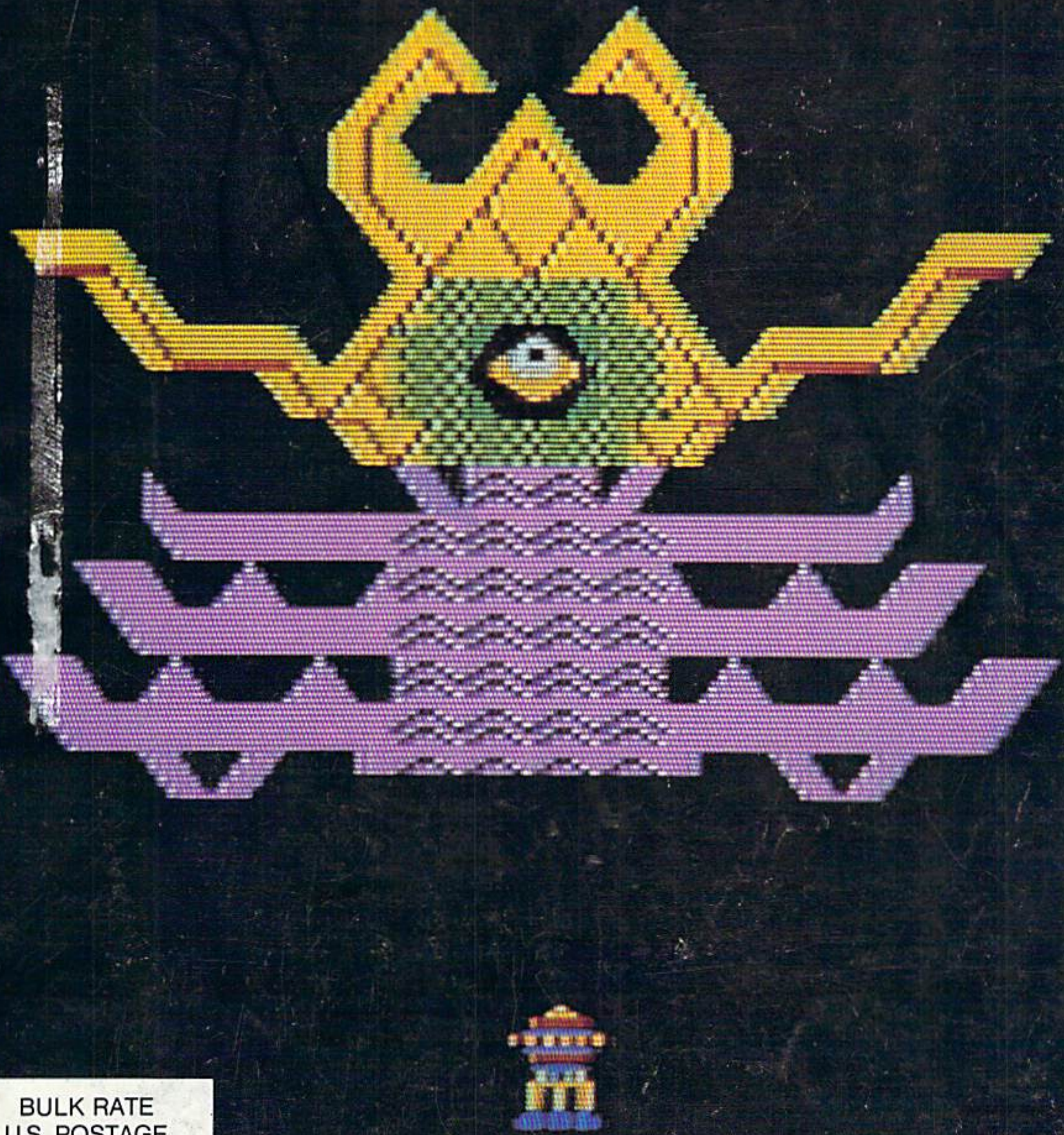
commodore
power/play
home computing

Summer 1983 Volume II Number 2
$2.50 U.S.
$3.50 Canada

And Even More Games for the Commodore 64

Butterfield Fools with Boolean

England Takes the VIC Seriously



BULK RATE
U.S. POSTAGE
PAID
Permit No. 1705
Phila., PA

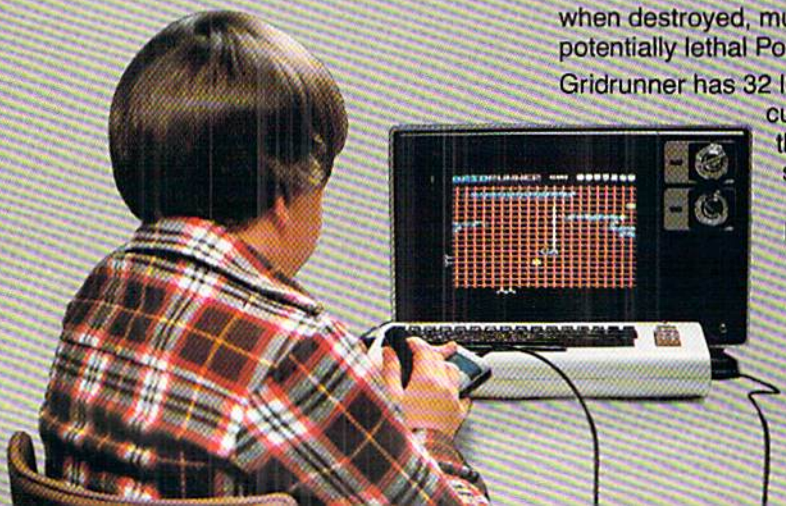
Commodore Business Machines, Inc.
1200 Wilson Drive · West Chester, PA 19380

IS GRIDRUNNER™ UNBEATABLE?

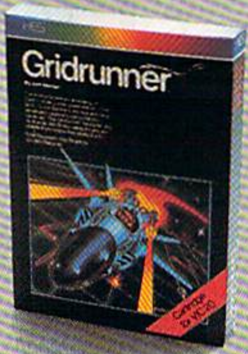
No one, not even the author, has ever achieved the last Gridrunner. It is an extremely fast-paced arcade-quality game designed to test your coolness under fire and challenge your reflexes.

