85 02 A0 00 98
0418 91 01 C8 D0 FB E6 02 A6
0420 02 E0 0B D0 F3 A2 BC A0
0428 07 20 E8 06 A2 00 BD D1
0430 08 F0 06 9D F1 08 E8 D0
0438 F5 20 CF FF C9 0D F0 08
0440 9D F1 08 E8 E0 19 D0 F1
0448 A0 00 B9 94 08 F0 07 9D
0450 F1 08 E8 C8 D0 F4 AD F8
0458 08 C9 30 F0 04 C9 31 D0
0460 C4 AD F9 08 C9 3A D0 BD
0468 A2 7E A0 08 20 E8 06 A2
0470 00 BD D9 08 F0 06 9D 15
0478 09 E8 D0 F5 20 CF FF C9
0480 0D F0 08 9D 15 09 E8 E0
0488 1A D0 F1 A0 00 B9 9A 08
0490 F0 07 9D 15 09 E8 C8 D0
0498 F4 AD 1D 09 C9 30 F0 04
04A0 C9 31 D0 C4 AD 1E 09 C9
04A8 3A D0 BD A5 77 8D E2 08
04B0 A5 78 8D E3 08 A9 C9 85
04B8 77 A9 08 85 78 20 C0 FF
04C0 A2 0F 20 C9 FF AD F8 08
04C8 CD 1D 09 F0 10 A9 49 20
04D0 D2 FF AD F8 08 20 D2 FF
04D8 A9 0D 20 D2 FF A9 49 20
04E0 D2 FF AD 1D 09 20 D2 FF
04E8 A9 0D 20 D2 FF 20 CC FF
04F0 20 10 07 A9 F1 85 77 A9
04F8 08 85 78 20 C0 FF 20 10
0500 07 A9 15 85 77 A9 09 85
0508 78 20 C0 FF 20 10 07 A2
0510 AB A0 08 20 E8 06 20 F9
0518 06 A2 06 20 C9 FF AD E4
0520 08 20 D2 FF AD E5 08 20
0528 D2 FF 20 CC FF AD E4 08
0530 D0 11 20 FF 06 A2 06 20
0538 C9 FF AD E5 08 20 D2 FF
0540 20 CC FF A9 01 8D F0 08
0548 D0 46 AD E8 08 8D EA 08
0550 AD E9 08 8D EB 08 AD EE
0558 08 D0 03 4C A5 06 AD EA
0560 08 AE EB 08 20 53 07 A2
0568 06 20 C9 FF A9 01 20 D2
0570 FF 20 D2 FF AD EC 08 20
0578 D2 FF AD ED 08 20 D2 FF
0580 20 CC FF A0 01 20 FF 06
0588 99 6B 09 F0 03 C8 D0 F5
0590 20 F9 06 18 6D E4 08 8D
0598 EE 08 90 03 EE EE 08 F0
05A0 12 20 F9 06 8D E9 08 8D
05A8 ED 08 AD E4 08 8D E8 08
05B0 8D EC 08 AD F0 08 F0 07
05B8 A9 00 8D F0 08 F0 8B A0
05C0 01 B9 6B 09 C9 3A D0 51
05C8 C0 01 D0 03 4C 8B 06 EE
05D0 EA 08 D0 03 EE EB 08 AD
05D8 EA 08 CD E8 08 90 0B AD
05E0 EB 08 CD E9 08 90 03 4C
05E8 8B 06 A2 06 20 C9 FF A9
05F0 00 20 D2 FF A9 01 20 D2
05F8 FF 20 D2 FF AD EA 08 20
0600 D2 FF AD EB 08 20 D2 FF
0608 20 CC FF C8 B9 6B 09 C9
0610 20 F0 F8 C9 3A F0 F4 D0
0618 A8 A9 81 D9 6B 09 B0 4A
0620 A9 9B D9 6B 09 90 43 A9
0628 80 D9 6B 09 F0 23 A9 99
0630 D9 6B 09 90 1C A9 8A D9
0638 6B 09 B0 2E A9 90 D9 6B
0640 09 90 27 A9 8C D9 6B 09
0648 F0 20 A9 8D D9 6B 09 F0
0650 19 A2 06 20 C9 FF B9 6B
0658 09 20 D2 FF B9 6B 09 F0
0660 03 C8 D0 ED 20 CC FF 4C
0668 4A 05 B9 6B 09 C9 22 D0
0670 1A A2 06 20 C9 FF B9 6B
0678 09 20 D2 FF 20 CC FF C8
0680 B9 6B 09 C9 22 F0 04 C9
0688 00 D0 E6 A2 06 20 C9 FF
0690 B9 6B 09 20 D2 FF 20 CC
0698 FF B9 6B 09 F0 04 C8 4C
06A0 C1 05 4C 4A 05 A2 06 20
06A8 C9 FF A9 00 20 D2 FF 20
06B0 D2 FF 20 CC FF A2 A0 A0
06B8 08 20 E8 06 A9 C2 85 77
06C0 A9 08 85 78 20 C3 FF A9
06C8 C4 85 77 A9 08 85 78 20
06D0 C3 FF A9 C6 85 77 A9 08
06D8 85 78 20 C3 FF AD E2 08
06E0 85 77 AD E3 08 85 78 60
06E8 86 01 84 02 A0 00 B1 01
06F0 F0 06 20 D2 FF C8 D0 F6
06F8 60 20 FF 06 8D E4 08 A2
0700 05 20 C6 FF 20 E4 FF 48
0708 20 CC FF 68 8D E5 08 60
0710 A2 0F 20 C6 FF A2 00 20
0718 E4 FF 9D 39 09 C9 0D F0
0720 03 E8 D0 F3 20 CC FF AD
0728 39 09 C9 32 B0 01 60 A9
0730 0D 20 D2 FF 20 D2 FF A2
0738 00 BD 39 09 20 D2 FF C9
0740 0D F0 03 E8 D0 F3 68 68
0748 AD E2 08 85 77 AD E3 08
0750 85 78 60 8D E6 08 8E E7
0758 08 A9 20 20 D2 FF 20 D2
0760 FF 20 D2 FF A0 00 8C EF
0768 08 A2 00 AD E6 08 38 F9
0770 B4 07 8D E6 08 AD E7 08
0778 C8 F9 B4 07 90 08 8D E7
0780 08 E8 88 4C 6B 07 88 AD
0788 E6 08 79 B4 07 8D E6 08
0790 8A D0 09 2C EF 08 30 09
0798 A9 20 D0 07 A2 80 8E EF
07A0 08 09 30 20 D2 FF C8 C8
07A8 C0 08 90 BD AD E6 08 09
07B0 30 4C D2 FF 10 27 E8 03
07B8 64 00 0A 00 93 20 20 20
07C0 20 20 20 4D 41 43 48 49
07C8 4E 45 20 4C 41 4E 47 55
07D0 41 47 45 20 55 4E 43 4F
07D8 4D 50 41 43 54 4F 52 0D
07E0 0D 0D 42 59 3A 20 44 41
07E8 56 49 44 20 45 56 41 4E
07F0 53 0D 0D 0D 45 4E 54 45
07F8 52 20 46 49 4C 45 20 4E
0800 41 4D 45 20 57 49 54 48
0808 20 54 48 45 20 44 52 49
0810 56 45 20 4E 55 4D 42 45
0818 52 0D 0D 50 52 45 43 45
0820 44 49 4E 47 20 49 54 2E
0828 0D 0D 45 58 41 4D 50 4C
0830 45 3A 20 20 20 30 3A 46
0838 49 4C 45 4E 41 4D 45 0D
0840 0D 44 4F 20 54 48 45 20
0848 53 41 4D 45 20 57 49 54
0850 48 20 54 48 45 20 4F 55
0858 54 50 55 54 20 46 49 4C
0860 45 20 4E 41 4D 45 2E 0D
0868 0D 0D 0D 49 4E 50 55 54
0870 20 46 49 4C 45 20 4E 41
0878 4D 45 20 3F 20 00 0D 0D
0880 4F 55 54 50 55 54 20 46
0888 49 4C 45 20 4E 41 4D 45
0890 20 3F 20 00 2C 50 2C 52
0898 22 00 2C 50 2C 57 22 00
08A0 93 12 44 4F 4E 45 0D 0D
08A8 0D 0D 0D 93 57 4F 52 4B
08B0 49 4E 47 20 4F 4E 20 4C
08B8 49 4E 45 2E 2E 2E 2E 0D
08C0 0D 00 36 00 35 00 31 35
08C8 00 31 35 2C 38 2C 31 35
08D0 00 35 2C 38 2C 35 2C 22
08D8 00 36 2C 38 2C 36 2C 22
08E0 40 00 00 00 00 00 00 00
```

©

Statistical Test Of Commodore And Radio Shack RND

Brian Flynn

This article provides a statistical test of the randomness of your BASIC's random number generator. Versions of the program for TRS-80 Color Computers with Extended Color BASIC and for PET/CBM, VIC, and 64 computers are provided. To use the TRS-80 version with non-Extended BASIC, you must substitute the value of square root of N for SQR(N) in lines 6110 and 6120, since non-Extended Color BASIC has no square root function. (SQR(1000)=31.6228.) Alternatively, the Color BASIC manual lists a square root routine on page 116. The only changes necessary to use the PET/CBM version on the VIC-20 or Commodore 64 are to adapt the PRINT statements to the smaller screen sizes.

As presented, the program takes several hours to sort each subsequence. Thus, several days would be required for a complete program run. Each of the following options significantly reduces the required execution time:

1. *Replace the sort routine (Module 5) with a faster sorting routine. (See "All Sorts of BASIC Sorts,"* **COMPUTE!**, *December 1982, #31.)*
2. *Compile the program before running it. (Of course, to do this you must have a BASIC compiler.)*
3. *Reduce the number of fractions specified in the DATA statement of line 2020.*

The phrase "Kolmogorov-Smirnov" brings to mind the vision of a big white dog, a beautiful princess, and a bearded, virile, vodka-drinking czar. In reality, however, "Kolmogorov-Smirnov" is not this imaginary troika from pre-Bolshevik days, but rather a statistical test, named after two Russian mathematicians, for trying to determine how well values from a sample match values from a specified population.

The test is often used to examine the degree of randomness of sequences of fractions generated by the computer from a uniform distribution. This article explains the Kolmogorov-Smirnov test in more detail, and then uses the test to evaluate the quality of the random number generator in Microsoft's BASIC compiler for the TRS-80 and Commodore computers.

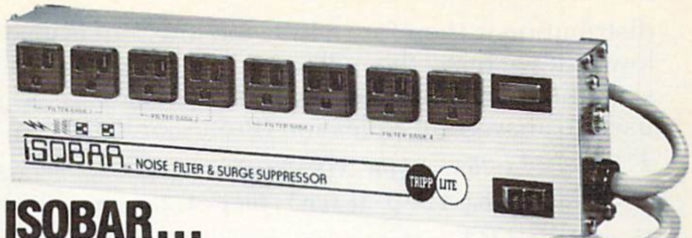
Kolmogorov-Smirnov Test

The command "RND(0)" in TRS-80 BASIC generates a fraction from a uniform distribution between 0 and 1, exclusive. In this distribution, graphed in Figure 1, the probability of drawing a fraction between 0.0 and 0.1, in a one-shot selection, is equal to the probability of drawing a fraction between 0.1 and 0.2, or 0.2 and 0.3, and so on. In each case, the probability is 1/10 since the distribution is divided into ten equal parts.

Now, the Kolmogorov-Smirnov test uses cumulative rather than absolute relative frequency distributions. Referring again to the uniform distribution of Figure 1, note that the probability of drawing a fraction less than or equal to 0.2 is 1/10+1/10, or 0.2. Similarly, the probability of drawing a fraction less than or equal to 0.3 is 1/10+1/10+1/10, or 0.3. In general, the probability of selecting a fraction less than or equal to some number X is simply X, where X ranges from 0 to 1. The distribution based upon these cumulative probabilities is graphed in Figure 2.

The essence of the Kolmogorov-Smirnov test is comparing theoretical and empirical cumulative frequency distributions. An example of the latter type of distribution is based upon the following sequence of ten fractions, rounded to three decimal places, generated by executing "RND(0)" on a Honeywell computer: 0.789, 0.528, 0.871, 0.097, 0.276, 0.434, 0.711, 0.535, 0.776, and 0.918. If the sample sequence is random, then the empirical cumulative frequency distribution, based upon observed values sorted in ascending order, should approximate the theoretical one. These distributions are compared in Table 1 and Figure 3.

These two displays reveal that the observed fractions are a little too high, and that the empirical



ISOBAR NOISE FILTER & SURGE SUPPRESSOR
TRIPP LITE

distribution is therefore a little too low. But is it so low that we reject the null hypothesis of a random sequence? The following two test statistics are used to answer this question (Professor Knuth, p. 45):

$$K^{+} = \sqrt{n}\max\{j/n - F(X_j)\} \text{ and}$$

$$K^{-} = \sqrt{n}\max\{F(X_j) - (j-1)/n\}, \text{ for } j = 1, 2, \ldots, n.$$

The symbol K^+ is the maximum vertical distance between the two curves when the empirical distribution is higher than the theoretical distribution, and K^- is the maximum distance when the empirical distribution is lower. Further, n is the sample size, ten in this case. And $F(X_j)$ is the theoretical cumulative frequency for the j^{th} observation. For example, $F(X_1) = 0.097$ since 9.7% of all values from a uniform distribution are ≤ 0.097. Similarly, $F(X_2) = 0.276$, and so on.

For our data, $K^+ = 0.259$ and $K^- = 0.740$. Referencing Kolmogorov-Smirnov critical values (Professor Knuth, p. 44), both of these statistics fall in the acceptance region for the null hypothesis at the 10% level of significance, using a two-tail test (5% of the distribution's area is under each tail). Hence, we can't label the observed sequence of fractions "nonrandom."

A Practical Application

The quality of the random number generator in Microsoft's BASIC is examined here, using the computer program listed at the end of the article. Specifically, the degree of randomness of the sequence of the first 50,000 fractions generated by RND(0) is investigated. This is done by performing the Kolmogorov-Smirnov test on each successive interval of 1,000 fractions within the total sequence. Hence, 50 values of the K^+ statistic and 50 values of the K^- statistic are tallied.

Test results, summarized in Table 2, reveal that the K^+ and K^- values always fall within the middle 98 percentile portion of the distribution. And they fall within the middle 90 percentile part 92 out of 100 times. These results suggest that the random number generator is a good one.

As an additional check, however, the Kolmogorov-Smirnov test is applied once again, this time to the 50 K^+ values and to the 50 K^- values from before. As Professor Knuth indicates (p. 45), this enables us "... to detect both local and global nonrandom behavior." Test results, using

$$F(X) = 1 - e^{-2X^2}$$

as the cumulative frequency distribution for the K values, are:

	K^+	K^-
Based on 50 K^+'s	0.217	0.650
Based on 50 K^-'s	0.875	0.111

In all four cases, the null hypothesis of a random sequence is not rejected at the 2% level of significance, in a two-tail test. At the 10% level of significance H_o is rejected one out of four times, with 0.111 the guilty value.

The Kolmogorov-Smirnov test is useful in examining the randomness of sequences of fractions generated by RND(0). But remember, no random number generator is perfect. And just because a sequence passes one statistical test does not mean that it will pass a second.

References

Knuth, Donald E. *The Art of Computer Programming*, Vol. 2. Reading: Addison-Wesley Publishing Company, Inc., 1971.

Lapin, Lawrence L. *Statistics for Modern Business Decisions*. New York: Harcourt Brace Jovanovich, Inc., 1973, pp. 422-426.

Table 1: Sample And Theoretical Cumulative Relative Frequencies

Fraction	Sample Cumulative Frequency	Theoretical Cumulative Frequency
0.097	0.1	0.097
0.276	0.2	0.276
0.434	0.3	0.434
0.528	0.4	0.528
0.535	0.5	0.535
0.711	0.6	0.711
0.776	0.7	0.776
0.789	0.8	0.789
0.871	0.9	0.871
0.918	1.0	0.918

Note that the theoretical cumulative frequency always equals the value of the observed fraction. This is because the probability of drawing a fraction less than or equal to, say, 0.276, is 0.276, where the population is the uniform distribution between 0 and 1.

Table 2: Kolmogorov-Smirnov Test Results

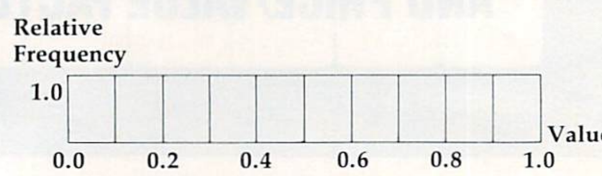
H_o: The sequence is random
H_A: The sequence is nonrandom

Level Of Significance	Critical Values Lower	Critical Values Upper	Number Of Times In 50 Trials That H_0 Is Rejected K^+	K^-
2%	0.066	1.511	0	0
10%	0.156	1.219	4	4
50%	0.375	0.828	26	26

Note: The level of significance is the probability of rejecting H_o when H_o is in fact true.

Figure 1.

Uniform Distribution Between 0 And 1



Universal's Hot as a firecracker deals for July!

TITLE	RETAIL	COST
VIC-20		
Martian Raider (New) (T)	19.95	14.95
Shark Trap (New) (T)	19.95	14.95
Choplifter (SAVE!!!) (C)	44.95	29.95
Rat Hotel (New) (C)	39.95	29.95
Trashman (SAVE!!!) (C)	44.95	29.95
Home/Office (T)	29.95	21.95
Sword of Fargoal (T)	29.95	21.95
Rescue at Rigel (T)	29.95	21.95
HES Writer (C)	39.95	29.95
VIC Forth (C)	59.95	44.95
Exterminator (T)	24.95	18.95
Krazy Kong (T)	12.95	9.95
Quick Brown Fox (C)	65.00	48.95
Deadly Duck (C)	34.95	25.95
Touch Typing Tutor (T)	19.95	14.95
Annihilator (T)	19.95	14.95
Kongo Kong (T)	19.95	14.95
Trek (T)	19.95	14.95
Flash 'N Math Addition (T)	15.95	11.95
Flash 'N Spell (T)	15.95	11.95
Serpentine (SAVE!!!) (C)	44.95	29.95
COMMODORE 64		
Home Accountant (D)	74.95	54.95
Temple of Apshai (D)	39.95	29.95
Upper Reaches of Apshai (D)	19.95	14.95
Curse of Ra (D)	19.95	14.95
Jumpman (New...GREAT!)(D)	39.95	29.95
Coco (D)	49.95	36.95
Gridrunner (C)	39.95	29.95
HES Writer (C)	44.95	34.95
Retro Ball (C)	39.95	29.95
HES Mon 64 (C)	39.95	29.95
Zork 1 (D)	39.95	29.95
Zork 2 (D)	39.95	29.95
Zork 3 (D)	39.95	29.95
Deadline (D)	49.95	36.95
Starcross (D)	39.95	29.95
Suspended (New) (D)	49.95	36.95
Weather War II (T)	19.95	14.95
Medicine Man (T)	21.95	16.95
3D 64 Man (T)	19.95	14.95
Krazy Kong (T)	19.95	14.95
Quick Brown Fox (C)	65.00	49.95
Color Craft (T)	29.95	21.95
Color Craft (D)	34.95	25.95
Meteor Madness (New) (T)	24.95	18.95
Meteor Madness (New) (D)	29.95	21.95
Fast Eddie (D)	34.95	26.95
Turmoil (D)	34.95	26.95
Squish 'Um (New) (D)	34.95	26.95
Type Attack (D)	39.95	29.95
Robbers of the Lost Tomb (D/T)	24.95	18.95
Wall Street (D/T)	24.95	18.95
Money Manager (D/T)	24.95	18.95
Data Manager (D/T)	24.95	18.95
Adventure Pack 1 (T)	19.95	14.95
Adventure Pack 2 (T)	19.95	14.95
Grave Robbers (T)	19.95	14.95
Trek-64 (T)	15.95	12.95
Annihilator (T)	21.95	15.95
Kongo Kong (T)	24.95	18.95
Word Pro 3+ (GREAT!!!) (D)	89.95	71.95

FOR FAST DELIVERY, send certified checks, money orders, or use your Master Charge or Visa and call **1-800-343-8019.** From inside New Hampshire call **1-603-542-6175.** Personal or company checks require two to three weeks to clear. All prices are subject to change without notice. Please include $2.00 per package for postage and handling.

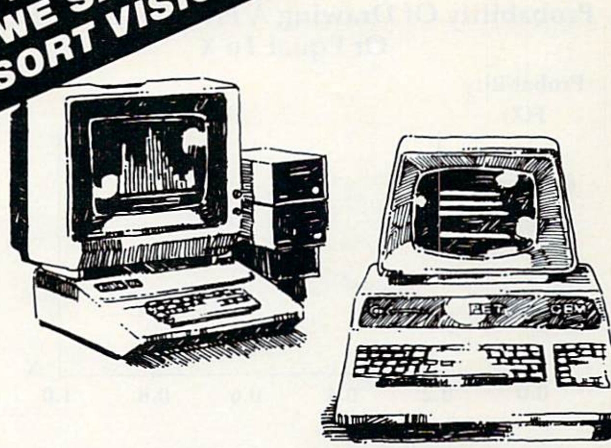
CALL NOW 1-800 343-8019

UNIVERSAL SOFTWARE
The Best Software for Less
P. O. Box 955
Claremont, NH 03743

OPEN MON THRU SAT 8:00 AM to 10:00 PM E.D.T.

We also carry Apple, Atari & Timex-Sinclair Software—ask for our list.

YES! WE SAID SORT VISICALC



VERSACALC™

Everything you always wanted to do with Visicalc, (but thought you couldn't).

If you use Visicalc™ but you are bumping into its limitations, then you need Versacalc™! Now it is possible for people untrained in Visicalc to perform weekly updating without constant instruction. Versacalc runs within Viscalc but uses no extra memory; in fact, it effectively increases memory by letting you call in modules from disk as needed.

Now You Can:

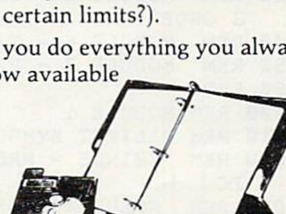
- ☐ SORT a Visicalc screen on any column, ascending or descending; all related formulas and labels are sorted too!
- ☐ Put the entire disk CATALOG on the screen at once!
- ☐ Easily do Year-To-Date accumulations!
- ☐ "Pound" formulas to expose formulas in place on the screen![1]
- ☐ Append two Visicalc files![1]
- ☐ 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!

A Tutorial section makes clear such features as @LOOKUP, DIF, @NA, and @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?).

Versacalc, software to help you do everything you always wanted to do with Visicalc is now available for the following hardware:

Apple II $100 IBM PC[1] $150
PET &CBM*[1] $125 *Specify DOS
Plus shipping. MC and Visa accepted.



Anthro-Digital

103 Bartlett Avenue, Pittsfield, MA 01201
Telephone (413) 448-8278, Telex 467622

The Company That Interfaces People and Computers

Apple II is a trademark of Apple Computer, Inc.
Versacalc is a trademark of Versacalc Enterprises, Inc.
Visicalc is a trademark of Visicorp, Inc.

See your dealer or contact us.

Also available from Anthro-Digital: Visicalc Formatting Aids, The Executive Secretary, The Personal Secretary, and Executive Speller, Hebrew II and Hebrew II Plus, S-C Macro Assembler, QuickTrace, Amper-Magic, The Rental Manager, F.A.R.M. accounting packages.

Figure 2.

Probability Of Drawing A Fraction Less Than Or Equal To X

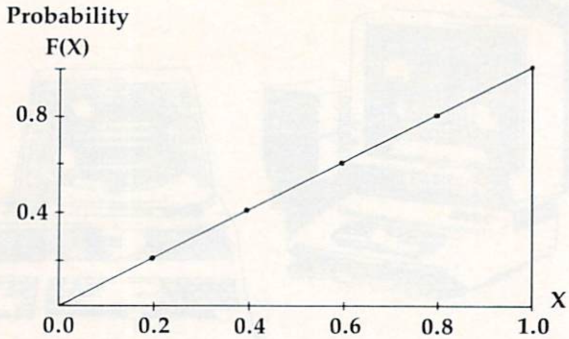
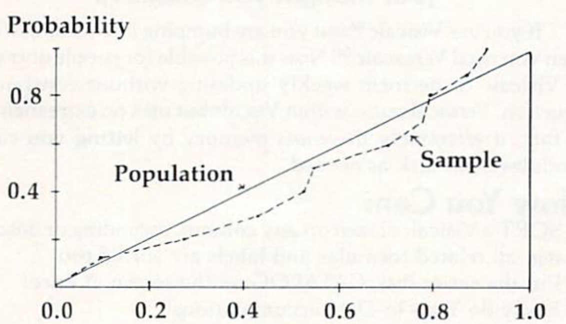


Figure 3.

Comparison Of Theoretical And Sample Cumulative Relative Frequency Distributions



Color Computer Version.

```
4Ø REM SALIENT SYMBOLS AND ARRAYS
5Ø REM  MODULE 1
6Ø REM FIRST LEVEL SUBROUTINES
7Ø REM  MODULE 2 - INITIALIZE
8Ø GOSUB 2ØØØ
9Ø REM  MODULE 3 - PERFORM TEST
1ØØ GOSUB 3ØØØ
11Ø REM SECOND LEVEL SUBROUTINES
12Ø REM  MODULE 4 - GENERATE A SEQUENCE OF FRA
    CTIONS
13Ø REM  MODULE 5 - SORT FRACTIONS IN ASCENDIN
    G ORDER
14Ø REM  MODULE 6 - TALLY TEST STATISTICS
15Ø REM  MODULE 7 - PRINT RESULTS
16Ø END
1ØØØ REM MODULE 1
1Ø1Ø REM SALIENT SYMBOLS
1Ø2Ø REM  KMINUS = PROFESSOR KNUTH'S K- STATIST
    IC
1Ø3Ø REM  KPLUS = PROFESSOR KNUTH'S K+ STATISTI
    C
1Ø4Ø REM  N = NUMBER OF FRACTIONS IN A SUBSEQUE
    NCE
1Ø5Ø REM  T = TOTAL NUMBER OF FRACTIONS
1Ø6Ø REM ARRAYS
1Ø7Ø REM  U = VECTOR OF VALUES FROM A UNIFORM D
    ISTRIBUTION
2ØØØ REM MODULE 2
2Ø1Ø REM  TOTAL NUMBER OF FRACTIONS GENERATED &
     NUMBER IN EACH SUBSEQUENCE
2Ø2Ø DATA 5ØØØØ,1ØØØ
2Ø3Ø READ T,N
2Ø4Ø DIM U(N)
2Ø5Ø REM HEADING
2Ø6Ø CLS
2Ø7Ø PRINT"THIS PROGRAM PERFORMS THE KOLMOGOROV
    -SMIRNOV (KS) TEST OF"
2Ø8Ø PRINT"RANDOMNESS ON A SEQUENCE OF FRACTION
    S FROM A UNIFORM"
2Ø9Ø PRINT"DISTRIBUTION BETWEEN Ø AND 1."
21ØØ PRINT
211Ø PRINT"THIS IS DONE BY APPLYING THE KS TEST
     TO SUBSEQUENCES"
212Ø PRINT"OF THE TOTAL SEQUENCE:"
213Ø PRINT
214Ø PRINT" TOTAL NUMBER OF FRACTIONS GENERATED
    = ";T
215Ø PRINT"NUMBER IN EACH SUBSEQUENCE = ";N
216Ø PRINT
217Ø PRINT"CHANGE THE ELEMENTS IN THE DATA STAT
    EMENT OF LINE 2Ø2Ø"
218Ø PRINT"FOR DIFFERENT VALUES."
219Ø PRINT
22ØØ PRINT"HIT 'ENTER' TO PROCEED":INPUT Z$
221Ø RETURN
3ØØØ REM MODULE 3
3Ø1Ø CLS
3Ø2Ø BK$ = "
    "
3Ø3Ø PRINT TAB(2Ø)"KOLMOGOROV-SMIRNOV TEST"
3Ø4Ø FOR I=1 TO T STEP N
3Ø5Ø REM  PRINT SUBSEQUENCE
3Ø6Ø PRINT @64,BK$
3Ø7Ø PRINT @64,"FRACTIONS :";I;" TO ";I+N-1
3Ø8Ø REM  GENERATE SEQUENCE OF FRACTIONS
3Ø9Ø PRINT @192,"** GENERATING FRACTIONS ...
    "
31ØØ GOSUB 4ØØØ
311Ø REM  SORT FRACTIONS
312Ø PRINT @192,"** SORTING FRACTIONS ...
    "
313Ø GOSUB 5ØØØ
314Ø REM  TALLY KS STATISTICS
315Ø PRINT @192,"** TALLYING TEST STATISTICS ..
    ."
316Ø GOSUB 6ØØØ
317Ø REM  PRINT RESULTS
318Ø GOSUB 7ØØØ
319Ø NEXT I
32ØØ RETURN
4ØØØ REM MODULE 4
4Ø1Ø FOR J = 1 TO N
4Ø2Ø U(J) = RND(Ø)
4Ø3Ø NEXT J
4Ø4Ø RETURN
5ØØØ REM MODULE 5
5Ø1Ø REM  SUBSTITUTE "QUICK SORT" HERE FOR FAST
    ER PROGRAM EXECUTION
5Ø2Ø FOR J=1 TO N-1
5Ø3Ø FOR L=1 TO N-J
5Ø4Ø IF U(L+1)<U(L) THEN HOLD=U(L+1):U(L+1)=U(L
    ):U(L)=HOLD
5Ø5Ø NEXT L,J
5Ø6Ø RETURN
6ØØØ REM MODULE 6
6Ø1Ø REM  PROFESSOR KNUTH'S K+ AND K- STATISTICS
6Ø2Ø KPLUS=Ø
6Ø3Ø KMINUS=Ø
6Ø4Ø FOR J=1 TO N
6Ø5Ø QPLUS=J/N - U(J)
6Ø6Ø QMINUS=U(J) - (J-1)/N
6Ø7Ø IF QPLUS>KPLUS THEN KPLUS=QPLUS
6Ø8Ø IF QMINUS>KMINUS THEN KMINUS=QMINUS
6Ø9Ø NEXT J
61ØØ REM  APPLY PROFESSOR KNUTH'S MULTIPLICATIV
    E TERM
611Ø KPLUS=SQR(N)*KPLUS
612Ø KMINUS=SQR(N)*KMINUS
613Ø RETURN
7ØØØ REM MODULE 7
7Ø1Ø PRINT @32Ø,BK$
7Ø2Ø PRINT @384,BK$
7Ø3Ø PRINT @32Ø,"K+ = ";KPLUS
7Ø4Ø PRINT @384,"K- = ";KMINUS
7Ø5Ø RETURN
```

Commodore Version.