As the pilot of the Gridrunner, a combat ship, you must annihilate the various enemies traveling along the "Grid." High scores are possible only through the mastery of the patterns of the X/Y Zappers and the Gridsearch Droids which, when destroyed, mutate into potentially lethal Pods.

Gridrunner has 32 levels of difficulty (20 levels in the VIC 20 version). To this date, the 13th level has been the highest achieved.



Gridrunner is available for VIC 20™ and Commodore 64™.



Can you beat Gridrunner? See your local computer or games dealer and find out.

Human Engineered Software
71 Park Lane
Brisbane, CA 94005

HES
a division of USI

VIC 20 and **Commodore 64** are trademarks of Commodore Electronics Ltd.

GROW WITH US.

VIC 20™ and Commodore 64™ expansion products from Micro Systems Development.

A The Interbus Series. Three interfaces for the VIC 20 and Commodore 64: one for IEEE 488, one for RS 232 and one for Parallel.

The VIE and CIE are IEEE 488 interfaces for the VIC 20. When plugged into the expansion port, the cartridge is "transparent," that is, the user can still attach other peripherals without any interference. Devices such as 4040, 8050, 2031, 2032, 4022 and 8023 can be controlled. The IEEE software can be called by using the 'SYS' command, even in the middle of a BASIC program.

The V232 and C232 are serial interface cartridges which allow connection of various input/output devices such as printers, modems, plotters, etc. to VIC 20 or Commodore 64 computers. Features include: positive and negative voltage swings to meet full EIA standards, straps and jumpers to allow reconfiguration to meet pinouts for any RS232 device, and software selectable reconfiguration such as baud rate, parity, and duplex mode.

The VPI and CPI are parallel interfaces for the VIC 20 and Commodore 64. These interfaces provide direct BASIC use of the parallel printer bus and give "invisible" access to the bus. The VPI can be used only on the VIC 20 and uses the expansion port. The CPI will work with both the VIC 20 and Commodore 64 and does not use the expansion port. The CPI also has switches for setting insertion or deletion of line feed, conversion of Commodore ASCII into standard ASCII or visa versa, addresses printer to device 4, 5, 6 or 7, and allows normally unprintable Commodore characters to be printed in a recognizable form.

B Expandoport Series. Expandoport 3 and Expandoport 6 are three- and six-slot expansion boards for the VIC 20. Each slot on the Expandoport 6 has a switch for controlling power to that connector. The switch allows the use of cartridges which respond to the same memory space. The Expandoport 6 also has a fuse and reset switch. The fuse prevents excessive current drain from the VIC 20 and protects it from 'shorts'. The reset switch allows the user to 'Restart' the VIC 20 without turning power off. This feature allows RAM, which is located in the ROM expansion area, to be protected during 'Restart'.

Expandoport 4 is a four port expansion board for the Commodore 64. It has the same features as the Expandoport 6 and even allows for the use of varying width cartridges.

C Terminal Pak Series. The VTE 40 Terminal Emulator (VTE 40) is a hardware and software package which converts the VIC 20 into a 40-column communications terminal. The VTE 40 cartridge is complete. Various set-up parameters such as baud rate, parity, duplex, and bits per character can be selected through a 'menu' format. VTE 40 features are: 40 x 25 text display, user definable communication specs, smooth or normal scroll, print information to printer or disk, generation of control codes, selective omission of data, continuous status line.

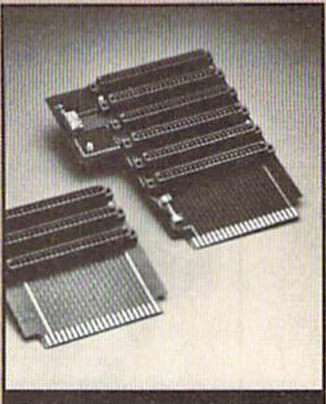
The CTE/VTE Terminal Emulator (CTE/VTE) is a software program which converts the VIC 20 or Commodore 64 into a terminal. The user can 'software select' the baud rate compatible with the modem used. Full upper and lower case characters are supported.

D Audio Link. An audio cassette adapter interface for the VIC 20. Features include: use of regular cassette recorders, conversion of VIC 20 digital data to audio and visa versa, normal and inverted cassette signal, remote on/off control and control of external devices.

E Monitor Link. This cable assembly allows the VIC 20 and/or the Commodore 64 to interface with a monitor instead of a TV. The Monitor Link provides separate video and audio output. This enables the sound output to go directly into a stereo system for unmatched audio quality. The Monitor Link is great for applications where a TV is not desired. It allows the Commodore 64 user to have high quality resolution on black and white monitors.

VIC 20™ and Commodore 64™, CBM™ and PET™ are registered trademarks of Commodore Business Machines.
Expandoport, Audio Link, Interbus, Terminal Pak, VRAM and Monitor-Link are copywrited by Micro Systems Development, Inc.

A Interbus Series.


B Expandoport Series.

C Terminal Pak Series.


D Audio Link.

E Monitor Link.

Call toll free for nearest dealer

1-800-527-5285

Lifetime warranty available upon return of Product warranty card

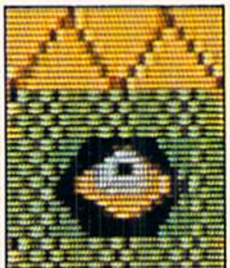
MSD

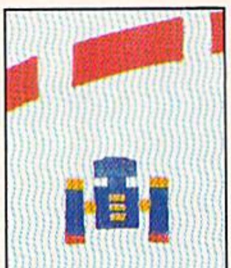
MICRO SYSTEMS DEVELOPMENT, INC.
11105 SHADY TRAIL • SUITE 104 • DALLAS, TEXAS 75229

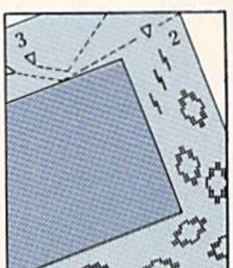
commodore
power/play
home computing

features

Summer 1983 Volume II Number 2

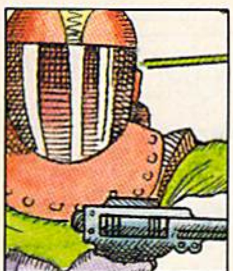

New Cartridge Games


OOPS


Omega Race

departments

SID


Bit-Mapped


England

staff

Publishing Manager
Neil Harris
Editor
Diane LeBold
Technical Editor
Jim Gracely
Associate Editor
Betsy Byrne
Staff Writers
Jeff Bruette
Jeff Hand
Paul Higginbottom
John O'Brien
Mike Smith
Michael Tomczyk
Contributing Writers
Alex Bakman
Jim Butterfield
Mike and Annette Hinshaw
Jim Lockridge
David Malmberg
Barbara J. Moody
David Owens
Carl Robertson
Dean S. Rossa
Gerald and Betty Schueler
John Watkin
Technical Staff
John Campbell
Rick Cotton
Andy Finkel
Bill Hindorff
Cyndie Merten
Advertising Coordinator
Sharon Steinhofer
Circulation Manager
John O'Brien
Circulation Assistant
Kathy Reigel
Graphic Design
Neumann Greenberg Schlenker
Cover Photo:
Kevin Raber
Printing
International Lithographing

Power/Play is published four times a year by the Computer Systems Division, Commodore Business Machines, Inc., 1200 Wilson Drive, West Chester, PA 19380. Volume II, Number 2.

Subscription Information: U.S. subscriber rate is $10.00 per year. Canadian subscriber rate is $15.00 per year. Overseas $17.00 per year. Questions concerning subscriptions should be directed to Commodore Business Machines, Magazine Subscription Department, Box 651, Holmes, PA 19043. Phone 800-345-8112. (Pennsylvania phone 1-800-662-2444.)

For Even More Information And Just Plain Fun

Watch These Upcoming Issues!

Commodore: The Microcomputer Magazine. Our June/July issue is devoted to programming. Whether you're a full-fledged techie or a stumbling novice, you'll find something in this issue that is meant for you.

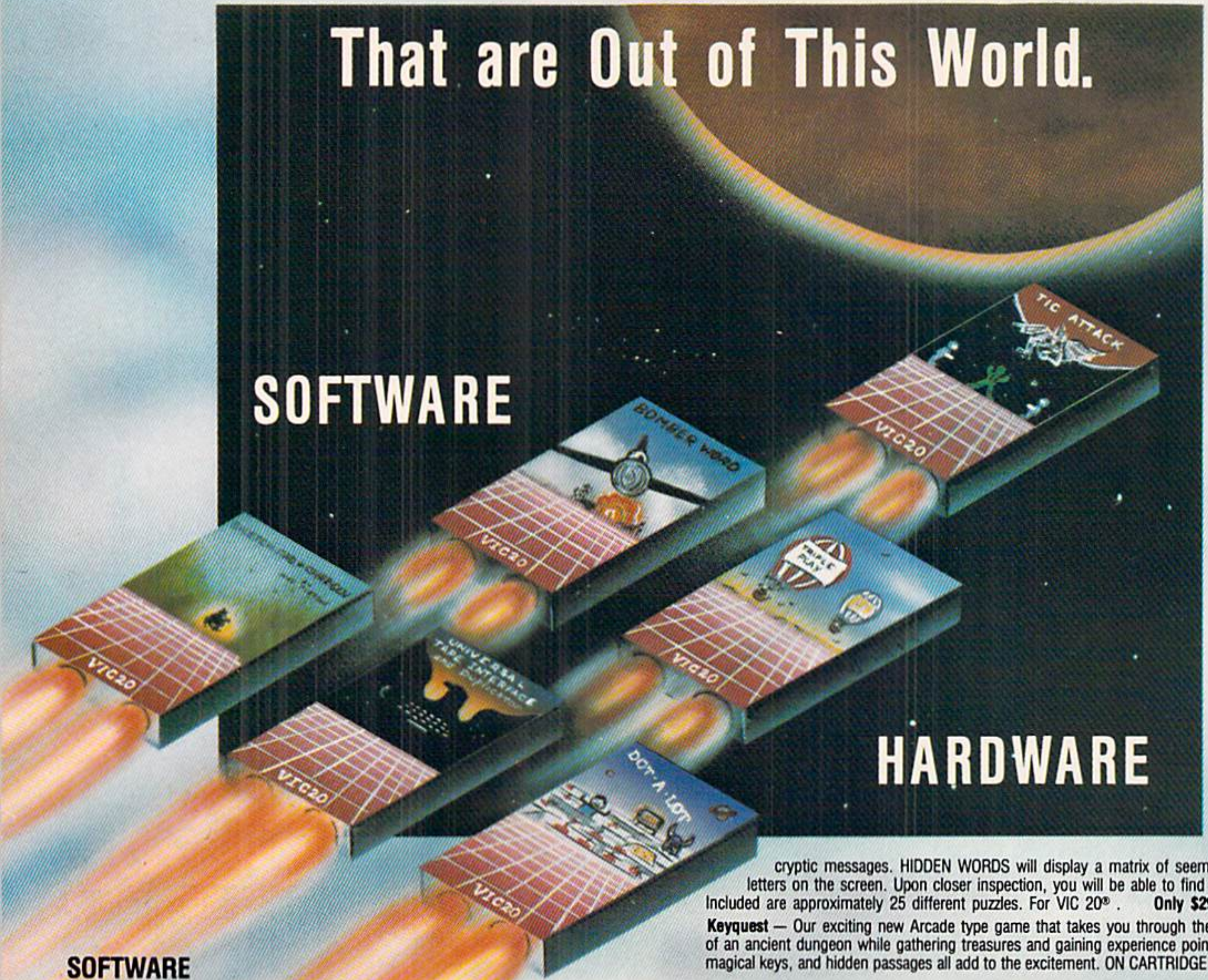
Power/Play. Watch for us again in the Fall with more fun, games and home applications for our family users.

special keys

When you're typing in program listings for the VIC 20 and Commodore 64 you occasionally come across a reversed graphic symbol that may not be familiar to you. If you find these symbols in a program listing, refer to this table to find out what key they represent.