```
40 REM SALIENT SYMBOLS AND ARRAYS
50 REM  MODULE 1
60 REM FIRST LEVEL SUBROUTINES
70 REM  MODULE 2 - INITIALIZE
80 GOSUB 2000
90 REM  MODULE 3 - PERFORM TEST
100 GOSUB 3000
110 REM SECOND LEVEL SUBROUTINES
120 REM  MODULE 4 - GENERATE A SEQUENCE OF FRA
   CTIONS
130 REM  MODULE 5 - SORT FRACTIONS IN ASCENDIN
   G ORDER
140 REM  MODULE 6 - TALLY TEST STATISTICS
150 REM  MODULE 7 - PRINT RESULTS
160 END
1000 REM MODULE 1
1010 REM SALIENT SYMBOLS
1020 REM  KMINUS = PROFESSOR KNUTH'S K- STATIST
   IC
1030 REM  KPLUS = PROFESSOR KNUTH'S K+ STATISTI
   C
1040 REM  N = NUMBER OF FRACTIONS IN A SUBSEQUE
   NCE
1050 REM  T = TOTAL NUMBER OF FRACTIONS
1060 REM ARRAYS
1070 REM  U = VECTOR OF VALUES FROM A UNIFORM D
   ISTRIBUTION
2000 REM MODULE 2
2010 REM  TOTAL NUMBER OF FRACTIONS GENERATED &
    NUMBER IN EACH SUBSEQUENCE
2020 DATA 50000,1000
2030 READ T,N
2040 DIM U(N)
2050 REM HEADING
2060 PRINT"{CLEAR}"
2070 PRINT"THIS PROGRAM PERFORMS THE KOLMOGOROV
   -"
2080 PRINT"SMIRNOV (KS) TEST OF RANDOMNESS ON A
   "
2090 PRINT"SEQUENCE OF FRACTIONS FROM A UNIFORM
   "
2100 PRINT"DISTRIBUTION BETWEEN 0 AND 1.":PRINT
2110 PRINT"THIS IS DONE BY APPLYING THE KS TEST
    TO"
2120 PRINT"SUBSEQUENCES OF THE TOTAL SEQUENCE:"
2130 PRINT
2140 PRINT"TOTAL NUMBER OF FRACTIONS = ";T
2150 PRINT"NUMBER IN EACH SUBSEQUENCE = ";N
2160 PRINT
2170 PRINT"CHANGE THE ELEMENTS IN THE DATA STAT
   E-"
2180 PRINT"MENTS OF LINE 2020 FOR DIFFERENT VAL
   UES."
2190 PRINT
2200 PRINT"HIT 'RETURN' TO PROCEED"
2210 GET Z$:IF Z$<>CHR$(13) THEN 2210
2220 RETURN
3000 REM MODULE 3
3010 PRINT"{CLEAR}"
3020 BK$ = "
         "
3030 PRINT TAB(8)"KOLMOGOROV-SMIRNOV TEST"
3040 FOR I=1 TO T STEP N
3050 REM  PRINT SUBSEQUENCE
3060 PRINT"{04 DOWN}";BK$
3070 PRINT"{02 UP}FRACTIONS :";I;"TO";I+N-1
3080 REM  GENERATE SEQUENCE OF FRACTIONS
3090 PRINT"{04 DOWN}** GENERATING FRACTIONS ...
   "
3100 GOSUB 4000
3110 REM  SORT FRACTIONS
3120 PRINT"{UP}** SORTING FRACTIONS ...  "
3130 GOSUB 5000
3140 REM  TALLY KS STATISTICS
3150 PRINT"{UP}** TALLYING TEST STATISTICS ..."
3160 GOSUB 6000
3170 REM  PRINT RESULTS
3180 GOSUB 7000
3190 PRINT"{HOME}{DOWN}"
3200 NEXT I
3210 RETURN
4000 REM MODULE 4
4010 FOR J = 1 TO N
4020 U(J) = RND(0)
4030 NEXT J
4040 RETURN
5000 REM MODULE 2
5010 REM  SUBSTITUTE "QUICK SORT" HERE FOR FAST
   ER PROGRAM EXECUTION
5020 FOR J=1 TO N-1
5030 FOR L=1 TO N-J
5040 IF U(L+1)<U(L) THEN HOLD=U(L+1):U(L+1)=U(L
   ):U(L)=HOLD
5050 NEXT L,J
5060 RETURN
6000 REM MODULE 6
6010 REM  PROFESSOR KNUTH'S K+ AND K- STATISTIC
   S
6020 KPLUS=0
6030 KMINUS=0
6040 FOR J=1 TO N
6050 QPLUS=J/N - U(J)
6060 QMINUS=U(J) - (J-1)/N
6070 IF QPLUS>KPLUS THEN KPLUS=QPLUS
6080 IF QMINUS>KMINUS THEN KMINUS=QMINUS
6090 NEXT J
6100 REM  APPLY PROFESSOR KNUTH'S MULTIPLICATIV
   E TERM
6110 KPLUS=SQR(N)*KPLUS
6120 KMINUS=SQR(N)*KMINUS
6130 RETURN
7000 REM MODULE 7
7010 PRINT"{04 DOWN}";BK$
7020 PRINT"{02 UP}K+ = ";KPLUS
7030 PRINT"{DOWN}";B$
7040 PRINT"{02 UP}K- = ";KMINUS
7050 RETURN
```

©

How The VIC/64 Serial Bus Works

Jim Butterfield, Associate Editor

The Serial bus connects VIC or Commodore 64 to its major peripherals, especially disk and tape. The workings of this interface have been a source of bafflement to most of us. We know that it's somehow related to the IEEE-488 bus which is used on PET and CBM computers. But it has fewer wires, and it's slower. For anyone interested in interfacing details, this article will clear up the mystery.

Ground Rules

To understand the workings of this bus, you must work through a few concepts. Later, we'll get technical for those who want it.

The bus, like the IEEE, has two modes of operation: Select mode, in which the computer calls all devices and asks for a specific device to remain connected after the call ("Jones, would you stay in my office after the meeting?"); and Data mode, in which actual information is transmitted ("Jones, I've decided to give you a raise"). Select mode is invoked by the use of a special control line called "Attention," or ATN.

By using Select mode, you can call in any device you choose, but you may need to do more before you transmit data. You might have several disk files in progress – writing some and reading others – and when you select the disk, device 8, you'll still need to specify which "part" of the disk you want to reach: subchannel 3, subchannel 15, or whatever. To do this, we use a "secondary address" which usually signals a subsystem within a specific device. That goes in as part of the command during Select mode. Finally, we may need to send other control information: the name of the file we wish to open, for example. That's not data; it's device setup information, so we also send it in Select mode.

But the main part is: you select a device, and then you send to it or receive from it. Finally, you shut it off. All devices are connected, but only the one you have selected will listen or talk.

Some Technical Ground Rules

If you're not into volts and signals and things, the rest of this article may not do much for you. I want to talk about technical aspects of the bus.

First, all the data flows over two wires; they are called the Clock line and the Data line. There are other wires used for control purposes, but the data uses only the two main ones.

All wires connect to all devices. The wires don't go "one way"; any device can put a ground on a signal line, and all other devices will see it. Indeed, that's the secret of how it works: each wire serves as a common signal bus.

When no device puts a ground on a signal line, the voltage rises to almost five volts. We call this the "false" logic condition of the wire. If any device grounds the line, the voltage drops to zero; we call this the "true" condition of the line. Note that if two devices signal "true" on a line (by grounding it), the effect is exactly the same as if only one has done so: the voltage is zero and that's that. We can summarize this as an important set of logic rules:

– A line will become "true" if one or more devices signal true;

– A line will become "false" only if all devices signal false.

Remember that we have several lines, but the important ones for information transmission are the Clock line and the Data line. Let's watch them work.

Transmission: Step Zero

Let's look at the sequence when a character is about to be transmitted. At this time, both the Clock line and the Data line are being held down to the true state. With a test instrument, you can't tell who's doing it, but I'll tell you: the talker is holding the Clock line true, and the listener is holding the Data line true. There could be more than one listener, in which case all of the listeners are holding the Data line true. Each of the signals might be viewed as saying, "I'm here!"

Step 1: Ready To Send

Sooner or later, the talker will want to talk, and

COMPU SENSE!

C-64	**Compute**	$399.00
VIC-20®	**Personal Computer**	147.00
VIC-1515	**Printer**	334.95
VIC-1530	**Datasette**	67.50
VIC-1541	**Disk Drive**	347.00
VIC-1010	**Expansion Module**	139.95
VIC-1311	**Joystick**	9.95
VIC-1312	**Game Paddles**	19.95
	Telephone Modem	99.95
VIC-1210	**VIC 3K Memory Expander Cart.** Plugs directly into the VIC's expansion port. Expands to 8K RAM total.	34.95
VIC-1110	**VIC 8K Memory Expander Cart.** 8K RAM expansion cartridge plugs directly into the VIC.	52.50
CM102	**24K Memory Expander Cart.**	119.95
VIC-1011A	**RS232C Terminal Interface** Provides interface between the VIC-20 and RS232 telecommunications modems. Connects to VIC's user port.	39.95
	PETSPEED - Basic Compiler for Commodore. Compile any Pet Basic program. The only optimizing compiler. Programs compiled with Petspeed run up to 40 times faster. Petspeed code is unlistable and compiled programs cannot be tampered with. No security device required for compiled programs. Available NOW for the Commodore 64.	140.00
	Star Gemini 10 Printer	360.00
	Star Gemini 15 Printer	450.00
	SND Monitor	347.00

CS1	**QUICK BROWN FOX** The Word Processor of this decade! For the VIC-20 and C-64.	$55.00

COMMODORE SOFTWARE

VIC-1211A	**VIC-20 Super Expander** Everything Commodore could pack into one cartridge - 3K RAM memory expansion, high resolution graphics plotting, color, paint and sound commands. Graphic, text, multicolor and music modes. 1024x1024 dot screen plotting. All commands may be typed as new BASIC commands or accessed by hitting one of the VIC's special function keys. Includes tutorial instruction book. Excellent for all programming levels.	$55.00
VIC-1212	**Programmer's Aid Cartridge** More than 20 new BASIC commands help new and experienced programmers renumber, trace and edit BASIC programs. Trace any program line-by-line as it executes, pause to edit. Special KEY command lets programmers redefine function keys as BASIC commands, subroutines or new commands.	$45.99
VIC-1213	**VICMON Machine Language Monitor** Helps machine code programmers write fast, efficient 6502 assembly language programs. Includes one line assembler/disassembler.	$48.99

CARDCO

Atari Adapter - play your 2600 games on the VIC-20	$79.95
CARDBOARD 6 An expansion interface for the VIC-20. Allows expansion to 40K or accepts up to six games. May be daisy-chained for more versatility.	$87.50
CARDBOARD 3 Economy expansion interface for the Vic-20.	$35.95
CARD "?" CARD/PRINT Universal Centronics Parallel Printer Interface for the VIC-20 or CBM-64. Use an Epson MX-80 or OKIDATA or TANDY or just about any other.	$76.00
CARDETTE Use any standard cassette player/recorder with your VIC-20 or CBM-64.	$30.95
LIGHT PEN A light pen with six good programs to use with your VIC-20 or CBM-64.	$29.95
16K Memory Expander	$50.50

All CARDCO Products have a lifetime warranty.

BUSINESS USES FOR YOUR VIC-20®

SS	**Accounts Payable & Receivable**	$29.95
CW-107A	**Home Calculation Program Pack**	48.95
CPV-31	**Data Files** - your storage is unlimited	14.95
CPV-96	**Household Finance Package** - to keep records of all your household expenses	30.95
CPV-208	**Bar-Chart** - display your numerical data	8.95
CH	**Turtle Graphics** - learn programming	34.95
CH	**VIC Forth** - a powerful language for BASIC programming	49.95
CH	**HES MON** - a 6502 machine language monitor with a mini-assembler	34.95
CH	**HES Writer** - time-saving word process tool	34.95
CH	**Encoder** - keep your personal records away from prying eyes	34.95
CT-21	**Statistics Sadistics** - statistical analysis	14.95
CT-121	**Total Time Manager 2.0** - creates personal or business schedules	15.95
CT-124	**Totl Label** - a mailing list & label program	13.95
CT-125	**Totl Text BASIC**	15.95
CT-126	**Research Assistant** - keep track of data	17.50
CT-140	**Totl Text Enhanced**	29.95
CM-152	**Grafix Designer** - design graphic characters	12.95
CQ-5	**Minimon** - allows you to program, load, save, or execute machine language programs	13.95
CS	**Home Inventory** - lists your belongings	17.95
CS	**Check Minder** - keep your checkbook right	14.95
CS	**General Ledger** - a complete ledger	19.95

GAMES FOR YOUR VIC-20®

CH-G203	**Tank Wars**	$15.95
CH-G205	**Pinball**	13.45
CH-G206	**Simon** - It gets tougher as you get better. Great for kids of all ages.	13.45
CH-G207	**Fuel Pirates**	13.45
CH-G209	**Laser Blitz**	15.95
CH-G210	**Tank Trap**	15.95
CH-G211	**Concentration**	13.45
CH-G212	**Dam Bomber** - pilot the plane, avoid enemy	13.45
CH-C307	**Shamus** - search room after room for the shadow-eluding androids; 2 levels of intense arcade action	34.95
CH-C308	**Protector**	36.95
CPU-79	**Breakout**	7.95
CPU-85	**Hangman** - unbelievable graphics & sound	9.95
CPU-87	**Memory** - VIC challenges your memory	9.95
CPU-88	**Match** - hand & eye coordination	7.95
CPU-89	**Monks** - a devilish game of logic	7.95
CPU-108	**Bomber** - you must decide who you want to fly for, then pick a target & your experience level	9.95
CPU-109	**Amok** - the halls of Amok are populated by robots that obey one command - get the intruder!	20.95
CPU-153	**Tank vs. UFO** - the tank is moving back & forth along the base; shoot the UFO before it shoots you	9.95
CPU-194	**Snakman** - Pacman for the VIC	14.95
	Defender on Tri - you're the pilot of an experimental ship	17.95
	3-D Man - the popular arcade game, requires 3K	17.95
	Exterminator - a game full of bugs	20.95

We have over 400 programs for your VIC-20 and over 250 programs for your C-64!

Shipping & Handling Charges:
First two (2) items - $2.00 per item.
Three (3) or more items - $1.00 per item.
For orders over $100 total, surface shipping will be paid by CompuSense. Blue Label or special handling will be paid by the customer.
Additional $2.00 C.O.D. fee on all C.O.D. orders.
MasterCard and Visa accepted. Give card number and expiration date on order form.
Allow three (3) weeks for personal checks.

TO ORDER:
P.O. Box 18765
Wichita, KS 67218
(316) 263-1095

AMERICAN EXPRESS® MasterCard VISA

Prices subject to change.
VIC-20® is a registered trademark of Commodore

Write for FREE Catalog!

send a character. When it's ready to go, it releases the Clock line to false. This signal change might be translated as "I'm ready to send a character." The listener must detect this and respond, but it doesn't have to do so immediately.

The listener will respond to the talker's "ready to send" signal whenever it likes; it can wait a long time. If it's a printer chugging out a line of print, or a disk drive with a formatting job in progress, it might hold back for quite a while; there's no time limit.

Step 2: Ready For Data

When the listener is ready to listen, it releases the Data line to false. Suppose there is more than one listener. The Data line will go false only when all listeners have released it – in other words, when all listeners are ready to accept data.

What happens next is variable. Either the talker will pull the Clock line back to true in less than 200 microseconds – usually within 60 microseconds – or it will do nothing. The listener should be watching, and if 200 microseconds pass without the Clock line going to true, it has a special task to perform: note EOI.

Intermission: EOI

If the Ready for Data signal isn't acknowledged by the talker within 200 microseconds, the listener knows that the talker is trying to signal EOI. EOI, which formally stands for "End of Indicator," means "this character will be the last one." If it's a sequential disk file, don't ask for more: there will be no more. If it's a relative record, that's the end of the record. The character itself will still be coming, but the listener should note: here comes the last character.

So if the listener sees the 200 microsecond time-out, it must signal "OK, I noticed the EOI" back to the talker. It does this by pulling the Data line true for at least 60 microseconds, and then releasing it.

The talker will then revert to transmitting the character in the usual way; within 60 microseconds it will pull the Clock line true, and transmission will continue.

At this point, the Clock line is true whether or not we have gone through the EOI sequence; we're back to a common transmission sequence.

Step 3: Sending The Bits

The talker has eight bits to send. They will go out without handshake; in other words, the listener had better be there to catch them, since the talker won't wait to hear from the listener. At this point, the talker controls both lines, Clock and Data. At the beginning of the sequence, it is holding the Clock true, while the Data line is released to false. The Data line will change soon, since we'll send the data over it.

The eight bits will go out from the character one at a time, with the least significant bit going first. For example, if the character is the ASCII question mark, which is written in binary as 00011111, the ones will go out first, followed by the zeros.

Now, for each bit, we set the Data line true or false according to whether the bit is one or zero. As soon as that's set, the Clock line is released to false, signalling "data ready." The talker will typically have a bit in place and be signalling ready in 70 microseconds or less.

Once the talker has signalled "data ready," it will hold the two lines steady for at least 20 microseconds to allow the listener to read it. This timing needs to be increased to 60 microseconds if the Commodore 64 is listening, since the 64's video chip may interrupt the processor for 42 microseconds at a time, and without the extra wait the 64 might completely miss a bit.

The listener plays a passive role here; it sends nothing, and just watches. As soon as it sees the Clock line false, it grabs the bit from the Data line and puts it away. It then waits for the Clock line to go true, in order to prepare for the next bit.

When the talker figures the data has been held for a sufficient length of time, it pulls the Clock line true and releases the Data line to false. Then it starts to prepare the next bit.

Step 4: Frame Handshake

After the eighth bit has been sent, it's the listener's turn to acknowledge. At this moment, the Clock line is true and the Data line is false. The listener must acknowledge receiving the byte OK by pulling the Data line to true.

The talker is now watching the Data line. If the listener doesn't pull the Data line true within one millisecond – one thousand microseconds – it will know that something's wrong and may alarm appropriately.

Step 5: Start Over

We're finished, and back where we started. The talker is holding the Clock line true, and the listener is holding the Data line true. We're ready for step 1; we may send another character – unless EOI has happened.

If EOI was sent or received in this last transmission, both talker and listener "let go." After a suitable pause, the Clock and Data lines are released to false and transmission stops.

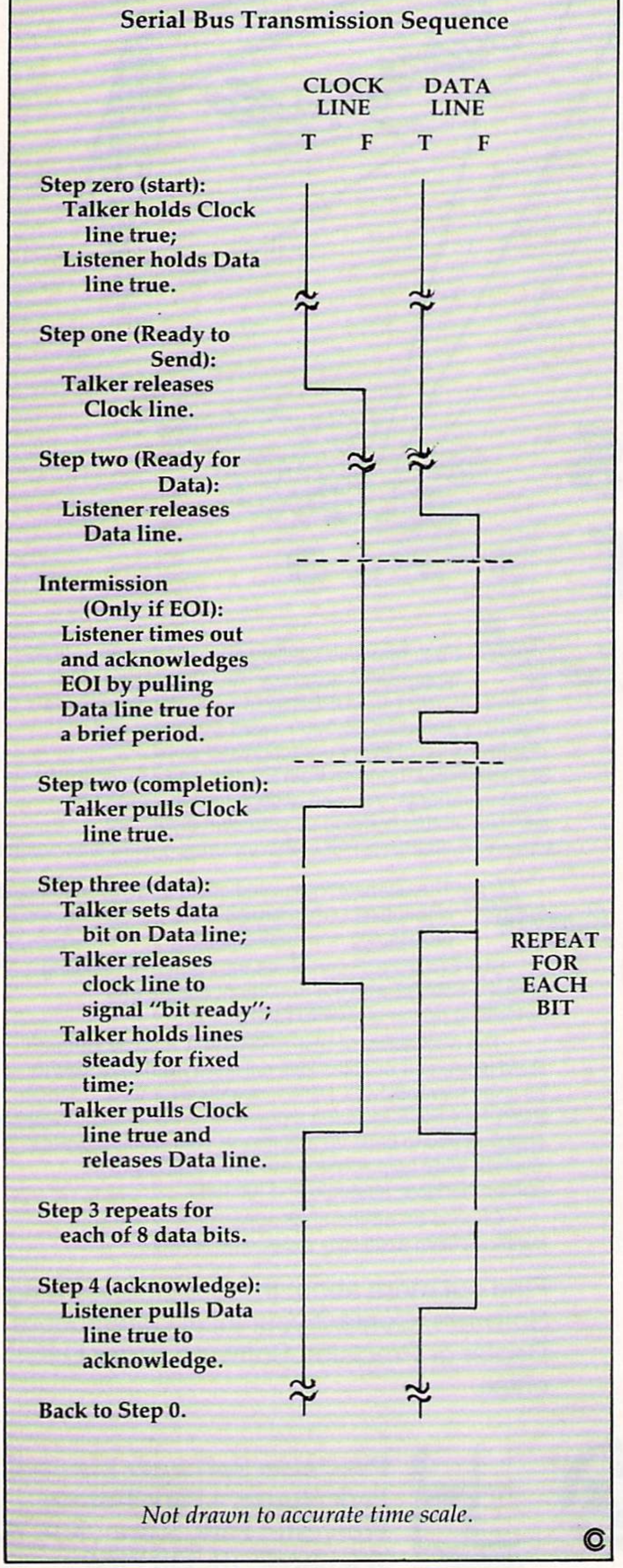
Attention!

This is all very well for a transmission that's under way, but how do we set up talker and listener? We use an extra line that overrides everything else, called the ATN, or Attention line.

Normally, the computer is the only device that will pull ATN true. When it does so, all other

devices drop what they are doing and become listeners.

Signals sent by the computer during an ATN period look like ordinary characters – eight bits with the usual handshake – but they are not data. They are "Talk," "Listen," "Untalk," and "Unlisten" commands telling a specific device that it will become (or cease to be) a talker or listener. The commands go to all devices, and all devices acknowledge them, but only the ones with the suitable device numbers will switch into talk and listen mode. These commands are sometimes followed by a secondary address, and after ATN is released, perhaps by a file name.



Not drawn to accurate time scale.

©

An example might help give an idea of the nature of the communications that take place. To open for writing a sequential disk file called "XX," the following sequence would be sent with ATN on: DEVICE-8-LISTEN; SECONDARY-ADDRESS-2-OPEN. When ATN switches off, the computer will be waiting as a talker, holding the Clock line true; and the disk will be the listener, holding the Data line true. That's good, because the computer has more to send, and it will transmit: X; X; comma; S; comma; W – the W will be accompanied with an EOI signal. Shortly thereafter, the computer will switch ATN back on and send DEVICE-8-UNLISTEN.

The file is now open; later, the computer will want to send data there. It will transmit, with ATN on, DEVICE-8-LISTEN; SECONDARY-ADDRESS-2-DATA. Then the computer releases the ATN line and sends its data; only the disk will receive the data, and the disk will know to put it onto the file called XX. The last character sent by the computer will also signal EOI.

After the computer has sent enough data for the moment, it will pull ATN on again and send DEVICE-8-UNLISTEN. Many bursts of data may go to the file; eventually, the computer will close the file by sending (with ATN on, of course) DEVICE-8-LISTEN; SECONDARY-ADDRESS-2-CLOSE.

ATN overrides everything in progress. A disk file might have lots of characters to give to the computer, but the computer wants only a little data. It accepts the characters it wants, then switches on ATN and commands the disk to Untalk. The disk has not sent EOI, but it will disconnect as commanded. Later, when it's asked to Talk again, it will send more characters.

ATN Sequences

When ATN is pulled true, everybody stops what they are doing. The processor will quickly pull the Clock line true (it's going to send soon), so it may be hard to notice that all other devices release the Clock line. At the same time, the processor releases the Data line to false, but all other devices are getting ready to listen and will each pull Data to true. They had better do this within one millisecond (1000 microseconds), since the processor is watching and may sound an alarm ("device not available") if it doesn't see this take place.

COMMODORE USERS

Join the largest, active Commodore users group. Benefit from:

— Access to hundreds of public domain programs on tape and disk for your **Commodore 64, VIC 20 and PET/CBM.**

— Informative monthly club magazine **THE TORPET.**

Send $1.00 for Program & Information Catalogue.
(Free with membership).

Membership Fees for 12 Months	Canada	—	$20 Can.
	U.S.A.	—	$20 U.S.
	Overseas	—	$30 U.S.

Toronto Pet Users Group
Department "S"
1912A Avenue Road, Suite 1
Toronto, Ontario, Canada M5M 4A1

* LET US KNOW WHICH MACHINE YOU USE *

VIC* OWNERS

VIK KONG – Barrels of fun. 6 Konglands, elevators, trapdoors. Machine code makes this an exciting game for everyone. – **$14.95**

LAUNCH 2031 AD – Command your ship through an asteroid belt to reach the other side. Machine language makes this a fast game. – **$14.95**

GRAND PRIX – Extensive use of hi-res graphics. Don't collide with the other cars or your gone!!. – **$14.95** –100% Machine code

CATERPILLAR – It's a battle between you and the spiders and caterpillars. 100% Machine code adds to the excitment of this great game. – **$14.95**

ALIEN HUNT – Try to break through the moving blocks to hit the ALIEN! – **$12.95**

*After a Word Processor? Inquire about ours for the Vic-Pet-64!

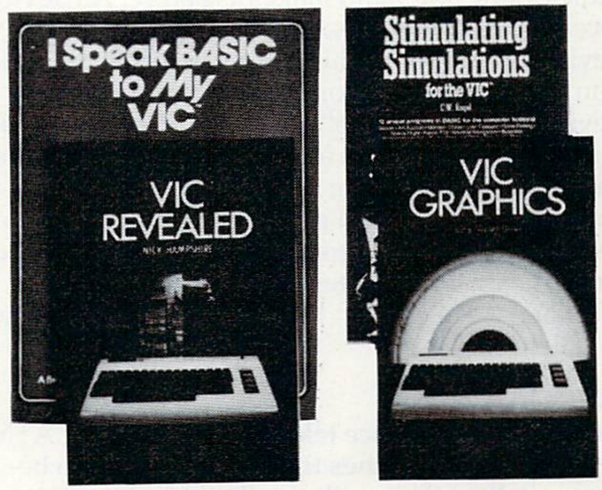
CANADIAN OR US FUNDS!!
Send cheque or M.O., M.C., Visa to:
TELEGAMES SOFTWARE
P.O. Box 152
Hampton, Ontario, Canada L0B 1J0

Write For Catalog .25

For Shipping Add $1.00
(416) 263-8064

*Trademark of Commodore

B. Dalton **The Source of Computer Books**



I SPEAK BASIC TO MY VIC™
Student Text
Aubrey B. Jones, Jr., $8.45 paperback

A field-tested computer literacy course that introduces students to BASIC language programming. Step-by-step instruction, exercises, practices.

VIC™ REVEALED
Nick Hampshire, $12.95 paperback

A comprehensive look at the VIC-20's hardware capabilities. 6502 Microprocessor, VIC System software, 6561 Video Interface Chip, 6522 Via & User Port, VIC I/O Functions.

STIMULATING SIMULATIONS FOR THE VIC™
C. W. Engel, $6.50 paperback

Twelve unique simulation programs in BASIC for the computer hobbyist — Soccer, Monster Chase, Space Flight, Business Management, and more.

VIC™ GRAPHICS
Nick Hampshire, $12.95 paperback

Thirty-eight complete graphics programs for the VIC-20. Theory of high resolution graphics, use of the multicolor mode, applications.

B. Dalton stocks a complete selection of computer books in 700 stores nationwide. Check your Yellow Pages.

People who know books know B. Dalton.

Name ______
Address ______
City ______ **State** ______ **Zip** ______
Phone ______
☐**Check/Money Order Enclosed**
☐**VISA** ☐**MasterCard** ☐**American Express**
Card No. ______ **Exp. Date** ______
Book Title ______ **Qty.** ______

Please send $1.75 for postage and handling.
Please add appropriate sales tax.

B. Dalton Bookseller
9340 James Ave. S.
Minneapolis, MN 55431

Under normal circumstances, transmission now takes place as previously described. The computer is sending commands rather than data, but the characters are exchanged with exactly the same timing and handshakes as before. All devices receive the commands, but only the specified device acts upon it. This results in a curious situation: you can send a command to a nonexistent device (try "OPEN 6,6") – and the computer will not know that there is a problem, since it receives valid handshakes from the other devices. The computer will notice a problem when you try to send or receive data from the nonexistent device, since the unselected devices will have dropped off when ATN ceased, leaving you with nobody to talk to.

Turnaround

An unusual sequence takes place following ATN if the computer wishes the remote device to become a talker. This will usually take place only after a Talk command has been sent. Immediately after ATN is released, the selected device will be behaving like a listener. After all, it's been listening during the ATN cycle, and the computer has been a talker. At this instant, we have "wrong way" logic; the device is holding down the Data line, and the computer is holding the Clock line. We must turn this around.

Here's the sequence: the computer quickly realizes what's going on, and pulls the Data line to true (it's already there), as well as releasing the Clock line to false. The device waits for this: when it sees the Clock line go true, it releases the Data line (which stays true anyway since the computer is now holding it down) and then pulls down the Clock line.

We're now in our starting position, with the talker (that's the device) holding the Clock true, and the listener (the computer) holding the Data line true. The computer watches for this state; only when it has gone through the cycle correctly will it be ready to receive data. And data will be signalled, of course, with the usual sequence: the talker releases the Clock line to signal that it's ready to send.

The logic sequences make sense. They are hard to watch with a voltmeter or oscilloscope since you can't tell which device is pulling the line down to true.

The principles involved are very similar to those on the PET/CBM IEEE-488 bus – the same Talk and Listen commands go out, with secondary addresses and similar features. There are fewer "handshake" lines than on IEEE, and the speed is slower; but the principle is the same.

INSIGHT: Atari

Bill Wilkinson

A mini-series on relocatable machine language begins in this month's column, plus a tip on a new product – an intelligent cable. Next month, the last part of the BAIT interpreter and more on relocatable machine language.

I have been working on a new project for **COMPUTE! Books**. By the time you read this, *COMPUTE!'s Atari BASIC Sourcebook* should be wending its way to your dealers' shelves and into your hands. Like *Inside Atari DOS*, the *Sourcebook* is a complete source listing of – what else? – Atari BASIC, along with a comprehensive explanation of how and why it all works.

Enough advertising. This month we will begin a mini-series on self-relocatable machine language. But before we begin all that, time out for some ruminations.

Machine Language Be Not Hard

Before we start investigating self-relocatable machine code on the 6502, I'd like to get up on my soapbox for a while and do a little preaching.

This month's sermon was inspired by a machine language program published in another magazine. The program seemed to me the epitome of poor programming techniques. And lest it seem that I am taking a cheap shot, let me hasten to add that the program works and works well. I am carping about the printed form of the program, not the results thereof.

In the tradition of any good preacher, then, let me give you some suggestions on how to write good, readable, maintainable machine language:

1. Always use plenty of comments (they cost nothing in the assembled code, unlike BASIC).

2. Never use absolute addresses (except in equates).

3. Never use absolute numeric constants (again, except in equates, though we might forgive an occasional constant 0 or 1).

4. Always use plenty of comments.

5. Always use long, meaningful names for labels. (Which makes more sense, ICCOM or IOCB.COMMAND ?).

6. Never branch to a location relative to the location counter (that is, never use "*+xx" or "*-xx").

7. Never use a comment that simply echoes the machine language code.

8. Always use plenty of comments.

9. Never change the location counter needlessly (that is, most programs should contain only one "*=", except for the use of "*=*+xx" to reserve space).

10. If possible, always define a label before its first use.

11. Always thoroughly document the entry and exit values for a subroutine, taking special care to note what happens to the CPU registers.

12. Always use plenty of comments.

Those of you with some OSS software will see that I have taken a small pot shot at our own manuals in commandment 5. Well, I never said we were perfect. (Great, maybe, but not perfect.)

And those of you with Atari's Macro Assembler may object to using long labels since, even though AMAC allows long labels, it ignores all but the first six characters. Sorry, but I still think this rule should be followed. You just have to be more inventive to insure that labels are unique in the first six characters. (For example, IOCB.AUX1 and IOCB.AUX2 look the same to AMAC, so use IOCB.1AUX and IOCB.2AUX .)

Anyway, rather than go through each of those commandments one by one, let's look at an example subroutine coded with both worst and best techniques.

Example 1: Worst Technique

```
; EXAMPLE 1 : print A register
  *= $1F00
  LDX #11
  STX $342    ; put 11 in location $342
  LDX #0
  STX $348
  STX $349
  JMP $E456    ; go to $E456
```

Example 2: Best Technique

```
;
; Example 2: Output the character in the A-register
;            to file channel (IOCB) number zero
;            (assumed to be the screen).
;
```

```
; Entry: A-register contains the character
; Exit: Status of all registers unknown
;
  *= LOWMEMORY
PRINTCHARACTER
  LDX #COMMAND.PUTBINARY
  STX IOCB.COMMAND ; command for CIO
  LDX #0  ; use a zero buffer length
  STX IOCB.LOLENGTH  ;tells CIO to output
  STX IOCB.HILENGTH  ;contents of A register
; next line commented out...not needed since X
  already =0
;  LDX #0  ; specify IOCB zero
   JSR CIO  ; let CIO do the real work
; Could check for errors here
   RTS      ; all done
```

Enough said? I refuse to decipher programs like Example 1. Of course, Example 2 wouldn't be very useful either unless equates for the various labels were supplied (as in IOCB.COMMAND= $342), but at least most readers could understand its intent.

Absolutely Not

Regular readers will no doubt recall the many occasions on which I have ranted about staying out of Page 6 or about putting code at LOMEM or about writing code that is not specific to a particular hardware/software configuration. But, to be fair, sometimes it is hard to follow all of the rules, especially when adapting a program from a book or magazine.

Often, the real secret to writing adaptable code is in learning to write self-relocatable code. The techniques we will begin discussing this month are designed specifically for use with the 6502 microprocessor. While there will be several references to Atari internal structure, most of what is presented here is appropriate to Apple and Commodore machines as well.

And I will answer one more question before we start on the hard stuff: *Why* should we want to write self-relocatable code? Sorry, we don't have room for that answer this month. Wait until next month. (It's a good answer, honest!)

Actually, there is just one rule to remember in writing self-relocatable code: *avoid references to absolute memory locations.*

Unfortunately, this is often a very hard rule to follow. Fortunately, there are many places where we can make an exception to this rule.

For starters, look at the subroutine in Examples 1 and 2 above. Is it self-relocatable? Your first impulse might be to say *no,* since it references $342, $348, $349, and $E456, which are all absolute locations. And even if you do it right and use the equated labels of Example 2, they are still absolute, no matter what they look like.

But. Within the context of any given machine, there are always certain locations which *never* change. In particular, hardware locations, locations in ROM, and locations in the RAM (or values used and defined by ROM subroutines) cannot possibly change. An exception to this is when you plug in a new set of ROMs, and you can ask the software vendors about the fun and games the Atari 1200XL's new ROMs are giving them.

In the example given, $E456 (CIO) is in the Atari's OS ROM space. It is a guaranteed entry point to the OS command implementation code. It won't change (even in the new 1200, etc.).

And locations $340 through $34F (as well as $350 through $3BF) are in the IOCB space defined by Atari for use with CIO. Again, they won't and cannot change.

Finally, the command used (11) and the zero buffer length are values defined by the OS ROMs to have certain meanings. And if Atari changes these meanings, we are *all* in trouble, because Atari BASIC, PILOT, and more won't work then.

Implicit Relocatability

The result of all this? No matter where you assemble that example (that is, no matter where the "*=" places the code), the resultant machine object code will be precisely the same! Presto. That example is self-relocatable.

Surprisingly, a lot of the subroutines used with Atari BASIC follow the mold shown here: they simply set up some values in the Atari-specified memory locations and call an Atari-specified OS routine. They are implicitly self-relocatable.

So what is *not* relocatable? Generally, the prime culprits are:

1. References to RAM locations defined within the user's own code (for example, LDA, STA, INC, etc.).

2. Jumps (JMPs) to locations in the user's own code.

3. Calls (JSRs) to locations in the user's own code.

Let's make up an example just to illustrate potential problems.

```
      *= $600
SAVEX *= *+1
MESSAGE .BYTE 'This is the message',0
;
; this is the same code as the examples above
;
PRINTCHARACTER
      LDX #COMMAND.PUTBINARY
      STX IOCB.COMMAND   ; command for CIO
      LDX #0   ; use a zero buffer length
      STX IOCB.LOLENGTH   ; tells CIO to output
      STX IOCB.HILENGTH   ; contents of A register
      JMP CIO   ; let CIO do the real work
;
; call here to print contents of 'MESSAGE'
; Entry conditions: none
; Exit conditions: none, no registers saved
;
```


ATARI 8

```
PRINTMESSAGE
        LDX #0
        STX SAVEX    ; initialize message pointer
MSGLOOP
        LDX SAVEX    ; get current message pointer
        LDA MESSAGE,X    ; get next character of msg
        BEQ QUIT    ; but quit if it's last char
        JSR PRINTCHARACTER    ; else print it
        INC SAVEX    ; point to next character
        JMP MSGLOOP    ; and do another character
;
QUIT
        RTS          ; we are done
```

Do you see the problem areas? If we move this routine somewhere else in memory, the addresses of MESSAGE, PRINTCHARACTER, MSGLOOP, and SAVEX all change, and the object code associated with them changes also. This routine is definitely *not* self-relocatable.

But let's tackle each of the problem labels one at a time and see how we can change the references to each to make the code self-relocatable.

MSGLOOP is the easiest label to "fix." For example, if we change the line JMP MSGLOOP to BNE MSGLOOP, the label MSGLOOP is no longer a problem (since *all* branch instructions are always, by nature, self-relocatable).

And we could save the X-register on the stack (via TXA and PHA) and later retrieve and increment it similarly (via PLA, TAX, and INX), thus eliminating the need for SAVEX.

The PRINTCHARACTER routine could easily be eliminated in its entirety by placing its code in-line in the middle of the PRINTMESSAGE routine. This is a good solution only if PRINTCHARACTER is not called by any other routine. It may also be an adequate solution if the routine being placed in-line is fairly small (as is PRINTCHARACTER) so that you can keep two or more copies around, if necessary.

But what do we do about MESSAGE, which is too big to put in a register? Or what would we do if PRINTCHARACTER was a long routine? And, most importantly, what do we do with a hunk of self-relocatable code once we have managed to produce it?

Next month we'll tackle those questions and others.

A Handy Product?

Do you do much work on *both* Apple II and Atari computers? If so, you could probably use a handy-dandy little device which we recently acquired.

Allen Prowell of Fresno, California, built us what amounts to an intelligent cable between our Apple and our Atari. It plugs into the joystick port on the Atari and into the game port on the Apple. It transfers ASCII files in either direction (doing "light conversion" on return characters, etc.). *Very fast.* It is much more convenient and reliable than using RS-232, and it moves over 1000 characters a second, including disk accesses.

As I said, this is a specialized product, but if you need it, call Allen (209) 227-4917. Using our C/65 and MAC/65 on both Atari and Apple, we have converted an 8K program in as little as two hours, including the transfers, assemblies, etc.

Coming Attractions

I think next month's column will be fairly long, what with the last part of BAIT and Part 2 of self-relocatable machine language. If I have room, though, I will introduce you to a new Atari graphics mode. Also, coming soon, information on some strange and wonderful new products for the Atari. ©

Jim Butterfield, Associate Editor

Part III
Numeric Output

There's a quick method of generating decimal output on the 6502. It's a notable departure from conventional methods, and it would be worthwhile to lay down a few general ideas.

Shift Transfer

Suppose we have two bytes, OLD and NEW. OLD contains a value, and NEW contains zero. We want to transfer the contents of OLD to NEW, and set OLD to zero. That's not hard by conventional coding (a couple of Load and Store commands), but we're going to look at another method.

Suppose that we shift each bit out of OLD, and then shift it into NEW. Using a left shift, we would code: ASL OLD (arithmetic shift left), which puts the extra bit into the carry; and then ROL NEW, which slides the carry bit into the new byte. If we repeat this eight times, OLD will have moved to NEW, bit by bit. It seems like a slow way of doing it, but it does indeed achieve what we want.

The same method, of course, would move a two-byte OLD to NEW, or as many bytes as we need. Each bit shift would consist of one ASL followed by several ROL commands until the job is done.

A New Way To Rotate

The ROL (Rotate Left) command is compact and handy. It takes the contents of the Carry flag and moves it into the low-order bit position of the operand; all other bits move over to make room, and the high-order bit falls out into the Carry. Now let's do the same thing without using the ROL command.

The ROL command might be considered the same as multiplying by two plus adding a carry, if necessary. We often use the left Shift and Rotate commands for multiplication. But there's another way to multiply: we can use repeated addition.

We can do exactly the same as ROL NEW by coding: LDA NEW: ADC NEW: STA NEW. The original number is doubled, which gives the left shift, and the carry is automatically added in. A new carry condition is generated. All we seem to have done is use three instructions where one would have done.

The Gimmick

But here's the gimmick: we can make the ADC instruction add in a different manner by switching to *decimal* mode. In decimal mode, addition automatically produces BCD numbers. And BCD numbers can be printed as if they were hexadecimal, which greatly simplifies the output calculation.

Let's work this out in principle. First, a warning: on many machines, decimal mode is poisonous to the operating system and to the interrupt routines. Remember to restore binary mode when you're finished; and if your machine uses interrupt, lock it out for the duration of your calculation.

Let's look at simple coding to change a one-byte OLD to NEW:

```
LDA  #$00
STA  NEW       (clear NEW)
LDX  #$07      (eight bits)
ASL  OLD       (grab a bit)
LDA  NEW       (slip it..)
ADC  NEW       (..into..)
STA  NEW       (...NEW)
DEX            (count down)
BPL  BIT       (next bit)
     ...
```

If we are in binary mode, the above routine will copy OLD to NEW unchanged. But if we switch to decimal mode, OLD will be converted to BCD as it is moved to NEW.

A warning here: the result might not fit. A one-byte binary number might need to be converted to three decimal digits (for example, 250). In this case, we'd need to have two bytes available in NEW to hold the result, since BCD holds only two decimal digits per byte. Be sure your coding provides for sufficient space for the answer.

An Example

Let's write the outline of a routine to convert a series of 16-bit numbers to decimal and output

them. We'll write the code in compact form so as to emphasize the logic flow.

Set up Y to reach several numbers:

LDY #0

Copy a number into the work area:

A: LDA TABLE,Y:STA WORK:LDA TABLE+1,Y: STA WORK1

Move Y to reach the next number, clear output area:

INY:INY:LDA #0:STA OUT1:STA OUT2:STA OUT3: STA ZSUP

Get ready to move 16 bits from WORK to OUT:

LDX #15

Move bit out of WORK:

B: ASL WORK:ROL WORK1

Switch to decimal mode:

SEI:SED

Move bit (decimally) into OUT:

LDA OUT1:ADC OUT1:STA OUT1
LDA OUT2:ADC OUT2:STA OUT2
LDA OUT3:ADC OUT3:STA OUT3

Clear decimal mode:

CLD:CLI

Repeat for next bit:

DEX:BPL B

Prepare to output three bytes (six digits):

LDX #2

Get bytes, high order first, for output:

C: LDA OUT1,Y:PHA

Output high order digit:

PHA:LSR:LSR:LSR:LSR:JSR PUT:PLA

Output low order digit:

AND #$0F:JSR PUT

Go for next byte:

DEX:BPL C

Print RETURN:

LDA #$0D:JSR PRINT

Go back for another number:

CPY #10:BCC A

Quit:

RTS

Zero suppress output subroutine:

PUT: CMP ZSUP:BNE D

Fill with space:

LDA #$20:BNE E

Convert numeric, kill zero suppression:

D: ORA #$30:STA ZSUP

Print and return:

E: JMP $FFD2

Let's put the above into a PET/CBM/VIC/C64 environment to see it work:

```
100 DATA 160,0, 185,80,3, 141,64,3
110 DATA 185,81,3, 141,65,3, 200,200
120 DATA 169,0, 141,66,3, 141,67,3, 141,68,3
130 DATA 141,69,3, 162,15, 14,64,3, 46,6
    5,3, 120, 248
140 DATA 173,66,3, 109,66,3, 141,66,3, 1
    73,67,3
150 DATA 109,67,3, 141,67,3, 173,68,3
160 DATA 109,68,3, 141,68,3, 216, 88, 20
    2, 16,216
170 DATA 162,2, 189,66,3, 72, 74, 74, 74, 74
180 DATA 32,184,3, 104, 41,15, 32,184,3,
     202, 16,236, 169,13, 32,210,255
190 DATA 192,10, 144,155, 96, 205,69,3,
    208,4, 169,32
195 DATA 208,5, 9,48, 141,69,3, 76,210,255
200 FOR J=848 TO 968:READ X
210 T=T+X:POKE J,X
220 NEXT J
230 IF T<>10738 THEN STOP
300 SYS 848
```

The numbers that are printed won't have any special meaning, but you'll see that conversion is taking place, and that zero suppression works nicely.

Converting binary numbers to BCD in preparation for output isn't really a gimmick. It's a sensible way to do an otherwise difficult job. ©

C. Regena

Planning Color Sets

In a previous column we looked at defining characters for graphics. Let's expand on that idea and discuss in more detail how to plan the color sets for high-resolution graphics.

To define colors for your graphics, use the CALL COLOR statement. The form is CALL COLOR(s,f,b) where s is the set number, f is the foreground color, and b is the background color. Each of the numbers can be from 1 to 16. Each graphics character you define can have two colors (a foreground color and a background color) chosen from the list of 16 colors.

The Color Sets

There are 16 color sets. Each color set contains eight character numbers (ASCII codes). The table shows which ASCII character codes are in which color set. You may find it handy to mark off these sets on the "Character Codes" table on the BASIC Reference Card that came with your computer. Just make a mark after every eighth number, then number the sets so you can tell at a glance which character is in which set – and which other characters are in the same set.

Color Sets

Set	Character Codes	Set	Character Codes
1	32-39	9	96-103
2	40-47	10	104-111
3	48-55	11	112-119
4	56-63	12	120-127
5	64-71	13	128-135
6	72-79	14	136-143
7	80-87	15	144-151
8	88-95	16	152-159

Now try this short program to see how the CALL COLOR statement works:

```
100 PRINT"HELLO THERE!"
110 PRINT"THIS IS A SAMPLE."
120 CALL COLOR(5,7,1)
130 GOTO 130
```

RUN the program. Lines 100 and 110 just print some words on the screen. By the way, we didn't use a CALL CLEAR statement, so the program will also still be on the screen. The screen turns green when the program starts to run. Line 120 says to change all characters in set number 5 to a red foreground (color 7) and a transparent background (color 1). Line 130 holds the colors on the screen until you press FCTN 4 to CLEAR or stop the program (SHIFT C on the TI-99/4 console). You will notice when you RUN the program that the screen turns green, and then all the letters in Set 5 (@,A,B,C,D,E,F,G) turn red. Color 1 for the transparent background means that the background for the character will be the screen color.

Stop the program by pressing CLEAR. Change line 120 to

```
120 CALL COLOR(5,6,1)
```

The letters turn blue. Go ahead and try different colors for the second number in parentheses.

Now experiment with background color. Add these lines to your program:

```
130 FOR DELAY=1 TO 100
140 NEXT DELAY
150 CALL COLOR(6,7,16)
160 FOR DELAY=1 TO 100
170 NEXT DELAY
180 CALL COLOR(6,16,7)
190 GOTO 130
```

Lines 130-140 and 160-170 are delay loops. RUN the program. Line 120 changes the letters in Set 5 to whatever color you specified. Line 150 changes the letters in Set 6 (H,I,J,K,L,M,N,O) to a red (7) foreground and a white (16) background. Each character will look like a red letter on a white square. After the delay loop, line 180 changes the letters in Set 6 to a white foreground and a red background – now white letters on red squares. Line 190 branches to the delay loop in line 130, so the letters in Set 6 blink red on white then white on red.

Screen Changes

Notice that as soon as you use a CALL COLOR statement, *all* characters in that set change color – those already on the screen and any that you may later print or draw on the screen. Careful planning is necessary so you know exactly which characters you are defining to be certain colors.

If you would like to change the screen color, use CALL SCREEN(c), where c is a color number from 1 to 16. For example, add line 90 and run your program:

```
90 CALL SCREEN(12)
```

Keep in mind that anywhere you have used the color number 1, for transparent, it really means the screen color.

Now try another special effect. Add line 125:

```
125 CALL COLOR(1,2,8)
```

This changes all characters in Set 1 to black on cyan (instead of black on transparent). RUN the program. The "space" is Character 32 in Set 1, and all spaces have been turned to cyan. The screen is light yellow from line 90, so you get a border around a cyan rectangle with various colors of letters from the rest of the program.

The default value of all character sets is black on transparent, so the letters on the screen are black on the screen color of yellow. If you would like a complete cyan rectangle with black letters on the cyan background, the character sets would need to be changed to black on cyan.

Keep in mind that it does make a difference in your programming whether you print first then define the colors, or define the colors and then print. Plan your program so that the computer will perform the actions in exactly the order you want.

Here is another sample program. Type NEW (enter), and then try this program. Watch carefully.

```
100 CALL CLEAR
110 CALL VCHAR(10,5,42,9)
120 CALL VCHAR(10,10,42,9)
130 CALL HCHAR(14,6,42,4)
140 CALL VCHAR(10,17,42,9)
150 CALL VCHAR(10,24,33,6)
160 CALL VCHAR(18,24,33)
170 CALL COLOR(2,7,1)
180 GOTO 180
```

The computer is quite fast, but you can see that the screen clears, the characters are drawn in black, and then some of the characters turn red. If you prefer to have the asterisks printed in red from the start, the CALL COLOR statement must come before the CALL VCHAR and CALL HCHAR statements. Delete line 170 and add

```
105 CALL COLOR(2,7,1)
```

RUN the program and you can see the difference.

Invisible Characters

Another thing you can try is to draw your characters invisibly and then make them appear all at once. This is quite effective if you have a lot of CALL HCHAR and CALL VCHAR statements drawing an intricate picture. For this program, make the following changes:

```
105 CALL COLOR(2,1,1)
106 CALL COLOR(1,1,1)
170 CALL COLOR(2,7,1)
175 CALL COLOR(1,2,1)
```

First the characters in Sets 2 and 1 are made invisible by setting both foreground and background to transparent. Next the characters are drawn with CALL HCHAR and CALL VCHAR statements. You won't be able to see this process. Last, line 170 colors the asterisks red, and line 175 colors the exclamation points black so the greeting appears all at once.

When defining your own graphics characters, you may use any character number. If you want to keep the alphabet intact, you will probably use character numbers beyond 95. Group your characters so that all characters of the same color will be in the same set.

Remember that there are eight characters per set. If you are using many different colors or need to conserve memory, you will also need to plan the number of characters you can design in each set. For example, if you have a dog that uses nine characters, could you redraw him in eight characters so only one CALL COLOR statement would be needed?

Refer to the table to determine which characters are in which set. For example, if you are designing character number 134, it will be in Set 13, which contains characters 128-135. Your CALL COLOR statement will use set number 13.

If you are not using the small letters in character codes 97-122 (available on the TI-99/4A console, but not on the TI-99/4), use those numbers to define your graphics characters, then PRINT the characters rather than using HCHAR and VCHAR to draw them on the screen. PRINT TAB(10);"hikn" will be much faster than four separate CALL HCHAR statements to put up characters 104, 105, 107, and 110. By the way, your listing will say "hikn" with the small letters, but when your program is run those letters will be substituted by the graphics characters as you defined them. If you want to use the PRINT method on characters numbered higher than 126, you may use a statement such as PRINT CHR$(132)&CHR$(133)&CHR$(137).

Teeth Wisdom

The following program illustrates the use of color sets in an educational program. "Teeth Wisdom" draws the teeth and their names on the screen in high resolution graphics. After the user knows the names, he or she presses ENTER and the labels clear. The names will be reprinted in a random order. For a quiz, certain teeth will "blink" and the user must press the correct answer. The order will be random.

The teeth are drawn white on a light red background, and the gums are light red on a transparent background. Although all the teeth are white, they are defined in different color sets so that only certain teeth will blink during the quiz. The central incisors use characters 96-100; the lateral incisors, 104-107; the cuspids, 112-117; the bicuspids, 120-127; and the molars 128-134. The gums use characters from 136 to 157.

Since so many graphics characters are defined, DATA statements rather than individual CALL CHAR statements are used. The DATA in lines 240 to 330 are character definitions. Be careful to type these lines exactly as shown. The round symbols are zeros and not the letter O. When there are two or more commas in a row, it means that a character is defined as a null string. At the end of a data list such as line 250, the "" (double quotes) marks are necessary to indicate a null string, but in a series such as in line 260, the quote marks may be omitted between commas. These null strings correspond to unused character numbers.

Lines 180-230 let the character number C vary from 94 to 157 and READ in a string then define character C with graphics definition C$. The CALL COLOR statements blink the asterisks on the title screen while the characters are being defined. Lines 340-390 define the colors for the teeth and gums.

Lines 590-690 PRINT the graphics on the screen, which is faster than using individual CALL HCHAR or CALL VCHAR statements for this many special characters. Within the quotation marks are the lowercase letters – release the ALPHA LOCK key to type these symbols in. Line 610 uses the symbol found on the face of the "C" key and is typed by pressing FCTN and C. Other symbols requiring the FCTN key are in lines 640 and 650.

For The TI-99/4 Console

If you have the TI-99/4 console, you will not be able to type in these lines. You can use the method found in line 600 to print the characters, listing each character number. *Note:* If a program like this has been typed in on the TI-99/4A console, it *will* work correctly on the TI-99/4 console (read it in from cassette or diskette).

In the quiz, lines 900 and 910 blink the particular teeth while the computer waits for a response. A random number (I) is chosen, and the corresponding color set is I+8 for the CALL COLOR statements.

Program Structure

Lines	
100	Title.
110-170	Clear screen; print title screen.
180-230	Define graphics characters 94 through 157 by READing the definitions from DATA; blink asterisks on screen green and white.
240-330	DATA containing graphics definitions.
340-360	Define color sets 9 through 13 as white on light red for teeth.
370-390	Define color sets for light red on transparent for graphics surrounding teeth.
400-510	Clear screen; print instructions; define strings as groups of characters for later printing.
520-560	READ in names of five groups of teeth as N$ array and set the W$ array elements equal to the N$ array elements.
570	Prints message to press ENTER and waits for response.
580-690	Clear screen; print teeth with labels.
700	Prints message to press ENTER and waits for response.
710-760	Clear message and clear labels.
770	Prints quiz title.
780-850	Randomly print names of teeth on screen from the W$ array of five names. A(I) will be the correct corresponding answer.
860-1060	Perform quiz.
870-880	Randomly choose teeth.
890-920	Blink teeth blue and white while waiting for response.
930-940	If number 1-5 is pressed, show which number was pressed, otherwise return to line 890.
950-990	If answer is incorrect, sound "uh-oh" and return for another response.
1000-1030	If answer is correct, play arpeggio.
1040	Clears answer chosen.
1050-1060	Set A element to zero so that tooth will not be chosen again; return to next problem.
1070-1100	Print option to try again; wait for response; branch appropriately.
1110-1140	If user wants to try again, set W$ array elements equal to names of teeth, branch to beginning of exercise.
1150	Stop.
1160-1190	Subroutine to print "PRESS <ENTER>" and wait for response.
1200-1210	Clear screen and END.

Teeth Wisdom

```
100 REM  TEETH FOR TI
110 CALL CLEAR
120 PRINT TAB(4);"*****************
    **"
130 PRINT TAB(4);"*";TAB(22);"*"
140 PRINT TAB(4);"*  TEETH  WISDOM
     *"
150 PRINT TAB(4);"*";TAB(22);"*"
160 PRINT TAB(4);"*****************
    **":::::::
170 PRINT :::
180 FOR C=94 TO 157
190 CALL COLOR(2,13,1)
200 READ C$
210 CALL CHAR(C,C$)
```