	Key	Appears As		Key	Appears As
	CLR/HOME			Commodore 5	
shift	CLR/HOME			Commodore 6	
	CRSR down			Commodore 7	
shift	CRSR up			Commodore 8	
	CRSR right		CTRL	RVS/ON shift M	
shift	CRSR left		CTRL	RVS/ON N	
	CTRL 1		CTRL	RVS/ON shift N	
	CTRL 2		CTRL	RVS/ON H	
	CTRL 3		CTRL	RVS/ON I	
	CTRL 4			f1	
	CTRL 5			f2	
	CTRL 6			f3	
	CTRL 7			f4	
	CTRL 8			f5	
	CTRL 9			f6	
	CTRL 0			f7	
	Commodore 1			f8	
	Commodore 2			INST/DEL	
	Commodore 3		shift	INST/DEL	
	Commodore 4				

Products for VIC 20® and CBM 64®



SOFTWARE

Word Wizard For The Vic 20® — (Requires at least 8K memory expansion.) A user friendly WORD PROCESSOR with optional joystick control. Easy edit and string manipulation commands that follow the standard format. Full use of function keys for ease of use. 100% machine language with Delete Word, Search functions and Full Justification. Use VIC Graphic printer, or any centronics compatible printer connected to the user port. On tape (support disk). **$34.95.**

ZAP! — Climbing the corporate ladder could be fun except for all that falling paperwork. This Hires arcade type games allows up to 4 players to advance through each floor and change levels to scale the corporate ranks. Be careful, it's easy to be ZAPPED! CARTRIDGE for VIC 20.® **$29.95.**

Bomber Word — A unique graphic word game on cartridge that provides the full thrill of arcade action. Complete with six modes of play options for added enjoyment. Play against the computer or another player. 6 to adult. For VIC 20®. **$29.95.**

Tic Attack — A fast action arcade game on Cartridge that challenges all of your dexterity. Written in machine language for special audio & visual effects. Over 100 levels of play. High score indication. For VIC 20®. **$29.95**

Dot-A-Lot — As you wander through the maze of life collecting Berries, you happen upon some magical fruit. Pick one and the treasures appear, but the Meanies are out today looking to spoil your fun. Defeat them and continue on to a higher level. An ever changing maze plus arcade type animation and sound will provide a real winning CARTRIDGE for the VIC 20®. **$29.95**

Triple Play — Three word games that are both fun and educational. The games that are included are CROSSWORDS (requires at least 8K expansion). Five complete puzzles are included and each puzzle has up to 100 different words. CRYPTO-SOLVE will help you solve those cryptic messages found in newspapers, books, and magazines with a systematic computer technique. Included are approximately 50 different puzzles. You can even enter your own cryptic messages. HIDDEN WORDS will display a matrix of seemingly random letters on the screen. Upon closer inspection, you will be able to find many words. Included are approximately 25 different puzzles. For VIC 20®. **Only $29.95 for all 3**

Keyquest — Our exciting new Arcade type game that takes you through the many levels of an ancient dungeon while gathering treasures and gaining experience points. Monsters, magical keys, and hidden passages all add to the excitement. ON CARTRIDGE for VIC 20®. **$34.95**

Sketch Pad & Char-Gen — A high resolution drawing program that will allow you to save your pictures to tape. Also included is a simple to use character generator that will allow you to design a different character for every printable key. Create game creatures, foreign alphabets, secret symbols or other special characters. One set is included. On tape for the VIC 20®. **$24.95**

HARDWARE

Expand-O-Ram — 16K Expansion Board for the VIC 20® with reset, memory write protect, full memory allocation, plus TWO slots. Like having 2 products in 1. Can be used as a cartridge development system too. **$119.00**

Universal Tape Interface & Duplicator — (Use on the Commodore 64® and VIC 20®). With this device, you can easily load, save or even duplicate tapes with your standard recorder. Full 3 LED indication of Data transfer. A reliable way to Load, Save and Duplicate. NOTE: Duplication requires 2 standard cassette recorders. **Only $49.50**

Tymac Buffered Parallel Cable with Driver — For the VIC 20® & Commodore 64®. This cable assembly plugs into the USER Port and provides a simple and inexpensive way to connect a PARALLEL Printer to your computer. **Only $29.95**

Driver Cartridge for VIC 20® — Take full advantage of the capabilities of your Parallel Printer including full Commodore graphics and formatting. Available for SEIKOSHA, C. ITOH, OKIDATA, and others. Specify printer. **Only $29.95**

Tymac "Connection" — A truly intelligent parallel interface for the VIC 20® and Commodore 64®. It will make your printer operate like the COMMODORE. Printer including graphics, text symbols, tabs, and virtually every other printer function. Plugs into the serial socket. Available for most popular parallel printers. **$119.00**

MasterCard VISA *Dealer and Distributor Inquiries Invited*

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

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.

letters

To the Editor,

In your Fall, 1982, issue you said you couldn't restore the expanded VIC back to the unexpanded VIC without the hassle of removing the memory expander. Actually, it is possible with the following program:

To turn off this program, type SYS 64802

Sincerely,
Craig Bruce

```
10 POKE44,16:POKE46,16:POKE48,16:POKE56,16:POKE642,
   16:POKE52,30:POKE644,30
20 POKE56,30:POKE648,30:POKE36866,150:POKE36869,240
30 POKE4096,0
40 PRINT"♥":NEW
50 REM**TYPE SYS64802 TO CANCEL EFFECTS***
```

To the Editors:

My company has resisted the 'computer invasion' for some time. We, rightly or wrongly, scoffed at the idea of cost/effectiveness, etc. This was until we broke down and bought the Commodore 64! To tell you the truth, I was truly petrified of the thing, thinking its first readout would be "you're fired!" Well, I am writing you this letter to tell you that it has not been that way—that the 64 and I are on VERY good terms indeed.

Even without the disk drive that is on order, we have found the machine to be sturdy, totally flawless, and forgiving. Only twice have we gotten perturbed, and this due to our turning off the machine before saving a program.

In any event, this serious business machine has been stolen by the company president and now is safely nestled in my home, doing spreadsheets that your guidebook showed me how to erect. I've even got my kid jealous, since the game I've created (about 9K's worth) is a full-bore re-creation of the history he's studying.

Thank you again for the affordable serious computer. (P.S. Why DO people buy Apples at $1,000 more for less computing power?) C

Sincerely,
James P. Boshnack
Twenty-First Century Concepts
Houston, Texas

And More Beginnings

I thought I was excited about the last issue, but this one may have it beat. That's because this issue marks the beginning of two new departments headed up by two people who are not only fun to read but who will help you learn a great deal about your computer.

Betsy Byrne, president of the New Mexico Commodore User Group—not to mention the mother of four children—has plunged enthusiastically into running the new "Kids' Corner". You probably remember Betsy's very funny account, in last issue's User Group Spotlight, of how she and her family became involved with Commodore computers, as well as her informative review of *Gortek and the Microchips* in that same issue. Betsy wants programs, reviews or interesting computer-related stories by, for or about kids. If that sounds good to you, turn to page 27 for full instructions on how and where to send your stuff. Oh, and while you're back there, don't forget to take a look at what Betsy's put together for this issue. Her Commodore Kids are pretty amazing.

And what can I say about David Malmberg? Those of you who have read his articles on Turtle Graphics and LOGO in some of the other leading computer magazines will undoubtedly be as delighted as I am about having him as a regular contributor. David will be writing our "No More Pencils, No More Books..." column, covering the many different facets of learning at home with your computer. I'm not sure what David has planned for upcoming issues, but whatever it is, you can bet you'll learn a lot—and learn it painlessly. His "First Look at LOGO" in this issue gives you a terrific start on using Commodore's new LOGO package to create exciting graphic designs on your Commodore 64. Stay tuned for more.

And yet another beginning. By the time you get this issue Commodore will have moved down the road to our new quarters in the great mirror-walled building in the woods outside West Chester. And for the first time in the history of our magazines, the whole publishing department will be in one place. It's hard to imagine what that will be like. I'll probably gain weight, since I won't be putting in as much mileage every day running from one end of the building (where Jeff Hand has been all this time) to the other (where they hid Jim Gracely when he came in February). It crossed my mind that we may all find out we can't stand each other, once we're in such close proximity.

Whatever comes of it, please note that the new Commodore Business Machines address is now 1200 Wilson Drive, West Chester, PA 19380. I heard a rumor that we do have phones there, but I don't know what the number is, yet.

Tired of hearing all this great news? Well, sit right back down because I'm not finished yet. With last issue both our magazines began to be sold in 800 Waldenbooks bookstores across the country. That means those of you who haven't managed to send in your subscription yet will have an easier time finding us. So look for us in Waldenbooks, and if we're not there, be sure to ask the manager why.

Our list of Commodore user groups is fast approaching 300, worldwide. And, what do you know, we finally got all the groups entered on a data base! That means we can start communicating with them via the mail, which we hope to do periodically. (In fact, some of the groups may have already gotten Betsy Byrne's questionnaire, requesting information on what they'd like to see in the Kids' Corner department of this magazine.) By the way, if you're thinking about starting a user group, I'd suggest you take a look at Jeff Hand's article (called, coincidentally, "Starting a User Group") in the April/May issue of *Commodore: the Microcomputer Magazine.*

See you in the fall, after the mosquitoes have subsided. C

—Diane LeBold
Editor

we're glad you asked

Our technical staff is glad to answer your questions. Please direct them to the attention of Jim Gracely, in care of this magazine.

Q Page 114 of the *VIC 20 Computer Guide* says that variable names may be a letter, a letter followed by a number, or two letters. I have used variable names such as "COUNTER" and "FIRSTNAME$" without any syntax errors. Is this an error in the *Computer Guide*? The only time I get a syntax error is if I use a BASIC command or function in the name: "FACTOR", for instance, cannot be used because it contains the operator "OR".

A No, this is not an error in the *VIC 20 Computer Guide.* As far as BASIC is concerned, a variable name has one of three forms: a letter, a letter followed by a number, or two letters. Regardless of what you name a variable, BASIC will reduce it to one of these three forms. The reason for this is that BASIC only allots two bytes of storage for each variable name. Any characters left over are ignored. Be careful if you're going to use long variable names: "COUNTER" and "COLUMN" both start with "CO" and BASIC will think that they are the same variable!

Q I have been using programmable characters on the VIC 20 for some time. However, with 8K or more of expansion RAM the formulas in the *VIC 20 Programmer's Reference Guide* no longer work! Please explain how to use programmable characters with 8K or more expansion RAM.

A Now here's a popular question. Turn to page 215 of the *Programmer's Reference Guide* and find the formula in the section entitled "Character Memory Location". The formula looks like this:

```
POKE 36869, PEEK(36869)
AND 15 OR (X*16)
```

Scratch out that formula (it's wrong) and replace it with this one:

```
POKE 36869, PEEK(36869)
AND 240 OR X
```

The "X" in the formula comes from the table on page 216. This new formula will work with or without memory expansion.

The table on pages 83 and 84 doesn't work with 8K or more expansion RAM because the location of screen memory moves. Register 36869 contains a part of the screen memory address in bits 4, 5 and 6. When 8K or more expansion RAM is added, screen memory moves from 7680 to 4096 and the start-up value of register 36869 changes from 240 to 192. You can still use the table on pages 83 and 84 with 8K+ RAM by subtracting 48 from each of the values in the "NUMBER" column.

The POKE values for locations 52 and 56 can still be used but they will severely limit your program size. Instead of protecting the programmable characters it is a better idea to move the beginning of BASIC to a location after (below) your characters. See the next question on moving the beginning of BASIC for the correct procedure.

Q On page 8 of the Fall 1982 *Power/Play* there is a tip for scrolling the screen backwards. I tried typing it in a number of ways and I can't get it to work.