```
220 CALL COLOR(2,16,1)
230 NEXT C
240 DATA 0000001F1010101,000000FF
250 DATA 00000F1F3F7F7F7F,000083C7C
    7E7EFEF,0000E0F0F8FCFEFE,FFFFFF
    FFFFFFFFFF,EFEFEFEFEFEFEFEF,0,,""
260 DATA 70F8FCFCFEFEFEFF,FEF8C,0E1
    F3F3F7F7F7FFF,7F1F03,,,,,000000
    00000000F8,0101010101,FCFEFFFFF
    EFCF83
270 DATA 00000000000000001F,3F7FFFFF7
    F3F1F0C,8080808080,,,0F1F1F1F1
    F1F0E,E0F8F8F8F8F,071F1F1F1F1F0F
280 DATA F0F8F8F8F8F87,1F3F3F3F3F
    1F,F0F8F8F8F8F,0F1F1F1F1F0F
    ,F8FCFCFCFCF8,7FFFFFFFFFFFF7F
290 DATA E0F0F0F0F0E,070F0F0F0F
    07,FEFFFFFFFFFFFFE,0303030303030
    301,FFFFFFFFFFFFFFFF,C0C0C0C0C0
    C08,""
300 DATA FFFFFCF0E0C08,FFFF7E181,FF
    FF3F0F070301,FFFEFEFCFCF8F8F8,F
    F7F7F3F3F1F1F1F,F0F0F0F0F0F0F
310 DATA 0F0F0F0F0F0F0F,E0E0E0E0C
    0C0808,0707070703030101,0000000
    00071FFF,0000031FFFFFFFFF,00FFFF
    FFFFFFFFFFFF
320 DATA 0000C0F8FFFFFFFF,00000000
    070F8FF,01070F1F3F3F7FFF,80E0F0
    F8FCFCFEFF,010103030307070F,8080C0
    C0C0E0E0F
330 DATA 0F0F1F1F1F3F3F3F,F0F0F8F8F
    8FCFCFC,3F7F7F7F7F7F7F7F,FCFE
    FEFEFEFEFE
340 FOR C=9 TO 13
350 CALL COLOR(C,16,10)
360 NEXT C
370 CALL COLOR(14,10,1)
380 CALL COLOR(15,10,1)
390 CALL COLOR(16,10,1)
400 CALL CLEAR
410 CALL COLOR(2,2,1)
420 PRINT "YOU WILL SEE A DIAGRAM O
    F"
430 PRINT :"THE TEETH WITH THE NAME
    S"
440 PRINT :"OF THE TEETH."
450 A$=CHR$(128)&CHR$(129)
460 B$=CHR$(130)&CHR$(131)
470 PRINT ::"WHEN YOU KNOW THE NAME
    S,"
480 PRINT :"PRESS <ENTER>."
490 D$=CHR$(132)&CHR$(133)&CHR$(134
    )
500 PRINT ::"THE LABELS WILL CLEAR
    AND"
510 PRINT :"YOU WILL BE GIVEN A QUI
    Z.":::: 
520 FOR C=1 TO 5
530 READ N$(C)
540 W$(C)=N$(C)
550 NEXT C
560 DATA CENTRAL INCISORS,LATERAL I
    NCISORS,CUSPIDS,BICUSPIDS,MOLAR
    S
570 GOSUB 1160
580 CALL CLEAR
590 PRINT TAB(8);"^_CENTRAL INCISOR
    S"
600 PRINT TAB(5);CHR$(145)&CHR$(146
    )&CHR$(147)&CHR$(148)&CHR$(149)
610 PRINT TAB(4);CHR$(150);"e`abe";C
    HR$(151);"__LATERAL INCISORS"
620 PRINT "  ";CHR$(150);"phcdcjs";C
    HR$(151)
630 PRINT " ";CHR$(152);"qrieeektu";
    CHR$(153);"CUSPIDS"
640 PRINT " ";CHR$(154);"xyeeeeez{";
    CHR$(155);"BICUSPIDS"
650 PRINT " ";CHR$(156);"|{,}e";CHR$
    (136)&CHR$(137)&CHR$(138);"e{,}"
    ;CHR$(127)&CHR$(157)
660 PRINT " e";A$;CHR$(139);"
    {3 SPACES}";CHR$(140);B$;"e";"MO
    LARS"
670 PRINT " e";A$;CHR$(141);"
    {3 SPACES}";CHR$(142);B$;"e"
680 PRINT " ";D$;CHR$(143);"
    {3 SPACES}";CHR$(144);D$
690 PRINT " eee{5 SPACES}eee":::
700 GOSUB 1160
710 CALL HCHAR(23,16,32,13)
720 CALL HCHAR(10,10,32,18)
730 CALL HCHAR(12,13,32,18)
740 CALL HCHAR(14,15,32,7)
750 CALL HCHAR(15,15,32,9)
760 CALL HCHAR(17,15,32,6)
770 PRINT TAB(8);"NAME THE TEETH"::
780 FOR C=1 TO 5
790 RANDOMIZE
800 I=INT(5*RND+1)
810 IF W$(I)="" THEN 800
820 PRINT TAB(9);C;W$(I)
830 A(I)=C
840 W$(I)=""
850 NEXT C
860 FOR C=1 TO 5
870 I=INT(5*RND+1)
880 IF A(I)=0 THEN 870
890 CALL KEY(0,K,S)
900 CALL COLOR(I+8,6,10)
910 CALL COLOR(I+8,16,10)
920 IF S<1 THEN 890
930 IF (K<49)+(K>53)THEN 890
940 CALL HCHAR(18+K-48,11,62)
950 IF K-48=A(I)THEN 1000
960 CALL SOUND(150,330,0)
970 CALL SOUND(150,262,0)
980 CALL VCHAR(19,11,32,5)
990 GOTO 890
1000 CALL SOUND(150,262,0)
1010 CALL SOUND(150,330,0)
1020 CALL SOUND(150,392,0)
1030 CALL SOUND(200,523,0)
1040 CALL VCHAR(19,11,32,5)
1050 A(I)=0
1060 NEXT C
1070 PRINT ::"TRY AGAIN? (Y/N)"
1080 CALL KEY(0,K,S)
1090 IF K=78 THEN 1200
1100 IF K<>89 THEN 1080
1110 FOR C=1 TO 5
1120 W$(C)=N$(C)
1130 NEXT C
1140 GOTO 580
1150 STOP
1160 PRINT TAB(14);"PRESS <ENTER>"
1170 CALL KEY(0,K,S)
1180 IF K<>13 THEN 1170
1190 RETURN
1200 CALL CLEAR
1210 END
```

©

Atari Sound Experimenter

Matt Giwer

If you've wanted more control over your Atari's sound, here's a solution. You can use this program to experiment, to add sound to other programs (via the SOUND or POKE instructions), and to govern all four voices and all aspects of special effects.

Sound is one of the most important capabilities of the Atari computer. Not only does it permit four-part harmony if you are so inclined, but sound is an essential ingredient in games. It transports you into the world of the game, filling your ears with the sound of a laser cannon, letting you hear force shields as they collapse around you.

Unfortunately, the sound commands are among the most difficult to experiment with. The SOUND instruction can sometimes be clumsy and inconvenient; for one thing, the sounds stay on until you turn them off with another SOUND instruction. Also, you can't achieve the full range of sound with the BASIC instruction, since using it changes any settings in AUDCTL (the register which controls sound effects).

Sound control is a complicated matter, and simple programs cannot offer you complete control over the sounds. Joysticks couldn't govern four channels with nine registers.

This program takes a little practice to get used to, but it permits total control over all sound registers plus AUDCTL, turns the channels on individually, and shuts them all off at once when you need silence. When you are satisfied with the sounds, you can display the appropriate BASIC statements in either the POKE or the SOUND format.

An Overview

Let's first briefly summarize the Atari sound system. (For complete details, see the *Atari Personal Computer System Hardware Manual*, pages III.12 through III.14.) There are four independent sound channels whose distortion, frequency, and volume can be independently controlled. These are addressed by the SOUND instruction with the numbers 0 through 3. The *Hardware Manual* refers to them as 1 through 4. The sound data can be independently POKEd into registers 53760 through 53767. The odd numbers control volume and distortion, and the even numbers control the frequency. Register 53768 is AUDCTL, which controls all of the sound channels in one way or another. If you use the BASIC SOUND instruction, any changes you may have made to AUDCTL are reset – AUDCTL is set to zero. Thus you do not have full control of the sounds with the SOUND instruction.

This program attempts to give you easy control over all of these parameters. Compromises to reduce complexity have been made in favor of the notation and numbers used in the SOUND instruction. Thus you may use the *BASIC Reference Manual* for further information.

```
AUDCTL (REG)ISTER 4
                     9 BIT POLY: (B7):  0
clock Ch.0 w/1.79 MHz: (B6):            0
clock Ch.2 w/1.79 MHz: (B5):            0
      clock Ch.1 w/Ch.0: (B4):          0
      clock Ch.3 w/Ch.2: (B3):          0
clock Ch.0 w/Ch.2 HiP: (B2):            0
clock Ch.1 w/Ch.3 HiP: (B1):            0
                     15 kHz: (B0):      0
         SOUND (REG)ISTER 0
           (DIS)TORTION:               10
           (FRE)QUENCY:               100
           FORCE OUTPUT:                0
                (VOL)UME:               8
X:
D:  ?█
REG DIS FRE FRC VOL
OFF CH
PDIS    SDIS          POKE 53768, 0
POKE    53761, 168    POKE 53760, 100
POKE    53763, 0      POKE 53762, 0
POKE    53765, 0      POKE 53764, 0
POKE    53767, 0      POKE 53766, 0
```

The figure shows the display that you will see upon RUNning and entering the commands. The first eight lines, numbered B7 through B0, are the bits in the AUDCTL Register. To change bit seven to 1, type B7 and RETURN. To change it back to zero, type B7 and RETURN again. These

are technical changes that give no indication of what the new sound will be like. Experimentation is best. Suffice it to say that using B1 through B4 turns on both of the sound channels associated with bit seven.

To discuss the next five lines of the figure, we have to jump down to the lines labeled D: and X:. There are two types of entries to make to this program, those which are purely commands and those which require numbers. If you need to enter a number, enter the number first and push RETURN. If it is a pure command, simply enter the command and RETURN. If you wish to work with sound channel zero, type the following sequence: 0, RETURN, REG, RETURN. A 0 will appear after SOUND (REG)ISTER on the display. For a pure tone, type 10, RETURN, DIS, RETURN, and a 10 will appear after (DIS)TORTION:. Similarly, 100, RETURN, FRE, RETURN, and 8, RETURN, VOL, RETURN, will complete this part of the display.

To hear this sound, type 0, RETURN, CH, RETURN, and to turn it off, type OFF, RETURN. To see the POKE values for this sound, type PDIS, RETURN, and the list of nine POKEs will appear on the screen. Copy these POKEs into your program, and you will duplicate the sound that you hear. The top right POKE is AUDCTL. The next four rows are channels 0 through 3 – the left column is the distortion and volume, and the right is the frequency for each channel.

If AUDCTL is 0 – which is the same as bits B0 through B7 being all 0 – then the SOUND instruction may be used. To see the SOUND instructions, type SDIS, RETURN, and the POKEs will be replaced with SOUNDs.

The "force" output is in the odd-numbered POKE registers and produces a click from the TV. It is turned off and on by use of FRC, RETURN. If you have set any of the AUDCTL bits, you must use the POKEs to duplicate the sounds. The sound channels must be turned on individually by the CH command. OFF turns off all channels. If you make a change and want to hear it, type the channel number and CH again. This may seem cumbersome, but otherwise the sounds would always be on.

Atari Sound Experimenter

```
80 DIM S(5,8),IN$(50)
90 FOR I=0 TO 8:FOR J=0 TO 5:S(J,I)=
   0:NEXT J:NEXT I
100 REG=5000:DIS=5100:FRE=5200:FRC=5
   300:OFF=5400
102 CLD=5900:CLX=6000:VOL=6100:POKAU
   D=6200:CH=6300:START=6400:REGDIS
   =6500:BUZZ=6600
104 PDIS=6700:SDIS=6800:EDIS=6900
1000 REM DISPLAY
1002 GRAPHICS 0:POKE 752,1
1008 POSITION 2,0:? "AUDCTL (REG)IST
   ER 4"
1010 POSITION 2,1:? "{11 SPACES}9 BIT
    POLY:(B7):"
1020 POSITION 2,2:? "clock Ch.0 w/1.
   79 MHz:(B6):"
1030 POSITION 2,3:? "clock Ch.2 w/1.
   79 MHz:(B5):"
1040 POSITION 2,4:? "{4 SPACES}clock
    Ch.1 w/Ch.0:(B4):"
1050 POSITION 2,5:? "{4 SPACES}clock
    Ch.3 w/Ch.2:(B3):"
1060 POSITION 2,6:? "clock Ch.0 w/Ch
   .2 HiP:(B2):"
1070 POSITION 2,7:? "clock Ch.1 w/Ch
   .3 HiP:(B1):"
1080 POSITION 2,8:? "{15 SPACES}15 kH
   z:(B0):"
1090 POSITION 2,9:? "{5 SPACES}SOUND
    (REG)ISTER"
1100 POSITION 2,10:? "{6 SPACES}(DIS
   )TORTION:"
1110 POSITION 2,11:? "{7 SPACES}(FRE
   )QUENCY:"
1120 POSITION 2,12:? "{6 SPACES}FORC
   E OUTPUT:"
1126 POSITION 2,13:? "{10 SPACES}(VOL
   )UME:"
1128 POSITION 2,14:? "X:"
1130 POSITION 2,15:? "D:"
1140 POSITION 2,16:? "REG DIS FRE FR
   C VOL"
1150 POSITION 2,17:? "OFF CH"
1160 POSITION 2,18:? "PDIS SDIS"
1500 GOSUB START
2000 REM JUMP TABLE
2008 FOR ZZZ=1 TO 2 STEP 0
2010 POSITION 5,15:POKE 752,0:INPUT
   IN$:POKE 752,1
2020 TRAP 2040:A=VAL(IN$):TRAP 40000
2030 POSITION 5,14:? A:GOSUB CLD
2040 IF IN$="REG" THEN GOSUB REG
2042 IF IN$="DIS" THEN GOSUB DIS
2044 IF IN$="FRE" THEN GOSUB FRE
2046 IF IN$="FRC" THEN GOSUB FRC
2048 IF IN$="OFF" THEN GOSUB OFF
2049 IF IN$="CH" THEN GOSUB CH
2058 IF IN$="VOL" THEN GOSUB VOL
2060 IF IN$="B7" THEN S(4,7)= NOT (S
   (4,7)):POSITION 30,1:? S(4,7):G
   OSUB CLD
2061 IF IN$="B6" THEN S(4,6)= NOT (S
   (4,6)):POSITION 30,2:? S(4,6):G
   OSUB CLD
2062 IF IN$="B5" THEN S(4,5)= NOT (S
   (4,5)):POSITION 30,3:? S(4,5):G
   OSUB CLD
2063 IF IN$="B4" THEN S(4,4)= NOT (S
   (4,4)):POSITION 30,4:? S(4,4):G
   OSUB CLD
2064 IF IN$="B3" THEN S(4,3)= NOT (S
   (4,3)):POSITION 30,5:? S(4,3):G
   OSUB CLD
2065 IF IN$="B2" THEN S(4,2)= NOT (S
   (4,2)):POSITION 30,6:? S(4,2):G
   OSUB CLD
2066 IF IN$="B1" THEN S(4,1)= NOT (S
   (4,1)):POSITION 30,7:? S(4,1):G
   OSUB CLD
2067 IF IN$="B0" THEN S(4,0)= NOT (S
   (4,0)):POSITION 30,8:? S(4,0):G
   OSUB CLD
2070 IF IN$="PDIS" THEN GOSUB PDIS
2072 IF IN$="SDIS" THEN GOSUB SDIS
```

```
2980 IF FAIL=1 THEN GOSUB BUZZ
2989 FAIL=0
2990 NEXT ZZZ
5000 REM REG REGISTER SET
5010 IF A<0 OR A>3 THEN FAIL=1
5020 IF A>0 OR A<4 THEN POSITION 24,
 9:? A
5030 C=A:REM S(C,B)
5040 GOSUB REGDIS
5088 GOSUB CLD:GOSUB CLX
5090 RETURN
5100 REM DIS DISTORTION LEVEL
5110 IF A<0 OR A>14 THEN FAIL=1:GOTO
  5180
5112 IF INT(A/2)-A/2<>0 THEN FAIL=1:
 GOTO 5180
5120 IF A=0 THEN D1=0
5121 IF A=2 THEN D1=32
5122 IF A=4 THEN D1=64
5123 IF A=6 THEN D1=96
5124 IF A=8 THEN D1=128
5125 IF A=10 THEN D1=160
5126 IF A=12 THEN D1=192
5127 IF A=14 THEN D1=224
5130 POSITION 21,10:? A
5140 S(C,1)=D1:S(C,5)=A
5170 S(C,8)=A
5180 GOSUB CLD:GOSUB CLX
5190 RETURN
5200 REM FRE FREQUENCY STORE
5210 IF A<0 OR A>255 THEN FAIL=1
5218 POSITION 21,11:? "{8 SPACES}"
5220 POSITION 21,11:? A
5230 S(C,2)=A
5280 GOSUB CLD:GOSUB CLX
5290 RETURN
5300 REM FRC  SET FORCE BIT
5310 IF A=0 THEN S(0,3)= NOT S(0,3)
5320 IF A=1 THEN S(1,3)= NOT S(1,3)
5330 IF A=2 THEN S(2,3)= NOT S(2,3)
5340 IF A=3 THEN S(3,3)= NOT S(3,3)
5350 POSITION 21,12:? S(C,3)
5380 GOSUB CLD
5390 RETURN
5400 REM OFF TURN OFF SOUND
5410 POKE 53761,0:POKE 53763,0:POKE
 53765,0:POKE 53767,0
5480 GOSUB CLD
5490 RETURN
5900 REM CLD CLEAR D POS.
5910 POSITION 5,15:? "{20 SPACES}"
5990 RETURN
6000 REM CLX CLEAR X POS.
6010 POSITION 5,14:? "{21 SPACES}":A=
 0
6090 RETURN
6100 REM VOL VOLUME SET
6110 IF A<0 OR A>15 THEN FAIL=1:GOTO
  6180
6120 POSITION 21,13:? "{12 SPACES}"
6122 POSITION 21,13:? A
6130 S(C,4)=A
6180 GOSUB CLD:GOSUB CLX
6190 RETURN
6200 REM POKAUD POKE AUDCTL VALUE
6208 SUM=0
6210 IF S(4,0)=1 THEN SUM=SUM+1
6211 IF S(4,1)=1 THEN SUM=SUM+2
6212 IF S(4,2)=1 THEN SUM=SUM+4
6213 IF S(4,3)=1 THEN SUM=SUM+8
6214 IF S(4,4)=1 THEN SUM=SUM+16
6215 IF S(4,5)=1 THEN SUM=SUM+32
6216 IF S(4,6)=1 THEN SUM=SUM+64
6217 IF S(4,7)=1 THEN SUM=SUM+128
6220 POKE 53768,SUM
6290 RETURN
6300 REM CH TURN ON SOUND CHANNELS
6310 GOSUB POKAUD
6320 IF A=0 THEN POKE 53761,S(0,1)+S
 (0,4):POKE 53760,S(0,2)
6322 IF A=1 THEN POKE 53763,S(1,1)+S
 (1,4):POKE 53762,S(1,2)
6324 IF A=2 THEN POKE 53765,S(2,1)+S
 (2,4):POKE 53764,S(2,2)
6326 IF A=3 THEN POKE 53767,S(3,1)+S
 (3,4):POKE 53766,S(3,2)
6380 GOSUB CLX:GOSUB CLD:GOSUB REGDI
 S
6390 RETURN
6400 REM START  SET UP
6410 FOR I=1 TO 8:POSITION 30,I:? "0
 ":NEXT I
6490 RETURN
6500 REM REGDIS  DISPLAY OF REGISTER
6505 POSITION 21,12:? "{3 SPACES}"
6506 POSITION 21,12:? S(C,3)
6510 POSITION 21,11:? "{6 SPACES}"
6511 POSITION 21,11:? S(C,2)
6520 POSITION 21,10:? "{6 SPACES}"
6521 POSITION 21,10
6522 IF S(C,1)=224 THEN ? "14"
6523 IF S(C,1)=192 THEN ? "12"
6524 IF S(C,1)=160 THEN ? "10"
6525 IF S(C,1)=128 THEN ? "8"
6526 IF S(C,1)=96 THEN ? "6"
6527 IF S(C,1)=64 THEN ? "4"
6528 IF S(C,1)=32 THEN ? "2"
6529 IF S(C,1)=0 THEN ? "0"
6530 POSITION 21,13:? "{6 SPACES}"
6531 POSITION 21,13:? S(C,4)
6590 RETURN
6600 REM BUZZ
6610 ? "{BELL}"
6690 RETURN
6700 REM PDIS  DISPLAY OF POKE DATA
6705 GOSUB EDIS
6710 POSITION 20,18:? "POKE 53768, "
 ;SUM
6720 POSITION 2,19:? "POKE 53761, ";
 S(0,1)+S(0,4):POSITION 20,19:?
 "POKE 53760, ";S(0,2)
6730 POSITION 2,20:? "POKE 53763, ";
 S(1,1)+S(1,4):POSITION 20,20:?
 "POKE 53762, ";S(1,2)
6740 POSITION 2,21:? "POKE 53765, ";
 S(2,1)+S(2,4):POSITION 20,21:?
 "POKE 53764, ";S(2,2)
6750 POSITION 2,22:? "POKE 53767, ";
 S(3,1)+S(3,4):POSITION 20,22:?
 "POKE 53766, ";S(3,2)
6780 GOSUB CLD
6790 RETURN
6800 REM SDIS  DISPLAY OF SOUND DATA
6810 POSITION 2,19:? "SOUND 0, ";S(0
 ,2);", ";S(0,8);", ";S(0,4)
6820 POSITION 2,20:? "SOUND 1, ";S(1
 ,2);", ";S(1,8);", ";S(1,4)
6830 POSITION 2,21:? "SOUND 2, ";S(2
 ,2);", ";S(2,8);", ";S(2,4)
6840 POSITION 2,22:? "SOUND 3, ";S(3
 ,2);", ";S(3,8);", ";S(3,4)
6880 GOSUB CLD
6890 RETURN
6900 REM EDIS  ERASE PDIS &SDIS
6910 POSITION 20,18:? "{18 SPACES}"
6920 POSITION 2,19:? "{35 SPACES}"
6930 POSITION 2,20:? "{35 SPACES}"
6940 POSITION 2,21:? "{35 SPACES}"
6950 POSITION 2,22:? "{35 SPACES}"
6990 RETURN
31990 END
32000 SAVE "D2:SOUND.DEV"
```

©

Commodore REM Revealed

John L. Darling

Did you know that you can prevent someone from easily listing your program? This is one of several hidden secrets of the REM statement. Did you ever try putting shifted or reverse video characters behind a REM? The results you get when you LIST may come as a surprise. Try these experiments to learn about the tricks you can play with REMs. For VIC, 64, and all PET/CBM models.

There are quite a few hidden surprises in the REM statement. Many are just plain fun, but a few can be put to good use. Let's go exploring.

The REM statement was designed to provide a way to add remarks or comments in a program. During execution of the program, all the characters on a line following the REM are ignored. Thus, the only time the remarks are seen is when the program is LISTed.

Also note that, for program operation, it doesn't make any difference whether the characters following the REM are enclosed in quote marks or not, but it sure can change the results you get when you LIST the program. First, let's look at the REM when quotes are not used. The results you get when the program is LISTed will be determined by the following rules:

1. Non-shifted characters appear as typed in.
2. Shifted characters are converted to BASIC commands if the ASCII code for the character is equivalent to a BASIC command token.
3. Reverse fields are stripped from any character.

Before we examine these rules, you may want to put your computer into lowercase mode by typing POKE 59468,14 on the PET/CBM or by hitting shift-Commodore key on the VIC and 64. It is easier to discuss upper- and lowercase letters than it is to describe graphic symbols. Reverse video characters are produced by pressing the RVS key and then the character. The OFF key gets you out of reverse video. (On the VIC and 64, the RVS ON and RVS OFF keys are CTRL-9 and CTRL-0.)

To illustrate these rules, type in the following four lines and then LIST.

```
10 rem a b c d e f
20 rem A B C D E F
30 rem {RVS}a b c d e f{OFF}
40 rem {RVS}A B C D E F{OFF}

list

10 rem a b c d e f
20 rem atn peek len str$ val asc
30 rem a b c d e f
40 rem atn peek len str$ val asc
```

Line 10 demonstrates Rule 1. All the characters are LISTed just as they were entered. This is the normal effect that we're all used to.

Line 20 doesn't look much like the original, does it? It illustrates Rule 2: the shifted letters are interpreted as BASIC command tokens.

Lines 30 and 40 show Rule 3 in action. They look just like lines 10 and 20 because the reverse field was stripped when the lines were entered.

List Blocking

Now we get to the question of how to prevent someone from easily LISTing your program. Let's examine Rule 2 a little more closely. Certain characters become "tokens" which cause unusual effects. One will cause the LIST operation to terminate with a "syntax error" message when it is encountered. These tokens are equivalent to a shifted-L on the VIC, 64, Original and Upgrade PETs. In BASIC 4.0 this character is the shifted-[.

This can be verified by the following line.

VIC, 64, PET Original/Upgrade	BASIC 4.0
10 rem L	**10 rem [**

When you attempt to list the line, the result will be:

10 rem
?syntax error
ready.

For newer CBM models with the "business style" (nongraphics) keyboard, the shifted-[is not directly available from the keyboard. The "[" and "shift-[" appear to the computer as the same character. You can get around this by typing the following line in direct mode (without a line number):

print "10 rem "chr$(219)

When you hit RETURN, the program line should appear. (The chr$(219) will produce a graphics character.) Position the cursor on this line and hit RETURN.

Up to now, it's just been fun, but there is a reason you might want to use this line. If this special REM line is the first line in a program, it prevents a normal LISTing. Let's assume that the first line in a large program is line 100. Inserting this special REM line ahead of the program causes the LIST operation to terminate as soon as it encounters the special shifted character. However, LIST 100- will allow the program to be displayed normally.

Consider the following situation. A quiz program has the answers in DATA statements at the end of the program listing. Inserting the special REM line just ahead of these DATA statements will prevent the answers from being displayed during a LIST. Don't forget that REM statements are ignored during program execution, so they won't affect the actual program operation.

Quote Mode

Now, let's examine the quote mode. A new set of rules applies when the REM characters are enclosed in quotes:

1. Shifted and non-shifted characters LIST as they were typed in.
2. Reverse video characters are preserved when inside quotes (they are not stripped, as is the case in the non-quote mode).
3. Some reverse video characters and combinations of characters behave as print control commands when LISTed.

Rules 1 and 2 produce results that you would normally expect during the LIST operation. They LIST exactly as typed in. No examples are provided for these rules, but try a few experiments to verify this for yourself.

Here are some interesting examples of Rule 3 in action. (The comments in brackets are the resultant action produced during LIST.)

Note: A dot matrix printer was used to list the examples with reverse video characters.

```
rem "T       [delete]
rem "T       [insert]
rem "M       [return]
rem "M       [shifted return] = *
rem "MS      * + [home]
rem "MS      * + [clear screen]
rem "MQ      * + [cursor down]
rem "MQ      * + [cursor up]
rem "M]      * + [cursor right]
rem "M]      * + [cursor left]
```

When these characters are inside a REM" statement, strange things are going to happen.

To enter the following tests, first type the line number, the REM, the quote symbol, and then RETURN. Next, edit the line by positioning the cursor past the quote mark, press the "RVS" key and then the letters. This allows you to put the reverse video characters on the screen line.

```
10 rem"help !tttt

list

10 rem"he
ready.
```

The four reverse "t" characters achieve the same thing that would occur if the DEL key was pressed during an edit operation, deleting the last four characters. Adding more reverse "t" characters (15 total) on the test line will cause the entire line to disappear *after* it is LISTed on the screen.

Notice that many of the cursor controls shown require the "M" (shifted RETURN) character to be the first character. This is important, for without the "shifted return" most of the cursor controls or special control codes will not be executed. As soon as this character is encountered, a shifted RETURN will be generated. All characters following the shifted-M will be printed as if they were in a PRINT statement, rather than in a REM. Consequently, if any of these characters are cursor controls, they will produce a cursor control action as if they were inside the quotes following a PRINT statement.

If the reverse *t*'s in the previous example were replaced with reverse "MS" characters, then the LIST operation would list that line up to the ! and then the cursor will go to the top of screen since "MS" is interpreted as a HOME command. If this was listed to a Commodore printer and the paging mode was on, the printer would eject a page after LISTing that line.

A Program Within A Program

Let's try one final example to illustrate how the reverse field shifted-M works in combination with other characters. To avoid errors, here is a complete key sequence that will produce the following line:

```
1,0,SPACE,R,E,M,",",DEL,RVS,SHIFT-M,
SHIFT-S,Q,Q,Q,Q,OFF,I,SPACE,T,H,I,N,K,
SPACE ,I,SPACE,A,M,SPACE,S,RVS,Q,OFF,I,
RVS,Q,OFF,C,RVS,Q,OFF,K,RVS,S,OFF,",
SHIFT-L
```

(For 4.0 BASIC, replace the final SHIFT-L with SHIFT-[. For "business style" keyboards, use the same technique as in the preceding example. Use chr$(34) in place of the quotes.)

```
10 rem"MSqqqqi think i am sqiqcqkS"L
```

Can you guess the results? If you type the line correctly, the following will happen after you LIST:

1. 10 rem" will be printed.
2. A "clear screen" will be printed, blanking the screen and also the previous 10 rem" .
3. Four "cursor downs" will be printed.
4. The message "I think I am sick" will be printed with the I, C, K characters on different lines.
5. A "cursor-home" will occur.
6. " '@' will be printed on the top line followed by a "?syntax error" message on the next line. (Note that the special shifted character is no longer enclosed in quotes.)
7. Finally, the "ready" message will appear with the cursor above the "I think I am s" line.

The above line could be inserted in most programs, and it will not affect the program execution performance in the least. You just can't get a normal LISTing of the program.

There are a lot more combinations to try, so have fun. It's like having a program inside another program. The second program requires a LIST command for execution instead of a RUN command. ©

Alspa Computer, Inc.

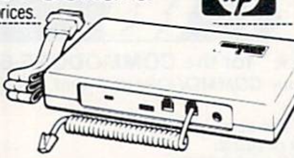
The price-performance leader. Includes Z80A, 1 or 2 full 8" drives (double density, double sided), 3 serial and 1 parallel port, and winchester port. Prices start at less than $2000. Networking Available. DEALER / OEM inquiries invited.

SPECIALS on INTREGATED CIRCUITS

6502	7.45	10/ 6.95	50/ 6.55	100/ 6.15
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-L300		1.95	25/ 1.85	100/ 1.75
2716 EPROM		5.90	5/ 5.75	10/ 5.35
2532 EPROM		6.90	5/ 6.45	10/ 5.90
6116 2K×8 CMOS RAM		5.90	5/ 5.45	10/ 5.10
4116 RAM			8 for 14	
Zero Insertion Force 24 pin Socket (Scanbe)				2.00

Hewlett Packard

Write or call for prices.



Anchor Automation Signalman Modems

FREE SOURCE MEMBERSHIP WITH SIGNALMAN

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

Mark I RS232	(99)	79
Mark II for Atari 850	(99)	79
Mark IV for CBM/PET with software	(169)	99
Mark V for Osborne (software available)	(129)	93
Mark VI for IBM Personal Computer	(250)	180
Mark VII Auto Dial/Auto Answer	(179)	119
Mark VIII Bell 212 Auto Dial/Answer	(399)	319

DC HAYES Smartmodem	**219**
DC Hayes Smartmodem 1200	545

Apple Emulator for Commodore 64	89
Screenmaker 80 COLUMN CARD for C64	149
FROGGER for C64 or VIC	25
Solid Oak 2 Level Stand for C64 or VIC	29
C64/VIC Switch (networking)	125
BACKUP V1.0 tape copier for C64 or VIC	20
CARDBOARD/6 Motherboard - VIC	79
CARDAPTER/1 Atari VCS Adapter - VIC	69
CARDPRINT Printer Interface - C64/VIC	64
CARDBOARD/3s Motherboard - VIC	32
CARDETTE/1 Cassette Interface - C64/VIC	32
CARDRITER Lightpen - C64/VIC	32
CARDRAM/16 RAM Expansion - VIC	64

We carry Apple II+ from Bell & Howell

Apple Emulator for Commodore 64	89
16K RAM Card for Apple	59
Solid Oak 2 Level Stand for Apple	29
Super Serial Card	149
MPC RAM/80 column card for IIE	139
Z80 Softcard and CP/M (Microsoft)	235
Parallel Printer Interface/Cable	79
Apple Dumpling (Microtek) Printer Interface	115
Apple Dumpling with 16K Buffer	160
Grappler + Interface	140
TG Products for Apple in stock	
DC Hayes Micromodem II	299
Videx 80 Column Card	239
Hayden Software for Apple 20% OFF	
PIE Writer Word Processor	120

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN $245 base price

Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

CBC / PET / C64 COMMUNICATIONS!

COMPACK $115

Intelligent Terminal Package

ACIA Hardware / STCP Software

VE-2 IEEE to Parallel Interface 110

Includes case, power supply, full 8-bit transmission, and switch selectable character conversion to ASCII.

VIDEO ENHANCER for Commodore 64 89

Realize video quality equal or better than composite monitor using standard color TV.

SCREENMAKER 80 Column Adpater for C64 149

Provides big screen capability for business applications.

VIC 20 Products		VIC Sargon II Chess	32
BACKUP V1.0	20	**VIC GORF**	32
VIC RAM Cards in stock		Meteor Run (UMI)	39
VIC SuperExpander	52	VIC Radar Ratrace	24
VIC 16K RAM	95	Amok (UMI)	20
Thorn EMI Software		Snakman	15
HES Software		Rubik's Cube	13
VIC Omega Race	32	Programmers Reference	15
Spiders of Mars (UMI)	39	**FROGGER**	25
Programmers Aid	45	VIC Adventure Series	
VICTORY Software for VIC and C64			
Street Sweepers	12	Maze in 3-D	12
Night Rider	11	Cosmic Debris	12
Treasures of Bat Cave	12	Grave Robbers Advent.	11
Games Pack I	12	Games Pack II	12
Victory Casino	8	Adventure Pack I	12
Adventure Pack II	12	Trek	11

Commodore 64 Programmers Reference Guide	16
MicroChess for C64 or PET	19
Compute!'s First Book of PET/CBM	11
C64 or VIC SWITCH	125
POWER ROM Utilities for PET/CBM	78
WordPro 3+/64	69
WordPro 4+ - 8032, disk, printer	295
SPELLMASTER spelling checker for WordPro	170
VISICALC for PET, ATARI, or Apple	189
PETRAX PET to Epson Graphics Software	40
SM-KIT enhanced PET/CBM ROM Utilities	40
Programmers Toolkit - PET ROM Utilities	35
CALC RESULT for C64	135
PET Spacemaker II ROM Switch	36
COPYWRITER Word Processor for C64	69
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, ADA1450, SADI in stock	
ZRAM - CBM 64K RAM, Z80, CP/M	550
Programming the PET/CBM (Compute!) — R. West	20
Compute! First Book of VIC	11
Whole PET Catalog (Midnight Gazette)	8
PET Fun and Games (Cursor)	11
Color Chart Video Board for PET	125
REVERSAL (Spracklen) Apple or Atari	25
SARGON II — Apple or TRS-80	26
Apple II User's Guide (Osborne)	12
Introduction to Pascal (Sybex)	13
Pascal Handbook (Sybex)	16
Musical Applications of Micros (Chamberlin)	20
Starting FORTH	14
Discover FORTH	12
User Guide to the Unix System	13
6502 Assembly Language Subroutines	11
COMAL Handbook	9
KAMIKAZE (Hayden Software-Apple)	28

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.25	50/ 2.10	100/ 2.05
Scotch (3M) 5" ds/dd	10/ 3.15	50/ 2.90	100/ 2.85
Scotch (3M) 8" ss/sd	10/ 2.40	50/ 2.20	100/ 2.15
Scotch (3M) 8" ss/dd	10/ 2.95	50/ 2.70	100/ 2.65

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

BASF 5" or 8"	10/ 2.00	20/ 1.95	100/ 1.85
NEW BASF Qualimetric Disks also in stock.			
Wabash 5"ss/sd	10/ 1.80	50/ 1.75	100/ 1.70
Wabash 5" ss/dd	10/ 2.00	50/ 1.95	100/ 1.90
Wabash 8" ss/sd	10/ 2.00	50/ 1.95	100/ 1.90

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages	10 for $5	Hub Rings 50 for $6
Disk Library Cases	8"—3.00	5"—2.25
Head Cleaning Kits	11	

CASSETTES—AGFA PE-611 PREMIUM

C-10	10/ .61	50/ .58	100/ .50
C-30	10/ .85	50/ .82	100/ .70

DATASHIELD BACKUP POWER SOURCE 225

Battery back up Uninterruptible Power Supply with surge and noise filtering. The answer to your power problems.

Zenith ZVM-121 Green Phosphor Monitor	100
BMC 12A 12" Green Monitor	80
VOTRAX Personal Speech System	280
VOTRAX Type-N-Talk	160
VOICE BOX Speech Synthesizer (Apple or Atari)	
CompuServe Subscription (5 hours free)	32
Prowriter Parallel Printer	389
USI CompuMOD 4 R F Modulator	39
Daisywriter 2000	1050
Many printers available (Star, Brother, OKI, etc.)	
We Stock AMDEK Monitors	
Amdek DXY-100 Plotter	600
A P Products	15% OFF
Watanabe Intelligent Plotter 990	6-pen 1290
ISOBAR 4 Outlet Surge Suppressor/Noise Filter	**49**
We stock Electrohome Monitors	
dBASE II (8" format)	325

ALL BOOK and SOFTWARE PRICES DISCOUNTED

Panasonic TR-120M1P 12" Monitor (20 MHz)	149
Panasonic CT-160 Dual Mode Color Monitor	285

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

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

ZENITH data systems

Z29 Terminal (new detached keyboard)	**680**
ZT-1 Intelligent Communications Terminal	**369**
Z100 16-bit/8-bit Systems in stock	**CALL**
We stock entire Zenith line.	

ATARI SPECIALS

WE STOCK ENTIRE LINE—write for prices.

Atari 1200	549	**QIX**	34
Voice Box	100	**Anchor Modem—Atari**	79
FROGGER	25	**Atari Graphics** (Compute!)	11
Thorn EMI Software		**First Book of Atari**	11
EduFun Software		**APX Software**	

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

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

OAK STAND-C64, VIC, Apple, Atari 29

Beautiful natural solid oak two-level stand. Rests on table above computer. Holds disk drives/cassette deck, as well as your monitor/TV.

KMMM Pascal for PET/CBM/C64 $85

A subset of standard Pascal with extensions. Includes Machine Language Pascal Source Editor, Machine Language P-Code Compiler, P-Code to machine language translator for optimized object code, Run-time package, Floating Point capability, User Manual, and sample programs.
Requires 32K Please specify configuration.

EARL for PET (disk file based) $65

Editor, Assembler, Relocater, Linker

Generates relocatable object code using MOS Technology mnemonics. Disk file input (can edit files larger than memory).

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 SUPER-SORT), universal wedge, Extramon, etc.

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

SUBSORT for PET/CBM $35

Excellent general purpose machine language sort routine.

THE WHOLE PET CATALOG $9

A two year 320 page compendium of the Midnite Software Gazette for Commodore computer users. Contains 500 reviews of commercial products, 700 education programs (reviewed and organized by course), 200 reviews of free games, info on over 1800 free programs, list of PET and VIC user groups, and many pages of helps and hints.

COMAL Package for CBM $25

Includes software on disk, and Comal Handbook

SuperGraphics 2.0

NEW Version with TURTLE GRAPHICS

SuperGraphics, by John Fluharty, provides a 4K machine language extension which adds 35 full featured commands to Commodore BASIC to allow fast and easy plotting and manipulation of graphics 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 ($A000 or $9000) $45

Volume discounts available for schools.

FLEX-FILE for PET/CBM Computers — NEW VERSION 2 now for C64

FLEX-FILE is a set of flexible, friendly programs to allow you to set up and maintain a data base. Includes versatile Report Writer and Mail Label routines, and documentation for programmers to use Data Base routines as part of other programs.

RANDOM ACCESS DATA BASE

Record size limit is 256 characters. The number of records per disk is limited only by record size and free space on the disk. File maintenance lets you step forward or backward through a file, add, delete, or change a record, go to a numbered record, or find a record by specified field (or partial field). Field lengths may vary to allow maximum information packing. Both subtotals and sorting may be nested up to 5 fields deep. Any field may be specified as a key. Sequential file input and output, as well as file output in WordPro and PaperMate format is supported. Record size, fields per record, and order of fields may be changed easily.

MAILING LABELS

Typical mail records may be packed 3000 per disk on 8050 (1400 in 4040). Labels may be printed any number wide, and may begin in any column position. There is no limit on the number or order of fields on a label, and complete record selection via type code or field condition is supported.

REPORT WRITER

Flexible printing format, including field placement, decimal justification and rounding. Define any column as a series of math or trig functions performed on other columns, and pass results such as running total from row to row. Totals, nested subtotals, and averages supported. Complete record selection, including field within range, pattern match, and logical functions can be specified.

FLEX-FILE 2 by Michael Riley $110

Please specify equipment configuration when ordering.

DISK I.C.U. $40

Intensive Care Unit by L.C. Cargile

COMPLETE DISK RECOVERY SYSTEM FOR CBM DRIVES

- edit disk blocks with ease
- duplicate disks, skipping over bad blocks
- complete diagnostic facilities
- unscratch scratched files
- check and correct scrambled files
- recover improperly closed files
- extensive treatment of relative files
- optional output to IEEE488 printer
- comprehensive user manual (an excellent tutorial on disk operation and theory).

Furnished on copy-protected disk with manual.
Backup disk available, $10 additional.

PROGRAM YOUR OWN EPROMS $75

Branding Iron EPROM Programmer for PET/CBM software for all ROM versions. Includes all hardware and software to program or copy 2716 and 2532 EPROMs.

PORTMAKER DUAL RS232 SERIAL PORT $63

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

CBM Software

BASIC INTERPRETER for CBM 8096	**$200**
PEDISK II Systems from cgrs Microtech available.	
FILEX IBM 3741/2 Data Exchange Software available.	
JINSAM Data Base Management System for CBM.	
COPY-WRITER Word Processor for PET/CBM	**$159**
CASH MANAGEMENT SYSTEM	**$45**
Petspeed BASIC Compiler	120
Integer BASIC Compiler	120
CMAR Record Handler	110
UCSD Pascal (without board)	135
Wordcraft 80 or 8096	265
BPI Accounting Modules	280
Professional Tax Prep System	575
ASERT Data Base	375
Dow Jones Portfolio Management	110
Assembler Development	80

FORTH for PET now for C64

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

Features include:

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

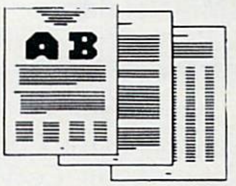
For Commodore 64, or any 16K/32K PET/CBM with ROM 3 or 4, and CBM disk drive. Please specify configuration when ordering.

Metacompiler for FORTH **$30**

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

PageMate 60 COMMAND WORD PROCESSOR

by Michael Riley



Paper-Mate is a full-featured word processor for Commodore computers. Page-Mate incorporates 60 commands to give you full screen editing with graphics for all 16K or 32K machines (including 8032), all printers, and disk or tape drives. Many additional features are available (including most capabilities of WordPro 3).

Page-Mate functions with all Commodore machines with at least 16K, with any printer, and either cassette or disk.

To order Page-Mate, please specify machine and ROM type.

Page-Mate (disk or tape) for PET, CBM, VIC, C64 **$40**

SM-KIT for PET/CBM $40

Enhanced ROM based utilities for BASIC 4. Includes both programming aids and disk handling commands.

Commodore 64

Hunter-Killer - Commodore 64	**15**
- authentic naval warfare game (complete with sonar)	
Submarine Warfare (Clockwork Computers)	29
WordPro 3+/64	75
Vanilla PILOT with Turtle Graphics	27
- also includes sound, Toolkit, joystick support	
Commodore 64 Programmer Reference Guide	16
CCI Submarine Warfare	24
Laser Command	15
EARLY GAMES for Young Children	**25**
PETSPEED Compiler C64	120
CALC RESULT Spread Sheet Package	**135**
1000 Miles (Mille Bornes Game	9
MicroChess	**19**
Adventure (disk)	9
Draw Poker	5
MAE Assembler - C64	85
Assembly Language Tutorial - C64/VIC	27
Abacus Software in stock	
Synthy-64 music and sound synthesizer	26
Tiny BASIC Compiler	17
ScreenGraphics-64 adds BASIC Graphics	22
Victory Software for Commodore 64 in stock	
Adventure Pack I (Victory Software)	12
Adventure Pack II (Victory Software)	12
Annihilator	16
Chomper Man	16
Educational Pack I	10
Grave Robbers (Victory Software)	12
Kongo Kong	16
Strategy Pack I	16
TREK	12

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

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

COMPUTE!'s First Book Of VIC

Authors: **COMPUTE!** Magazine contributors
Price: $12.95
On Sale: Now

Finally, it's VIC's turn!

Users of other popular personal computers have been enjoying their **COMPUTE! Books**: *COMPUTE!'s First Book Of PET/CBM, Programming The PET/CBM*, and others.

Now, there's a book devoted exclusively to the Commodore VIC-20™ Computer: *COMPUTE!'s First Book Of VIC.*

The editors of **COMPUTE!** Magazine – the leading resource for the VIC-20 – gathered together the best VIC-20 articles published since the summer of 1981 and added some new material. The result is more than 200 pages of valuable information – information that goes beyond the instruction manuals. In the **COMPUTE!** tradition, it is carefully edited to be easily understood and useful for beginners and experts alike.

COMPUTE!'s First Book Of VIC is spiral-bound to lie flat, and includes ready-to-type program listings and articles such as "The Joystick Connection: Meteor Maze," "STARFIGHT3," "Train Your PET To Run VIC Programs," "Renumber BASIC Lines The Easy Way," "High Resolution Plotting," "Custom Characters For The VIC," "VIC Memory – The Uncharted Adventure," and "A Simple Monitor For The VIC."

At only $12.95, less than most computer manuals, *COMPUTE!'s First Book Of VIC* is among the best resources a VIC user can own.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2 shipping and handling. Outside the U.S. add $5 for air mail, $2 for surface mail. All orders prepaid, U.S. funds only.
VIC-20 is a trademark of Commodore Electronics Limited.

HARMONY VIDEO & COMPUTERS
800-221-8927

JVC • PANASONIC • SONY • PROFEEL • ZENITH • HITACHI • QUASAR

G.E. • SANYO • SONY • SONY PROJECTION • TDK • MAXELL • FUJI • SCOTCH • SONY • RCA

COMMODORE

VIC 20	89.95
COMMODORE 64	369.95
VIC 2N DATASETTE	59.95
COMMODORE DISC DRIVE	289.95
1525P PRINTER	289.95
COMMODORE MONITOR	249.95
VIC TELEPHONE MODEM	89.95
RS 232 TERMINAL INTERFACE	CALL
IGEE-488 INTERFACE	CALL
VIC 8K MEMORY PAC	39.95
VIC 16K MEMORY PAC	89.95
MOTHER BOARD	89.95
VIC 3K SUPER EXPANDER	
VIC PROGRAMMER'S AID	CALL
VIC MON	CALL
INTRO TO BASIC I & II	69.95

APPLE 2 PLUS
999.95

APPLE 2E
CALL

EPSON
MX 80 PRINTER
399.95

EPSON
FX 80 PRINTER
599.95

ATARI

ATARI	400 W16K	159.95
ATARI	800 W48K	449.95
ATARI	1200 XL W64K	589.95
ATARI	410 RECORDER	76.95
ATARI	1010 RECORDER	77.95
ATARI	810 DISC DRIVE	389.95
ATARI	830 ACOUSTIC TEL. MODEM	159.95
ATARI	850 INTERFACE	169.95
ATARI	822 THERMOL PRINTER	CALL
ATARI	1020 40 COL. PRINTER	219.95
ATARI	1025 80 COL. PRINTER	379.95
ATARI	16K MEMORY EXPANDER	89.95
ATARI	WORD PROCESSOR	69.95
ATARI	BASIC REF. GUIDE	CALL
ATARI	PROGRAMMER KIT	CALL
ATARI	ENTERTAINER KIT	29.95

WE CARRY A FULL LINE OF SOFTWARE FOR ATARI AND COMMODORE
CALL FOR LOWEST PRICES

ALL SONY T.V.'s IN STOCK CALL FOR LOWEST $ PRICES $

ALL VCR's IN STOCK CALL FOR LOWEST $ PRICES $

VIDEO TAPE (By case of 10 only)

	VHS T120	VHS T160	HIGH GRADE	BETA L500	BETA L750	BETA L830
TDK	8.85	14.75	10.75	6.50	7.99	10.25
FUJI	9.65	14.75	11.75	6.50	7.99	
MAXELL	8.40	13.50	10.25	6.50	7.99	10.25
MEMOREX	8.75	13.50	10.50	6.50	7.99	10.25
SCOTCH	8.50	13.50	10.25	6.50	7.99	
SONY			10.25	6.99	8.40	10.25
JVC	9.50	13.50	11.50			
RCA	8.90					
PANASONIC	8.50					

TDK PROFESSIONAL T120	15.95
MAXELL PROFESSIONAL L750 OR T120	15.95
MEMOREX PROFESSIONAL	16.95

FOR THE LOWEST PRICES ON VIDEO CALL 800-221-8927

To order simply dial toll free 800-221-8927 with your Master Card or VISA and your order will arrive via UPS or send certified check or money order only to: HARMONY VIDEO AND ELECTRONICS, 2357 Coney Island Ave., Brooklyn, N.Y. 11223, and add approximate shipping postage and insurance charges. Credit cards for phone order only. For customer service please dial (212) 627-8960. All pries and availability subject to change without notice. All orders shipped out of state. Dealer inquires invited!!! For sales info dial (212) 627-6989. Open Sun. 10-4 Mon.-Thur. 9-7 Fri. 9-3.

800-221-8927

COMMODORE ATARI APPLE TEXAS INSTRUMENTS EPSON

VIC MUSICIAN

Blake Wilson

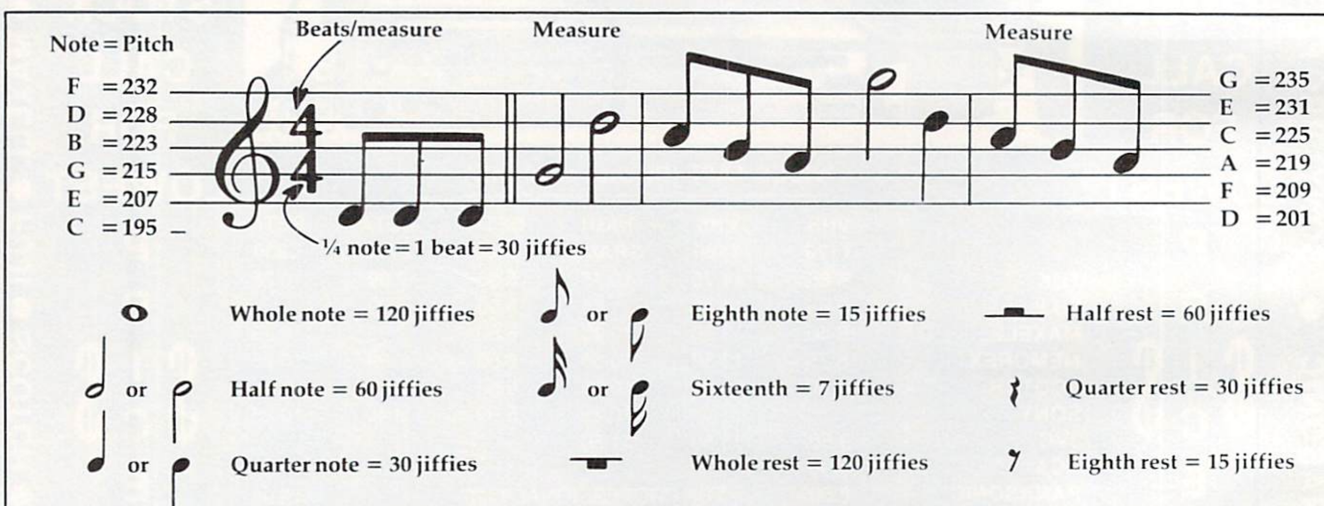
You can have your VIC playing a song in the background while some other BASIC program is RUNning. It will even play while you're programming. This article will also show you how to take music from a printed score and enter it into your VIC.

I have always believed that music in any form was above my head. I own a computer retail store, and several customers have asked how they could produce music that would run *continuously*, without delays, during a program. My wife helped me to understand just what all those incomprehensible symbols on a piece of sheet music actually mean. I wanted the explanation in terms that I (and VIC) could understand. The only musical instrument that I can play is the VIC. Here's her response:

lines). The lower set is often for harmony; usually we can ignore them.

Compare the data at line 190 with the musical score above. Notice that the first item is duration. The first note is filled in and has a shaft and one flag. That means that it is an eighth note and will be played for 15 jiffies. The next data item represents the pitch – it is 201 because the note is tangent to the lowest line. The next two notes are the same. The fourth note is open and has a shaft. So it is a half note and gets 60 jiffies. Since that note is on the third line, its pitch is 'G' = 215. See how it works?

Inspect a musical score and write data for each note, first duration and then pitch. Place the data in the program to replace lines 190 through 260. The actual line numbers mean nothing, but



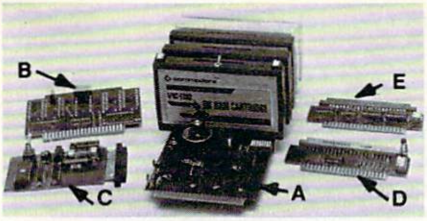
Don't be concerned that some shafts go up and some down, or that notes may be joined with arcs or bar-like flags. These exist only to confuse computer people. You need be concerned only with the duration of a note (filled in or open, whether it has a shaft or not, and the number of bars or flags. By the way, a dot after a note increases its duration by 50 percent) and its pitch (as determined by its altitude on the staff). Any pitch of less than 128 is a rest or silence.

If notes are stacked, just take the highest one. There are generally two staffs (groups of

the first 71 data items representing the program must come first.

The program is limited to 61 notes due to the size of the cassette buffer. In practice, the 61 notes should be enough for most programs. I use only 37 notes in the sample program. If your creation uses fewer than 61 notes, end your data with 1,1. This instructs the computer to replay from the first.

Once the data has been POKEd into the buffer and you've done a SYS to 830, the music plays continuously even if you execute NEW or write



VIC-20* OWNERS

Expand your System with these Exclusive Factory Direct Products

A. Dataspan-50

The **Dataspan-50** expansion board is the cornerstone for expanding the VIC-20 to its maximum capabilities. It is the most advanced, yet easiest to use expansion board available. Unlike other expansion boards, the Dataspan-50 has the following **exclusive** features:

- 5 slot combination **rotary** and **rocker switch** selectable expansion board conveniently **covers all switching needs** now and in the future without using common hazardous slot-by-slot power switching.
- Combination switches allow **total control** between computer cartridges (memory expansion, Programmer's Aid*, Vic-Mon* and other utilities) and game cartridges.
- Dataspan-50 allows **stacking of memory cartridges** up to **29K** in BASIC and **40K** in machine language.
- **Fully buffered** by five hi-tech, low power, integrated circuits that help **prevent erratic operation and loss of data** common in typical unbuffered expansion boards and isolates the VIC's* microprocessor from accidental damage.
- **Highest quality** circuit board with **gold contacts** throughout.
- **Fused** to protect the VIC-20* power supply.
- Master **reset button** eliminates turning computer off and on.
- Independent write-protection on two slots.

Dataspan-50 Kit **$69.95**
Dataspan-50 Assembled **$84.95**
(Suggested Retail Price Assembled $109.95)

D. Mother Switcher

Now make any bare bones expansion board fully block selectable.

- Master reset button.
- Write-protection switch.
- Cartridges piggyback on Mother Switcher.

Mother Switcher Assembled **$11.95**
(Suggested Retail Price $17.95)

Kits for Experienced Builders only!
All assembled units have full 90-Day Limited Guarantee.
*Trademark Commodore Bus. Machines

NOTES: These prices are effective June 1, 1983 and are subject to change without notice. All kits supplied with **complete assembly and operating instructions.**

B. Champagne Memory on a Beer Budget

Highest quality glass epoxy 16K memory board with gold fingers provides full block switching and write-protection on **each** 8K block. All block switches are conveniently located at the **top edge** of the board.

DataRAM
a] Bare memory board [RAM/ROM] **$13.95**
b] Bare memory board Kit **$17.95**
includes all components except RAM/ROM chips

DataRAM 8
c] Board with 8K RAM - Complete Kit **$34.95**
d] Board with 8K RAM Assembled **$37.95**
(Suggested Retail Price Assembled $47.95)

DataRAM 16
e] Board with 16K RAM - Complete Kit **$48.95**
f] Board with 16K RAM Assembled **$54.95**
(Suggested Retail Price Assembled $69.95)

C. Datablast-16

Finally a low cost, high quality 2716 EPROM programmer for the VIC-20! Put your most often used machine language programs into EPROMs.

- On-board 25 volt power supply.
- Can be used with COMPUTE!'s, "Micromon" or our software below.
- Program/read mode switch.
- We suggest you use the EPROMs with our DataRAM memory board.

Datablast-16
a] Kit with low-cost ZIF* socket **$49.95**
b] Assembled with low-cost ZIF* socket **$59.95**
(Suggested Retail Price Assembled $79.95)
c] Kit with Textool ZIF* socket **$55.95**
d] Assembled with Textool ZIF* socket **$61.95**
(Suggested Retail Price Assembled $85.95)
e] Software for Datablast-16 (tape) .... **$ 9.95**
f] 2716 EPROMs .... **$ 4.50**
*Zero insertion force.

TERMS: No. C.O.D. orders. Shipping and Handling **$3.00.** VISA/MASTERCARD - Add 3%. Most orders shipped within 48 hours. (Personal checks - allow 2 weeks.)

Digital Interface Systems Co.
P.O. Box 8715
Portland, OR 97207
(503) 295-5890

CB-2

The CB-2 is a complete hardware and software package that allows you to easily and efficiently make a back-up copy of your valuable software library. Now you can protect your investment!

Unique features:
- Allows connection for one or two Datasette* recorders (or equivalent). Two recorders required for **simple** back-up copies.
- Exclusive state-of-the-art circuitry lets you **actually hear and see** tape data being loaded or saved.
- Special wave shaping circuitry makes a back-up copy as good or better than the original.
- CB-2's **Super Blocksaver** software and interface card allow you to make a back-up copy of your cartridge programs.

a] CB-2 Assembled **$89.95**
(Suggested Retail Price $114.95)
b] Special CB-2 System for Dataspan-20/50 owners **$79.95**
[Please specify which model when ordering]
(Suggested Retail Price $104.95)

E. RAMraider

- Makes your 3K or Superexpander* cartridge a full 4K RAM.
- Recaptures your RAM for BASIC and moves it into Expansion memory (lower half of Blocks 1, 2, or 3).

RAMraider Kit **$24.95**
RAMraider Assembled **$34.95**

RAMcharger

- Turn your Commodore 8K cartridge into a full 16K cartridge.
- Full block switching capabilities.
- Sockets allow future EPROM substitution.

RAMcharger Kit **$28.95**

What Makes The VIC Tic?

If you think computer hardware means nuts and bolts, this book is for you! Written especially for the beginner by VIC enthusiasts.

What Makes The VIC Tic? .... **$ 7.95**
(Shipping included.)

Write For FREE Catalog — *Write For FREE Catalog*

VISA

NEW
VIC SOFTWARE CBM 64

MasterCard

Great VIC Software

PARATROOPER a High Resolution game that doesn't let you make any mistakes. You are in your command. Helicopters fill the sky, (and we mean fill the sky!), dropping paratroopers. Your mission is to keep 3 paratroopers from hitting the ground on either side of your gun. But that's just the beginning. You score by hitting the helicopters or the paratroopers, but if you miss a shot it subtracts from your score. Therefore, you must make every shot count to make a high score! IT HAS FOUR FAST ACTION LEVELS TO CHALLENGE THE BEST PLAYER. The High Resolution graphics helicoptors are fantastic. They look exactly like helicopters! The paratroopers are super realistic. Their chutes open and then they drift down to earth. If this weren't enough the sounds are fantastic. There are helicopter blades whirring and you can hear the howitzer pumping shells. This game really show off the sound and graphic capabilities of your VIC. PARATROOPER IS OUR #1 SELLING ARCADE GAME, you've got to see this game to believe it. **$19.95**

SPACE PAK Can you survive? 3 space games with the sights and sounds of an arcade. The excitement builds as the action is un-ending. lBlast away at everything in sight. The alien attacks will stop at nothing to destroy you. Prepare for battle, there is no escape, only you can help. Can you survive? Hi-Res, color, graphics and sound. Joystick or keyboard. 3 Games — Rocket Race, Fence-A-Tron and Raiders. **$19.95**

COSMIC CRUZER Hot action and 3 challenging scenarios. Move your cruzer into the tunnel - fire missiles and drop bombs. Hit the fuel dumps to get more fuel. Move as quick as you dare to hit the surface-to-air missiles. If you are good enough you will make it to the asteroidz field and then try to destroy the base. No one has destroyed the base yet. Will you be the first. **$19.95**

VIC ALL STARS We took the best selling VIC programs and put them in a package to save you $35. If purchased seperately it would cost you $85. You get **Paratrooper, Target Command, Head On, Cattle Round-up, Snake Out, Trapper, Double Snake Out and Artillery.** All eight games for $49.95. Hurry because at this price they won't last long. Limited quantity. 8 Games. **$49.95**

COMMODORE 64 SOFTWARE

Let the COMPUTERMAT turn your 64 into a home arcade!

COLOR • GRAPHICS • SOUND
ON CASSETTE

(Disk Versions Available — Add $5.00)

MUSIC MAKER - $19.95
Put sheet music notes into your 64, plays 3 voices. Program, plus 2 sample songs.

EDUCATION PAK - $24.95
4 Programs
Geography Match
Math Adventure
Ruler & Micro

TREASURE PAK - $14.95
3 Programs
Adventure
Caves of Silver
Shuttle Voyage

GAME PAK - $14.95
3 Programs
Dragon Chase
Deflect
Flip It

Joystick and Keyboard versions included.

COMPUTERMAT
Box 1664 • Dept. C • Lake Havasu City, Az. 86403
(602) 855-3357

another program. The music will continue until:

1. You type a RUN/STOP RESTORE
2. Attempt to use the cassette
3. Or get so tired of it that you shut off the VIC.

The program is driven by the *hardware interrupt*. Each jiffy (1/60 of a second), a counter ($033D=829) is reduced by one. If this decrement results in a value of zero, the next duration is placed in the counter, the next note is placed in the sound generator ($900C=36876), and the note count is increased by two and stored at ($033C=828). The computer then goes on its way, updating the realtime clock and scanning its keyboard.

The computer is not slowed appreciably. Most of the time, unless the note is to be changed, the time wasted is 120 microseconds per second or 120 parts per million. If the note must be changed, then the time lost is a few thousandths of a second.

Note: A complete listing of musical pitch values, including sharps and flats, is in the *VIC-20 Programmer's Reference Guide*.

Program 1: VIC Musician

```
10 REM ******* VIC MUSICIAN **********
30 REM *** CONTINUOUSLY PLAYS *******
40 REM ** TUNE FROM STAR WARS ******
60 POKE36878,15 : REM SET VOLUME TO MAX
70 FORI=830TO976:READC:POKEI,C:NEXT
80 SYS 830 :REM STARTS MUSIC
90 REM ******* SET-UP FOLLOWS *******
100 DATA120,169,5,141,60,3,169,6,141,61,3,
    169
110 DATA133,133,0,169,3,133,1,169,93,141
120 DATA20,3,169,3,141,21,3,88,96
130 REM **** MUSIC PGM FOLLOWS ******
140 DATA206,61,3,208,28,72,152,72,172,60
150 DATA3,200,177,0,141,61,3,200,177,0,201
160 DATA1,240,12,141,12,144,140,60,3,104
170 DATA168,104,76,191,234,160,255,208,243
180 REM *** MUSIC DATA FOLLOWS ******
190 DATA15,201,15,201,15,201,60,215,60,228
    ,15
200 DATA225,15,223,15,219,60,235,35,228,15
210 DATA225,15,223,15,219,60,235,35,228,15
220 DATA225,15,223,15,225,60,219,35,201,15
230 DATA201,60,215,60,228,15,225,15,223,15
240 DATA219,60,235,35,228,15,225,15,223,15
250 DATA219,60,235,35,228,15,225,15,223,15
260 DATA219,60,235,1,1
```

Program 2: Disassembled Machine Language For VIC Musician

```
033E 78         SEI          ;PROHIBIT INTERRUPTS (DON'T BOTHER ME)
033F A9 05      LDA #$05     ;FIRST NOTE IS SILENCE
0341 8D 3C 03   STA $033C    ;STORE NOTE
0344 A9 06      LDA #$06     ;WAIT 6 JIFFIES
0346 8D 3D 03   STA $033D    ;STORE DELAY
0349 A9 85      LDA #$85     ;ADDRESS LOW OF START OF MUSIC DATA
034B 85 00      STA $00      ;ZERO PAGE POINTER
034D A9 03      LDA #$03     ;ADDRESS HI START OF MUSIC DATA
034F 85 01      STA $01      ;ZERO PAGE POINTER
0351 A9 5D      LDA #$5D     ;ADDRESS LOW OF MUSIC PROGRAM
0353 8D 14 03   STA $0314    ;VECTOR INTERRUPT TO MUSIC PGM
0356 A9 03      LDA #$03     ;ADDRESS HI OF MUSIC PROGRAM
0358 8D 15 03   STA $0315    ;VECTOR INTERRUPT TO MUSIC PGM
035B 58         CLI          ;OK TO INTERRUPT AGAIN
035C 60         RTS          ;BACK TO BASICS
035D CE 3D 03   DEC $033D    ;REDUCE DELAY COUNTER
0360 D0 1C      BNE $037E    ;UNLESS DELAY IS FINISHED, EXIT
0362 48         PHA          ;SAVE ACCUMULATOR ON STACK
0363 98         TYA          ;TRANSFER Y TO ACCUMULATOR
0364 48         PHA          ;SAVE IT TOO
0365 AC 3C 03   LDY $033C    ;GET POINTER TO LAST NOTE FREQUENCY
0368 C8         INY          ;MOVE POINTER TO NEXT DELAY
0369 B1 00      LDA ($00),Y  ;GET NEXT DELAY
036B 8D 3D 03   STA $033D    ;SET DELAY DURATION
036E C8         INY          ;MOVE POINTER TO NOTE FREQUENCY
036F B1 00      LDA ($00),Y  ;GET NOTE FREQUENCY OR TONE
0371 C9 01      CMP #$01     ;IS FREQUENCY=1 (START OVER FLAG)
0373 F0 0C      BEQ $0381    ;IF SO, PLAY IT AGAIN SAM
0375 8D 0C 90   STA $900C    ;START NOTE PLAYING
0378 8C 3C 03   STY $033C    ;STORE NOTE FREQUENCY POINTER
037B 68         PLA          ;RETRIEVE Y FROM STACK
037C A8         TAY          ;PUT Y BACK IN Y
037D 68         PLA          ;RETRIEVE ACCUMULATOR
037E 4C BF EA   JMP $EABF    ;GO ON ABOUT YOUR BUSINESS
0381 A0 FF      LDY #$FF     ;SET NOTE POINTER TO START -1
0383 D0 F3      BNE $0378    ;BRANCH ALWAYS TO STORE POINTER
```

©

Timex/Sinclair Screenscrolls

Glen Martin

As anyone who has worked with a ZX-81 or TS-1000 knows, the scroll facility could be expanded and improved. It cuts down the display file, and screen POKEs (which are difficult at best) are made nearly impossible. This article provides the solution.

Here are four machine language routines which provide left, right, up, and down scrolls. The scrolls may be used singularly or in any combination. When entered together, they occupy a mere 106 bytes, including the six bytes needed for the REM statement they are stored in.

These utilities will keep the display file expanded to its maximum size, making screen POKEs possible. If preferred, the programs could be assembled above RAMTOP so they would not be affected by NEW or LOAD. *Note:* These programs will not work with less than 3¼K of RAM, because of the unexpanded display file.

The programs are POKEd in by a short BASIC hex loader, which can also be used for your own machine language routines. Although no previous knowledge of machine language is required to use these programs, I have included mnemonics as well as hex listings. This will be helpful to those readers who understand the Z-80 instruction set and wish to modify or disassemble the programs.

Step-by-step Instructions

The first step in using these scrolls is to type in a REM statement to store them in. The REM statement must be at least 100 characters long, preferably a few bytes longer, so if a mistake is made the program won't POKE up into BASIC. After you have entered the REM statement, enter POKE 16510,0 in the direct mode. This POKEs the first line number to 0 so that it cannot be accidentally edited or deleted. Next, type in this hex loader program:

```
5 LET A=16514
15 FOR X=1 TO LEN A$-1 STEP 2
2Ø POKE A, 16* CODE A$(X)+ CODE A$
   (X+1)-476
25 LET A=A+1
3Ø NEXT X
```

Now comes the tricky part. Type in 10 LET A$ = " and the hex listing of Program 1 (Upscroll) (e.g., 10 LET A$ = "2A0C40…10FBC9"), ending with quotation marks. Be extremely careful here, since any mistake could crash the computer. *Note:* There are no spaces in line 10; the hex codes are simply typed in one after the other.

After you have entered line 10 and double-checked for errors, run the program. When it has finished, enter "PRINT A" in direct mode. This should return a value of 16538. If it does not, reenter line 10 and rerun the program.

The variable "A" is simply a pointer to the address at which the next block of code is to be assembled. Therefore, to enter the next block into your computer you must change line 5 to 5 LET A = 16538. Then enter the hex listing for Program 2 (Downscroll) into line 10 and rerun the program. This time, A should be equal to 16567.

Change line 5 to 5 LET A = 16567 and enter the listing for Program 3 (Rightscroll) into line 10. Run the program and check that A is 16588. Change line 5 again to 5 LET A = 16588 and enter the code for Program 4 (Leftscroll). Run the program; A should now equal 16614.

Test Program

Now that the machine language has been entered, lines 5 through 30 can be deleted and a short test program can be run. Enter the test program shown below and run it:

```
5 INPUT A$
1Ø LET L= LEN A$
15 IF L>31 THEN GOTO 5
```

```
20 PRINT AT 11,(31-L)/2;A$
25 IF INKEY$="5" THEN RAND USR 16588
30 IF INKEY$="6" THEN RAND USR 16538
35 IF INKEY$="7" THEN RAND USR 16514
40 IF INKEY$="8" THEN RAND USR 16567
45 GOTO 25
```

The program will wait for an input, so enter a string of less than 32 characters. The string will be printed in the middle of the screen. You should be able to move it around using the arrow keys (5-8).

This program was written as a simple test; the scrolls could certainly be put to better use in programs for games, word processors, or anything else you might come up with. I have used variations of these scrolls in a machine language arcade-type game I am working on and have found that they work quite well. If you want to convert the code listings into decimal, but are not familiar with hexadecimal, the codes from 00 to FF are given beside their decimal equivalent in Appendix A (p. 137) of the *Sinclair BASIC Manual*.

Let's take a brief overview of the techniques used in these scroll routines. In the horizontal scrolls, everything is shifted up or down one byte, and the first/last column is filled with spaces. A POKE can be entered so that the empty column is filled with a different character. The accumulator is used as a temporary store variable, as in a bubble sort.

In the vertical scrolls, the main instruction is ldir/lddr. With this instruction, the hl register pair is loaded with the start address of the block to be transferred. The de register pair is loaded with the address the block is to be moved to, and the bc register pair is loaded with the length of the block. With these scrolls, everything is shifted up/down 33 bytes (the length of one line + one byte for the "enter" character). The empty line is filled with spaces, but a POKE can also be used here to fill the empty line with another character.

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

Program 1: Upscroll

```
      2A 0C 40   ld hl,(400C)
      23         inc hl
      E5         push hl
      11 21 00   ld de,0021
      19         add hl,de
      D1         pop de
      01 B5 02   ld bc,02B5
      ED B0      ldir
      EB         ex de,hl
      06 20      ld b,20
Blnk  36 00      ld (hl),00
      23         inc hl
      10 FB      djnz Blnk
      C9         ret
```

Program 2: Downscroll

```
      2A 10 40   ld hl,(4010)
      11 43 00   ld de,0043
      ED 52      sbc hl,de
      E5         push hl
      11 21 00   ld de,0021
      ED 52      sbc hl,de
      D1         pop de
      01 B5 02   ld bc,02B5
      ED B8      lddr
      EB         ex de,hl
      06 20      ld b,20
Blnk  2B         dec hl
      36 00      ld (hl),00
      10 FB      djnz Blnk
      C9         ret
```

Program 3: Rightscroll

```
      2A 0C 40   ld hl,(400C)
      06 16      ld b,16
Lp1   C5         push bc
      06 20      ld b,20
      3E 00      ld a,00
Lp2   23         inc hl
      4F         ld c,a
      7E         ld a,(hl)
      71         ld (hl),c
      10 FA      djnz Lp2
      23         inc hl
      C1         pop bc
      10 F1      djnz Lp1
      C9         ret
```

Program 4: Leftscroll

```
      2A 10 40   ld hl,(4010)
      11 43 00   ld de,0043
      ED 52      sbc hl,de
      06 16      ld b,16
Lp1   C5         push bc
      06 20      ld b,20
      3E 00      ld a,00
Lp2   2B         dec hl
      4F         ld c,a
      7E         ld a,hl
      71         ld (hl),c
      10 FA      djnz Lp2
      2B         dec hl
      C1         pop bc
      10 F1      djnz Lp1
      C9         ret
```

©

Part VI

Commodore 64 Video – A Guided Tour

Jim Butterfield, Associate Editor

This month we explore a fairly advanced technique: split screens on the Commodore 64. It's a new aspect of the computer, combining things we have already learned into a new set of capabilities. We'll demonstrate, via a BASIC program, an amazing visual display.

We'll need to venture into more technical waters now, but with a little effort, we can perform some minor miracles on the Commodore 64 screen. All the limitations we have learned may be set aside with a little creative "cheating." We'll have to venture into machine language; but even if you're not a ML tyro, it's worth knowing that the job can be done.

We have learned a number of limitations, largely based on the idea that the screen can do a lot of things, but only one at a time:

- We can have only one background color, unless we are in multicolor mode; and even in that case, we're restricted to our choice of colors.
- We can obtain information only from one 16K memory quadrant.
- We can use only one character set.
- We can be in character mode or bit map (hi-res) mode, but not both.
- We may have only eight sprites on the screen at one time.

In fact, we have a more general set of rules. We may be in only one mode at a time – multicolor is either on or off; extended color is either on or off; and so on. It seems impossible to mix screen modes and have the best of both worlds, but we can do it.

Here's the trick: the "Raster Register," address D012 together with the high bit of D011, can do more than tell us where the screen is being painted at this instant. We may store an "interrupt" value there, and tell the computer: "Advise me when you get to this part of the screen." At this point, we can switch screen characteristics: color mode, high resolution, background color, character set, memory bank – whatever you want. Of course, we need to put it all back when we return to the top of the screen.

The Task

We're going to write a quick program to split the screen into two parts, each with a different characteristic. It won't be perfect; we're just trying to show the technique, not to polish up all the loose ends. The fine points will come later. First, let's plan.

If we set a new "interrupt" into our machine, we'll need to make some careful distinctions. First, when an interrupt happens, we must establish: who caused this one? Was it the raster, or the traditional interrupt source of 1/60 second timing? Second, if it was a raster, which part of the screen is involved – the top or the "switch" point?

The Interrupts

Let's start to lay out the machine language program. All interrupts will come here, and we'll need to sort them out. We'll put the program into the cassette buffer.

```
033C  AD  19  D0  INT    LDA  $D019
033F  29  01             AND  #$01
0341  F0  19             BEQ  REGULR
```

The interrupt has happened and has come here. Check the Raster Interrupt Bit in D019 – was this one caused by the raster? We'll need to mask out the bit we want with an AND. If we get nothing, it's a regular interrupt – go there.

```
0343  8D  19  D0         STA  $D019
```

It is indeed a raster interrupt, and we must shut off the alarm. We do this by storing the bit back where it came from (there's a 1 in the A register right now). Amazingly, this turns the bit off.

```
0346  A2  92              LDX  #$92
0348  A0  15              LDY  #$15
```

We'll prepare the registers, assuming we are doing the top-of-screen work. The hex 92 is decimal 146 – the scan line that hits about mid-screen; that's where we will want the next interrupt to take place. Note that hex 92 is considered a "negative" byte; we will use this fact in just a moment. Now, let's see if we are correct about being at mid-screen:

```
034A  AD  12  D0          LDA  #$D012
034D  10  04              BPL  MID
```

We look at the raster scan. If it's less than 127, we're near the top of the screen, and we don't see the "negative" byte. So we skip ahead. If, however, we are at the middle of the screen, we'll see a "negative" value. We won't branch; instead, we'll fix up the registers for mid-screen work:

```
034F  A2  01              LDX  #$01
0351  A0  17              LDY  #$17
```

Both streams join again at this point. X contains the raster location where we will want the next interrupt: if we're at the top, we want to be interrupted at the middle (hex 92); if we're at the middle, we will want to be interrupted at the top (hex 01). Y contains information on the character set we want to choose: graphics or text. Let's proceed:

```
0353  8E  12  D0  MID     STX  $D012
```

Place the next interrupt point into the raster register. The next interrupt will now hit at the right time.

```
0356  8C  18  D0          STY  $D018
```

Place the "character set" value – hex 15 for graphics, hex 17 for text – into the appropriate register.

```
0359  4C  BC  FE          JMP  $FEBC
```

We've done our job. We may now exit. Don't give an RTI; instead, go to a routine that cleans things up nicely, at FEBC. And what of our regular interrupt?

```
035C  4C  31  EA  REGULR  JMP  $EA31
```

It goes to the normal address, to which regular interrupts go.

We have more to do after we get this program into memory. We must also detour the interrupt vector to our new program, and fire up the raster interrupt control.

Back To BASIC

Ready to put all this in BASIC? Here we go:

```
100 FOR J=828 TO 862:READ X
110 T=T+X:POKE J,X
120 NEXT J
130 IF T<>3958 THEN STOP
200 DATA 173,25,208,41,1,240,25,141,25,208,162,146,
     160,21,173,18
210 DATA 208,16,4,162,1,160,23,142,18,208,140,24,208,
     76,188,254,76,49,234
300 POKE 56333,127
310 POKE 788,60:POKE 789,3
320 POKE 56333,129:POKE 53274,129
```

Let's look at the last three lines. Line 300 kills the interrupt for a moment, so that we can mess with the interrupt vector without running into disaster. Line 310 changes the interrupt vector to point at our newly POKEd program. Line 320 restores the interrupt, and adds an extra one: the raster interrupt.

An Amazing Split

When the program is run, an amazing thing happens: the screen becomes graphic at the top and text at the bottom. Impossible, you say? Not for us clever – and careful – people. The effect is permanent: you may NEW the program and start something else and the split screen will still be there.

You shouldn't use cassette tape with this program in place – it's there in the buffer. And you may find that LOAD and SAVE don't work quite right. RUN-STOP/RESTORE will put everything back to its former state.

The Unsolved Problem

But it's not perfect (I warned you). Every once in a while, the barrier seems to creep slightly, and then correct itself. Maybe it's computer hiccups. It seems worse when you are using the keyboard. What's happening? And how can we fix it? Stay tuned.

Atari Artifacting

Judson Pewther

These tools for exploring artifacting can create some of the most beautiful graphics you've ever seen from your Atari.

Even if you are not already familiar with the phrase *television artifacts*, you have probably noticed that the colors of points and lines drawn in Atari's graphics mode 8 are not always what they are supposed to be. (False colors may also appear in graphics mode 0.) These comments apply unless you are using a high resolution color monitor with digital input.

Although the *BASIC Reference Manual* claims that only "one color and two different luminances" are available in GRAPHICS 8, in actual fact six distinguishable color/luminance combinations are possible because of TV artifacting.

While the *BASIC Reference Manual* does not mention this very interesting fact, it is fully documented in *De Re Atari*, Appendix IV, which gives the definition: "The term TV artifacts refers to a spot or 'pixel' on the screen that displays a different color than the one assigned to it." And as further explained, TV artifacts are caused by the way in which color and luminance information is modulated onto an NTSC television signal.

Let's summarize the effects of artifacting in GRAPHICS 8:

1. The effect is maximized by plotting a light color (high luminance) on a dark background, or dense dark patterns on a light background.

2. The color of a pixel is not affected by its Y-coordinate.

3. The color displayed by a pixel depends not only on its assigned color, but also on whether its X-coordinate is even or odd, and on the color assigned to its horizontal neighbors.

4. Horizontal resolution has a practical limit of 160 rather than 320. Thus, two horizontally contiguous pixels tend to form a single pixel of uniform color.

What colors are actually produced? This can depend on the particular TV monitor being used, and on the exact setting of its controls. The setting of the tint control will make the biggest difference.

The major effects of plotting white (the assigned color) pixels on a black background are summarized in the following table. N is the number of horizontally contiguous white pixels. X is the X-coordinate(s) of these pixels in terms of "even" and "odd." COLOR is the approximate actual color displayed by these pixels, assuming normal settings on the TV monitor.

N	X	Color
1	even	green
1	odd	blue
2	even-odd	orange
2	odd-even	light blue
3	—	nearly white
4	—	white

The Table Illustrated

The short program below illustrates artifacts by drawing two series of nearly vertical white lines on a black background. Colored horizontal bands are produced in accordance with the rules in the previous table. No actual white is produced in this example, because there are at most only pairs of horizontally contiguous "white" pixels. Notice in particular that the solid-color bands are created either because all the "even" pixels give a solid green, or all the "odd" pixels give a solid blue.

Lines 199 to 250 can be added to allow the user to easily step the assigned hue through all 16 possibilities, while preserving the 0 luminance setting for the background and the 14 luminance setting for the plotted lines. The background color may be nearly invisible because it is at 0 luminance, but the colors in the horizontal bands will change greatly. Remember that in GRAPHICS 8 the hue associated with the COLOR 1 statement and with the lines that were drawn is the background hue as determined in the SETCOLOR 2,hue,0 statement. Even when we are not seeing the assigned hue because of TV artifacting, changing the assigned hue changes the displayed hues.

Best results are obtained by adjusting TV brightness and contrast to a low or minimum value. TV color may be boosted somewhat, but too much boost blurs the picture. However, the tint control may be adjusted freely from one extreme to the other to vary the colors. These comments apply generally to any program where TV artifacts are used.

Program 1: TV Artifacts

```
10 GRAPHICS 8:COLOR 1
20 SETCOLOR 1,0,14:SETCOLOR 2,0,0
30 FOR X=0 TO 318 STEP 4
40 PLOT X,0:DRAWTO X+1,159:NEXT X
50 FOR X=0 TO 308 STEP 4
60 PLOT X,0:DRAWTO X+9,159:NEXT X
199 REM *** CHANGE HUE ***
200 H=0:OPEN #1,4,0,"K:"
210 ? :? "ASSIGNED HUE IS NOW ";H
220 ? "HIT H KEY TO CHANGE HUE"
230 GET #1,X:IF X<>72 THEN 230
240 H=H+1:IF H=16 THEN H=0
250 SETCOLOR 2,H,0:GOTO 210
```

TV artifacts are really a failure of resolution, but a very interesting failure. And the colors produced can add dazzle to graphics art programs. Although these false colors may at times be annoying, and although the failure in horizontal resolution is certainly an annoyance, TV artifacts compensate considerably for the fact that only two intensities of a single color are officially available in GRAPHICS 8.

Moiré Patterns

Program 2 is a graphics art program which relies on artifacts for its beauty. It also makes use of a technique for creating enhanced moiré patterns.

You are probably already familiar with the simple type of moiré pattern produced by a program like the following:

Program 2: Simple Moiré

```
10 GRAPHICS 24
20 SETCOLOR 1,0,14:SETCOLOR 2,0,0
30 FOR X=0 TO 318 STEP 3
40 COLOR 1:PLOT 159,0:DRAWTO X,191
60 NEXT X
70 GOTO 70
```

To see a somewhat different moiré pattern with a more uniform distribution of light and enhanced contrasts in the details, add the following line and run the program again:

```
50 COLOR 0:PLOT 159,0:DRAWTO X+1,191
```

This program step draws a black line which cancels out half (or more) of the "white" pixels which were plotted in the previous step, line 40. This basic idea is varied and elaborated in Program 3: Pyramid.

Program 3 is designed so that slow typists (like myself) will not have to type in the whole thing just to see what it does. The first half of the program (lines 100 through 470) is almost entirely for the purpose of letting the user control the parameters of the pattern in order to see better how the various effects are achieved. To eliminate some typing, replace the first half of the program with the single line: 100 GRAPHICS 24. Then begin typing at line 500.

The program is essentially self-explanatory, but it might be worthwhile to point out a few things. Lines 500 to 540 select a set of random parameters for the pattern that is about to be drawn. WHITE and BLACK are associated with the subroutine for drawing a set of vertical lines at line 1000 in the program. They are dual purpose variables: if equal to 0 or 1, then a set of "even" or "odd" lines will be drawn, but if greater than 1 the subroutine will not be called. So, the probability is .25 that WHITE will call the subroutine, since it is a random integer ranging from 0 to 7. The same applies to BLACK.

WHITE lays down a colored background for the pattern, but has a slightly different effect if the old pattern has not been wiped out by line 730. BLACK erases all colors in the pattern except for black and another color, just before the program recycles to select a new set of random parameters.

Line 535 works in conjunction with line 740 to insure that the new values of MODE, APEX, and SPACE are not exactly the same as the old values.

Line 550 prevents the attract mode from setting in as long as the program continues to recycle through new variations.

Except for the user option to hold a pattern indefinitely (lines 450 and 720), there are no forced time delays. It takes about a minute for the program to make one cycle, which should be more than enough time to observe a variation of the pattern. If you wish to freeze a particular pattern, program execution may be stopped and restarted by hitting CTRL 1.

Finally, although the program isn't especially fast, I think you will find that many of its variations are as spectacular as anything you have yet seen on your Atari.

Program 3: Pyramid

```
100 GRAPHICS 0:POSITION 9,2:? "*** T
    HE PYRAMID ***"
112 ? "ADJUST TV CONTRAST AND"
113 ? "BRIGHTNESS TO MINIMUM.":?
114 ? "ADJUST TV COLOR AND TINT"
115 ? "TO SUIT INDIVIDUAL TASTE.":?
120 ? "YOUR CHOICE:"
130 ? "   (0) RANDOM PARAMETERS"
140 ? "   (1) USER CONTROLLED PARAMETERS"
150 INPUT CHOICE:IF CHOICE=0 THEN GR
    APHICS 24:GOTO 500
160 IF CHOICE<>1 THEN 150
```

```
170 ? :? "DRAW BACKGROUND OF VERTICA
L LINES?"
180 ? "   (0) EVEN LINES"
190 ? "   (1) ODD LINES"
200 ? "   (2) NO BACKGROUND"
210 INPUT WHITE:IF WHITE=2 THEN 230
220 IF WHITE<>0 AND WHITE<>1 THEN 210
230 ? :? "DRAW PYRAMID IN MODE"
240 ? "   (0) FROM THE CENTER OUTWARD"
250 ? "   (1) FROM LEFT TO RIGHT"
260 INPUT MODE
270 IF MODE<>0 AND MODE<>1 THEN 260
280 ? :? "FOR APEX OF PYRAMID USE"
290 ? "   (0) ONE POINT"
300 ? "   (1) TWO POINTS"
310 INPUT APEX
320 IF APEX<>0 AND APEX<>1 THEN 310
330 ? :? "SPACING OF RAYS FROM APEX?"
340 ? "(USUALLY AN INTEGER: 2 TO 6)"
350 INPUT SPACE:IF SPACE<1 THEN 350
360 ? :? "DRAW BLACK VERTICAL LINES?"
370 ? "   (0) EVEN LINES"
380 ? "   (1) ODD LINES"
390 ? "   (2) NO LINES"
400 INPUT BLACK:IF BLACK=2 THEN 420
410 IF BLACK<>0 AND BLACK<>1 THEN 400
420 ? :? "WHEN FINISHED"
430 ? "   (0) GOTO RANDOM PARAMETERS"
440 ? "   (1) HOLD THE PATTERN"
450 INPUT HOLD
460 IF HOLD<>0 AND HOLD<>1 THEN 450
470 IF CHOICE=1 THEN GRAPHICS 24:GOT
O 600
498 REM *** RANDOM PARAMETERS***
500 WHITE=INT(RND(0)*8)
510 MODE=INT(RND(0)*2)
520 APEX=INT(RND(0)*2)
530 SPACE=2+INT(RND(0)*5)
535 IF MODE=M AND APEX=A AND SPACE=S
    THEN 510
540 BLACK=INT(RND(0)*8)
550 POKE 77,0:REM RESET ATTRACT MODE
598 REM *** PROGRAM EXECUTION ***
600 SETCOLOR 2,0,0:SETCOLOR 1,0,14
620 IF WHITE>1 THEN 640
630 COLOR 1:B=WHITE:GOSUB 1000
640 IF MODE=0 THEN GOSUB 2000
650 IF MODE=1 THEN GOSUB 3000
660 REM *** DRAWTO SIDES ***
670 FOR K=191 TO 1 STEP -SPACE
680 COLOR 1:PLOT 159,APEX:DRAWTO 318
,K:PLOT 159,APEX:DRAWTO 0,K
690 COLOR 0:PLOT 159,0:DRAWTO 318,K-
    1:PLOT 159,0:DRAWTO 0,K-1:NEXT K
700 IF BLACK>1 THEN 720
710 COLOR 0:B=BLACK:GOSUB 1000
720 IF HOLD=1 THEN 720
730 IF RND(0)<0.2 THEN GRAPHICS 24
740 M=MODE:A=APEX:S=SPACE
750 GOTO 500
999 REM *** SUBROUTINE VERTICAL LINES
1000 FOR K=B TO 319 STEP 2
1010 PLOT K,0:DRAWTO K,191:NEXT K
1020 RETURN
1999 REM *** SUB STARTS FROM CENTER
2000 FOR K=0 TO 158 STEP SPACE
2010 COLOR 1:PLOT 159,0:DRAWTO 159+K
     ,191:PLOT 159,0:DRAWTO 159-K,191
2020 COLOR 0:PLOT 159,APEX:DRAWTO 16
     0+K,191:PLOT 159,APEX:DRAWTO 15
     8-K,191
2030 NEXT K:RETURN
2999 REM *** SUB DRAWS LEFT TO RIGHT
3000 FOR K=0 TO 318 STEP SPACE
3010 COLOR 1:PLOT 159,APEX:DRAWTO K,191
3020 COLOR 0:PLOT 159,0:DRAWTO K+1,1
     91:NEXT K:RETURN
```

©

All About The Commodore USR Command

John L. Darling

Have you wondered how to use BASIC's USR command? This article explores this useful feature with examples for every Commodore computer.

This introduction to the USR function will form a basis for more complex applications. We'll explore passing double precision integers between BASIC and machine language.

Here's how the USR function works. Both the USR and SYS commands are like the BASIC GOSUB command. Instead of transferring control of a BASIC program to a BASIC subroutine, USR and SYS cause control to go to a machine language subroutine.

But unlike the SYS function, the USR function has the additional capability of transferring numbers or information to or from a machine language subroutine.

The USR command format is n = USR(v), where n is any variable name, and v is any variable whose value is to be transferred into the machine language subroutine. Upon return to BASIC, the machine language subroutine will place the newly computed value into the variable n. The transfer of values and information between BASIC and machine language is accomplished via the floating-point accumulator (FAC). The FAC is five consecutive bytes in memory that are used for storing floating-point numbers (numbers which can have decimal points in them). Address information for the machine language subroutine is specified in locations 1 and 2 (785 and 786 in the Commodore 64), and is stored in standard LBHB (Low Byte, High Byte) format.

For example, the command nv = USR(ov) in a BASIC program would first transfer the value of *ov* to the FAC. Then the program branches to the machine language subroutine whose starting address is stored in locations 1 and 2 (785 and 786 in the 64). Before leaving the machine language subroutine, the programmer stores the newly computed value in the FAC, and issues an RTS command (ReTurn from Subroutine). Upon reentering BASIC, the value in the FAC is held by the variable *nv*, and the BASIC program continues from where it left off.

The only thing preventing you from taking full advantage of the USR function is the conversion of the floating point data in the FAC. The number must first be in a format your machine language program can use, and then the computed value must be reformatted back into the FAC for the return trip to your BASIC program. The secret lies in knowing the location of two important subroutines in the ROM code. One of these subroutines converts the contents of the FAC into a double precision integer stored at \$61 and \$62 on the PET, or \$64 and \$65 on the VIC and 64. The second subroutine converts the double precision integer in the A and Y registers to a floating point number in the FAC.

Helpful Subroutines

These two subroutines are easy to use. *Note:* These examples are for PET Upgrade BASIC. Refer to the table reference for the appropriate locations on your computer.

1. Convert the FAC to a double precision integer.