A The instructions for using this tip were not very clear. It is a nice little trick and here is the way to enter it into your VIC 20 or COMMODORE 64:

```
PRINT"[CLR HOME][CRSR
  DOWN][SHIFT CURSOR
LEFT][SHIFT INST/DEL]"
  :POKE218:158
```

This is the way to use this command in the direct mode. It will scroll the screen one line backwards (down). To get a better idea of what you can use this for, try entering the following two line program into your VIC 20 or Commodore 64.

```
10 FORX=1TO10:PRINT"████":
   POKE218,158:NEXT
20 PRINT"QQQQQQQQQQQQQQQQ
   QQQQQQQQQQQQQ":GOTO10
```

(That's 29 [CURSOR DOWN]s)

Now move the cursor to the middle of the screen, type some words or characters and then type RUN.

Q How do I move the beginning of BASIC so that I can protect a part of BASIC RAM?

A Moving the beginning of BASIC is not hard to do but there are a couple of rules that must be followed.

The operating systems of the VIC 20 and the Commodore 64 require the first memory location of BASIC to be a "0". The registers 43 and 44 must contain the value of the second memory location of BASIC. If BASIC starts at 2048, location 2048 must contain a "0" and registers 43 and 44 are poked with the values for 2049 (43=1 and 44=8). Once registers 43 and 44 have been set, a CLR must be performed to reset all of the other memory pointers, and a NEW command is used to "straighten-up" the new BASIC location.

A simple program can do all of this for us:

```
10 BG = LOCATION + 1
20 HB = INT(BG/256)
30 LB = BG - HB * 256
40 POKE LOCATION,0
50 POKE 43,LB : POKE
   44,HB : CLR
60 NEW
```

Replace the variable "LOCATION" with your new beginning location for BASIC. After you have typed in the program (and saved it!), type RUN. Now type LIST. If you typed in the program correctly, nothing will be listed! Everything between the beginning of BASIC RAM and your new beginning location is protected (and hidden!). C

Creating Attractive Screen Titles

by Michael S. Tomczyk
Reprinted from *Commodore Magazine,* August/September 1982

Displaying an attractive screen title is always a nice way to start a BASIC program. You can even let the user choose his own title, and display it dramatically when the program starts!

One thing to remember when choosing a title is to try to keep the length of the title less than 22 characters including spaces, because the VIC displays 22 characters on each line and your title will "spill over" to the next line if it's longer.

Centering your title

There are lots of ways to make attractive titles, but one of the easiest techniques is centering your title at the top of the screen when the program begins.

To center your title in the middle of a line, the first thing to do is count the number of characters in your title. Let's say your title is "VIC MAGICIAN."

□□□□□□□□□□□□□□□□□□□□□□
□□□□□VIC MAGICIAN□□□□□

This title has 12 characters including the space. Now subtract that total from 22 (the number of characters on one line). The answer (22-12) is 10. That means you have 10 spaces "left over" after PRINTing your title on the line. To center the title, you want to have an equal number of spaces on each side of the title, so next we divide the "left over" spaces by 2. Since 10/2 = 5 we know to put 5 spaces on each side of our title if we want to center it.

The following one-line program centers the title "VIC MAGICIAN" on the screen by CLEARing the screen and then PRINTing the title five spaces over from the left margin (To make each space in your program, press the long space bar once, quickly. Any space you include inside the quotation marks will appear as a space on the screen when the title is PRINTed. Notice that you don't have to worry about the spaces on the right side of the title because they're automatically left blank.)

Type this line and hit the RETURN key:

```
10 PRINT"SHIFT CLR/HOME SPACE SPACE
SPACE SPACE SPACE VIC MAGICIAN"
```

The VIC uses "shorthand" to display the CLEAR command inside quotation marks, so when you hold down SHIFT and press CLR/HOME the VIC displays a reverse heart, which means "CLEAR." The spaces you typed appear as blank spaces, like this:

```
10 PRINT"♥     VIC MAGICIAN"
```

To display your title, type the word RUN and press the RETURN key.

If you have a problem or make a mistake, type LIST and hit RETURN, then retype the line.

Using the SPC Command

There's another way to put spaces in your BASIC programs... by using the SPC command. If you type PRINT SPC (5) your program will insert five spaces. Remember the SPC command is never enclosed in quotation marks but is always outside of the quotation marks... and you must always use the PRINT command before the SPC command. Here's how you can use the SPC command to PRINT the line you just typed:

```
10 PRINT"SHIFT CLR/HOME"SPC(5)
"VIC MAGICIAN"
```

Here's another example of the SPC command:

```
10 PRINTSPC (5) "MOVES 5 SPACES"SPC(3)
"MOVES 3 SPACES"
```

Displaying Your Title in Reverse

Now that you've created a title. Let's "dress it up" a little by reversing the title colors. To do this, you'll have to retype line 10. This time, just before you type the title, hold down the CTRL key and press the RVS ON key. This makes the title print in reverse when you RUN the program. Notice that when you hold down CTRL and press RVS ON, the VIC displays a REVERSE R. Type the following and hit the RETURN key:

TYPE THE "9" KEY

10 PRINT" [SHIFT] [CLR/HOME] [SPACE] [SPACE] [SPACE] [SPACE] [SPACE] [CTRL] [RVS ON] VIC MAGICIAN"

The line you typed should look like this on the screen:

10 PRINT" ♥ R VIC MAGICIAN"

Now type [RUN] and hit [RETURN] to see the new title. What if you want to reverse the whole line, including spaces? Type this line exactly as shown and hit [RETURN], then type [RUN] and [RETURN] to see it:

10 PRINT" [SHIFT] [CLR/HOME] [CTRL] [RVS ON] [5 spaces] VIC MAGICIAN [5 spaces]"

This time we put the REVERSE ON command at the beginning of the first blank space and we added five blank spaces on the right side of our title. When you want to print reverse spaces (solid blocks) on the screen, you have to include all the spaces that you want to appear as solid blocks. That's why we include the right side spaces here.

Letting the User Choose a Title

One of the nicest cosmetic touches you can add to a title is letting the user choose his own title. The following program asks the user to type in a title, then centers it at the top of the screen. You can use these two lines at the beginning of any BASIC program.