```
JSR $DBA7 ;CONVERT FAC TO AN INTEGER IN
               $61 AND $62
LDA $61    ;MSB OF INTEGER
LDY $62    ;LSB OF INTEGER
```

The A and Y registers contain the converted integer value of T in the BASIC equation, S = USR(T). It can be saved and used by your machine language program when needed.

When you need to transfer a value from the machine language program back to BASIC, the

double precision integer must be placed in the FAC in the proper format. The following code will accomplish this nicely.

2. Convert a double precision integer in A, Y to FAC.

```
LDA  $...    ;MSB OF INTEGER
LDY  $...    ;LSB OF INTEGER
JSR  $D26D   ;CONVERT INTEGER IN A,Y TO FAC
RTS          ;RETURN TO BASIC
```

In the equation, S = USR(T), S will contain the converted value provided by the machine language program. These two ROM subroutines should be sufficient for most user applications.

To illustrate the use of these subroutines, let's use the USR function to simulate the PEEK instruction. This allows us to evaluate the subroutines by comparing the test results with the results of a known instruction. The particular example chosen is useful only as a learning tool.

Simulated PEEK

If you wish to try the example on your computer, use the following procedure:

For PET/CBM with Upgrade or 4.0 BASIC:

1. Access the Monitor by SYS 1024.

2. Display the proper block of memory by typing .M 033C 0352.

3. Type in the code from Program 1 or 2 as appropriate for your PET. Do a monitor save on it by typing .S"@0:USR.M",08,033C,0352 for disk or .S"@0:USR.M",01,033C,0352 for tape.

4. Enter the BASIC Program 5 and SAVE it as USR.B.

5. RUN the BASIC program.

The only reason for saving these programs is to avoid reentering the data in case of an unrecoverable crash due to a typing error.

If it is necessary to load the programs again, always load the machine language program first, and then the BASIC program. Otherwise, the BASIC address pointers will be incorrect and your program will not run properly.

For the VIC-20 and Commodore 64:

1. Type in Program 3 or 4 as appropriate, then add the lines from Program 5 (64 owners note line 120). The numbers in the DATA statements comprise a machine language program and *must* be typed in correctly.

2. Before RUNning the program, SAVE it to tape or disk.

3. RUN the program.

If you are curious about what the machine language does, here is a disassembly of the Upgrade BASIC version.

```
$033C  20 A7 DB  START JSR  $DBA7    ;FAC TO INTEGER
                                      IN $61, $62
$033F  A5 61           LDA  $61      ;T-MSB
$0341  85 FC           STA  $FC      ;TEMP SAVE MSB
$0343  A5 62           LDA  $62      ;T-LSB
$0345  85 FB           STA  $FB      ;TEMP SAVE LSB
$0347  A0 00           LDY  #$00     ;INDIR INDEX
$0349  B1 FB           LDA  ($FB),Y  ;DATA AT $FB,FC
                                      ADDR
$034B  A8              TAY           ;S-LSB
$034C  A9 00           LDA  #$00     ;S-MSB = 0
$034E  20 6D D2        JSR  $D26D    ;CONVERT A,Y
                                      TO FAC
$0351  60              RTS           ;RETURN TO
                                      BASIC
$0352                  .END
```

Making USR Work

Now let's look at all seven steps that are necessary to make the USR function work. These steps, as used in the example, will be explained in detail for the Upgrade BASIC version. Again, you can refer to the table for the appropriate locations on your particular computer.

• STEP 1: POKE the USR Jump Address in Locations 1 and 2.

The first step is to put the machine language start address in locations $01 and $02. The least significant address byte (LSB) is stored in location 1, and the most significant address byte (MSB) is stored in location 2. In the example, the machine language start address is located at $033C. Therefore, locations $00 through $02 should contain the following 6502 code:

```
ADDR    CODE       MNEMONIC
$0000   4C 3C 03   JMP  $033C
```

In BASIC, address values must be stated in decimal. The conversion for the LSB address byte $3C is 3*16 + 12 = 60 and the MSB address byte $03 is 0*16 + 3 = 3, providing the POKEd values in line 110. It is not necessary to POKE location 0 ($310 on the 64) as it was initialized to the proper value when power was applied.

• STEP 2: Determine T in S = USR(T).

In your BASIC program, establish a value for T in the equation S = USR(T). To use the ROM subroutines provided in this article, T must be an integer with a value between 0 and 65535. This is the full range of 2-byte integers and in hex is equivalent to $0000 through $FFFF. In the example, lines 130 - 150 are used to input and test an integer number between 0 and 65535.

• STEP 3: Execute S = USR(T).

When S = USR(T) is executed, it is equivalent to SYS(828). In both cases, control is transferred from BASIC to your machine language program. The USR function differs from the SYS command in that the FAC can be used to pass real data to and from the machine language program. Note that 828 = Loc 1 + 256 x Loc 2 (828 = 60 + 3 x 256).

• STEP 4: Convert FAC to Positive Integer.

While in your machine language program, convert the value in the FAC to a two-byte integer. The JSR to $DBA7 at $033C converts the FAC to a double precision integer whose MSB is located in $61 and LSB is located in $62.

• STEP 5: Execute ML Program.

In the example, the instructions located from $033F - $034A are used to get the value that we wish to pass to BASIC.

• STEP 6: Convert Positive Integer to FAC and Exit.

The instructions in $034B through $0351 put the integer value in the "A" and "Y" registers. Since the simulated PEEK value is really only a single precision integer, the MSB is set to zero. The JSR to $D26D will convert the values in the A and Y registers to floating point and place them in the FAC. Finally, the RTS will return control to BASIC.

• STEP 7: Verify S in S=USR(T).

The real variable S, in the equation S=USR(T), will now be assigned the value placed in the FAC by your machine language program. Lines 170 and 180 in Program 5 display both the value of S and the actual PEEK value to verify that the simulation is correct.

That's all there is to it. When you break it down, step by step, it's not that difficult. Perhaps the USR function will now find a place in your programming arsenal.

Floating Accumulator Locations For PET/CBM, VIC, And 64

Computer	Loc. of FAC to Integer Routine	Result Left in Location	Loc. of Integer to FAC Routine	USR Vector Location
VIC	$DC9B	$64,$65	$D391	$01,$02
64	$BC9B	$64,$65	$B391	$311,$312
Upgrade PET	$DBA7	$61,$62	$D26D	$01,$02
4.0 PET	$CDD1	$61,$62	$C4BC	$01,$02

Program 1: PET Upgrade BASIC Version

```
033C 20 A7 DB A5 61 85 FC A5
0344 62 85 FB A0 00 B1 FB A8
034C A9 00 20 6D D2 60 00 FF
```

Program 2: PET 4.0 BASIC Version

```
033C 20 D1 CD A5 61 85 FC A5
0344 62 85 FB A0 00 B1 FB A8
034C A9 00 20 BC C4 60 00 FF
```

Program 3: VIC-20 Version

```
10 FOR A=828 TO 849:READ D:POKE A,D:NEXT
20 DATA 32,155,220,165,100,133,252,165
30 DATA 101,133,251,160,0,177,251,168
40 DATA 169,0,32,145,211,96
```

An Explanation Of LBHB

The Low Byte, High Byte (LBHB) data storage format is a method many microcomputers use to store large numbers. Because a byte can hold a number no larger than 255, two or more consecutive bytes are needed to represent numbers larger than 255. The LBHB format involves a method in which numbers are broken down, then stored in memory with the least significant byte (LSB) first, followed by the most significant byte (MSB).

A number between 256 and 65535 is stored in RAM memory using two consecutive bytes. The *second* byte (the most *significant* byte) is derived by dividing the original number by 256, and then storing the *integer* (no fractions) value into the MSB. The remainder of this division is then stored in the first or *least significant* byte. Thus you use the following formula for reading LBHB numbers in memory:

number = LSB + (MSB * 256)

For example, let's say that you wanted to USR to address 828 (the cassette buffer in most Commodore machines). You would need to put 828 into addresses 1 and 2 and it would have to be in this LBHB format.

Here's how it's done:

1) Divide 828 by 256 and store the resulting integer byte 2.

828 / 256 = integer 3

2) Store the remainder of this division in byte one:

828 - (256 * 3) = 60

The Two Methods

To automatically store numbers into and read numbers from memory using the LBHB format, use these two formulas:

To read a LBHB number, where N = number:

N = byte 1 + (256 * byte 2)

To store a LBHB number, where N = number to be stored:

NN = INT(N / 256): POKE byte 1, N - (NN * 256): POKE byte 2, NN

BEGINNING PROGRAMMERS

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

Program 4: Commodore 64 Version

```
10 FOR A=828 TO 849:READ D:POKE A,D:NEXT
20 DATA 32,155,188,165,100,133,252,165
30 DATA 101,133,251,160,0,177,251,168
40 DATA 169,0,32,145,179,96
```

Program 5: USR Demonstration

```
100 REM SAVE BEFORE RUNNING
110 POKE 1,60:POKE 2,3:REM JMP $033C
120 REM FOR C-64, USE POKE 785,60:POKE 786
    ,3
130 PRINT "SIMULATED PEEK":PRINT
140 PRINT "INPUT AN ADDRESS BETWEEN 0 AND ~
    65535"
150 INPUT T:IF T<0 OR T>65535 OR INT(T)<>T
     THEN 140
160 S=USR(T):REM SYS 828 ($003C)
170 PRINT S"= PEEK("T")",PEEK(T):PRINT
180 GOTO 140
```

©

Commodore Programmer's Alarm Clock

Bruce Jaeger

You'll appreciate this program if you've ever lost track of time while at your computer. It will act as a countdown timer and print "QUIT!" on screen and sound a bell when the time comes to stop. For VIC, 64, and PET/CBMs.

Have you ever sat down at your computer after dinner to "just touch up that program a bit," only to find again that you've lost all notion of time and you've just missed the first half of that movie you've waited for all week? Or you're supposed to pick someone up at 6:00, and by the time you look up from the screen it's 7:30? Me too!

That's why "Programmer's Alarm Clock" came about. When you first sit down at your desk, load and run the program. It will ask you for the alarm time, and for the current time of day.

That's all. You can run games, develop programs, write computer articles, whatever. But when the alarm time comes, the word "QUIT!" comes up on the screen and there's a healthy beep from the CB2 speaker.

Programmer's Alarm is a machine language routine located in the second cassette buffer, and is accessed 60 times a second by the interrupt routine that updates TI$ and does other housekeeping chores. The program merely compares the previously stored alarm time with the time-of-day, and lets you know when they match.

The program as written is for the PET/CBM Upgrade ROM set, 4.0 ROMs, 64, and VIC.

Program 1:
Alarm Clock – Upgrade BASIC PET Version

```
160 REM *** "ALARM FOR 3.0 PETS"
    ***
170 GOSUB280 :{3 SPACES}REM LOAD MACHINE
    LANGUAGE
180 T=141 :{6 SPACES}REM TIMER LOCATION,
    3.0 ROMS
190 PRINT"{CLR}SET ALARM TIME"
200 PRINT"{DOWN}( HHMMSS )"
210 INPUT "{DOWN}{2 SPACES}000000
    {8 LEFT}";TI$
220 H=PEEK(T):L=PEEK(T+1)
230 POKE 1022,H:POKE1023,L
240 PRINT"{DOWN}INPUT TIME OF DAY."
250 PRINT"{DOWN}( HHMMSS )"
260 INPUT "{DOWN}{2 SPACES}000000
    {8 LEFT}";TI$
270 PRINT"{CLR}":SYS826:END
280 FORX9=0TO 115 :READX8:POKE826+X9,X8:
    NEXTX9:RETURN
290 DATA 120,165,144,141,172,3,165,145,1
    41,173,3,169,79,133,144,169,3
300 DATA 133,145,88,96,165,141,205,254,3
    ,208,83,165,142,205,255,3,208
310 DATA 76,169,145,141,35,128,169,149,1
    41,36,128,169,137,141,37,128,169
320 DATA 148,141,38,128,169,161,141,39,1
    28,169,16,141,75,232,169,15,141
330 DATA 74,232,169,150,141,72,232,160,2
    55,162,255,136,240,6,202,208,253
340 DATA 76,135,3,169,0,141,75,232,141,7
    2,232,141,46,230,120,173,172
350 DATA 3,133,144,173,173,3,133,145,88,
    76,46,230,46,230
```

```
UNDERLINE = SHIFT,
E 3 = COMMODORE KEY,
{ }= SPECIAL.
REFER TO LISTING CONVENTIONS
```

Program 2: **Alarm Clock – 4.0 PET Version**

```
160 REM ** ALARM FOR 4.0 PETS **
170 GOSUB280 :{3 SPACES}REM LOAD MACHINE
    LANGUAGE
180 T=141 :{6 SPACES}REM TIMER LOCATION,
    4.0 ROMS
190 PRINT"{CLR}SET ALARM TIME"
```

Notes For Commodore Alarm Clock

Since Commodore provides a realtime clock in all of its computers, the versions presented here differ only in the location of the machine language routine and the location of the interrupt request vectors which continually check the internal clock. Since the internal clock is affected by using the cassette, the VIC and 64 versions of this program will give unpredictable results if you use the cassette unit. Disk operation and TOOLKIT seem unaffected.

Also, if the alarm time is set for after 12:00 and you set the time of day to a point before 12:00, then you must use military time (1300 for one o'clock, etc.). Otherwise, the two times will not match, and the alarm will not sound.

This program is a good one to study if you are interested in learning about simple machine language and interrupt-driven routines. Since the program is so short, it is fairly simple to understand and adapt for use in other programs.

```
200 PRINT"{DOWN}( HHMMSS )"
210 INPUT "{DOWN}{2 SPACES}ØØØØØØ
    {8 LEFT}";TI$
220 H=PEEK(T):L=PEEK(T+1)
230 POKE 1Ø22,H:POKE1Ø23,L
240 PRINT"{DOWN}INPUT TIME OF DAY."
250 PRINT"{DOWN}( HHMMSS )"
260 INPUT "{DOWN}{2 SPACES}ØØØØØØ
    {8 LEFT}";TI$
270 PRINT"{CLR}":SYS826:END
280 FORX9=ØTO 115 :READX8:POKE826+X9,X8:
    NEXTX9:RETURN
290 DATA 120,165,144,141,172,3,165,145,1
    41,173,3,169,79,133,144,169,3
300 DATA 133,145,88,96,165,141,205,254,3
    ,208,83,165,142,205,255,3,208
310 DATA 76,169,145,141,35,128,169,149,1
    41,36,128,169,137,141,37,128,169
320 DATA 148,141,38,128,169,161,141,39,1
    28,169,16,141,75,232,169,15,141
330 DATA 74,232,169,150,141,72,232,160,2
    55,162,255,136,240,6,202,208,253
340 DATA 76,135,3,169,Ø,141,75,232,141,7
    2,232,141,46,230,120,173,172
350 DATA 3,133,144,173,173,3,133,145,88,
    76,85,228,85,228
```

UNDERLINE = SHIFT,
[ ] = COMMODORE KEY,
{ }= SPECIAL.
REFER TO LISTING CONVENTIONS

Program 3: Alarm Clock – VIC Version

```
80 REM ** ALARM CLOCK FOR VIC **
90 GOSUB195
100 PRINT"{CLR}SET ALARM TIME"
110 PRINT"{DOWN}( HHMMSS )"
120 INPUT"{DOWN}{2 SPACES}ØØØØØØ{8 LEFT}
    ";TI$
130 POKE953,PEEK(16Ø)
140 POKE954,PEEK(161)
150 PRINT"{DOWN}INPUT TIME OF DAY"
160 PRINT"{DOWN}( HHMMSS )"
170 INPUT"{DOWN}{2 SPACES}ØØØØØØ{8 LEFT}
    ";TI$
180 PRINT"{CLR}":SYS826:END
195 FORG=826TO953:READE:POKEG,E:NEXT:RET
    URN
200 DATA 120, 173, 20, 3, 141, 183, 3, 1
    73
210 DATA 21, 3, 141, 184, 3, 169, 83, 14
    1
220 DATA 20, 3, 169, 3, 141, 21, 3, 88
230 DATA 96, 173, 160, Ø, 205, 185, 3, 2
    08
240 DATA 89, 173, 161, Ø, 205, 186, 3, 2
    08
250 DATA 81, 169, 145, 141, 17, 30, 169,
     149
260 DATA 141, 18, 30, 169, 137, 141, 19,
     30
270 DATA 169, 148, 141, 20, 30, 169, 161
    , 141
280 DATA 21, 30, 169, 15, 141, 14, 144,
    169
290 DATA 139, 141, 10, 144, 166, 255, 16
    4, 255
300 DATA 136, 208, 253, 202, 208, 248, 1
    69, Ø
```

```
310 DATA 141, 14, 144, 120, 173, 183, 3,
     141
320 DATA 20, 3, 173, 184, 3, 141, 21, 3
330 DATA 169, 0, 141, 17, 150, 141, 18,
     150
340 DATA 141, 19, 150, 141, 20, 150, 141
     , 21
350 DATA 150, 88, 76, 191, 234, 255, 0,
     255
```

UNDERLINE = SHIFT,
[ ] = COMMODORE KEY,
{ }= SPECIAL.
REFER TO LISTING CONVENTIONS

Program 4: Alarm Clock – 64 Version

```
70 REM ** ALARM CLOCK FOR C-64 **
80 S=54272:FORR=STOS+24:POKER,0:NEXT
95 GOSUB195
100 PRINT"{CLR}SET ALARM TIME"
110 PRINT"{DOWN}( HHMMSS )"
120 INPUT"{DOWN}{2 SPACES}000000{8 LEFT}
     ";TI$
130 POKE956,PEEK(160)
140 POKE957,PEEK(161)
150 PRINT"{DOWN}INPUT TIME OF DAY"
160 PRINT"{DOWN}( HHMMSS )"
170 INPUT"{DOWN}{2 SPACES}000000{8 LEFT}
     ";TI$
180 PRINT"{CLR}":SYS49152:END
195 FORG=49152TO49284:READE:POKEG,E:NEXT
     :RETURN
200 DATA 120, 173, 20, 3, 141, 186, 3, 1
     73, 21, 3, 141
210 DATA 187, 3, 169, 25, 141, 20, 3, 16
     9, 192, 141
220 DATA 21, 3, 88, 96, 173, 160, 0, 205
     , 188, 3
230 DATA 208, 92, 173, 161, 0, 205, 189,
     3, 208, 84
240 DATA 169, 145, 141, 17, 4, 169, 149,
     141, 18, 4
250 DATA 169, 137, 141, 19, 4, 169, 148,
     141, 20, 4
260 DATA 169, 161, 141, 21, 4, 169, 15,
     141, 24, 212
270 DATA 169, 9, 141, 5, 212, 169, 6, 14
     1, 6, 212
280 DATA 169, 34, 141, 1, 212, 169, 70,
     141, 0, 212
290 DATA 169, 33, 141, 4, 212, 169, 255,
     160, 255, 136
300 DATA 208, 253, 202, 208, 248, 169, 0
     , 141, 24, 212
310 DATA 120, 173, 186, 3, 141, 20, 3, 1
     73, 187, 3
320 DATA 141, 21, 3, 88, 76, 49, 234, 13
     4, 223, 32
330 DATA 223, 0, 223, 32, 223, 32, 223,
     32, 223, 0
```

UNDERLINE = SHIFT,
[ ] = COMMODORE KEY,
{ }= SPECIAL.
REFER TO LISTING CONVENTIONS

©

STARS

George Trepal

This short graphics program draws stars – separate or concentric. It is designed for the TRS-80 Color Computer, but its simplicity makes it a candidate for conversion to any machine. You'll learn how to create many types of patterns and also some interesting tricks you can use with other Color Computer programs.

This routine for the TRS Color Computer draws star-like patterns. It's a no-frills program which is easy to convert to other computers.

The stars can have as many points and sides as you want. However, the resolution of the TV sets an upper limit of visibility at about 25. The points used to draw the stars are stored in arrays X and Y; since 25 is the upper limit, these arrays are DIMensioned to 25 in line 10. Lines 20 and 30 simply clear the screen and ask for the number of sides desired. After you've typed in the program, a good number to start with is 17.

Line 40 puts the Color Computer into its highest resolution mode.

Lines 50 to 90 use polar coordinates and the computer draws an imaginary circle. It then finds points that equally divide the circumference into N equal parts. N is the number of sides you input in line 30. The Color Computer is not able to plot points given in polar form, so they have to be converted to rectangular (also called Cartesian) coordinates. Each of the points is stored in the X and Y arrays. If you want to know more, you'll find this discussed in high school algebra books.

The 96s in line 70 are special instructions for the Color Computer. The highest resolution screen is 264 by 192 separate dots, called pixels. Since I like big pictures, I'm telling the computer to take up the whole screen when it draws its circle. A circle with a diameter of more than 192 would be too big for the screen. Half of 192 is 96. In other words, 96 is the image size and the radius of the circle. We'll get back to this in a minute.

Now we have all the coordinates stored in arrays, and the screen is still blank. Lines 100 to 130 draw lines between the dots. In line 120, the 128 and 96 refer to the point (128, 96) which is the center of the screen. That's where we want the center of our circle. All the other points in the arrays are in relation to the circle center.

Line 140 locks the computer in a loop so the program continues running, and the picture stays on the screen.

Now that we have a nice 14-line program that draws pretty pictures, here are a few suggestions for improvements.

Concentric Stars

Remember line 70 with its 96? Instead of 96, let's put in a variable R (for radius). Let's add a line: 15 R=96. If you run the program, there is no change at all. Let's add another line: 135 R=R/2: GOTO 50. Now when the program is run, the machine draws the star as expected. Next it draws another star half as big inside the first star. Then it draws a star half as big as the second star inside the second star, and so on forever or until you press the BREAK key.

Of course, you need not divide R by 2. You could use 1.4 or any other number you like. You don't need to draw an infinite number of concentric stars either, if you set up a counter. A good counter could be made by adding these two lines.

```
35 INPUT "HOW MANY STARS"; HM
95 C=C+1: IF C=HM THEN GOTO 140
```

Multiple Stars

So much for multiple concentric stars. Let's modify line 120. Remember that coordinates 128,96 are the center of the screen. They are also the center of the figure. If you change them, the location of the figure will change. If you duplicate line 120 (call it line 125) with the 96 and 128 reversed, the figure will appear twice on the screen. Maybe you'd like lots of little stars on the screen in different places. You could do this by making lots of duplicates of line 120 with different center coordinates. Or you could store the coordinates in a DATA statement.

To use a DATA statement, first change the 96s in line 120 to DY (for data Y coordinate) and change the 128s to DX. Add line 95 READ DX: READ DY and put your DATA statement wherever you want.

POKE For Speed And Sound

Do you want to make the program draw faster? If the program is running, press the BREAK key. Then carefully type POKE 65495,0, and press the ENTER key. The program will now run twice as fast. This POKE (sometimes called vitamin E, by Color Computer users) doubles the rate of the internal clock, but it will not work with all Color Computers, especially the early models. The drawback to this POKE is that sound routines no longer work, printers print garbage, modems don't work, and you can't CLOAD or CSAVE. To get the computer back to normal you can: (1) POKE 65494,0 and ENTER, (2) press the reset button, or (3) turn off the machine.

If you want to use the magic POKE and have sound with the graphics, the way to do it is POKE 65494,0: SOUND 1,1: POKE 65495,0.

Stars

```
10 DIM X(25),Y(25)
20 CLS
30 INPUT "NUMBER OF SIDES";N
40 PMODE 4,1:PCLS:SCREEN 1,1
50 FOR I=1 TO N
60 A=I*(3.14159/(N/2))
70 X=96*COS(A):Y=96*SIN(A)
80 X(I)=X:Y(I)=Y
90 NEXT I
100 FOR J=1 TO N
110 FOR K=J TO N
120 LINE(X(J)+128,Y(J)+96)-(X
    (K)+128,Y(K)+96),PSET
130 NEXT:NEXT
140 GOTO 140
```
©

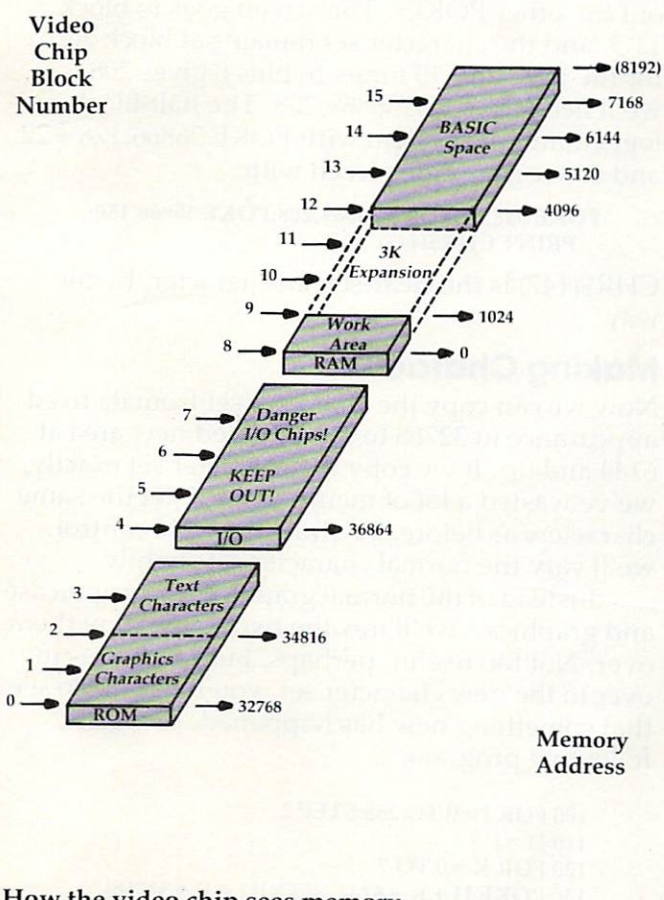
Part III

Visiting The VIC-20 Video

Jim Butterfield, Associate Editor

This month our traveller stakes a claim – but discovers he must first find empty land, and must subsequently register his address.

We know that the VIC-20 video chip gets two things from memory: "screen memory" and "the character set." But it sees the computer's memory in an unusual way:



How the video chip sees memory.

Suppose we want to lay out our own screen and characters. It seems simple enough: choose the locations for screen memory and character set, and POKE the block numbers (screen block times 16 plus character block) into address 36869. If the screen is positioned at an exact block boundary, we put a low number (such as 22) into 36866, otherwise we place a high number there (such as 22 plus 128, or 150). The 22, by the way, is for 22 columns – standard for the VIC.

However, we have two major tasks to perform. First, we must make sure that the memory we are using to feed the video chip isn't needed by somebody else. Second, we must tell the VIC-20 operating system about our new screen location. Changing the video chip isn't enough – the parts of the computer that print to the screen must be told that the screen is somewhere else.

Let's try an example: we'd like to put our own character set into a tiny 5K VIC. Things will get a little crowded, since we need to use 2K for the extra character set. But we can make it work.

Finding Room

Almost all the spare RAM memory of the computer is assigned to BASIC. This is to allow you to write programs as large as possible. We must take memory away from BASIC to make room for the new video stuff.

BASIC memory is a single continuous block. It goes from Start-of-BASIC (whose address is logged in locations 43 and 44) to Limit-of-BASIC (whose address is logged in locations 55 and 56). No breaks: you can't pop a screen in the middle and have BASIC memory skip around it. You can find the Start-of-BASIC address on your machine by typing PRINT PEEK(43) + PEEK(44)*256; or the Limit-of-BASIC address by typing PRINT PEEK (55) + PEEK(56)*256. Remember these; they are a good way to check the values after you've changed things around.

Making Room

We have a choice. We can move down the Limit-of-BASIC, which will give us room at the top. We can move up the Start-of-BASIC, which will make room at the bottom. Or we can do both, if we don't mind the extra work. Whatever we do, we must realize that we're trimming back the area available for BASIC.

If we move down the Limit-of-BASIC, we must say CLR after we do so. This gets rid of variables and strings that might be in embarrassing places. Don't forget this.

Moving the Start-of-BASIC upwards takes a good deal of care. Rule 1: We must POKE a value of zero into the first available location. Rule 2: We must set the Start-of-BASIC pointer so that it points to the next location behind the zero. Rule 3: When we're finished, we must type NEW to make sure that BASIC is cleanly set up in the new memory area.

How do we set up these pointers? Divide the desired address by 256: the remainder goes into the first byte, and the quotient into the second byte. For example: we want to move the Limit-of-BASIC down to 6144. 6144 divided by 256 gives 24 with zero remainder, so we POKE 55,0:POKE 56,24:CLR.

Another example: we want BASIC to start at 5120. First, place the zero: POKE 5120,0. Now, the pointer must be set to 5121 (behind the zero); since 5121 divided by 256 gives 20 with a remainder of 1, we POKE 43,1:POKE 44,20:NEW.

Planning

We want to set up a complete character set, including the reverse characters. That will take 2K of memory – we could do it in 1K if we were willing to skip the reverse characters. Let's plan to put this at the top of memory, starting at block 14.

The screen takes up half a block, of course, and it seems to make sense to set this up just below the characters; so we'll pick block 13.5 (we can set the screen on a half-block boundary, remember?). This calls for a Limit-of-Memory of 5632. You may have noticed, by the way, that the Limit-of-Memory pointer is set one location beyond the last usable value. In other words, BASIC can use 5631, but it can't use 5632, the Limit value.

Arithmetic time. 5632 divided by 256 gives 22 with zero remainder; so type:

```
POKE 55,0:POKE 56,22:CLR
```

and the space is allocated. You can try PRINT FRE(0) and see what a puny amount of memory you have left.

We haven't yet told the video chip to use this area. We're not ready to point the chip towards the new character set area; we haven't put any characters there yet. So let's move characters in – but wait a moment.

The new character set would go over top of the present screen location. This would give us an odd-looking screen. We could live with that part, but the screen would also do odd things like scrolling, which would move the character set we had so carefully placed. We'd better move the screen to a clear area first.

Moving The Screen: Video And System

The character set can remain as block zero for the moment; we'll want to shift the screen to block 13.5, with POKEs to 36869 and 36866. But we need to do two extra things at the same time: tell the computer system where to find the new screen, and clean up the screen area.

The POKEs to 36869 and 36866 tell the video chip all it needs to know about delivering the screen memory to the video output circuits. But unless we tell the computer system about the change, it will continue to put new characters into the old screen area. We tell it with a POKE to location 648. Here's how the arithmetic goes.

Divide the new screen memory address by 256, and POKE the result into address 648. Our example puts the screen at 5632, which gives 22 when divided by 256; so we'll POKE 648,22. But we need to do everything together. Let's work out the other POKEs. The screen goes to block 13.5, and the character set remains at block zero for the moment. 13 times 16 plus 0 gives 208, so we'll need to POKE 36869,208. The half-block is logged into the system with POKE 36866,128 + 22, and so we move the screen with:

```
POKE 648,22:POKE 36869,208:POKE 36866,150:
  PRINT CHR$(147)
```

CHR$(147) is the clear-screen character, by the way.

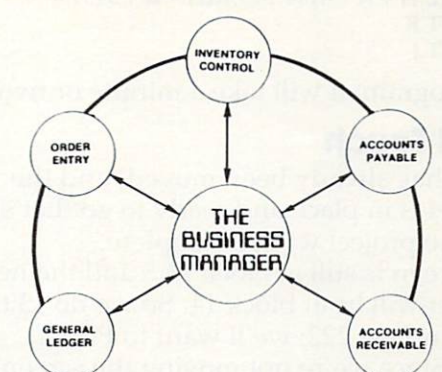
Making Characters

Now we can copy the character set from its fixed appearance in 32768 to our planned new area at 6144 and up. If we copy the character set exactly, we've wasted a lot of memory; we'll get the same characters as before. To show we have control, we'll vary the normal character set slightly.

Instead of the normal graphics set – uppercase and graphics – we'll mix the two as we copy them over. Not too useful, perhaps, but when we cut over to the new character set, you'll be able to see that something new has happened. Enter the following program:

```
100 FOR J = 0 TO 255 STEP 2
110 J1 = J*8
120 FOR K = 0 TO 7
130 POKE J1 + K + 6144,PEEK(J1 + K + 32768)
140 NEXT K
```

THE BUSINESS MANAGER...



FULLY INTERACTIVE ACCOUNTING SYSTEM. NO DISKETTE CHANGING! FINALLY ALL THE FEATURES YOU'VE BEEN WAITING FOR – IN ONE COMPLETE INTERACTIVE SYSTEM.

HARDWARE REQUIREMENTS:

- Commodore Model 8032 or SuperPET or 8096 Micro-computer
- Commodore Model 8050 One Megabyte Dual Disk Drive
- Commodore Model 4022 (or Larger) Printer
- Commodore PET — IEEE cable
- Commodore IEEE — IEEE cable

FULL SOFTWARE SYSTEM

ONLY $995.00

QUESTAR INTERNATIONAL INC.

MANUAL ONLY – $40.00
REFUNDABLE UPON
SYSTEM PURCHASE.

DISKSHARE 9000

WORKS WITH ALL COMBINATIONS PET/CBM 100% HARDWARE. NO SOFTWARE REQUIRED! UP TO 8 USERS SHARE DISK AND/OR PRINTER. CONNECTS IN MINUTES – NO TOOLS REQUIRED. INCREASES PRODUCTIVITY – REDUCES COSTS. IDEAL FOR EDUCATION AND BUSINESS. CABLE LENGTHS UP TO 50 FEET.

ONLY $749.00 (U.S.)
8 USER SYSTEM ONLY $1149

Multi User System

DEALER INQUIRIES WELCOME.

QUESTAR INTERNATIONAL INC.
Suite 102, 7270 Woodbine Avenue
Markham, Ontario Canada L3R 1A4
Tel (416) 475 8044

```
150 FOR K=8 TO 15
160 POKE J1+K+6144,PEEK(J1+K+34816)
170 NEXT K
180 NEXT J
```

Run this program; it will take a minute or two.

The Final Touch

The screen has already been moved, and the character set is in place and ready to go. Let's cut it in, and the project will be complete.

The screen is still at block 13.5 and the new character set will be at block 14. So we do 13 times 16 plus 14 and get 222; we'll want to POKE 36869,222. Since we're not moving the screen this time, the "half-block" value in 36866 is still good, we won't need to change that. We're ready – enter:

```
POKE 36869,222
```

Now try typing or listing the previous program, and look at the odd combination of characters we've created. We must tie things together neatly – BASIC, the Operating system, the video chip – to make it all work properly. But with good planning, we can make the screen do marvelous things.

©

Atari Laser Gunner II

A Vertical Blank Enhancement

Thomas A. Marshall

The improvements to this previously published game – and the author's explanations of the techniques he used – easily justify a second look at Laser Gunner. This version, Laser Gunner II, mixes machine language and BASIC to make Laser Gunner (originally published in November 1982) an even more exciting game. The enhancements include having both missiles on screen simultaneously and smoothing out the animation, even as missiles are fired.

The concept of Laser Gunner (**COMPUTE!**, November 1982) is excellent, but anything can be improved. For example, all other motions stop when the missiles are fired. For continuous and smooth motion, the computer could process the missile horizontal positioning during the vertical blank (VB) period.

The VB is the time during which the television's electron beam is turned off while it returns from the lower right corner of the screen to the top left. Depending on the graphics mode and other interrupts, there are approximately 7980 *machine cycles* available during a single VB. (A machine cycle is the smallest measurement of time on your computer's internal "clock.")

Bringing VB Into The Picture

To utilize the VB, we first have to tell the Operating System (OS) where to go. We do this by performing a Vertical Blank Interrupt (VBI) through the Set Vertical Blank Vector (SETVBV) routine. Before jumping to the SETVBV, we have to load the least significant byte (LSB) in the Y-register and the most significant byte (MSB) in the X-register of our VB machine language routine.

Into the accumulator we can place either a 6 or a 7. Six is for deferred mode; the OS does its housekeeping operations before it executes our code. Seven is for immediate mode; the OS executes our code first during the VB. Since we will be checking the collision registers, we will be loading a 6 into the accumulator. The BASIC program initializes the SETVBV through the USR statement on line 1460. To return control to the OS, we jump back through $E45F.

The BASIC and the machine language (ML) programs interact through several PEEKs and POKEs. The ML program checks the STRIG(0), location $0284, for the press of a button, and moves both missiles horizontally. Since the player/missile graphics are defined in strings, it is easier to have BASIC draw and erase the missiles by PEEKing the flags that the ML program sets.

In the enhanced version, both missiles appear on the screen at the same time. This requires the additional coding located at $06D7. The missiles are defined as

BIT	7	6	5	4	3	2	1	0
	M3		M2		M1		M0	

Since it is difficult for Atari BASIC to selectively turn bits off and on, we will use ML to change the bits. The AND instruction is used to set bits to zero (off). ANDing a bit with zero sets the bit to zero. The ORA instruction is used to set bits to one (on). By ORAing a bit with one, we set the bit to one. The flipping of the missile bits is done in the subroutines at lines 1300-1330. The original Laser Gunner BASIC program with the vertical blank enhancements appears below.

All the lines after 1280 are new, and the other major changes are from lines 630 to 735, and from lines 880 to 900. In addition, to speed up the vertical motion of the defender, the vertical step size was increased by two. The changes for this enhancement are in lines 110, 530, 540, 560, 630, 640, and 650.

Further Enhancements

The programming technique of performing graphics movement during the vertical blank enhances Laser Gunner almost to the level of difficulty of professional arcade games. Further program execution speed can be achieved by removing the REMs and moving the part of the program that does most of the action to the beginning. This shortens the memory that BASIC has to search to find line number references. An additional enhancement would be to add a sound routine during the VB each time the trigger is pressed.

Laser Gunner II

```
0 REM LASER GUNNER.V2{17 SPACES}
10 GOSUB 1400
20 RESTORE
100 DIM PM$(2048):GRAPHICS 2+16
110 DIM ALIEN$(11),PLAYER$(11),NULL$
  (11),EXPLODE$(12*9),TARGET(20)
120 FOR I=1 TO 11:NULL$(I)=CHR$(0):N
  EXT I
130 LEVEL=15:CNT=15:REM DECREASE LEV
  EL FOR A HARDER GAME
140 A=ADR(PM$):REM RAW ADDRESS
150 PMBASE=INT(A/1024)*1024:REM NEAR
  EST 1 K BOUNDARY
160 IF PMBASE<A THEN PMBASE=PMBASE+1
  024:REM IF BELOW STRING, GO TO N
  EXT 1K BOUNDARY
170 S=PMBASE-A:REM START OF PMBASE I
  N STRING (OFFSET)
180 POKE 559,46:REM SET DOUBLE-LINE
  RES.
190 POKE 54279,PMBASE/256:REM TELL A
  NTIC WHERE PMBASE IS
200 POKE 53277,3:REM TURN ON PLAYER/
  MISSILE DIRECT MEMORY ACCESS(DMA)
210 PM$=CHR$(0):PM$(2048)=CHR$(0):PM
  $(2)=PM$:REM CLEAR OUT ALL P/M M
  EMORY
220 POSITION 4,0:? #6;"laser gunner"
230 ? #6:FOR I=1 TO 10:? #6;"■":NEXT
   I:POSITION 0,0
240 REM STRING POS OF PLAYER 0-3, AN
  D MISSILES IN STRING:
250 P0=S+512:P1=P0+128:P2=P1+128:P3=
  P2+128:MS=S+384
260 PM$(P2+32)=CHR$(255):PM$(P2+127)
  =CHR$(255):PM$(P2+33,P2+127)=PM$
  (P2+32):REM CREATE WALL
270 PM$(P3,P3+127)=PM$(P2,P2+127):RE
  M CREATE "ZONE"
280 POKE 53250,92:REM POSITION PLAYE
  R 2, THE WALL
290 POKE 53251,60:REM POSITION PLAYE
  R 3, THE ZONE
300 POKE 53258,0:POKE 53259,3:REM RE
  M MAXIMUM WIDTH
310 POKE 706,14:POKE 707,66:REM SET
  COLOR OF PLAYERS 2 AND 3
320 DATA 0,8,28,62,255,62,255,62,28,
  8,0
330 FOR I=1 TO 11:READ A:ALIEN$(I)=C
  HR$(A):NEXT I:REM PLACE INTO STR
  ING, HENCE INTO P/M MEMORY
340 AY=32:REM ALIEN VERTICAL LOCATION
350 PM$(P1+AY,P1+AY+11)=ALIEN$:REM P
  LACE INTO STRING INTO P/M MEMORY
360 POKE 705,6*16+10:REM SET COLOR O
  F ALIEN TO PURPLE
370 POKE 53249,180:REM SET HORIZONTA
  L POSITION
380 POKE 53257,1:REM SET ALIEN TO DO
  UBLE-WIDTH
390 REM SET UP EXPLODE$, USE FOR EXP
  LOSION OF ALIEN
400 FOR I=1 TO 108:READ A:EXPLODE$(I
  )=CHR$(A):NEXT I:REM EXPLODE DATA
410 DATA 8,28,62,255,54,255,62,28,8,
  8,28,62,235,54,235,62,28,8,8,28,
  54,227,34,227,54,28,8
420 DATA 8,24,34,227,34,227,18,24,8,
  8,24,34,194,32,163,18,8,8
430 DATA 0,0,0,0,24,24,0,0,0,0,0,0,3
  2,8,24,0,4,0,0,0,0,36,0,16,0,36,
  0,0,128,10,128,0,16,0,16,65
440 DATA 0,9,0,0,32,0,32,0,8,0,0,0,6
  4,0,0,64,0,4,0,0,0,0,0,0,0,128,0
450 RY=INT(78*RND(0)+32):MH=190+RY*2
  :REM ATTRACT MODE:
455 POSITION 9,5:? #6;"PRESS":POSITI
  ON 9,6:? #6;"START"
460 FOR I=32 TO 110:PM$(P1+I,P1+I+11
  )=ALIEN$:IF I=RY THEN PM$(MS+RY+
  10,MS+RY+10)=CHR$(12)
470 IF I>RY THEN POKE 53253,MH-I*2
480 IF PEEK(53279)>6 THEN NEXT I
490 PM$(MS+RY+10,MS+RY+10)=CHR$(0)
500 FOR I=110 TO 32 STEP -1:PM$(P1+I
  ,P1+I+11)=ALIEN$:IF PEEK(53279)>
  6 THEN NEXT I
510 IF PEEK(53279)>=7 THEN 450
515 POSITION 9,5:? #6;"{5 SPACES}":P
  OSITION 9,6:? #6;"{5 SPACES}"
520 IF PEEK(53279)=3 THEN FOR I=0 TO
   4:POKE 53248+I,0:NEXT I:GRAPHIC
  S 0:END
530 DATA 0,0,224,48,120,63,120,48,22
  4,0,0
540 FOR I=1 TO 11:READ A:PLAYER$(I)=
  CHR$(A):NEXT I
550 PY=60:REM SET PLAYER'S VERITCAL
  LOCATION
560 PM$(P0+PY,P0+PY+11)=PLAYER$
570 PM$(P1,P1)=CHR$(0):PM$(P1+127,P1
  +127)=CHR$(0):PM$(P1+2,P1+127)=P
  M$(P1)
580 AY=INT(78*RND(0)+32):PM$(P1+AY,P
  1+AY+11)=ALIEN$:REM RESET ALIEN
590 POKE 53256,1:REM PLAYER 0 DOUBLE
  -WIDTH
600 POKE 53248,64:REM HORIZONTAL POS
  ITION OF PLAYER 0
610 POKE 704,26:REM COLOR OF PLAYER 0
620 POKE 53260,1:REM MISSILE 0 DOUBL
  E-WIDTH
630 ST=STICK(0):IF ST<>15 THEN DIR=S
  T:F=2:SOUND 0,100,0,8
635 IF PEEK(CMPFLG)=1 THEN PM$(TMS,T
  MS)=CHR$(0):POKE CMPFLG,0:REM TH
  E MISSILES HIT EACH OTHER
636 IF PEEK(COLFLG)=1 THEN POKE COLF
  LG,0:GOTO 900:REM THE ALIEN MISS
  ILE HIT THE WALL OR ZONE
640 PY=PY-(DIR=14)*(PY>32)*F+(DIR=13
  )*(PY<110)*F:F=1:REM UPDATE PLAYER
650 PM$(P0+PY,P0+PY+11)=PLAYER$:SOUN
  D 0,0,0,0
660 IF PEEK(M0FLG)=1 THEN GOSUB 1310
  :REM ERASE THE PLAYER'S MISSILE
670 IF PEEK(TRIGFLG)=0 THEN GOSUB 13
  10:POKE M0FLG,0:TMS=MS+PY+5:GOSU
  B 1300:POKE TRIGFLG,1:REM THE TR
  IGGER WAS PRESSED
```

```
720 IF PEEK(HITFLG)<>0 THEN 790:REM
    NO COLLISION
725 REM THE PLAYER'S MISSILE HIT THE
    ALIEN
730 SCR=SCR+10:POSITION 11-LEN(STR$(
    SCR))/2,5:? #6;SCR
735 PM$(TMS,TMS)=CHR$(0):POKE M0FLG,
    1:POKE HITFLG,1:POKE 53278,0
740 AY=AY+1:P=PEEK(705):REM PRESERVE
     COLOR OF ALIEN
750 FOR I=0 TO 11:Z=I*9:PM$(P1+AY,P1
    +AY+9)=EXPLODE$(Z+1,Z+9)
760 POKE 705,PEEK(53770):POKE 53279,
    0:SOUND 0,I*2,0,15-I:FOR W=1 TO
    2:NEXT W:NEXT I
770 POSITION 5,5:PRINT #6;"
    {10 SPACES}":REM ERASE SCORE
780 SOUND 0,0,0,0:POKE 705,P:GOTO 570
790 IF AY=PY THEN 870:REM TOO CLOSE
    FOR COMFORT
800 IF TARGET=0 THEN GOSUB 950:TARGE
    T=TARGET(INDEX):REM SELECT A TAR
    GET
810 IF AY<>TARGET THEN 840
820 CNT=CNT-1:IF CNT THEN 630
830 CNT=LEVEL:GOTO 870
840 AY=AY+SGN(TARGET-AY):REM MOVE TO
    WARDS TARGET
850 PM$(P1+AY,P1+AY+11)=ALIEN$
860 GOTO 630
870 IF ABS(AY-PY)<10 THEN GOSUB 970
875 IF PEEK(ALIEFLG)=0 THEN 630
880 POKE ALIEFLG,0:TM1S=MS+AY+5:GOSU
    B 1320:TTAY=AY:GOTO 630
900 P=ASC(PM$(P2+TTAY+5))*2-256:GOSU
    B 1330:POKE 53278,0:REM CUT HOLE
     IN WALL
910 IF P<0 THEN 990:REM WALL DESTROYED
920 PM$(P2+TTAY+5,P2+TTAY+5)=CHR$(P)
930 GOTO 630
940 REM PICK A TARGET
950 INDEX=INDEX+1:TARGET(INDEX)=INT(
    78*RND(0)+32):RETURN
970 IF INDEX=1 THEN 950
980 TARGET=TARGET(INT(INDEX*RND(0)+1
    )):RETURN
990 REM DESTRUCTION OF PLAYER
1000 FOR I=1 TO 100:Z1=TTAY+5+I:Z2=T
     TAY+5-I
1005 PM$(TMS,TMS)=CHR$(0):POKE M0FLG
     ,1:POKE M0PFLG,72
1010 IF Z1<126 THEN PM$(P2+Z1,P2+Z1)
     =CHR$(0)
1020 IF Z2>30 THEN PM$(P2+Z2,P2+Z2)=
     CHR$(0)
1030 IF Z1<126 OR Z2>30 THEN NEXT I
1040 FOR I=30 TO 1 STEP -1:FOR J=0 T
     O 20 STEP 3:SOUND 0,J+1,10,8:PO
     KE 707,PEEK(53770):NEXT J:NEXT
     I
1050 SOUND 0,0,0,0:SOUND 1,0,0,0:POK
     E 707,14:FOR W=1 TO 50:NEXT W:P
     OKE 707,0
1060 FOR I=0 TO 15 STEP 0.2:SOUND 0
     I,8,I:POKE 704,16+I:NEXT I
1070 SOUND 0,0,0,0
1080 Z1=PY:Z2=PY:INCR=0
1090 Z1=Z1+INCR*(Z1<128):Z2=Z2-INCR*
     (Z2>=0):POKE 704,PEEK(53770)
1100 PM$(P0+Z1,P0+Z1)=CHR$(255):PM$(
     P0+Z2,P0+Z2)=CHR$(255):POKE 532
     79,0
1110 INCR=INCR+0.5:IF Z1<127 OR Z2>0
      THEN 1090
1120 FOR I=1 TO 100:POKE 704,PEEK(53
     770):NEXT I
1130 FOR I=0 TO 7:POKE 53248+I,0:NEX
     T I:GRAPHICS 18
1140 POSITION 4,0:PRINT #6;"laser gu
     nner":POSITION 3,5:PRINT #6;"yo
     ur score was:";
1150 POSITION 10-LEN(STR$(SCR))/2,7:
     PRINT #6;SCR
1160 FOR I=15 TO 0 STEP -0.2:SOUND 0
     ,10+10*RND(0),0,I:SOUND 1,100+1
     0*RND(0),16,I
1170 SETCOLOR 4,3,14*RND(0):NEXT I
1280 RUN
1299 REM  M0 SET
1300 Q=USR(ANORA,ASC(PM$(TMS,TMS)),3
     ,2):PM$(TMS,TMS)=CHR$(Q):RETURN
1309 REM  M0 CLEAR
1310 Q=USR(ANORA,ASC(PM$(TMS,TMS)),1
     2,1):PM$(TMS,TMS)=CHR$(Q):RETUR
     N
1319 REM  M1 SET
1320 Q=USR(ANORA,ASC(PM$(TM1S,TM1S))
     ,12,2):PM$(TM1S,TM1S)=CHR$(Q):R
     ETURN
1329 REM  M1 CLEAR
1330 Q=USR(ANORA,ASC(PM$(TM1S,TM1S))
     ,3,1):PM$(TM1S,TM1S)=CHR$(Q):RE
     TURN
1400 TRIGFLG=1546:HITFLG=1547:M0FLG=
     1548:TMS=1:TM1S=1
1410 ALIEFLG=1550:COLFLG=1551
1420 ANORA=1753:CMPFLG=1553
1430 IF PEEK(1753)=104 THEN RETURN
1440 GRAPHICS 18:? #6;"INITIALIZING"
1450 RESTORE 1500:GOSUB 1500
1460 A=USR(1536):RETURN
1500 FOR I=1536 TO 1552:READ A:POKE
     I,A:NEXT I
1509 REM  INIT 1536 TO 1552
1510 DATA 104,169,6,170,160,22,32,92
     ,228,96,1,1,1,72,1,0,180
1520 FOR I=1558 TO 1709:READ A:POKE
     I,A:NEXT I
1530 REM  MISSILE MOVING ROUTINE
1540 DATA 173,132,2,201,0,240,2,208,
     12,205,12,6,240,12,169,0,141,10
     ,6,240
1550 DATA 58,205,12,6,240,53,238,13,
     6,238,13,6,173,13,6,141,4,208,1
     73,8
1560 DATA 208,41,2,208,9,173,13,6,20
     1,190,144,27,176,15,173,13,6,20
     1,170,144
1570 DATA 18,169,0,141,30,208,141,11
     ,6,169,1,141,12,6,169,72,141,13
     ,6,173
1580 DATA 14,6,201,0,208,63,173,9,20
     8,41,1,208,21,173,9,208,41,12,2
     08,29
1590 DATA 206,16,6,206,16,6,173,16,6
     ,141,5,208,208,35,169,1,141,17,
     6,141
1600 DATA 12,6,169,72,141,13,6,208,5
     ,169,1,141,15,6,169,0,141,30,20
     8,169
1610 DATA 1,141,14,6,169,180,141,16,
     6,76,95,228
1620 FOR I=1753 TO 1791:READ A:POKE
     I,A:NEXT I
1630 REM  AND-OR ROUTINES
1640 DATA 104,104,104,141,215,6,104,
     104,141,216,6,104,104,201,1,208
     ,9,173,215,6
1650 DATA 45,216,6,76,249,6,173,215,
     6,13,216,6,133,212,169,0,133,21
     3,96
1660 RETURN
```

©

TI Mailing List

Doug Hapeman

This program can be used for developing small mailing lists, for families or for organizations. There are ten options, including printing a single label or an entire alphabetized mailing list. For the TI-99/4A.

Have you ever kept a file of addresses on index cards, hoping to organize them someday in an orderly fashion? It sounds simple, but in practice you know how difficult it is to organize and update a paper-based filing system. "TI Mailing List" offers you an easy method of creating, maintaining, and utilizing a mailing list file.

Without any programming experience you can keep an up-to-date, well-organized file. The program will prompt you step-by-step through the entry of names, addresses, and phone numbers. Then, with a few simple keystrokes, you can update your file, print lists in two different modes, or save your file on a storage device. It's that easy.

TI Mailing List is designed specifically as a family mailing list, but is flexible enough to accommodate a number of applications. The program will store last names, first names, children's names, addresses, and phone numbers.

The program is written in a Canadian format, that is, Province and Postal Code. However, the format can be easily adjusted to the American system as you type in the program.

Program Environment

The program is set up for 45 entries. After 45 entries you will be given the message *DATA FILE IS FULL*. This feature will prevent your program from crashing with a MEMORY FULL error message. If you have more than 45 addresses to enter, you may easily divide your list into two or more files – for example: (A - L) and (M - Z).

When you RUN the program, the initial title screen appears. The next display permits you to initialize the printer. Be sure to enter the proper name and spelling of the device you're using, because an improper name will cause the program to break when you attempt to address the device later in the program.

Ten Options

Once the computer "environment" is established, you are taken to the Main Index. Here you will discover ten options:

1 View Names List
2 Search For a Name
3 Add Names
4 Change Names
5 Delete Names
6 Alphabetize List
7 Save Data File
8 Load Data File
9 Print Labels/List
10 Finish Session

Of course, to create a mail list you would first choose option 3 (Add Names). The other options will enable you to update, maintain, and utilize an existing file. The program will guide you step-by-step through the procedure for each option. There are many helpful features, such as the Search, Change, and Delete. You can also enter names and addresses in any order, and then, by choosing the Alphabetize option, have the computer sort them for you.

The Data File

The program is written to both save and load data files for either cassette or disk storage. When you choose either the Save or Load option, you will be given any further step-by-step instructions.

Print Options

The program offers you two print options – one for mailing labels, and the other for the mailing list.

The Print Labels option will print the first name, followed by the last name, and then the address on lines two and three. For example:

John Doe
1234 Street Address
City Province Postal Code

The Print Mailing List option will print the last name first, followed by the first name and children's names, with the address on line two, and the phone number on line three. For example:

Doe, John Mary Joe/Sally
1234 Street Address City Province Postal Code
(p)-444/4456

Line spacing between addresses is flexible via a minor program change. If you wish to alter the line spacing, program lines 497 (labels) and 517 (list) may be adjusted by either increasing or decreasing the number of colons (:) at the end of each line. Each colon represents one line space. For example:

> #497 PRINT #2:TAB(5);NA$(I);" ";LN$(I):TAB(5); AD$(I):TAB(5);CP$(I);" ";PC$(I):::: (Add or delete colons here.)

In the Print Labels option, you may wish to print two labels per line instead of one. If so, you should adjust the line listing as follows:

> (Chg) #487 FOR I=1 to N STEP 2
> (Chg) #497 PRINT #2:TAB(5);NA$(I);" ";LN$(I); TAB(45);NA$(I+1);" ";LN$(I+1):TAB(5); AD$(I);TAB(45);AD$(I+1)
> (Add) #498 PRINT #2:TAB(5);CP$(I);" ";PC$(I); TAB(45);CP$(I+1);" ";PC$(I+1)::::

The Search option permits the printing of a single mailing label. After finding the name you are seeking, the display asks if you would like a mailing label printed. If yes, the program branches to the print routine and then returns to the search option.

TI Mailing List Program Structure

Line Nos.	
1-21	REMs and computer environment.
23-47	Main loop, main index.
49-73	Subroutine to view names.
75-109	Subroutine to search for a name.
111-181	Subroutine to add names.
183-285	Subroutine to change data.
287-331	Subroutine to delete names.
333-423	Subroutine to alphabetize list.
425-441	Subroutine to save data.
443-471	Subroutine to load data.
473-521	Subroutine to print.
523-533	Subroutine to finish session.

TI Mailing List

```
1 REM{4 SPACES}99/4A MAIL LIST
  {5 SPACES}
5 REM{3 SPACES}**COMPUTER ENVIRONMEN
  T**
7 DIM LN$(45),NA$(45),CH$(45),AD$(45
  ),CP$(45),PC$(45),TP$(45)
9 CALL CLEAR
11 PRINT " *{3 SPACES}99/4A MAILING
   LIST{3 SPACES}*"::::::::::
13 INPUT "{4 SPACES}PRESS ENTER TO B
   EGIN":X$
15 CALL CLEAR
17 PRINT "{5 SPACES}WHAT IS THE NAME
    OF":"{4 SPACES}YOUR PRINTING DEV
   ICE?"::"  (EXAMPLE: RS232.BA=4800)
   "::::::::::::
19 INPUT P$
21 G$="{7 SPACES}PLEASE WAIT...
   {7 SPACES}WHILE THE PRINTER IS WO
   RKING"
23 REM{3 SPACES}**MAIL LIST MENU**
25 CALL CLEAR
27 PRINT "{8 SPACES}MAIN INDEX"::::
29 PRINT "PRESS{3 SPACES}TO":::
31 PRINT "  1  =  VIEW NAMES LIST":"
    2  =  SEARCH FOR A NAME":"  3
   =  ADD NAMES":"  4  =  CHANGE NAMES"
33 PRINT "  5  =  DELETE NAMES":"  6
     =  ALPHABETIZE LIST":"  7  =  S
   AVE DATA FILE":"  8  =  LOAD DATA
   FILE"
35 PRINT "  9  =  PRINT LABELS/LIST"
   :" 10  =  FINISH SESSION"::::
37 INPUT P
39 IF P>10 THEN 37
41 IF P<1 THEN 37
43 CALL CLEAR
45 ON P GOSUB 51,77,113,185,289,335,
   427,445,475,525
47 GOTO 25
49 REM{4 SPACES}**VIEW NAMES LIST**

51 T=0
53 FOR I=1 TO N
55 T=T+1
57 PRINT NA$(I),LN$(I):CH$(I):AD$(I)
   :CP$(I):PC$(I):"(P)-";TP$(I):::
59 IF T<2 THEN 69
61 PRINT " *PRESS ENTER TO CONTINUE*
   ":" *""R"",ENTER FOR MAIN INDEX*"
63 INPUT X$
65 IF X$="R" THEN 73
67 T=0
69 NEXT I
71 INPUT "{7 SPACES}*END OF FILE*
   {9 SPACES}*PRESS ENTER TO CONTINU
   E*":X$
73 RETURN
75 REM{4 SPACES}**SEARCH NAMES**
77 INPUT "LAST NAME? ":Y$
79 FOR I=1 TO N
81 IF LN$(I)<>Y$ THEN 103
83 PRINT :::"  IS THE PERSON:"::"  "
   ;NA$(I):"  ";LN$(I)::
85 INPUT "  (Y/N)?":X$
87 IF X$="N" THEN 103
89 PRINT :::NA$(I),LN$(I):CH$(I):AD$
   (I):CP$(I):PC$(I):"(P)-";TP$(I)::
   :
91 INPUT "{3 SPACES}DO YOU WISH TO P
   RINT{6 SPACES}A MAILING LABEL?  (
   Y/N)":Z$
93 IF Z$<>"Y" THEN 97
95 GOSUB 495
97 INPUT "SEARCH MORE NAMES?  (Y/N)"
   :X$
99 IF X$="Y" THEN 77
101 GOTO 109
103 NEXT I
105 PRINT :::"  THE ";Y$:"  YOU ARE
    SEARCHING FOR":"  IS NOT IN THIS
     FILE."::::
107 GOTO 97
109 RETURN
111 REM{4 SPACES}**ADD NAMES**
    {5 SPACES}
113 A=N+1
115 FOR I=A TO 45
117 CALL CLEAR
119 PRINT ::::"ENTER DATA: ";"#";I;"
     (MAX:45)":::
121 PRINT "  *LAST NAME:"
123 INPUT LN$(I)
125 PRINT :"  *FIRST NAME(S):"
```

```
127 INPUT NA$(I)
129 PRINT :"  *CHILDREN:":"
    {3 SPACES}NOTE--DO NOT USE COMMA
    S!"
131 INPUT CH$(I)
133 PRINT :"  *STREET ADDRESS:"
135 INPUT AD$(I)
137 PRINT :"  *CITY/PROVINCE:":"
    {3 SPACES}NOTE--DO NOT USE COMMA
    S!"
139 INPUT CP$(I)
141 PRINT :"  *POSTAL CODE:"
143 INPUT PC$(I)
145 PRINT :"  *PHONE:"
147 INPUT TP$(I)
149 V=I
151 REM{3 SPACES}**VERIFY ENTRIES**
153 CALL CLEAR
155 PRINT "ENTRY";"#";V:::
157 PRINT "YOU ENTERED:"::"  ";LN$(V
    );",  ";NA$(V):"  ";CH$(V):"  ";A
    D$(V):"  ";CP$(V)
159 PRINT "  ";PC$(V):"  PHONE: ";TP
    $(V):::::::
161 INPUT "CHANGE ANYTHING?  (Y/N)":
    X$
163 IF X$<>"Y" THEN 171
165 C=N+1
167 CALL CLEAR
169 GOSUB 2Ø1
171 INPUT "ADD MORE NAMES?  (Y/N)":X
    $
173 N=N+1
175 IF X$="N" THEN 181
177 NEXT I
179 INPUT "{4 SPACES}*DATA FILE IS F
    ULL*{6 SPACES}*PRESS ENTER TO CO
    NTINUE*":X$
181 RETURN
183 REM{4 SPACES}**CHANGE DATA**
185 PRINT "  LAST NAME OF THE PERSON
    {3 SPACES}WHOSE DATA IS TO BE CH
    ANGED:"::::
187 INPUT C$
189 CALL CLEAR
191 FOR C=1 TO N+1
193 IF LN$(C)=C$ THEN 195 ELSE 239
195 PRINT "IS THE PERSON:":"  ";NA$(
    C):" ";LN$(C)::
197 INPUT "  (Y/N)?":X$
199 IF X$="Y" THEN 2Ø1 ELSE 239
2Ø1 PRINT :::::::"PRESS{3 SPACES}TO
    CHANGE"::
2Ø3 PRINT "  1  = LAST NAME":"  2
    =  FIRST NAME(S)":"  3  =  CHILD
    REN":"  4  = STREET ADDRESS"
2Ø5 R=C
2Ø7 R$="  *ENTER THE NEW DATA:"
2Ø9 PRINT "  5  =  CITY/PROVINCE":"
    6  =  POSTAL CODE":"  7  =  PHO
    NE":"  8 = NO CHANGE":::::::
211 INPUT P
213 CALL CLEAR
215 IF P<1 THEN 211
217 IF P>8 THEN 211
219 IF P=8 THEN 229
221 ON P GOSUB 245,251,257,263,269,2
    75,281
223 PRINT ::"MORE CHANGES FOR:":"  "
    ;NA$(R):"  ";LN$(R)::
225 INPUT "  (Y/N)?":Y$
227 IF Y$<>"N" THEN 2Ø1
229 PRINT :::"CHANGE DATA FOR OTHER
    NAMES?":::
231 INPUT "  (Y/N)":Z$
233 CALL CLEAR
235 IF Z$<>"N" THEN 185
237 RETURN
239 NEXT C
241 RETURN
243 REM{3 SPACES}**CHANGE LOOPS**
245 PRINT "LAST NAME WAS:"::LN$(R)::
    :R$
247 INPUT LN$(R)
249 RETURN
251 PRINT "FIRST NAME(S) WERE:"::NA$
    (R):::R$
253 INPUT NA$(R)
255 RETURN
257 PRINT "CHILDREN WERE:"::CH$(R)::
    :R$
259 INPUT CH$(R)
261 RETURN
263 PRINT "ADDRESS WAS:"::AD$(R):::R
    $
265 INPUT AD$(R)
267 RETURN
269 PRINT "CITY/PROVINCE WAS:"::CP$(
    R):::R$
271 INPUT CP$(R)
273 RETURN
275 PRINT "POSTAL CODE WAS:"::PC$(R)
    :::R$
277 INPUT PC$(R)
279 RETURN
281 PRINT "PHONE NUMBER WAS:"::TP$(R
    ):::R$
283 INPUT TP$(R)
285 RETURN
287 REM{4 SPACES}**DELETE NAMES**
289 INPUT "LAST NAME? ":X$
291 FOR I=1 TO N
293 IF LN$(I)<>X$ THEN 325
295 PRINT :::"IS THE PERSON:":"  ";N
    A$(I):"  ";LN$(I)::
297 INPUT "  (Y/N)?":Y$
299 IF Y$<>"Y" THEN 325
3Ø1 A=I
3Ø3 FOR D=A TO N
3Ø5 LN$(D)=LN$(D+1)
3Ø7 NA$(D)=NA$(D+1)
3Ø9 CH$(D)=CH$(D+1)
311 AD$(D)=AD$(D+1)
313 CP$(D)=CP$(D+1)
315 PC$(D)=PC$(D+1)
317 TP$(D)=TP$(D+1)
319 NEXT D
321 N=N-1
323 GOTO 327
325 NEXT I
327 INPUT "MORE DELETIONS? (Y/N) ":X$
329 IF X$="Y" THEN 289
331 RETURN
333 REM{3 SPACES}**ALPHABETIZE LIST*
    *{3 SPACES}
335 PRINT "{7 SPACES}PLEASE WAIT..."
    :::" THE LIST IS BEING ARRANGED"
    ::::::::::
337 B=1
339 B=2*B
341 IF B<=N THEN 339
343 B=INT(B/2)
345 IF B=Ø THEN 369
347 FOR Y=1 TO N-B
348 X=Y
349 I=X+B
351 IF LN$(X)=LN$(I)THEN 363
353 IF LN$(X)<LN$(I)THEN 365
355 GOSUB 381
357 X=X-B
359 IF X>Ø THEN 349
```

```
361 GOTO 365
363 GOSUB 373
365 NEXT Y
367 GOTO 343
369 RETURN
371 REM{3 SPACES}**ORDER FIRST NAMES
    **{3 SPACES}
373 IF NA$(X)<NA$(I)THEN 377
375 GOSUB 381
377 RETURN
379 REM{3 SPACES}**CHANGE ORDER**
381 N$=LN$(X)
383 LN$(X)=LN$(I)
385 LN$(I)=N$
387 N$=NA$(X)
389 NA$(X)=NA$(I)
391 NA$(I)=N$
393 N$=CH$(X)
395 CH$(X)=CH$(I)
397 CH$(I)=N$
399 N$=AD$(X)
401 AD$(X)=AD$(I)
403 AD$(I)=N$
405 N$=CP$(X)
407 CP$(X)=CP$(I)
409 CP$(I)=N$
411 N$=PC$(X)
413 PC$(X)=PC$(I)
415 PC$(I)=N$
417 N$=TP$(X)
419 TP$(X)=TP$(I)
421 TP$(I)=N$
423 RETURN
425 REM{3 SPACES}**SAVE DATA FILE**
    {5 SPACES}
427 GOSUB 467
429 OPEN #1:L$,INTERNAL,OUTPUT,FIXED
     150
431 PRINT #1:N
433 FOR I=1 TO N
435 PRINT #1:LN$(I),NA$(I),CH$(I),AD
    $(I),CP$(I),PC$(I),TP$(I)
437 NEXT I
439 CLOSE #1
441 RETURN
443 REM{4 SPACES}**LOAD DATA FILE**
    {6 SPACES}
445 GOSUB 467
447 OPEN #1:L$,INTERNAL,INPUT ,FIXED
     150
449 INPUT #1:N
451 FOR I=1 TO N
453 INPUT #1:LN$(I),NA$(I),CH$(I),AD
    $(I),CP$(I),PC$(I),TP$(I)
455 NEXT I
457 CLOSE #1
459 CALL CLEAR
461 PRINT " ";L$::"  THIS FILE HAS"
    ;N;"ENTRIES.":: "  *45 ENTRIES IS
     MAXIMUM*":::::::::
463 INPUT " *PRESS ENTER TO CONTINUE
    *":X$
465 RETURN
467 PRINT "{5 SPACES}WHAT IS THE NAM
    E OF":"{4 SPACES}YOUR STORAGE DE
    VICE?"::"(EXAMPLE: CS1OR DSK1.FIL
    E)":::::::::::
469 INPUT L$
471 RETURN
473 REM  **SUB TO PRINT LABELS/LIST*
    *
475 PRINT "PRESS{3 SPACES}TO PRINT":
    ::"  1{5 SPACES}MAILING LABELS":
    :"  2{5 SPACES}MAILING LIST":::::
    ::
477 INPUT P
479 IF P<1 THEN 477
481 IF P>2 THEN 477
483 PRINT ::::::::::::G$::::::::::
485 IF P<>1 THEN 505
487 FOR I=1 TO N
489 GOSUB 495
491 NEXT I
493 RETURN
495 OPEN #2:P$
497 PRINT #2:TAB(5);NA$(I);" ";LN$(I
    ):TAB(5);AD$(I):TAB(5);CP$(I);"
     ";PC$(I)::::
499 CLOSE #2
501 RETURN
503 REM{4 SPACES}**PRINT MAIL LIST**
505 FOR I=1 TO N
507 GOSUB 513
509 NEXT I
511 RETURN
513 OPEN #2:P$
515 PRINT #2:TAB(5);LN$(I);", ";NA$(
    I);"{6 SPACES}";CH$(I):TAB(5);AD
    $(I);"{3 SPACES}";CP$(I);" ";PC$(I)
517 PRINT #2:TAB(60);"(P)-";TP$(I)::
519 CLOSE #2
521 RETURN
523 REM{3 SPACES}**FINISH SESSION**
    {5 SPACES}
525 INPUT "{7 SPACES}DO YOU WISH TO
    {10 SPACES}TERMINATE THIS SESSION
    ?{5 SPACES}(Y/N)":X$
527 CALL CLEAR
529 IF X$<>"Y" THEN 25
531 PRINT "{6 SPACES}HAVE A NICE DAY
    !":::::::::
533 STOP
```
©

VIC Bitmapping

C. D. Lane

If you don't think that there's enough space in an unexpanded VIC to create exciting, high resolution graphics, you're in for a surprise.

Bitmapping, controlling each tiny dot on the TV screen, is the only way to gain total control over the video image. The finest, sharpest graphics result when you govern each point of light separately.

This article deals exclusively with BASIC programming and contains a ready-to-run demo program along with an engaging, high-resolution, two-player game called "Lines." Nevertheless, some of the concepts here might be new to the beginning VIC user so we've provided a brief dictionary of the more complicated terms.

Bitmapping allows us to turn any bit on the screen on or off, usually under the control of a program. Bitmapping the VIC's screen requires software routines to plot the bits, proper initialization of the video registers, and most of RAM to store the screen.

Sizes And Shapes

The video chip in the VIC-20 computer cannot address expansion RAM. This limitation leaves it 1K of RAM starting at address 0, 4K of RAM starting at 4096 ($1000 hex), 4K of ROM starting at 32768 ($8000), and 1K of nybble RAM starting at 37888 ($9400). The normal state of a 5K unexpanded VIC has the character memory in the 4K of ROM, the screen memory in 506 bytes of RAM at 7680 ($1E00), and the color memory in 506 bytes of nybble RAM at 38400 ($9600). (*Nybble* just means that items are stored in four-bit large spaces in memory, rather than the normal, eight-bit groups called *bytes*.)

There are two character sizes on the VIC 6560 video chip, 8 bits by 8 bits and 16 bits by 8 bits. The 8 x 8 characters are the norm, and the character ROM is set up to use these. The 16 x 8 characters are twice as tall, and are useful in bitmapping the screen since they double the graphic area using the same amount of screen memory. The 8 x 8 characters are advantageous when using shapes from the character ROM, and they are simpler to initialize and plot.

At most, 256 different characters of 64 bits each, or 16,384 bits, can be addressed with 8 x 8 characters. To get more screen area you must use 16 x 8 characters. With the larger characters, 256 characters of 128 bits – or 32,768 bits maximum, twice as many – can be addressed. The 16 x 8 characters can be selected by turning on the low bit of 36867 ($9003). (You can turn this bit on by POKEing it with any odd number.)

The screen can also be bitmapped in multicolor mode. Multicolor mode is covered in detail in the *VIC-20 Programmer's Reference Guide*. Multicolor mode reduces horizontal resolution to half of normal, so the characters are now 8 x 4 and 16 x 4, using two bits for each pixel on the screen. Pixels are located in a byte by using powers of 4, rather than the powers of 2 used in high resolution mode.

Usually you will need to reshape the screen for your graphic program, since you will probably be using less area than the normal screen. Memory location 36866 ($9002) controls the number of columns on the screen, and 36867 ($9003) controls the number of rows, as well as the double high characters. Be careful of the meaning of rows when using the larger characters since rows become twice as tall.

You can format the screen by using the combination of columns and rows, which results in the closest thing to a square shape, or a combination that uses all of the RAM available. An alternate scheme is to try to do both. The standard VIC screen uses this combination: with 22 columns by 23 rows, it achieves the "most square" screen possible, using as many of the 512 characters as it can (leaving only 6 characters unused).

Where It All Goes

When you're bitmapping the screen from BASIC, there is quite a lot to fit into a small space: the screen, character, and BASIC memories all must go into the 4K of RAM at address 4096 (except, of course, in an expanded VIC, where BASIC can be moved out of internal memory). There are several ways we can accommodate BASIC and character memories, all with the screen at address 7680. In the table below, the first column is the value to POKE into the character pointer register of the VIC chip; 52,56 are the high bytes of the end of RAM pointers.

			Usable RAM		# of Chars		Screen Shape	
36869	Start	52,56	Graphic	BASIC	8x8	16x8	Square	Maximum
252	4096	16	3.5K	none	256	224	21x21	16x28
253	5120	20	2.5K	1K	256	160	17x18	16x20
254	6144	24	1.5K	2K	192	96	13x14	12x16
255	7168	28	0.5K	3K	64	32	8x8	8x8

Moving the screen memory from 7680 to the top of the lower 1K of memory – for example, 768 ($0300), where the tape buffers are – adds 64 8x8 or 16 16x8 characters to the above figures (and adds 0.5K to the available RAM figures). However, this move is more safely accomplished via machine language. The character pointer can also be pointed at the first 1K of RAM (POKE 36869,248), but this is not very useful since page zero is used in both BASIC and machine language programming.

The case where the character pointer equals 255 is described in detail in the *Reference Guide*. This arrangement gives the user very little graphic area, but does allow mixing graphic characters with a subset of the ROM characters.

Programming The Bitmap

The first step in bitmapping the screen is to allocate RAM for the character memory in an area outlined in the chart. We must keep BASIC from using this memory by POKEing locations 52 and 56 with the high byte of the starting location we have chosen, and POKEing locations 51 and 55 with the low byte (which is zero for the examples in this article since they all begin on a page boundary).

To initialize the character memory, clear it by POKEing each byte with a zero. Next, initialize the screen memory by sequentially numbering the locations and POKEing each byte with its distance (in bytes) from the start of screen memory.

Program 1: Polargraph

```
10 POKE55,Ø:POKE52,2Ø:POKE56,2Ø:CLR:V=368
   64:M=512Ø:H=248:W=768Ø:R=384ØØ:K=
   63:S$="N"
20 U=2/3:T=1:F=Ø:Q=63:B=.Ø1:L=1:C=2:FORI=
   .TO7:T%(7-I)=2^I:NEXT:POKEV+1,37:
   POKEV+3,32
3Ø PRINTCHR$(147)TAB(6)"POLARGRAPH"CHR$(1
   7):INPUT"FREQUENCY";F
4Ø INPUT"OBJECT COLOR 1-8";C:IFC>8ORC<1GO
   TO4Ø
5Ø INPUT"SCREEN COLOR 1-8";T:IFT>8ORT<1GO
   TO5Ø
6Ø INPUT"SIZE Ø-63";Q:IFQAND192GOTO6Ø
7Ø INPUT"ECCENTRICITY <=1";U:IFABS(U)>1GO
   TO7Ø
8Ø INPUT"RESOLUTION";B:INPUT"CYCLES";L:IN
   PUT"SOLID Y/N";S$:PRINTCHR$(17)"P
   LEASE WAIT";
9Ø FORI=MTOW-512:POKEI,.:NEXT:POKEV+5,253
   :POKEV,11:POKEV+2,144:G=S$<>"Y"
1ØØ POKEV+15,17*T-9:FORI=.TO255:POKEW+I,I:
    POKER+I,C-1:NEXT:FORI=.TOL*2*STE
    PB
11Ø S=ABS(COS(F*I)*Q):X=COS(I)*S*U+K:Y=SIN
    (I)*S+K:GOSUB16Ø:IFGGOTO15Ø
12Ø IFABS(X-K)>ABS(Y-K)GOTO14Ø
13Ø S=SGN(K-Y):E=(K-X)/(K-Y)*S:X=X-E:FORY=
    YTOKSTEPS:X=X+E:GOSUB16Ø:NEXT:GOT
    O15Ø
14Ø IFX<>KTHENS=SGN(K-X):E=(K-Y)/(K-X)*S:Y
    =Y-E:FORX=XTOKSTEPS:Y=Y+E:GOSUB16
    Ø:NEXT
15Ø NEXT:WAIT198,1:GETA$:POKEV+5,24Ø:POKEV
    +15,24:POKEV,5:POKEV+2,15Ø:GOTO3Ø

16Ø Z=(YANDH)*15+Y+(XANDH)+M:POKEZ,PEEK(Z)
    ORT%(XAND7):RETURN
```

Program 2: Lines