10 PRINT" [SHIFT] [CLR/HOME] TITLE";:INPUTT$
20 PRINT" [SHIFT] [CLR/HOME] ";:FORX = 1TO(22-LEN (T$))/2:PRINT " [SPACE] ";:NEXT:PRINTT$

After you enter this program, type the word RUN and hit the [RETURN] key. The screen will clear and say, "TITLE?" Type in a title and hit [RETURN]. The title appears automatically centered at the top of your screen.

The key here is using the INPUT statement in line 10 to define the title typed in by the user as T$. Now, in line 20, we can use a special BASIC command called LEN to determine the LENgth of T$. LEN (T$) counts the number of characters in the title, which was INPUT in line 10 as "T$".

Remember the centering formula we used at the beginning of the article? Well, this whole formula is included in line 20 like this: (22-LEN(T$))/2. If we use VIC MAGICIAN as our title, the LENgth is 12 characters, and (22-12)/2 = 5. So this formula gives us the number of spaces on one side of our centered title. (If you're wondering why we have two parentheses after T$, it's because all BASIC formulas must have the same number of left and right parentheses and the two right parentheses balance out the two left parentheses. This is an over-simplification but it's important to remember because one of the most common mistakes when creating formulas for calculation is not balancing left and right parentheses.)

Next, we use a counting command to tell the VIC to count five spaces and PRINT our title. We do this by saying FOR X = 1 TO 5 (remember our whole formula equals 5), PRINT a SPACE. The semicolon means print everything next to each other, and the NEXT command means "keep PRINTing a SPACE until the upper limit (5) is reached." After our five spaces are PRINTed, we PRINT T$, which is our title.

An Easy Way to Reverse a User Defined Title

The following program reverses the user-defined title and displays it at the top of the screen, flush left:

10 PRINT" [SHIFT] [CLR/HOME] TITLE";:INPUTT$
20 PRINT" [SHIFT] [CLR/HOME] ";:PRINT" [CTRL] [RVS ON] " T$;:FORX=LEN (T$) TO21:PRINT "[CTRL] [RVS ON] SPACE ";:NEXT

Adding Colors to Titles

Another dimension you can add to your screen titles is color. The color command works just like the [RVS ON] command. Hold down the [CTRL] key and press the color key you want. The following program PRINTs a title in red, then returns the color to blue so the rest of the program will continue in blue. [CTRL] [RED] means hold down

vic magician

the [CTRL] key and press the key marked [RED] on the keyboard:

TYPE THE "3" KEY

```
10 PRINT" [SHIFT] [CLR/HOME] TITLE";:INPUTT$
20 PRINT" [SHIFT] [CLR/HOME] ";:PRINT" [CTRL]
   [RVS ON] [CTRL  RED] "
   T$;:FORX=LEN (T$) TO21:PRINT" [CTRL]
   [RVS ON] [SPACE] " ;:NEXT:PRINT" [CTRL] [BLUE] "
```

TYPE THE "7" KEY

Experiment!

There are lots of other ways to create interesting screen titles. For example, try centering the title in the middle of the screen (hint: use the [CRSR DOWN] key in a PRINT statement, and the same centering formula we used for centering the title horizontally). You can also create borders around your titles, on both sides or on top and bottom, by using a graphic character.

Try putting graphic lines or bars above or below your titles, like this:

```
5 PRINT "[SHIFT][CLR/HOME]";:FORG = 1TO22:
  PRINT"  ";:NEXT
10 PRINT"VIC MAGICIAN"
20 FORG = 1TO22:PRINT"  ";:NEXT
```

insert above title

title line(s) here

insert below title

Hold down the SHIFT key and type the "C" key to get this graphic symbol

Here's another variation to try. This program centers a one-line title on the screen. Try changing the number 10 in line 20 to move the title higher or lower on the screen:

CRSR DOWN WILL APPEAR AS A REVERSE Q on your screen.

```
10 PRINT" [CLR/HOME] TITLE";:INPUTT$
20 PRINT" [CLR/HOME] ":FORX = 1TO10:
   PRINT" [CRSR DOWN] ";:NEXT
30 FORX = 1TO(22-LEN(T$))/2:PRINT" [SPACE] "
   ;:NEXTX:PRINTT$
```

Notice that in line 20 we told the VIC to PRINT ten CURSOR-DOWNs, moving the position of the cursor 10 spaces down. Then in line 30 we used our centering formula to center the title in the middle of the screen. You can include spaces, cursor up or down movements, Control Reverse On and Off, and Control Color commands... as long as you put them in quotation marks. Just PRINT them like any letter, number or graphic symbol and they will appear in your program.

As a final variation you might want to "announce" your title with a sound effect. Try putting this one-line program on the line immediately before the line that PRINTs your title:

```
25 POKE36878,15:FORM = 200TO250:POKE
   36876,M:NEXTM:POKE36876,0
```

Now you're on your own. Try different colors, sound effects, screen positioning, and other variations. C

Foolin' with Boolean

by Jim Butterfield

There's something in your Commodore computer that's so simple—and so powerful—that you probably won't notice it unless you're told. It's called "boolean variables" and it can change your (programming) life.

At first, it seems confusing. A BASIC line such as X=Y=7 seems wrong. But if you try it, you will not get SYNTAX ERROR... the computer will happily accept the command and print READY. Well, if it works, maybe it means: set both X and Y to 7. Nope. PRINT X;Y will give values of 0.

Using parentheses helps make things more understandable. If we had typed: X=(Y=7) we can see that it makes sense if we add the following information: X will end up as either "true" or "false". If Y=7 then X will be "true" (that's represented by a value of −1), otherwise X will be "false" (a value of 0). X has a new meaning. It's not really a value... it's a true/false condition. We call this type of thing a "boolean" variable.

If X can be true or false, we should be able to say such things as:

IF X THEN PRINT "HELLO"

...which means, if X is true, perform the rest of the command. If you've been typing along, X will be false and so the command will not print HELLO. We can try the inverse by entering:

IF NOT X THEN PRINT "HI"

...and for "false" X the line will be completely executed and print HI.

Boolean Types

Any numeric value can be a boolean: integers or floating point variables, and even array elements. Arrays? That means you can have tables of "true" and "false" conditions.