```
1Ø POKE52,2Ø:POKE56,2Ø:CLR:V=36864:M=512Ø
   :S=768Ø:C=384ØØ:Q=816Ø:W=24Ø:GOTO 8Ø
2Ø I=1-I:FORY=.TO6STEP2:X=Y+I*8:POKEQ+1,H
   %(X):SYS(Q):IFPEEK(Q+19)=H%(X+1)T
   HENX=Y:Y=W
3Ø NEXT:POKEV+13,W:IFY>WTHENP%(I,.)=(X=4)
   -(X=.):P%(I,1)=(X=2)-(X=6)
4Ø X=P%(I,2)+P%(I,.):Y=P%(I,3)+P%(I,1):PO
   KEV+13,.:IFX>63ORY>159ORX*Y<.GOTO7Ø
5Ø P%(I,2)=X:P%(I,3)=Y:Y=(YANDW)*15+Y+(X*
   4ANDW)+M:X=T%(XAND3):IFPEEK(Y)AND
   XGOTO7Ø
6Ø POKEY,XORI*X*2ORPEEK(Y):GOTO2Ø:DATA6Ø,
   6Ø,62,62,6Ø,62,56,6Ø,32,32,,48,32
   ,48,48,32
7Ø POKEV+15-I,17-I:FORI=.TOM:NEXT:GOTO11Ø
   :DATA138,96,11Ø,1,114,142,2,114,1
   1Ø,212,,65
8Ø POKEV+1,3Ø:POKEV+3,21:POKEV,11:POKEV+2
   ,144:POKEV+5,253:DIMP%(1,3),H%(15
   ),T%(3)
9Ø FORI=.TO255:POKES+I,I:POKEC+I,8:NEXT:D
   EFFNR(X)=INT(RND(1)*X):FORI=.TO3:
   T%(3-I)=4^I
1ØØ NEXT:FORI=.TO15:READX:H%(I)=X+191:NEXT
    :FORI=.TO11:READX:POKEI+Q,X+31:NEXT
11Ø FORI=MTOS:POKEI,.:NEXT:POKEV+15,19+FNR
    (5):POKEV+14,16*FNR(5)+63
12Ø FORI=.TO1:P%(I,.)=.:P%(I,1)=FNR(2)*2-1
    :P%(I,2)=21*(I+1):P%(I,3)=8Ø:NEXT
    :I=1:GOTO2Ø
```

To plot, we will use cartesian (X,Y) coordinates, where Y is zero at the top of the screen and X is zero at the left of the screen, making the HOME position the origin (0,0). Both are bounded on the high end by the particular height and width chosen for the screen in bits. To plot a particular bit in memory from its X,Y coordinates, we must determine the actual character it resides in – which byte of that character, and which bit of that byte.

To determine what character the bit is in, drop the low order digits of the coordinates, where the number of low order digits equals log base 2 (dimension in pixels of the character). Next find the number of the character by multiplying the character's Y coordinate by the number of X characters in each row and add in the character's X coordinate. The low order Y bits not used earlier are the number of the byte in the character the bit is in. 128/ (2 to the low order X bits) locates the bit in the byte.

For example, in a 16 x 16 character screen with 8 x 8 characters, use INT(coordinate/8)*8 or simply (coordinate AND (255 - 7)) to get the character coordinates (but don't throw away the original values!). The location of the character is X + Y * 16 (the number of columns).

Location of byte in memory =
Start of character memory
+ Number of character the byte is in * Number of bytes in character
+ The low order Y bits (the byte in the character)

To set a bit, POKE byte, PEEK(byte) OR 128/(2 ↑ low order X bits)

To clear a bit, POKE byte, PEEK(byte) AND NOT 128/(2 ↑ low order X bits)

To plot bits faster, store the powers of 2 (or 4 for multicolor mode) in an array at initialization time, saving the time of computing the powers each time.

Two programs are included here to illustrate bitmapping the screen.

Polargraph

Polargraph (Program 1) prompts the user for various parameters and uses these to draw spiral curves and solid objects. The program uses polar coordinates and the SIN( ) function to calculate the shapes, translates these to cartesian coordinates and plots them on the VIC's linear memory in high resolution mode. Polargraph asks for the following parameters to control plotting:

Frequency? This controls the number of "leaves" on the design; higher numbers give more leaves. It also controls overlapping – whole numbers give non-overlapping "leaves" and rational numbers give more complex patterns. As the frequency decreases from 1 to 0, different types of cardioids are produced, degenerating into spirals, and eventually ending in a perfect circle at zero, the default value. Any value is legal for this parameter.

Object color? Screen color? The number the user enters is the number on the key that the color he wishes is on; black = 1, white = 2, and so on. Numbers from 1 to 8 are legal here. The default is a white object on a black screen.

Size? This controls how far out from the center the object extends. A value around zero will make a dot in the center of the screen, and a value of 63 (half the width in bits of the screen) will make the largest possible shape. The default value here is 63.

Eccentricity? This controls how elliptical the shape is. The default 2/3 produces the most symmetric result on the screen. The default is less than 1 since the VIC's screen is not square (it uses rectangular pixels). A value of 1 (circular) is best when sending the shapes to a printer. The shapes can be stretched both horizontally or vertically. Legal values are between -1 and 1.

Resolution? This controls how many dots are used to draw the figure. A high resolution makes a finer drawing, but takes longer to draw. A low resolution draws faster with less precision. The default is .01, any value is legal, and the usable range is .5 to .001.

Cycles? This is the number of times around the graph the program runs. Its setting is related to the frequency. A simple whole number frequency requires only 1 cycle. A frequency of 3.33 requires 3 cycles to complete the drawing; 1.25 requires 2; 2.125 requires 4, and so forth. The default is 1 cycle; any value is legal. This parameter can be used to force partial drawings, open curves, incomplete spirals, etc.

Solid? This parameter controls whether the shape is drawn with dots or lines. Lines make sense only with whole number frequencies. Line mode is slower. The default is "N" which is dot mode. Legal values are "Y" and "N" (yes and no).

The user does not have to enter every parameter; just RETURNing uses the defaults outlined above. Once a drawing is complete, typing any key will get it back to the parameter menu. The defaults become the values entered on the last run, so you can just change one or more parameters to see their effect while holding others constant.

Polargraph uses a 16 row by 16 column screen in 8 by 8 character format (256 characters), simplifying the mathematics of plotting due to the symmetry of the screen. The LINES program formats the screen in a more complex and less symmetric fashion.

The designs produced by Polargraph can be printed on the VIC 1515 (or other) printer with the routine in my article "Printing the Screen" in *COMPUTE!'s First Book of VIC* (**COMPUTE! Books**, 1982). You will need to remove the IF statement after the colon on the end of line 5 if you are using more than 64 graphic characters (as Polargraph does). The variables HIGH and WIDE will also need to be adjusted to the dimensions (in characters) of the screen you are using. These changes

Brief Definitions

Dan Carmichael, Assistant Editor

Some of the terms used in this article might be unfamiliar. Here's a short description of the main ideas:

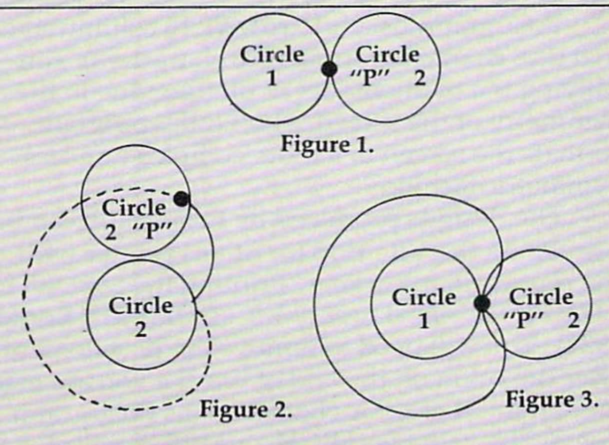
• **Bitmapping.** Bitmapping is a process whereby each tiny individual dot (pixel) on a TV screen or monitor is represented by its own "bit" in memory. When the corresponding memory bit is zero (off), the dot (or pixel) is off. When the bit is a one (on), the pixel is turned on. Each byte of memory (an address like 1525) is made up of eight bits.

When bitmapping the VIC-20, there are 32,384 separately programmable pixels. With each pixel assigned to one bit (or eight pixels to the byte), it would take 4048 bytes to bitmap the entire screen.

• **Cardioids.** A cardioid is a heart-shaped, closed curve that is produced by tracing a fixed point on one circle as it is rolled around the circumference of another equal, stationary circle.

Refer to figures 1 through 3. As circle 2 is rolled around the circumference of stationary circle 1 (Figures 1 and 2), the fixed point "P" on circle 2 begins to rotate and produce the heart-shaped cardioid curve. Figure 3 represents one complete revolution around fixed circle 1 and displays the completed, closed cardioid curve.

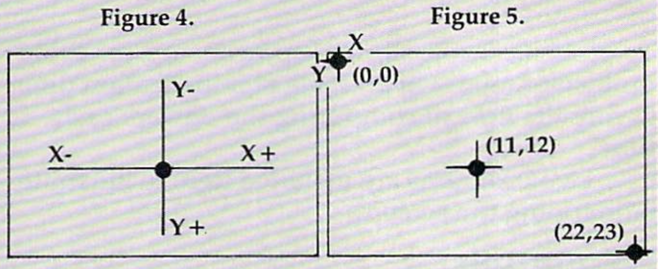
Cardioid curves are used in geometric applications for the classical problem of trisecting an angle. In "Bitmapping The Screen," various types of cardioids are produced in the Polargraph program.



"Drawing" The Cardioid Curve

• **Cartesian Coordinates.** The Cartesian Coordinate system is the common *x - y* coordinate system that is widely used in plotting charts and graphs. The *x* coordinate represents an imaginary horizontal plane, the *y* a vertical plane (refer to Figure 4). Positions are plotted by indicating their *x* and *y* coordinates (x,y). That is, you can locate anything by giving a horizontal and vertical number. It's like the way you can locate a particular street on a city map by looking for it within the square called "F-5" or "C-2."

In the article "VIC Bitmapping," *x* coordinate points begin at the left of the screen, *y* coordinate points begin at the top of the screen (see Figure 5). Coordinate 0,0 thereby becomes what computerists call the HOME position, the upper left corner of the screen. Screen positions are plotted by raising or lowering the *x* and *y* coordinates. Figure 5 illustrates plotted examples where the *x,y* coordinates are 11,12 (middle of the screen), and 22,23 (lower righthand corner).



Cartesian Coordinates On The TV Screen

• **Low Byte - High Byte.** For a complete definition of low byte - high byte (LBHB) addressing techniques refer to *An Explanation of LBHB* in "All About USR" in this issue.

• **Screen Memory.** Screen memory is the memory in the VIC that retains the image of what is displayed on the screen. Screen memory is RAM, and its contents are defined by the user. POKEing values into the screen memory will, in effect, display characters on the screen.

• **Polling The Keyboard.** "Polling" is the process of continually checking the status of a device (such as a keyboard, peripheral, etc.) to determine if anything has changed. *Polling the Keyboard* in the VIC means that the operating system of the VIC-20 checks the keyboard (60 times every second) to determine if any keys have been pressed.

will allow the printing of any large graphic screen.

The Lines Game

"Lines" (Program 2) is a two-player game in which each player independently guides his own line from the keyboard. The screen is formatted in double high character mode, with 16 columns by 10 rows (the equivalent of 20 rows in regular character format). The program makes the maximum use of the RAM available with BASIC and the screen in the same 4K. The screen is in multicolor mode with the two players' lines controlled by different color registers, so that they can run alongside each other without color interference, making for slightly more complicated plotting, but greater visual effect.

Each player controls a constantly growing colored line on the screen; he must not touch the walls, the other player's line, or his own line, or else his line disappears. The left line is controlled by keys A (left), D (right), W (up), and X (down). The right line is controlled by keys K (left), ; (semicolon, right), O (up), and . (period, down). The two players do not interfere with each other's control even though they both use the same keyboard.

The technique used to poll the keyboard is described in detail by Mike Bassman and Salomon Lederman in *COMPUTE!'s First Book of VIC*. The machine language subroutine in LINES has been moved into unused screen memory, in order to keep character memory free. It's easier to use the game if the keys that control the lines are marked with paper sticker arrows.

Both programs are as compact as possible in order to be fast and not exceed the 1K limit to program and variable space imposed by the large screen. Both programs are for an unexpanded VIC; however, they will also run with a 3K expander, all 3K of which is available for extra code.

Bitmapping the screen requires careful accounting of memory usage and small, efficient programs in an unexpanded VIC. With a little extra thought and work, it is possible to produce dazzling graphic displays and games without special hardware or software additions. ©

CAPUTE!

Modifications Or Corrections To Previous Articles

64 Video, Part III

Line 40 of the demonstration program from Part III of Jim Butterfield's "Commodore 64 Video – A Guided Tour" (p. 160, April 1983) should read:

```
40 FOR J=Ø TO 62:READ X:POKE J+832,X:NEX
   T J
```

Atari One-On-One

To use paddles with the Atari version of this game from the May 1983 issue (p. 48), change line 650 to read:

```
650 BLINE=250
```

Editype For The 64 And VIC

Reader Clifford Johnsen supplied changes which allow the VIC Editype mini word processing program from the April 1983 issue (p. 50) to run on the Commodore 64. Delete lines 265 and 266 and modify the following lines:

```
210 A$(K)=A$(K)+C$: C$="": IF LEN(A$(K))
   <4Ø THEN 12Ø
24Ø FOR U=1 TO 4Ø-LEN(A$(K)): PRINT CHR$
   (2Ø);: NEXT U
5Ø25 FOR B=1 TO 4Ø
```

For VIC or 64 users, the following change to line 180 suggested by John Stoddard will provide an underline cursor to eliminate confusion about where the next character will be printed:

```
18Ø PRINT C$;"{DOWN}{LEFT}";CHR$(32);CHR
   $(163);"{UP}{LEFT}";
```

Computer Literacy On The Timex/Sinclair

Program 4, "Big Letters," from this article in the April 1983 issue (p. 165) requires one correction. Change line 140 to read:

```
14Ø LET W$=W$(2 TO LEN W$)
```

Atari SuperFont Plus

Author John Slaby has found the following corrections to minor bugs in his improved version of SuperFont which was published in the February 1983 issue (p. 154):

```
12ØØ POKE 54286,192:GOSUB 39Ø:GOTO 52Ø
144Ø DATA 72, 169, 1ØØ, 141, 1Ø, 212
17ØØ IF A=58 OR (A>47 AND A<58) OR (A>64
   AND A<=9Ø) OR A=46 THEN 172Ø
```

Also, D. Chouiniere suggests that to eliminate the problem of scrolling of the display, line 500 should be changed to 525 and then line 500 should be deleted. ©

COMPUTE! Back Issues

Here are some of the applications, tutorials, and games from available back issues of **COMPUTE!**. Each issue contains much, much more than there's space here to list, but here are some highlights:

February 1981: Simulating PRINT USING, Using the Atari as a Terminal for Telecommunications, Attach a Printer to the Atari, Double Density Graphing on C1P, Commodore Disk Systems, PET Crash Prevention, A 25¢ Apple II Clock.

May 1981: Named GOSUB/GOTO in Applesoft, Generating Lower Case Text on Apple II, Copy Atari Screens to the Printer, Disk Directory Printer for Atari, Realtime Clock on Atari, PET BASIC Delete Utility, PET Calculated Bar Graphs, Running 40 Column Programs on a CBM 8032.

June 1981: Computer Using Educators (CUE) on Software Pricing, Apple II Hires Character Generator, Ever- expanding Apple Power, Color Burst for Atari, Mixing Atari Graphics Modes 0 and 8, Relocating PET BASIC Programs, An Assembler In BASIC for PET, QuadraPET: Multitasking?

July 1981: Home Heating and Cooling, Animating Integer BASIC Lores Graphics, The Apple Hires Shape Writer, Adding a Voice Track to Atari Programs, Machine Language Atari Joystick Driver, Four Screen Utilities for the PET, Saving Machine Language Programs on PET Tape Headers, Commodore ROM Systems, The Voracious Butterfly on OSI.

August 1981: Minimize Code and Maximize Speed, Apple Disk Motor Control, A Cassette Tape Monitor for the Apple, Easy Reading of the Atari Joystick, Blockade Game for the Atari, Atari Sound Utility, The CBM "Fat 40," Keyword for PET, CBM/PET Loading, Chaining, and Overlaying.

October 1981: Automatic DATA Statements for CBM and Atari, VIC News, Undeletable Lines on Apple, PET, VIC, Budgeting on the Apple, Switching Cleanly from Text to Graphics on Apple, Atari Cassette Boot-tapes, Atari Variable Name Utility, Atari Program Library, Train your PET to Run VIC Programs, Interface a BSR Remote Control System to PET, A General Purpose BCD to Binary Routine, Converting to Fat-40 PET.

December 1981: Saving Fuel $$ (multiple computers: versions for Apple, PET, and Atari), Unscramble Game (multiple computers), Maze Generator (multiple computers), Animating Applesoft Graphics, A Simple Printer Interface for the Apple II, A Simple Atari Wordprocessor, Adding High Speed Vertical Positioning to Atari P/M Graphics, OSI Supercursor, A Look At SuperPET, Supermon for PET/CBM, PET Mine Maze Game.

January 1982: Invest (multiple computers), Developing a Business Algorithm (multiple computers), Apple Addresses, Lowercase with Unmodified Apple, Cryptogram Game for Atari, Superfont: Design Special Character Sets on Atari, PET Repairs for the Amateur, Micromon for PET, Self-modifying Programs in PET BASIC, Tinymon: a VIC Monitor, Vic Color Tips, VIC Memory Map, ZAP: A VIC Game.

May 1982: VIC Meteor Maze Game, Atari Disk Drive Speed Check, Modifying Apple's Floating Point BASIC, Fast Sort For PET/CBM, Extra Atari Colors Through Artifacting, Life Insurance Estimator (multiple computers), PET Screen Input, Getting The Most Out Of VIC's 5000 Bytes.

August 1982: The New Wave Of Personal Computers, Household Budget Manager (multiple computers), Word Games (multiple computers), Color Computer Home Energy Monitor, Intelligent Apple Filing Cabinet, Guess That Animal (multiple computers), PET/CBM Inner BASIC, VIC Communications, Keyprint Compendium, Animation With Atari, VIC Curiosities, Atari Substring Search, PET and VIC Electric Eraser.

September 1982: Apple and Atari and the Sounds of *TRON*, Commodore Automatic Disk Boot, VIC Joysticks, Three Atari GTIA Articles, Color Computer Graphics, The Apple Pilot Language, Sprites and Sound on the Commodore 64, Peripheral Vision Exerciser (multiple computers), Banish INPUT Statements (multiple computers), Charades (multiple computers), PET Pointer Sort, VIC Pause, Mapping Machine Language, Editing Atari BASIC With the Assembler Cartridge, Process Any Apple Disk File.

January 1983: Sound Synthesis And The Personal Computer, Juggler And Thunderbird Games (multiple computers), Music And Sound Programs (multiple computers), Writing Transportable BASIC, Home Energy Calculator (multiple computers), All About Commodore WAIT, Supermon64, Perfect Commodore INPUTs, Atari Autonumber, Copy VIC Disk Files, Commodore 64 Architecture.

February 1983: How The Pros Write Computer Games, 12 Joysticks Compared, Slalom (a game in 3-D for multiple computers), Super Shell Sort For PET, Atari SuperFont Plus, Creating Graphics On The VIC, Joysticks And Sprites On The 64, Bi-Directional VIC Scrolling, Commodore 64 Video: A Guided Tour, The Atari Cruncher, Easy Apple Editing, VIC Custom Characters For Games.

March 1983: An Introduction To Data Storage (multiple computers), Mass Memory Now And In The Future, Games: Closeout, Boggler, Fighter Aces, Letter And Number Play (all for multiple computers), VIC Music, Direct Atari Disk Access, TRS-80 Color Computer Data Base, Apple Subroutine Capture, PET Quickplot, TI Graphics Made Easy, VIC and Atari Memory Management.

April 1983: Selecting The Right Word Processor (multiple computers), VIC and Atari Word Processor Programs, Typing Teacher, TI Matchem, Retirement Planner (multiple computers), Air Defense (multiple computers), Dr. Video (Commodore), Video 80 (Software for 80 Columns on the Atari), Color Computer Tester, Timex/Sinclair Sound, Estimating TI Memory, Magic Commodore BASIC.

Home and Educational COMPUTING! (Fall 1981 and Summer 1981 – count as one back issue): Exploring The Rainbow Machine, VIC As Super Calculator, Custom Characters, Alternate Screens, Automatic Line Numbers, Using The Joystick (Spacewar Game), Fast Tape Locater, Window, VIC Memory Map.

Back issues are $3 each or six for $15. Price includes freight in the US. Outside the US add $1 per magazine ordered for surface postage. $4 per magazine for air mail postage. All back issues subject to availability.

In the Continental US call
TOLL FREE 800-334-0868
(In NC Call 919-275-9809)

Or write to **COMPUTE!** Back Issues, P.O. Box 5406, Greensboro, NC 27403 USA. Prepayment required in US funds. MasterCard, Visa and American Express accepted. North Carolina Residents add 4% sales tax.

NEWS & PRODUCTS

Game Cartridges For TI, VIC And Atari

Romox has adapted some of its Atari games into cartridges for the TI-99/4A and VIC-20 computers. The games include *Ant Eater*, *Princess and the Frog*, and *Typo*. In addition, the company has released a new space adventure game for the Atari – *Attack at EP-CYG-4*.

The cartridges for the TI do not make use of the Texas Instruments GROM, so they are limited to 8K of memory. The VIC cartridges can make use of up to 32K. The suggested price for each game is $44.95.

• *Ant Eater* is a two-player survival game. The players control the ants, which must risk battle with the anteater to gather food and return it to their colony.

• *Typo* is an educational spelling and typing drill combined with a space maze. The drill consists of random letters, words and phrases, or the user can enter and be tested on his or her own list.

• In *The Princess and the Frog*, a two-player game, the object is to cross a field of jousting knights, navigate the castle moat, kiss the princess, and be transformed from frog to prince – all within the space of 60 seconds.

• *Attack at EP-CYG-4* puts you in command of a flying saucer assigned to attack the cities on the planet below. The planet has 20 areas to navigate and three levels of difficulty.

Romox, Inc.
501 Vandell Way
Campbell, CA 95008
(408)374-7200

Graphic Design Aids For TI

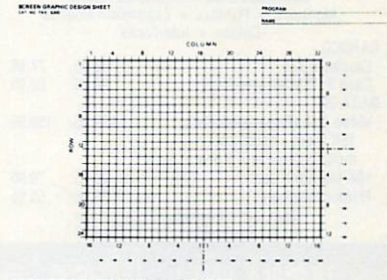
TENEX Computer Marketing Systems has designed two forms to assist the TI-99/4A programmer in graphic design.

The Screen Graphic design sheet is divided into 24 rows of 32 columns, allowing simple layout of text and characters. Another scale divides the sheet into 192 rows of 256 characters, assisting with the definition of sprite coordinates.

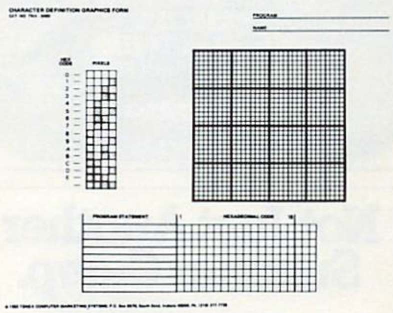
The Character Definition form displays a four-character by four-character matrix that can be used to design anything up to the largest sprite. The form also contains a pixel to hex code conversion chart, and space for program statements.

The forms are available in 40-sheet pads for $1.95 each.

Screen graphic design sheet for the TI.



Character definition graphics form for the TI.



TENEX Computer Marketing Systems, Inc.
P.O. Box 6578
South Bend, IN 46660
(219)277-7726

Action Games For Apple And Atari

Penguin Software produced two new games for the Apple and has converted another program for use on the Atari. Each game sells for $19.95.

Thunderbombs finds you in the midst of a swarm of alien dronebombs. Wipe out one wave of attackers and you're greeted with a new group of ships with different tricks. Requires 48K Apple with disk drive.

In *Crime Wave*, you represent the forces of law and order trying to stop a wave of bank robberies. Make your arrests before the robbers accumulate enough cash to buy a Robot Rammer, the latest in anti-peace officer technology. Requires 48K Apple with disk drive.

Spy's Demise is an arcade-type game in which you solve a puzzle by gathering dossiers, tapes, and microfilms, while eluding the embassy guards. Requires 32K Atari for disk version, or 24K Atari for tape version.

Penguin Software
830 4th Avenue
Geneva, IL 60134
(312)232-1894

No-Stick Joystick

Suncom has introduced a joystick-type game controller that has no stick. The Joy•Sensor is designed to simulate joystick movement through two activation panels.

When lightly touched by the player, sensors in the panels cause the designated movements on the screen. One sensor panel controls direction, the other controls firing. The firing panel accommodates right- or left-handed players and includes rapid fire.

Joy•Sensor, which sells for $34.95, is compatible with the TI-99/4A, Atari, and Commodore computers.

In addition, Suncom has introduced two new joysticks.

The Starfighter for the Apple uses thick-film resistive printing technology rather than the conventional potentiometers with a mechanical linkage. As a result, the "feel" of the Starfighter is described as smooth and pleasant. The joystick includes a throw adjuster that may be varied from 20 degrees to 40 degrees, fire buttons for right- and left-handed players, and dual axis centering trimmers. It sells for $49.95.

Suncom's Tac-2 joystick is made to accommodate the larger hand size and increased strength of adult game players. It includes a large, arcade-style ball top handle, a larger base, a longer cord, and a cone-shaped throw limiter. It includes right- and left-handed fire buttons, and, at $19.95, is compatible with Atari, Commodore, and Texas Instruments computers.

All three products include a two-year factory warranty.

Suncom, Inc.
605 E. Anthony Trail
Northbrook, IL 60062
(312)291-9780

The Suncom Joy•Sensor joystick simulator.

High-Speed Disk System For 64

cgrs Microtech has developed a high-speed floppy disk system for the Commodore 64. The PEDISK II is available in single or dual drive versions and can accommodate 3-inch, 5 ¼-inch, or 8-inch disk drives.

The system includes a controller circuit assembly, cable, and standard disk drives. It features IBM-CP/M compatibility and the ability to transfer data to computer memory at 250,000 bits per second.

Each PEDISK is supplied with the PDOS operating system, which includes a full set of utilities and BASIC commands. The C540-1, a single drive 5¼-inch system, retails for $595.

cgrs Microtech
P.O. Box 102
Langhorne, PA 19047
(215)757-0284

Apple Word Processor

Write Away, a word processor for the Apple IIe, can be used for home or office work and makes use of the features of any printer. A mailmerge/form letter feature and data base utilities are included with the program.

The program, which sells for $175, is compatible with most popular 80-column cards.

Midwest Software Associates
P.O. Box 301
St. Louis, MO 63074
(800)835-2264 ext. 467

TI-99/4A Directory

A wealth of information on and about the TI-99/4A computer is provided in Micronova's *Home Computer Directory for the 99/4A.*

The 52-page directory includes information on new TI equipment, TI hotlines and contacts, clubs and user groups, third-party software listings, technical information, and on-line data bases.

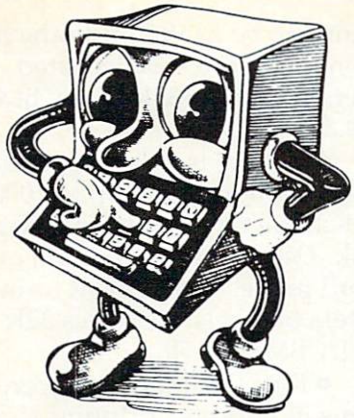
The directory is available for $4.95.

Micronova
P.O. Box 1058
Northampton, MA 01061

Timex/Sinclair Selections

D. Lipinski Software has introduced a group of programs for the 16K Timex/Sinclair computers. All programs are self-expanding to fit any memory configuration and are available on cassette for $10.

- *Tutor* is a quiz-making program that enables you to tailor tests for each student.
- *List* is a data entry and retrieval program for up to four categories of information.



INTEC
RAM BOARDS
Lifetime Warranty

ATARI

32K	400/800	**59.95**
48K	400	**84.95**
64K	400	**99.95**

PRINTERS

EPSON FX80 .	**529.95**
NEC 8023 . . .	**439.95**
Prowriter 8510	**369.95**

INTERFACES

Microbits MPP 1100 *Atari*	**79.95**
Grappler + *Apple* . . .	**129.95**

Ordering Information:

We accept M/C, VISA, Money Orders, and Cashier Checks. Sorry No C.O.D.'s.
SHIPPING: Add 3% UPS ($3.00 Minimum). APO/FPO 5% ($5.00 Minimum). Credit Cards add 3%. California Residents add 6% Sales Tax.

MICRO MERCHANT

290 N. 10th Street
P.O. Box 1516
Colton, Ca 92324

ORDER LINE

714-824-5555

Get Omni quality for as little as $1.99... even if all you want is a 10 pack.

Call toll-free for great savings on Omni's complete line of 5¼" and 8" premium disks. Each is certified error-free at a minimum of twice the error threshold of your system. Each is rated for more than 12 million passes without disk-related errors or significant wear. And each is precision fabricated to exceed all ANSI specifications with such standard features as reinforced hub rings and Tyvec sleeves. Get next day shipment and an unconditional, no hassle money-back guarantee.

ARCHIVE

152 Boston Turnpike
Shrewsbury, MA 01545
(800) 343-0314;
In Mass:
(617) 756-2960

Now, complete line of 8" and 5¼" disks, including 96 TPI and special format.

Call toll free
(800) 343-0314
In Mass: (617) 756-2960

Call if you're not sure which disk is compatible with your system. Call for prices on 96 tpi and special formats. We offer an unconditional money-back warranty. We're here to help.

Be sure to indicate system/drive name and model # below.

	5¼" disks Cost per 10 pack	5¼" disks Quantity	8" disks Cost per 10 pack	8" disks Quantity	Total Cost
Single side/single density	$19.90	______	$24.90	______	$______
Single side/double density	$23.90	______	$31.90	______	$______
Double side/single density	–––	______	$34.90	______	$______
Double side/double density	$37.50	______	$37.50	______	$______
Flip/Floppy reversible	$39.90	______	$39.90	______	$______
Plastic library case (in lieu of soft storage box)	$ 2.99	______	$ 3.49	______	$______
Shipping and handling ($2.00 first 10 pack, 40¢ additional 10 packs. Continental U.S. only.)					$______
5% sales tax (Mass only)					$______
☐Check ☐COD ☐Master Card ☐VISA				Total	$______

Card # ______________ Exp. ________ Name ______________________________

System/drive model # ______________ Address ______________________________

Tel. ______________ ______________ C-7

• *Check* is a checkbook program for 200-400 transactions. It includes several print options and allows correction of entry errors.

• *Inex* is an income and expense sheet that can keep track of more than 99 separate accounts.

• *Stock* keeps track of your stock portfolio and allows complete listings to screen or printer.

• *Inven* is an inventory accounting program for quantity purchased, sold, balance on hand, cost, replacement costs, and accrued values.

D. Lipinski Software
2737 Susquehanna Road
Roslyn, PA 19001

PET/64 Programs

Midwest Software has released five new programs for the Commodore PET and 64 computers.

• *Script Ease* is a 40-column word processing program designed to be useful for both children and more sophisticated users. Requires 32K PET/CBM; $39.50.

• *Datalog* is a data base that allows you to create up to 1000 200-character records on a 4040 disk. *Datalog* interfaces with most word processors to print form letters or labels. Requires 32K PET/CBM; $39.50.

• *Date Due* manages overdue items in libraries, printing a variety of reports. Requires 16K PET or 64, and disk drive; $39.50.

• *Ledger* is a financial package for personal or school accounts. It handles up to 300 transactions in any number of accounts. Requires 16K PET or 64 and disk drive; $29.50.

• *Multiple Choice* creates up to 150 question-and-answer sets per disk file. Any number of questions can be selected from the bank and randomized if desired. Tests can be taken on screen or printed on paper. Requires 16K PET or 64. Supplied only on disk, but can be saved and used on tape; $29.50.

Midwest Software
Box 214
Farmington, MI 48024
(313)477-0897

Joystick For Atari And VIC

AMIGA has introduced a precision joystick for use with Atari and VIC-20 computers. The Power-Stick, which has a short, 1½-inch handle for faster, more direct control, offers eight-way response.

The joystick is molded from high-impact plastic, and includes fire buttons on both sides of the controller. It comes with a carrying case and an extra-long cord. Power-Stick sells for $9.95 each.

AMIGA Corporation
3350 Scott Boulevard
Santa Clara, CA 95051
(408)748-0222

APPLE COUNTRY LTD.®

YOU'RE GONNA LOVE THESE ROCK BOTTOM PRICES!

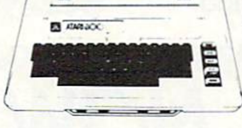
ACTION	99.00	68.95
ADVANCED MUSIC SYSTEM	29.95	21.95
AE	34.95	24.49
ANTI-SUB PATROL	29.95	21.95
APPLE CIDER SPIDER	39.95	27.95
ARCADE MACHINE	59.95	41.49
ARMOR ASSAULT	39.95	27.95
ATARI BOOKKEEPER KIT	249.95	179.95
ATARI MACRO ASSEM/TEXT ED	89.95	64.49
ATARI MICROSOFT BASIC II	89.95	64.49
ATARI MUSIC COMPOSER-ROM	39.95	29.95
ATARI PILOT EDUCATION-ROM	129.95	94.95
ATARI PROGRAMMER KIT	59.95	45.95
ATARI SPEED READING	74.95	54.95
ATARI TECH USER NOTES	29.95	21.95
ATARI TOUCH TYPING	24.95	17.95
ATARI WRITER	79.95	56.95
BANDITS	34.95	24.95
BANK STREET WRITER	69.95	49.95
BASIC COMPILER	99.95	68.95
BATTLE FOR NORMANDY	39.95	27.95
BATTLE OF SHILOH	39.95	27.95
BILESTOAD	29.95	21.95
THE BLADE OF BLACKPOOL	39.95	27.95
BOOK OF ATARI SOFTWARE'83	19.95	14.95
BUG/65	34.95	24.49
C/65	80.00	54.95
CAP'N COSMO	34.95	18.95
CASTLE WOLFENSTEIN	29.95	21.95
CENTIPEDE-ROM	44.95	31.95
CHOPLIFTER - ROM	44.95	31.95
COCO	49.95	34.49
COCO II	39.95	27.95
COLOR PRINT	39.99	27.95
COMMUNICATOR KIT	279.95	214.95
CONVERSATIONAL FRENCH	59.95	42.95
CONVERSATIONAL SPANISH	59.95	42.95
THE COSMIC BALANCE	39.95	27.95
CROSSFIRE-ROM	44.95	29.95
CYTRON MASTERS	39.95	27.95
DA FUZZ	44.95	31.95
DATA PERFECT	99.95	74.95
DATALINK	39.95	27.95
DAVID'S MIDNIGHT MAGIC	34.95	24.49
DEADLINE	49.95	34.49
DEFENDER	44.95	31.95
DE RE ATARI	19.95	14.49
DIG DUG	44.95	31.95
DISK MANAGER	29.95	21.95
DISK WORKSHOP	34.95	24.49
DISKETTE INVENTORY SYSTEM	24.95	17.49
DISKEY	49.95	34.49
DISKSCAN	40.00	28.00
DISKWIZ	29.95	21.95
DIVISION I	44.95	31.95
DNIEPER RIVER LINE	30.00	21.95
DODGE RACER	34.95	24.49
EASTERN FRONT (1941)	29.95	21.95
EDIT 6502-ROM	199.95	144.95
THE EDUCATOR KIT	164.95	109.95
FACEMAKER	34.95	24.49
FANTASTIC VOYAGE-ROM	34.95	26.49
FILE MANAGER +	99.95	68.95
FINANCIAL WIZARD	59.95	39.95
FLAME LORDS	34.95	24.95
FLASH GORDON-ROM	34.95	26.95
FLIP OUT	29.95	21.95
FIREBIRD-ROM	39.95	27.95
FORT APOCALYPSE	34.95	24.49
FROGGER	34.95	24.49
GALAXIAN	44.95	31.95
GHOST ENCOUNTERS	29.95	21.95
GHOSTLY MANOR	24.95	17.95
GLOBE MASTER	29.99	21.95
GORF-ROM	44.95	28.49
GRAPHIC GENERATOR	24.95	17.49
GRAPHIC MASTER	39.95	27.95
THE HOME ACCOUNTANT	74.95	54.95

JOURNEY TO THE PLANETS	29.95	21.95
JUMPMAN	39.95	27.95
KIDS AND THE ATARI-BOOK	19.95	13.95
KINDERCOMP	29.95	21.95
K-RAZY SHOOTOUT-ROM	49.95	34.49
KING ARTHUR'S HEIR	29.95	21.95
LEGIONNAIRE	35.00	25.00
LETTER PERFECT (40/80)	149.95	109.95
LETTER PERFECT UTILITY	29.95	21.95
LUNAR LEEPER	29.95	21.95
MASTER MEMORY MAP	6.95	4.95
MASTER TYPE	39.95	27.95
MATING ZONE	29.95	21.95
MATCH BOXES	29.95	21.95
MAURAUDER	34.95	24.49
MAX/65 (WITH OS/A+)	80.00	54.95
MICKEY IN GREAT OUTDOORS	49.95	35.95
MINER 2049'ER-ROM	49.95	34.49
MISSILE COMMAND-ROM	34.95	26.49
MONSTER SMASH	29.95	21.95
MORL	44.95	31.95
NAUTILUS	34.95	24.49
NECROMANCER	34.95	24.49
THE NIGHTMARE	29.95	21.95
OPERATION WHIRLWIND	34.95	24.49
OS-A+ & BASIC A+	80.00	54.95
PAINT	39.95	29.95
PINBALL	29.95	21.95
P.M.ANIMATOR	34.95	24.49
POKER-S.A.M.	24.95	17.95
PREPPIE II	34.95	24.49
PYRAMID PUZZLER	44.95	31.95
QIX	44.95	31.95
RASTER BLASTER	29.95	21.95
READING FLIGHT	44.95	31.95
ROUNDABOUT	29.95	21.95
SAMMY LIGHTFOOT	34.95	24.49
SAMMY THE SEA SERPENT	23.95	16.95
SEA DRAGON	34.95	24.49
SEA FOX	29.95	21.95
747 LANDING SIMULATOR	22.95	16.95
SHADOW WORLD	34.95	24.49
SHAMUS-ROM	44.95	31.95
SNEAKERS	29.95	21.95
SOFTWARE AUTO-MOUTH (SAM)	59.95	41.49
SPEED READ PLUS	59.95	41.49
SPEEDWAY BLAST	29.95	21.95
SPELLING BEE GAMES	39.95	27.95
SPELL WIZARD	79.95	54.95
STARCROSS	39.95	27.95
STAR MAZE	44.95	31.95
STAR RAIDERS-ROM	44.95	31.95
STAR WARRIOR	39.95	27.95
STORY MACHINE	34.95	24.49
SUPERMAN III	49.95	35.95
SURVIVOR	34.95	24.49
SWIFTY TACH MASTER	29.95	21.95
SYN ASSMBLER	49.95	34.49
TELECOM	69.95	49.95
TELETALK	49.95	36.95
TELETARI	39.95	27.95
TEMPLE OF APSHAI	39.95	27.95
TEXT WIZARD I	99.95	68.95
TIGERS IN THE SNOW	39.95	27.95
TIME WISE	29.95	21.95
TYPE ATTACK	39.95	27.95
TUTTI FRUTTI	24.95	17.95
VC	25.00	17.49
VISICALC	199.00	149.95
WARLOCK'S REVENGE	34.95	24.49
WAY OUT	39.95	27.95
WIZARD&PRINCESS HIRES ADV	32.95	22.95
WIZARD OF WOR - ROM	44.95	31.95
YOUR ATARI COMPUTER-BOOK	16.95	12.95
ZAXXON	39.95	27.95
ZORK I II OR III	39.95	27.95

AMDEK COLOR I PLUS MONITOR		319.00
APPLE EMULATOR	CBM64	79.50
CARDBOARD (6 CART + RESET)	VIC-20	70.00
THE CARD? (FOR PARALLEL PRTR)	CBM64&VIC-20	58.00
CARDETTE (CASSETTE INTFCE)	CBM64&VIC-20	22.00
CARDAPTER/1 ATARI 2600 INTFCE	VIC-20	50.00
CARDWRITER/1 LIGHT PEN W/CASS	CBM64&VIC-20	16.50
COMMODORE 64 HOME COMPUTER	CBM64	369.50
COMMODORECOLOR PLOTTER	CBM64&VIC-20	179.50
COMMODORE 1530 DATASETTE	CBM64&VIC-20	65.00
COMMODORE 1541 DISK DRIVE	CBM64&VIC-20	329.50
COMMODORE 1525 PRINTER	CBM64&VIC-20	329.50
COMMODORE 1600 MODEM	CBM64&VIC-20	95.50
COMMODORE COLOR MONITOR	CBM64&VIC-20	269.50
COMMODORE 1650 AUTODIAL MODEM	CBM64&VIC-20	169.50
MICRO EXPANSION CHASSIS	CBM64	35.75
VIDEO PAK 80 (80 COLUMN)	CBM64	129.00
VIDEO PAK 80 WITH CP/M	CBM64	219.00
ATARI 800 48K		$489-$100 REBATE=389.00
ATARI 1200 64K		$519-$100 REBATE=419.00
ATARI NUMERIC KEYPAD		94.95
ATARI PROGRAMMER KIT		49.95
410 RECORDER		72.95
810 DISK DRIVE		424.95
850 INTERFACE MODULE		169.95
C. ITOH PROWRITER I		394.95
C. ITOH PROWRITER II		649.95
C. ITOH STARWRITER		1325.00
CASSETTE 'N CARTRIDGE FILE		21.95
ELEPHANT SS/SD DISK		10/16.95
ELEPHANT SS/DD DISK		10/19.95
ELEPHANT DS/DD DISK		10/25.95
EPSON FX-80 W/TRACTOR		LOW!!
EPSON MX-100 F/T		LOW!!
FLIP'N-FILE		21.95
GEMINI-10 PRINTER		LOW!!
GEMINI-15 PRINTER		LOW!!
IN HOME 400 KEYBOARD		94.95
INTEC 32K RAM		69.95
INTEC 48K RAM		119.95
MOSAIC 64K RAM SELECT		149.95
HAYES SMARTMODEM 300 BD		194.95
HAYES SMARTMODEM 1200 BD		499.95
NEC 8023 PRINTER		459.95
NEC 12" HIRES GREEN SCRN		149.95
NEC 12" ECONO GREEN SCRN		79.95
NOVATION J-CAT MODEM		109.95
NOVATION SMART-CAT 103		179.95
NOVATION SMART-CAT 212		424.95
PERCOM SS/SD/1DR (88K)		419.95
PERCOM SS/DD/1DR (176K)		539.95
PERCOM SS/DD/2DRS (352K)		859.95
PERCOM DS/DD/1DR (352K)		649.95
PERCOM DS/DD/2DRS (704K)		939.95
SIGNALMAN MK II MODEM		79.95
USI 12" AMBER MONITOR		159.95
VERSAWRITER GRAPH TABLET		239.95
WICO JOYSTICK		21.95
WICO REDBALL JOYSTICK		24.95
WICO DELUXE JOYSTICK		29.95
WICO TRACKBALL		49.95
WICO 12 FT EXTENSION CORD		6.95

commodore

AGGRESSOR	CART/VIC20	27.50
ALIEN SOCCER	CASS/VIC20	10.25
ANDROMEDA CONQUEST	CASS/C64	12.50
APE ESCAPE	DISK/VIC20	27.50
BLADE OF BLACKPOOLE	DISK/C64	27.50
CAVE-IN	CART/VIC20	27.50
CHECKBOOK	CASS/VIC20	13.75
COMPUTER FOOTBALL STRATEGY	CASS/C64	11.00
COMPUTER STOCKS & BONDS	CASS/C64	13.75
CONCENTRATION	CASS/VIC20	11.00
CRITICAL MASS	DISK/C64	27.50
CROSSFIRE	CASS/VIC20	21.00
DEADLINE	DISK/C64	34.50
DEADLY DUCK	CART/VIC20	24.00
DEADLY SKIES	CART/VIC20	27.50
DEMON ATTACK	CART/VIC20	27.50
DRELBS	CASS&DISK/64	24.00
FACEMAKER	DISK/C64	24.00
FAST EDDY	CART/VIC20	27.50
FINAL ORBIT	CART/C64	24.00
FT. APOCALYPSE	CASS&DISK/64	24.00
FROGGER	DISK/C64	24.00
FRUIT FLY	CASS/VIC20	9.00
FUEL PIRATES	CASS/VIC20	11.00
GALACTIC BLITZ	CASS/VIC20	17.25
GALACTIC CROSSFIRE	CASS/VIC20	10.25
GOLD FEVER	CART/VIC20	27.50
GOLD MINE	DISK/VIC20	27.50
GRIDRUNNER	CART/64/VIC20	27.50
HANGMAN	8KCASS/VIC20	6.25
HARRIER	CASS/VIC20	24.00
HESMON MONITOR	CART/64&VIC20	27.50
HESWRITER W/P	CART/VIC20	27.50
HESWRITER 64	CART/C64	31.00
HEY DIDDLE DIDDLE	DISK/C64	21.00
IN SEARCH OF THE MOST AMAZING THING	DISK/C64	27.50
JUMP MAN	DISK/C64	27.50
KINDERCOMP	DISK/C64	21.00
MARTIAN RAIDER	DISK/VIC20	13.75
METEOR	8KCASS/VIC20	7.50
MONSTER MAZE	CART/VIC20	27.50
MULTISOUND SYNTH.	DISK/VIC20	13.75
NUMBERCHASER	16KCASS/VIC20	17.25
NUMBER CRUNCH	CART/VIC20	27.50
NUMBER GULPER	8K CASS/VIC20	17.25
PHAROH'S CURSE	CASS&DISK/64	24.00
PREDATOR	CART/VIC20	27.50
PROTECTOR	CART/VIC20	30.25
QUICK BROWN FOX W/P	CART/VIC20/64	44.75
RAID ON ISRAM	CASS/VIC20	13.75
REAGANOMICS	CART/VIC20	27.50
REPTON	DISK/C64	27.50
RESCUE AT RIGEL	16KCASS/VIC20	21.00
RETRO BALL	CART/C64	27.50
RICOCHET	8K CASS/VIC20	13.75
ROBOT PANIC	CART/VIC20	27.50
SCORPION	CART/VIC20	27.50
SHAMUS	CART/VIC20	27.50
SHARK TRAP	DISK/VIC20	13.75
SIDEWINDER	8K CASS/VIC20	21.00
SIMON	CASS/VIC20	11.00
6502 PROF DEV SYS	CASS/64/VIC20	21.00
SKI RUN	8KCASS/VIC20	12.50
SNAKE BYTE	CART/C64	24.00
SNOOPER TROOPS #1	DISK/C64	27.50
SPACE ATTACK	8K CASS/VIC20	6.25
SPIDER CITY	CART/64/VIC20	27.50
SPORTS SEARCH	CASS/VIC20	13.75
SQUISH'EM	CART/64/VIC20	24.00
STARCROSS	DISK/C64	27.50
SURVIVOR	CASS&DISK/C64	24.00
SWARM!	CASS/VIC20	21.00
SWORD OF FARGOAL	16KCASS/VIC20	21.00
SYN THE SOUND/MUSIC	CART/VIC20	41.00
TELENGARD	CASS/C64	15.75
TORG	CASS/VIC20	12.25
TURMOIL	CART/VIC20	27.50
TURTLE GRAPHICS	CART/VIC20	27.50
TURTLE GRAPHICS II	CART/C64	41.00
TYPE ATTACK	CART/64/VIC20	27.50
THE VEIN GAME	DISK/VIC20	24.00
VIC FORTH	CART/VIC20	41.25
VIC-MEN	8KCASS/VIC20	15.00
VIC MUSIC COMPOSER	CART/VIC20	30.00
VICTREK	CASS/VIC20	12.25
VIC VANGO	CASS/VIC20	9.00
VIDEO SEARCH	CASS/VIC20	13.75
WORD SEARCH	CASS/VIC20	13.75
ZORK I II OR III	DISK/C64	27.50



ATARI PERCOM RanaSystems Hayes FRANKLIN NEC IBM MOSAIC 64K RAM SELECT EAGLE COLUMBIA DATA PRODUCTS, INC TEXAS INSTRUMENTS OKIDATA

FOR FASTEST DELIVERY: CASHIER'S CHECK OR VISA/MASTERCARD (NO EXTRA CHARGE FOR CARDS. **INCLUDE NUMBER, EXPIRATION DATE, NAME, ADDRESS & PHONE**). PERSONAL CHECK ALLOW 2 WEEKS TO CLEAR. PURCHASE ORDER MUST INCLUDE CHECK. **SHIPPING & HANDLING:** CONTINENTAL U.S. 5% ($5 MIN), U.P.S. **STREET ADDRESS REQUIRED**; APO FPO ALASKA HAWAII & MONITORS 5% ($10 MIN); FOREIGN 15% ($15 MIN). **INCLUDE PHONE NUMBER WITH ALL ORDERS.** ALL ITEMS ARE NEW WITH MANUFACTURER'S WARRANTY. APPLE COUNTRY, LTD. CANNOT GUARANTEE THE MERCHANTABILITY OF ANY PRODUCT. PRICES ARE SUBJECT TO AVAILABILITY & CHANGE WITHOUT NOTICE. DUE TO OUR LOW PRICES, ALL SALES ARE FINAL. RETURNED MERCHANDISE IS SUBJECT TO A RESTOCKING FEE & MUST COME IN ORIGINAL UNDAMAGED CARTON WITH RMA NUMBER. NO SOFTWARE EXCHANGES. CALIFORNIA RESIDENTS ADD 6% TAX. SEND $1 FOR NEW FALL CATALOG (GOOD TOWARD FIRST PURCHASE). APPLE COUNTRY, LTD. IS A **DISCOUNT MAIL ORDER HOUSE FOR THE MICROCOMPUTER INDUSTRY** & IS A CALIFORNIA CORPORATION NOT AFFILIATED WITH APPLE COMPUTER INC. APPLE IS TRADEMARK OF APPLE COMPUTER INC. 6.1.83

Call us... we can help! (619) 765-0239

P.O. Box 1099,
Julian, Calif. 92036

Apple Country, Ltd. is a DISCOUNT MAIL ORDER HOUSE for the micro computer industry and is a California corporation not affiliated with Apple Computer Inc. Apple is a trademark of Apple Computer Inc.

Color Computer Games

Five new games for the TRS-80 Color Computer have been released by Radio Shack. The programs include adventure, arcade-style, and card games, and all include high-resolution graphics. Unless otherwise noted, all games require a 16K machine.

The games are:

- *Klendathu,* a space adventure based on the book *Starship Troopers* by Robert A. Heinlein. The players, fighting in the "Bug Wars" on the planet Klendathu, drop from their starship to the planet's surface where they must destroy as many bugs as possible before their special suits run out of energy. The game is available on cassette for $14.95.
- *Canyon Climber,* an action game in which the player must maneuver a climber through three levels of play, setting dynamite charges while avoiding mountain goats, Indian arrows, and falling rocks. *Canyon Climber* is available in a ROM Pak for $34.95.
- *Bridge Tutor,* an instructional program for beginning and average bridge players. The program includes 100 bridge hands to play or analyze. Messages signal when a wrong bid is made or a wrong card is played. The player can choose an offensive or defensive hand to play, or the computer can play all four hands. *Bridge Tutor* is available for $34.95 for computers with a minimum of 4K.
- *Doubleback,* a game in which players use a joystick to encircle random objects on the screen with a moving line. Available in ROM Pak for $24.95.
- *Card Games,* a program that gives the player a choice of six games – solitaire, solo poker, last pirate, go fish, blackjack, and war. *Card Games* sells for $19.95.

Tandy Corporation/Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102

Improving Atari's Memory

Mosaic Electronics has introduced the Mosaic 64K RAM Select, a board for the Atari 400 and 800 computers. The 64K RAM Select is bus compatible for use with the Atari 16K or Mosiac 32K RAM boards.

Atari 800 owners can use the board to simulate Atari 1200 architecture, or the board can be configured for bank selection. The 64K Select, which sells for $199, is compatible with 8K and 16K ROM cartridges. It installs without solder, but requires a cable kit from Mosaic.

Mosaic Electronics
P.O. Box 708
Oregon City, OR 97045
(503)655-9574

Automatic Firing Module

The Blaster, a plug-in, adjustable speed, automatic firing module for Atari and VIC-20, has been produced by Questar.

The unit plugs in between the computer and the joystick and can be adjusted to fire from one to 20 shots per second. The Blaster, which retails for $12.95, can be turned off for maze games or for games that restrict the firing rate.

Questar Controls, Inc.
670 N.W. Pennsylvania Avenue
Chehalis, WA 98532
(206)748-8614

The Blaster automatic firing module.

Fast Action Space Game

A huge solar power station in deep space is the setting for *Gridrunner*, a fast-paced game for VIC-20, Commodore 64, and Atari computers.

In the game, produced by Human Engineered Software, the power station comes under attack by more than 20 waves of Droids bent on destroying the station as a prelude to overtaking Earth.

Defense of the station against the Droids and their array of weapons is up to Gridrunner and his plasma cannon. The game sells for $39.95.

Human Engineered Software
71 Park Lane
Brisbane, CA 94005

Atari BASIC Compiler

Monarch Data Systems has introduced *ABC*, a BASIC compiler for the Atari that translates BASIC programs into integer P-code that runs up to 12 times faster than the original.

The program allows compiled programs to run without the BASIC cartridge, allows larger arrays, accepts most BASIC programs with little modification, allows DIM, GOTO, GOSUB, and RESTORE, compiles at 100 lines per minute, and includes a utility to generate relocatable code.

The suggested price of the compiler, which requires one disk drive and 40K, is $69.95.

Monarch Data Systems
P.O. Box 207
Cochituate, MA 01778
(617)877-3457

64 Math Teacher

The Math Teacher, a program of math problems for first-graders through junior high school, has been produced for the Commodore 64 by CompuTech.

The program, available on tape for $39.95, drills students in addition, subtraction, multiplication, and division at four skill levels.

CompuTech
P.O. Box 7000-309
Redondo Beach, CA 90277

Mini Thermal Printer

A miniaturized thermal printer is available from Panasonic. The EUY-3T printer offers a 40-character line while measuring just under five inches wide.

Other specifications include: dot-addressable graphic capability, a single 5-Vdc power supply, printing speed of 1.2 lines per

TIRED OF PLAYING GAMES?

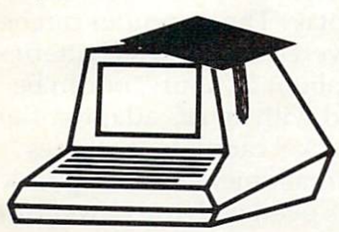
Get Down To Business With DATABOOKS-64

A Business Software Program

For The Commodore 64

DataBooks-64 is a complete single disk accounting system for a small business. Including General Ledger, Accounts Receivable, Accounts Payable, Billing and Invoicing - All in one program!

The Commodore 64 can do more than just play games. Its memory capacity and peripherals make it a powerful business tool. Its capabilities exceed some systems costing many times more! The DataBooks-64 accounting package also has hard-to-beat capability and flexibility, with many features not found in ANY other accounting package! Stop by a Commodore dealer and ask to see DataBooks-64 today!



LOOK AT THESE FEATURES!

- All data stored on one diskette - no "disk flipping" required.
- Sophisticated screen editor - works like a word processor.
- All posting is instantaneous - all reports are current up-to-the-minute.
- Full invoicing and billing
- Innovative "Quick Reference Guide" forms a direct link between the documentation and questions asked by the computer.
- Fully Menu driven.
- Fast - all files updated in only a few seconds.
- Access codes - protect your financial information
- Compatable with nearly any printer.

LOOK AT THE REPORTS PRODUCED!

- Ledger Totals Sheet (trial balance).
- Formal Balance Sheet.
- Profit & Loss (Income) Statement with Month and Year-to-date totals including percentages (ratios).
- Customer Invoices.
- Customer Statements.
- Receivables Invoice Register.
- Payables Invoice Register.
- Customer List.
- Vendor List.
- General Ledger (Month).
- General Ledger (Year).
- General Journal.
- Selective Journal (per account).

For more information and the name of your nearest dealer,
Please write or call (619) 223-4496

DMI Software Inc. 1866 Bacon Street San Diego, CA 92107

COMSTAR

INTRODUCING SOMETHING NEW AND USEFUL

THE ALOG PAGEWRITER

FOR ATARI 800/400

A Fast, Simple, Easy To Use and Inexpensive

WORD PROCESSOR

- Average Time To Master — 5 min
- Full Page Display Guide
- Help Screen With Command Summary
- Uses Standard ATARI Editing Keys
- Visible Adjustable Margins
- Stores Ten Pages on a Disc with No Swapping

Other Features Include:

Centering of Text • Set Tabs • Right Justification • Easy Text Insertion • Accepts Printer Control Commands

Requires at least 32k of memory and a 80 column printer.

List price, 39.95. Introductory price **$34.00.** Manual included.

DEALER INQUIRIES INVITED

AIR* SHIPPING WITHIN 2 DAYS

ATARI		VIC-20	
48K RAM (FOR 400)	$115	CARDBOARD (3 SLOT EXP.)	$ 33
64K RAM (FOR 400)	135	VIDEOPAK WITH 16K (40/80 COL)	250
ALIEN GROUP VOICE BOX (D.T)	119	VIDEOPAK WITH 64K (40/80 COL)	319
S.A.M. (D) 8K	46	PRINTER INTERFACE (PARALLEL)	55
BIT 3 80 COL. BOARD	289	KIDS AND THE VIC (BOOK)	17
TECHNICAL NOTES	25	16K RAM	85
BOX OF DISKS (10)	19	8K RAM	55
PROWRITER PRINTER	419	HES MON (ASSEMBLER) (C)	29
B KEY 400 (KEYBOARD)	99	HES WRITER (WORD PROC.) (C)	29
NEWPORT PROSTICK	27	TURTLE GRAPHICS (C)	29
MICROBITS MODEM	159	VIC FORTH (C)	49
PRINTER INTERFACE	89	QUICK BROWN FOX(WORD PROC.)(C)	54
THE ATARI ASSEMBLER (BOOK)	12	SHAMUS (C)	29
VAL FORTH (D) 24K	45	PROTECTOR (C)	33
PREPPIE (D.T) 16K	19	CHOPLIFTER (C)	34
EASTERN FRONT (D.T) 16K	24	APPLE PANIC (C)	34
MINER 2049ER (C)	35	VIC RABBIT (C)	35
STARBOWL FOOTBALL (D.T) 24K	25	UNWORD PROCESSOR (T) 5K	19
PILOT (C)	59	STARFIGHTER JOYSTICK	14
BASIC A + WITH OS/A + (D)32K	60	CARDETTE (CASS. INTERFACE)	33
ASTRO CHASE (D) 32K	24	CRUSH, CRUMBLE & CHOMP (T)21K	24
BAJA BUGGIES (D.T) 16K	23	SWORD OF FARGOAL (T) 21K	23
JUMP MAN (D) 32K	29	VICAT (T) 8K	20
CHOPLIFTER (C)	32	DEADLY DUCK (C)	28
ZAXXON (D.T)	29	TOTL MAIL LIST (T) 13K	19

HEAR ATARI SOUNDS THROUGH YOUR STEREO SPEAKERS WITH
STEREODAPTER — FOR ATARI 800
• NO ASSEMBLY REQUIRED • CAN USE STEREO HEADPHONES
• SHIELDED CABLE • ADJUST TONE & VOLUME WITH STEREO CONTROLS
STEREODAPTER WITH 16 FT CABLE $8 WITH 26 FT CABLE $10
DEALER INQUIRIES INVITED

C = CARTRIDGE D = DISK T = CASSETTE * MOST ITEMS

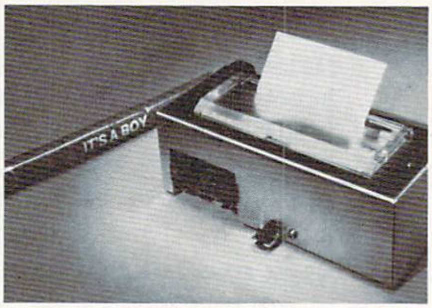
COMSTAR P.O. BOX 1730 GOLETA, CA 93116 (805) 964-4660

ORDERS ONLY: CALL TOLL FREE 800-558-8803
IN CALIF. (805) 964-4660 or send check, money order or credit card number and exp. date. Include $2.00 for shipping. Add 3% for Visa or MC (except Calif.). Calif. add 6% tax. There is a $2.50 charge for COD. Please include type of computer. (Checks—10 days to clear.)

second, bidirectional printing, and 3.18 mm characters.

The 14-ounce EUY-3T printers will retail for less than $100.

Panasonic Industrial Company
One Panasonic Way
Secaucus, NJ 07094



Panasonic's EUY-3T miniature thermal printer.

Radio Shack Portable Computer

Radio Shack has introduced a portable computer that includes built-in software, a built-in modem, and a full-size typewriter keyboard.

The TRS-80 Model 100 features instant power-on access to five ready-to-use programs contained in the 32K ROM. Programs are easily selected from the main menu by positioning the cursor and pressing ENTER.

The *Text* program allows the user to create and edit text files. *Schedl* serves as a note file, allowing the user to store and locate dates, times, appointments, or information. A character-by-character search function provides total recall.

Addrss organizes names, addresses, and telephone numbers. *Telcom* is the system's built-in communications program, which includes a direct-connect, auto-dial modem, and RS-232 interface. *Telcom* interfaces with *Addrss* to locate phone numbers.

The fifth built-in program is BASIC.

The Model 100 measures 2 x 11⅞ x 8½ inches, weighs about four pounds, and includes an 8-line, 40-character liquid crystal display. The computer can be powered by four AA batteries for about 20 hours, or can be used with an AC adapter. Built-in nickel cadmium batteries maintain memory for up to 30 days even with the power turned off.

The Model 100 is available with 8K RAM for $799, or with 24K RAM for $999. Optional 8K RAM add-ons can expand either version up to 32K.

Tandy Corporation/Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102

The TRS-80 Model 100 Portable Computer.

New Product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication.

Word Work

DesignWare has introduced two word-related educational games.

Spellicopter and *Crypto-Cube* are both available for the Apple II and Atari computers for $39.95.

In *Spellicopter*, the pilot

relies on keen memory, verbal, spelling, and navigational skills to rescue "stranded" letters and organize them into correctly spelled words.

Crypto-Cube gives puzzle lovers a challenge and a chance to build their vocabularies. The program has 50 puzzles built-in, and users can create their own word lists for the puzzle generator. The game features a rotating cube with crossword puzzle grids on each side. Players take turns uncovering letters to fill in the missing words.

DesignWare, Inc.
185 Berry Street
San Francisco, CA 94107

Learning Apple Pascal

Pascal for the Apple, by Iain MacCallum, is an introduction to Pascal as a first computer language. The 496-page book comes with a floppy disk to illustrate the written instruction. The book and disk are designed to be used concurrently.

Both experimental and practical programming are explored in the $33 package. A 48K Apple II, Apple IIe, or Apple II+ with two disk drives is required.

Prentice-Hall
Englewood Cliffs, NJ 07632

Workout On The Atari

BASIC Exercises for the Atari, by J. P. Lamoitier, takes Atari programmers through a range of applications including mathematics, physics, games, business, accounting, and statistics.

Each exercise in the $12.95, 252-page book takes the programmer from analysis of the problem through flowcharting, coding, and the final run.

Sybex
2344 Sixth Street
Berkeley, CA 94710
(415)848-8233

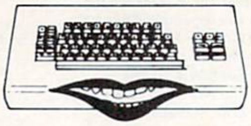
©

PRODUCT MART

PERSONAL PERIPHERAL PRODUCTS *presents:*

SPEAKEASY

VIC-20 SPEECH



COM-64 SPEECH

Cartridge, Instructions & Dictionary $79.00
JULY SPECIAL: Cassette Editor & Extension Speaker, Reg. $19.95, with **SPEAKEASY** ... **FREE**
Commodore 64 Adapter Board $12.95

BARE BONES BOARDS

	Assembled	Kit
8K Ram/VIC-20 Block Switched	$39.00	$29.95
16K Ram Expander for VIC-20	$64.95	$49.95
3 Slot Switched & Fused Board/VIC-20 .	$34.95	$24.95
NEW: 4 Slot Switched & Fused Board/COM-64	$44.95	$34.95
COMING SOON: Mighty Modem VIC 20/COM-64	$79.95	

ADD $2.00 Total Order Handling/Ill. Residents Add 6% Sales Tax

PERSONAL PERIPHERAL PRODUCTS
P.O. BOX 3423
FOX VALLEY MALL
AURORA, IL 60505 • (312) 961-2347
COM-64 & VIC-20 IS A TRADEMARK OF COMMODORE

RUNWAY 64

A FLIGHT SIMULATOR GAME FOR THE COMMODORE 64* COMPUTER. CASSETTE OR DISK. **$25.00**

RUNWAY 20

A FLIGHT SIMULATOR GAME FOR THE VIC-20* COMPUTER. REQUIRES 16K EXPANSION. COLOR, SOUND & JOYSTICK OPTION. CASSETTE OR DISK. **$25.00**

SKY PILOT

A FLIGHT SIMULATOR GAME FOR THE COMMODORE VIC-20* COMPUTER. CASSETTE OR DISK. REQUIRES 8K EXPANSION.
JOYSTICK OPTION **$18.00**

SUSIE SOFTWARE
709 WILSHIRE DRIVE
MT. PROSPECT, IL 60056
312/394-5165

*REGISTERED TRADEMARK OF COMMODORE BUSINESS MACHINES

VIC 20 RAM

24K RAM PAK $99.95
*16K RAM PAK $69.95
*8K RAM PAK $39.95
*3K RAM PAK $29.95
*8K EPROM BOARD (NO EPROMS) $39.95
*3 SLOT EXPANSION BOARD $24.95
*6 SLOT EXPANSION BOARD $49.95
*CASSETTE INTERFACE CABLE CONNECTS ANY CASSETTE RECORDER TO YOUR VIC $19.95
*PRINTER INTERFACE CABLE CONNECTS PRINTER TO YOUR VIC $49.95
SIX MONTHS LIMITED WARRANTY PARTS & LABOR
ALLOW THREE WEEKS FOR PERSONAL CHECKS
INCLUDE $3.00 SHIPPING AND HANDLING
ALL BOARDS TOP QUALITY
WITH GOLD PLATED CONTACTS

PAKS UNLIMITED
VISA — P. O. BOX 221 — MASTER CHARGE
MOORE, S. C. 29369
1-803-585-3400
24 HR. NO.

VIC-20-REG. T.M. COMMODORE BUS. MACH. INC.

VIC-20

Dealers call 212-499-5400 direct for pricing.

NEW ITEMS!

HARDWARE
UPA-20 Standard Centronics Cable $19.95
Driver Listing Included!
UCA-20 Universal Cassette Cable $19.95
Use any cassette machine!
16K RAM/ROM Board Jumpered for any 8K block
Socketed Board with support chips $24.95
Populated with 8K $49.95
Populated with 16K $79.95
2716 EPROMs for RAM/ROM Board ... @$7.50
3 Slot Memory Port Expander $19.95

SOFTWARE
WORDWIZ for the Unexpanded VIC-20 ... $14.95
A nice small Word Processor
Mailing List requires 8K minimum $14.95
Sorts, selects, and prints labels

2 FREE DISASSEMBLERS
when you send $1 for our CATALOG.

WORLD ELECTRONICS
177 27th Street
Brooklyn, N.Y. 11232
A Division of World International Trading Corp.

APPLE / ATARI / COMMODORE

MEASURE & CONTROL TEMPERATURE

DISPLAY GRAPHICS
HARD COPY OUTPUT
ALARMS & SETPOINTS
DISK FILE DATA STORAGE

1–256 Sensors
Precise to 1/100 Degree
Complete Software
$129.00 Package

American Data Cable, Inc.
2864 Ray Lawyer Drive, #205-352
P.O. Box 2212 • Placerville, CA 95667
(916) 622-3465

VIC & 64

BE A COPY C.A.D.
(CASSETTE AIDED DUPLICATOR) NOW YOU CAN MAKE BACKUP COPIES OF ALL THE COSTLY, NON-SAVEABLE CASSETTE PROGRAMS YOU BOUGHT.

OUR BACKUP V1.O UTILITY PROGRAM WILL LET YOU MAKE DUPLICATES THAT RUN.

BACKUP V1.O WILL WORK WITH A STANDARD 5K UNEXPANDED VIC. MEMORY EXPANSION IS REQUIRED TO COPY PROGRAMS LONGER THAN 3K BYTES.

$24.95
PLUS $2.00 SHIPPING & HANDLING

SOFTWARE PLUS
6201 SUITE C
GREENBACK LANE
CITRUS HEIGHTS, CA 95610

VISA, MASTERCARD, AND MONEY ORDERS
CA. RESIDENTS ADD 6% SALES TAX.

VIC IS A TRADEMARK OF COMMODORE

VIC-20
COMMODORE 64
APPLE II

THE RECIPE BOX

Now you can easily store and recall your favorite recipes on your Commodore or Apple computer. THE RECIPE BOX is a complete menu-driven disk system that comes with these additional features:
SEARCH BY INGREDIENT — Only have a pound of hamburger in the freezer? Let THE RECIPE BOX show you all the recipes that you have on file that use hamburger, or any other ingredient you choose.
SEARCH BY CATEGORY — Code your recipes as to breakfast, lunch, dinner, snacks, etc.
SEARCH BY CATEGORY/INGREDIENT — Any combination of the above.
AUTOMATIC MEASUREMENT — THE RECIPE BOX will automatically scale up or down the amount of ingredients you need according to how many servings you want.
SCREEN OR PRINTED OUTPUT — Have printed copies to use in the kitchen or give to friends.
THE RECIPE BOX requires one disk drive and will run on a 5K VIC-20, Commodore 64 or Apple II+. Please specify. Send check or money order for $19.95 to:

Aries Marketing Co.
P.O. Box 4196
4200 Shannon Drive
Baltimore, Md. 21205
Md. residents add 5% sales tax

Clear Plastic DUST COVER $7.95*
plus $2 shipping and handling

Fits standard keyboard and speech synthesizer.

Single Cassette Cables for TI Computers

Cassette Cable C-2000 .... $13.95
RS 232 Flat Ribbon C-2010 5ft. $24.95
RS 232 Flat Y (Printers) C-2020 5ft. $49.95
RS 232 Modem C-2030 5ft. $24.95
RS 232 Flat Y (Printer-Modem) C-2040 5ft. $49.95
TI-PIO Flat Centronics C-2050 5ft. $34.95
PIO Flat Centronics C-2060 5ft. $34.95

Guaranteed 90 days against workmanship defects. Repairable at low cost. Inquire about customized or hard-to-find cables. Add $2 shipping and handling.
*Florida residents add 5% sales tax.

Dealer inquiries invited.

GET COMPUTERIZED INC.
3550 S. Washington Ave., Dept. A
Titusville, FL 32780 ● (305) 268-4007

COPY CAT COMMODORE 64

COPIES DISK FILES
EXTRA FEATURES:

- Renames files
- Catalogs a diskett
- Scratches files
- Formats or news a disk
- Uses a M/L loader saver for fast speed
- Makes back-up copies of your programs

~~$49.80~~ NOW ONLY $19.95

Village Office Machines
4721 Euclid Avenue
Rolling Meadows, IL 60008
(312) 991-0880
COD

A Must For All Disk Drive Owners
DEALER INQUIRIES INVITED

VIC–20 USERS CARTRIDGE BACK-UP SYSTEM

PROTECT YOUR INVESTMENT.
BACK-UP YOUR CARTRIDGES ONTO CASSETTE.

SAVES WEAR ON YOUR CARTRIDGES.
SAVES WEAR ON YOUR **VIC–20'S** MEMORY PORT.

SYSTEM IS COMPRISED OF AN EASY TO USE PROGRAM AND A CARTRIDGE INTERFACE CARD. HIGH QUALITY GOLD CONTACTS AND SWITCHES. BACKED-UP CARTRIDGES RUN LIKE THE ORIGINALS (8K RAM REQUIRED).

INTRODUCTORY PRICE:
$49.95 POST PAID
(PA RESIDENTS ADD 6%)

SEND CHECK OR MONEY ORDER TO:
E–M TECHNOLOGIES
P.O. BOX 185
DOWNINGTOWN, PA 19335

6 MONTH REPLACEMENT GUARANTEE

Wasting Money? We Have the World's Most Cost Effective Development System.

- Includes Hexkit 1.0, a powerful 100% machine code editor debugger utility program that makes coding for 8-bit Micros a snap.
- Program from Commodore VIC-20 keyboard into built-in 4K ROM emulator
- Jumper to target ROM socket
- Test programs in circuit
- Built-in EPROM programmer and power supply
- Burns & runs EPROMS for the Commodore VIC-20, too
- Comprehensive manuals
- Fits EXPANSION PORT

Send for Free Brochure

VISA AND MASTERCARD ACCEPTED

PROMQUEEN CARTRIDGE ONLY $199 U.S. $269 CAN.

+ Prices	U.S.	Can.
Promqueen 64	$299.00	$399.00
8K board w/ 1 Eprom	$ 29.95	$ 39.95
16K board w/ 1 Eprom	$ 39.95	$ 49.95
8K ROM board w/ 1 Eprom-C64	$ 39.95	$ 49.95

Distributed in U.S. by **Arbutus Total Soft, Inc.**
4202 Meridian, Suite 214
Bellingham, WA 98226
Ph. 800-426-1253, in Washington 206-733-0404

Distributed in Canada by
IBC/Distribution Canada
4047 Cambie St., Vancouver, BC V5Z 2X9
Ph. 604-879-7812

NEW! MULTI-COLOR and HIGH RESOLUTION CHARACTER EDITORS For VIC-20 and CBM64

Multi-Color

-Character color and mode set for each character.
-Auxiliary, border, and screen color commands.
-8 Power magnification for editing area.
-Cursor controls, RETURN, HOME, CLR, all work within the editing area.
-4x5 animation area cycles through user-selected character collections.
-Saves set to all output devices.
-Loads set from all input devices.

VIC-20 Version
-64 characters edited at one time.
-2 character collections alternate in the animation area.
-Fits in an unexpanded VIC-20.

CBM64 Version
-256 characters edited! 192 available at one time.
-8 character collections availabe for the animation area.
-90 degree rotation command.
-Move command shifts pixels of one character, as a group, up, down, right, or left.

CASSETTE-$19.95 DISKETTE-$21.95 (CA RES ADD 6% TAX)
Please specify which version you'd like.
Just Another Software Co., Dept. C7
P.O. Box 893, Palo Alto, CA 94301

FREE

Write for your free catalog of

Software and Books

for your **PET, VIC** and **Commodore 64**

TIS, inc.
Box 921 Dept. C
Los Alamos, NM 87544

Software For Commodore 64

COMPELLO	$15.95
DATAFILE	$15.95
FLIGHT 64	$15.95
GUNSLINGER	$15.95
LOAN CALC	$ 9.95
SPACE CADET	$15.95
SPELLATHON	$19.95
SPRITE GEN	$15.95

-Prices are for cassette, add $2 for disk version.
-Check, MO, or COD for total order plus $2 shipping.

See your dealer or order direct

FANTASY COMPUTERWARE
BOX 461
SIOUX FALLS, SO. DAK. 57101
(605) 335-7684

VIC-20 & C 64 Stock Portfolio Manager

* * * * EASY TO USE * * * *

Stock	# Shares	Cost / Share	% Change	Dollar Change
IBM	100	112½	-1.6	-187.50
Kmart	200	20½	68.3	2,800.00

— PROGRAM FEATURES —

- Current Values (1) Individual Stock (2) Total Portfolio
- Records Dividend Payments
- Handles Stock Splits
- Buy & Sell Information By Date
- Works With or Without Printer
- VIC 20 Requires 16K RAM Expansion

Program on Cassette Tape: $29.95 / Disc: $34.95

SEND CHECK OR MONEY ORDER TO:

BASIC BYTE, INC.
P.O. BOX 924, SOUTHFIELD, MI 48037-0924
Phone Orders: 1-800-835-2246 Ext. 237
Kansas Residents: 1-800-362-2421 Ext. 237

VISA AND MASTERCARD ACCEPTED
Michigan Residents add 4% Sales Tax.

- DEALER INQUIRIES INVITED -

VIC-20

6 GREAT PROGRAMS ON 1 CASSETTE REG. $49.95 **$9.95**

16K RAM Reg. $89.95 **NOW $68.00**

Retirement Analysis	12.95
Car Maintenance	12.95
Loan Calculator	12.95
Gold/Silver Trader	12.95
Tax Free Savings	12.95
Stock Options	12.95
Equity Accelerator (8K)	19.95
Landlord (8K)	19.95
Insurance Comparison (8K)	19.95
Real Estate Investor (8K)	19.95

MC/VISA/MONEY ORDERS/CHECKS/COD
1000 ITEM CATALOG
Dealer Inquiries Invited

Space Shuttle Software
PO Box 252 - Cape Canaveral, FL 32920

GAME EXCHANGE

Apple - Atari
IBM-PC - VIC - TRS-80
Disks/Cartridges

Adventure games are great until solved. Arcade games can get stale. Exchange an unwanted game for one you would like to play.

Write For Information Or

Send your original manufacturer's game disk, documentation, a list of five games to choose your exchange from & $5.00 to:

National Home Computer Game Exchange
P.O. Box 20929
Columbus, OH 43220

SMALL BUSINESS PROGRAMS for ATARI®

GENERAL LEDGER SYSTEM

A complete system in one program. Designed by accountants to handle any combination of 200 Balance Sheet, Income and Expense accounts. Menu driven for easy entry. Maintains and prints to the screen or printer Cash Disbursements, Cash Receipts, Journal Entries, Trial Balance, Income Statement and Balance Sheet. Put your computer to work for you!

48K, DISKETTE ONLY, PRINTER OPTIONAL
ONLY $69.95

MAIL ORDER INVENTORY

With this new system you get results. Designed for quick inquiry and update. You maintain current inventory, reorder points, part costs, selling price and sales information. You also produce picking lists, time-to-buy reports, sales and inventory reports.

48K, DISKETTE ONLY, PRINTER OPTIONAL
ONLY $69.95

BULK RATE MAILER

Use this program to maintain your mailing lists and presort by zipcode add/delete/inquiry by partial fields of last name, city or zipcode.

24K, DISKETTE ONLY, PRINTER
ONLY $29.95

YOU SAVE MONEY! GET ALL THREE ONLY $150.00

TRADEWINDS SOFTWARE
1205 N. Genessee, L.A., CA 90046 (213) 656-2139
MasterCard — Visa — Checks — C.O.D.'s

EUCHRE FANS!!!

YOU'RE GONNA L♡VE OUR FIRST OFFERING

"EUCHRE FOR FOUR"
(For You Alone)

* * * * * *

EXCELLENT 4-HAND EUCHRE FOR YOU AND YOUR COMPUTER

* * * * * *

AVAILABLE ON DISKETTE OR CASSETTE FOR COMMODORE 64* OR EXPANDED VIC-20* (NEED 24K EXP.)
(PLEASE SPECIFY WHICH)

VISA **$29.95** MC
(NY State Residents add 7% Sales Tax)

HUTCHCRAFT ASSOCIATES
1132 Dryden Road Ithaca, NY 14850
607 347-4782
DEALER INQUIRIES WELCOME

*TRADEMARKS OF COMMODORE BUSINESS MACHINES

STOP LOOKING!

THIS IS **Ware It's At!** FOR THE

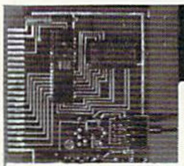
COMMODORE 64

```
BATTERIES INCLUDED: PaperClip 64
Sophisticated, comprehensive WP..........105.00
SOUTHERN SOLUTIONS: Businessman
Quality, understandable general ledger....105.00
ACCESS: Spritemaster
Complete, stick-driven animation utility.. 28.75
        Neutral Zone
Exotic 3-D space action drama............ 27.95
LUNA: Hyper-Hen
Split-second dodging of hawks, coyotes.... 15.95
        Firefighter 64
Graphic strategy game--against the clock!. 15.95
        Pedestrian 64
Cars, etc., will keep you hopping quickly. 15.95
ELECTRA PRODUCTS: Triga Command Joystick. 15.00
NUFEKOP: 3-D 64 Man; Exterminator 64;
Defender on Tri - rave reviews for these.. 15.95
```

Ware It's At!

Order Line - MC/VISA add 3% (409) 763-0291
Call: Leave name,address,order,card# and exp.date
P.O. Box 27
Free Catalog — Galveston, TX 77550 — PRICES SUBJECT TO CHANGE — Check,MO $1.50 S&H

VIC-20



EPROM PROGRAMMER

2716 • 2732 • 2732A DEVICES

COMPLETE SYSTEM
READY TO READ, VERIFY or PROGRAM YOUR EPROMS

ONLY **$79.95**

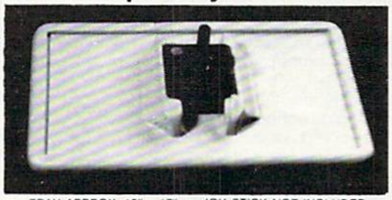
PLUGS DIRECTLY INTO VIC-20
NO ADDITIONAL PARTS OR ACCESSORIES NEEDED
SOFTWARE TAPE INCLUDED

ADAPTER KIT AVAILABLE FOR OTHER 6502 BASED COMPUTER SYSTEMS
PLEASE INQUIRE • ADD $3.50 FOR SHIPPING
MD RESIDENCE ADD 5% TAX

MWS ELECTRONICS — VISA, MC ACCEPTED
P.O. BOX 418
POCOMOKE, MD. 21851 — 301-632-0620

VIC-20 IS A REGISTERED TRADEMARK OF COMMODORE

New! Unique! Joy Stick Holder



TRAY APPROX. 10" x 17" — JOY STICK NOT INCLUDED

Only $9.95 each

*For use on - Atari 400/800, Vic-20 Computer, Radio Shack, Texas Instruments, Sears Video Arcade & Atari VCS

*Greater Accuracy *Reduces Fatigue *Just Snap in Place
*Real Arcade Action *Higher Scoring *Rests on lap

DEALER INQUIRIES INVITED

Send check or money order for $9.95 + $2.00 (Postage and Handling) - Total $11.95 each.

TO: **TREND-TEK CORP.**
P.O. Box 1393, N. Miami Beach, FL 33160-1393
PLEASE PRINT

Enclosed is my check for $ ____________
Please send me ____________ Joy Stick Holders.
Name ____________
Address ____________
City ________ State ________ Zip ________

TRAY COLOR MAY VARY — SORRY NO C.O.D.'s

★ ★ VIC 20 ★ ★

HARDWARE: BEATCO TAPE BACK-UP SYSTEM© - Does This Sound Familiar? Your dog just chewed up your "Super Star Raiders" tape. Or your budding computer genius just erased your new $20 "Whiz Kong" tape. So you get your back-up copy and make another working tape — right? Oh — you couldn't back-up those programs. Your just out the money! Well now you **can** - with the **BEATCO** and the cost is only $23.95. A second recorder is required for making back-up tapes. Hardware is compatable with existing Commodore cassette ports $23.95

SOFTWARE:
MEMORY CUBES - A high resolution graffic game which offers both educational and recreational enjoyment for both young and old alike. Has random reset of color and game board, so every game is different. Can be played by one or two players with exciting sound affects to tantalize your senses. A game of concentration that challenges the mind in an entertaining fashion. This is a truly professional product that is guaranteed satisfaction for all. Requires at least 3k memory expansion and can be run in any memory configuration above 3k using our "MEMBOOT" provided with game instructions. VIC $14.95

TRI-PAK - This pre-school to 2nd grade package offers three educational games to give endless hours of enjoyment for children. Big Math, Shape Mate, and Follow Me are geared towards youngsters in teaching simple addition & counting, identifying shapes & things, teaches ABC's with keyboard matching and sound exercises. VIC $19.95

Add $1.50 for shipping. Colorado residents add 6½% sales tax.
(Check - M.O. - Visa - MC (exp. date required)

HOME COMPUTER SERVICES
1150 Syracuse St. #2-20 / Denver, Colorado 80220
1 (303) 393-8109

ATTENTION TEXAS INSTRUMENTS TI-99/4A OWNERS

We have hundreds of 3rd party independent software programs on cassette and disk ready to run on your TI-99/4A. Games, business, and educational programming at discount prices as low as $8.95 ea. Plus all TI hardware and software at incredibly low, low prices, including the new TI-99/2 and CC-40 computers. We also have dust covers, heavy duty joysticks with TI adapters, and many more accessories. Call or write now for a FREE listing. We ship your order U.P.S. the same or next business day to insure fast service. Visa and MasterCard accepted (NO service charges) or C.O.D. is okay.

THE MUSIC WORKSHOP
59 E. Tioga St.
Tunkhannock, PA 18657
CALL 1-717-836-4522

NEW FOR YOUR VIC 20

NOW YOU CAN HAVE 35K OF RAM + IEEE 488 ON ONE CARTRIDGE!

3K RAM CARTRIDGE ONLY $39⁹⁵

All boards are fully socketed for future expansion. Add memory in 8K increments simply by inserting up to (4) HM 6264 RAMS. (Call for price) Add IEEE 488 Chip Set for $59⁹⁵. Available in any configuration from bare board to fully populated. Dealer inquiries invited.

WAVE Computers Inc.
P.O. Box 3883, Federal Way, Washington 98003
Add $2.00 Postage/Washington Residents Add Sales Tax
No COD's Please **Phone No. (206) 839-WAVE**

ATARI® DISK DRIVE USERS:

YOU'RE IN COMMAND WITH DOS-MOD

enhancements to Atari DOS 2.0S

FEATURES FROM SOPHISTICATED PROFESSIONAL OPERATING SYSTEMS

- create powerful command files
- quick one-line commands
- friendly full-screen use
- flexible memory examine/change
- improved wild-card capability
- nasty bugs exterminated
- completely compatible with DOS 2.0S and many more improvements

$25 ppd. RISK—FREE TRIAL!
includes easy-to-follow tutorial demo

SUNNY SOFTWARE
1058-C MARIGOLD COURT
SUNNYVALE, CA 94086

Advertisers Index

COMPUTE!'s FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, **COMPUTE!** cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: **COMPUTE!**, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new subscription to **COMPUTE!**: 12 monthly issues for $20.

Please print or type your full name and address. Limit one card per person.

Name

Address

City

State/Province Zip

Country

Please include zip code. Expiration: 9/30/83 C0783

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new subscription to **COMPUTE!**: 12 monthly issues for $20.

Please print or type your full name and address. Limit one card per person.

Name

Address

City

State/Province Zip

Country

Please include zip code. Expiration: 9/30/83 C0783

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new subscription to **COMPUTE!**: 12 monthly issues for $20.

Please print or type your full name and address. Limit one card per person.

Name

Address

City

State/Province Zip

Country

Please include zip code. Expiration: 9/30/83 C0783

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 11747
Philadelphia, PA 19101

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 11747
Philadelphia, PA 19101

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 27346 PHILADELPHIA, PA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 11747
Philadelphia, PA 19101

Introduce A Friend To COMPUTE!
Save $10.00 Off The Newsstand Price

One year, 12 issue subscriptions are $20.00 in the U.S., $25.00 (U.S. funds) in Canada.

PLEASE PRINT.

Please charge my:

☐ VISA ☐ MasterCard ☐ American Express

Acc't No. ______________________

Exp. ______ / ______

YOUR NAME ______________________

ADDRESS ______________________

CITY ______________________

STATE ______________ ZIP ______________

$______________ payment enclosed ☐ Bill me later

Please enter my ☐ RENEWAL ☐ NEW SUBSCRIPTION at the same time.

GIFT TO ______________________

ADDRESS ______________________

CITY ______________________

STATE ______________ ZIP ______________

☐ Renewal ☐ New subscription

SIGN CARD: ______________________

GIFT TO ______________________

ADDRESS ______________________

CITY ______________________

STATE ______________ ZIP ______________

☐ Renewal ☐ New subscription

SIGN CARD: ______________________

637101

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 914
Farmingdale, NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 2312 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!
P.O. Box 914
Farmingdale, NY 11737

IF PERSONAL COMPUTERS ARE FOR EVERYBODY, HOW COME THEY'RE PRICED FOR NOBODY?

$1395* APPLE® IIe 64K

$999* TRS-80® III 16K

$1355* IBM® PC 64K

A personal computer is supposed to be a computer for persons. Not just wealthy persons. Or whiz-kid persons. Or privileged persons.

But person persons.

In other words, all the persons whom Apple, IBM, and Radio Shack seem to have forgotten about (including, most likely, you).

But that's okay. Because now you can get a high-powered home computer without taking out a second mortgage on your home.

It's the Commodore 64. We're not talking about a low-priced computer that can barely retain a phone number. We're talking about a memory of 64K. Which means it can perform tasks most other home computers can't. Including some of those that cost a lot more. (Take another look at the three computers above.)

By itself, the Commodore 64 is all the computer you'll ever need. Yet, if you do want to expand its capabilities some day, you can do so by adding a full complement of Commodore peripherals. Such as disk drives. Modems. And printers.

You can also play terrific games on the Commodore 64. Many of which will be far more challenging than those you could ever play on a game machine alone.

And as great as all this sounds, what's even greater-sounding is the price. It's hundreds of dollars less than that of our nearest competitor.

So while other companies are trying to take advantage of the computer revolution, it seems to us they're really taking advantage of something else:

Their customers.

*Manufacturers' suggested list prices as of March 20, 1983. Monitor included with TRS-80 III only. Commodore Business Machines, P.O. Box 500R, Conshohocken, PA 19428; Canada–3370 Pharmacy Avenue, Agincourt, Ont., Can. M1W 2K4.

commodore COMPUTER