Keep in mind that in making boolean variables, you can use any true-or-false test. That includes strings, comparisons, and tests connected with OR or AND relationships. Here are some valid boolean calculations:

```
A=(X$="YES")
B=(R>20)
C=(M=2*INT(M/2))
D=(M=2 AND Y=1984)
```

Let's look at the last one more closely. It seems to be looking for a leap year (1984, month 2) using an AND expression. Each of the relationships on either side of the AND is an individual test... and wherever we have a test, we may replace it with a boolean value. So we might say:

```
A=(M=2)
B=(Y=1984)
D=(A AND B)
```

It's all logical—literally. When you look at an expression such as the one above, think: A is a boolean, either true or false; the same is true of B; therefore, if both A and B are true, then D will also be true.

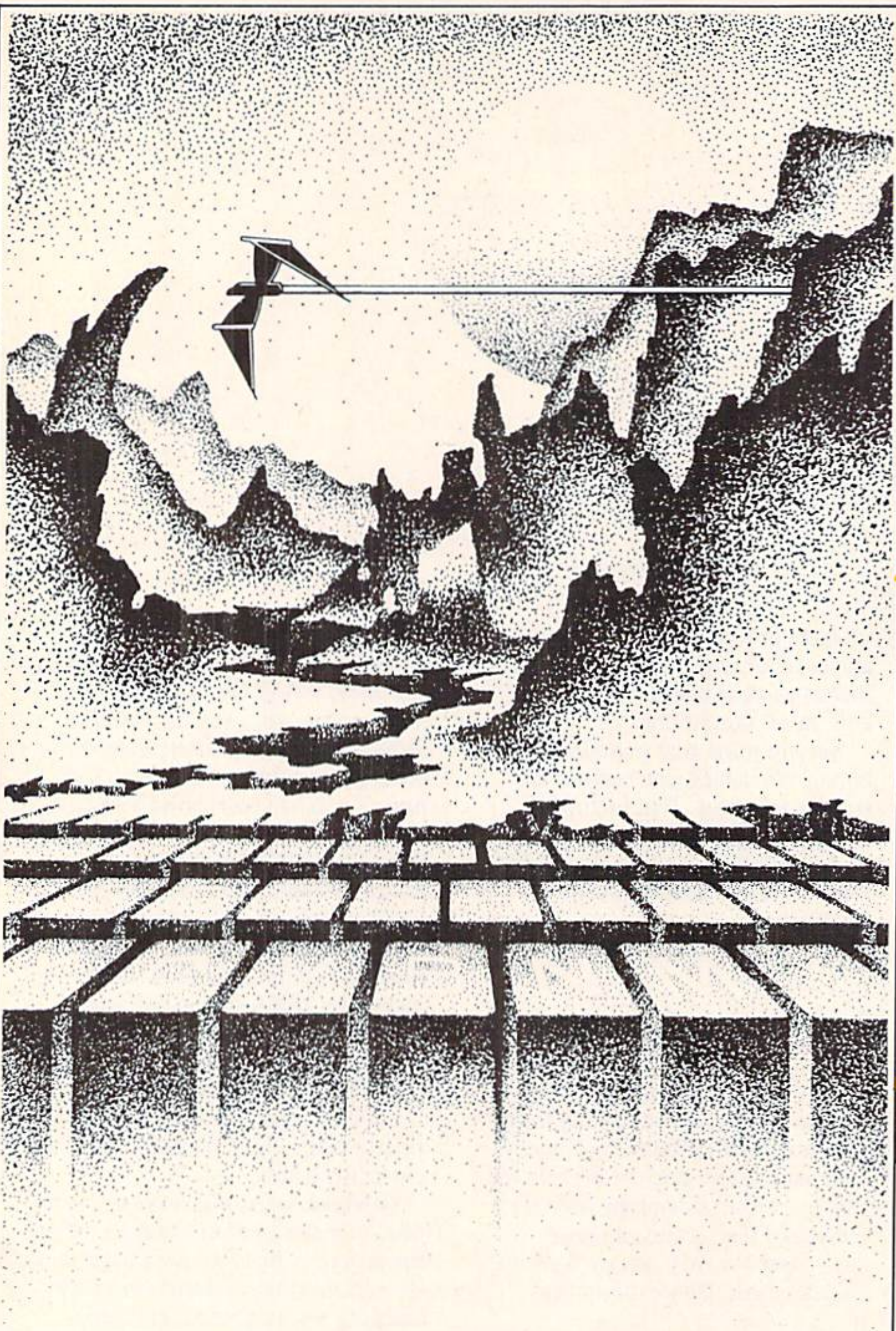
Some clever programmers (not you and I, of course) like to obscure things by leaving off the parentheses, and leaving out spaces. The previous expressions would then become D=AANDB. To confuse you further, the booleans are chosen with letters like P and K, so that we may end up with something like X=PORK (P OR K, get it?). Don't let it confuse you. You may be comforted to know that such programmers often get caught by their own cleverness; if you had two boolean values called T and Y, and tried to code something like IF TANDY THEN..., the machine would quite properly print SYNTAX ERROR. Why? Not because of any brand considerations, but because the computer would think you meant the TAN function and would run into trouble there.

Some Handy Uses

Let's talk about a number of situations in which boolean values are especially handy.

Deferred decisions: Sometimes you need to make a test at a certain part of the program, but you don't want to use the results of the test until later. For example, you're going to do some arithmetic on variable X; when you're finished, you want to take some action based on whether X was originally greater than 99. We've changed X, so the test must be deferred. Before the arithmetic, we do the test with B=(X>99). Now we go ahead with the arithmetic. Finally, we can act on the earlier test by writing IF B THEN....

Complex expressions: Sometimes we have a number of tests grouped together and it's hard to keep them all straight. We can make everything very orderly and very neat by assigning interesting conditions as boolean



Design & Illustration: Paul Stephenson

ADVENTURE.

THE KEY IS YOUR COMPUTER

CHOMPER MAN

High speed machine action. Don't let the bullies catch you in a game packed full of machine code.

Available for COMMODORE 64™

PLAYED WITH JOYSTICK OR KEYBOARD

$19.95

TREK

The classic game that has fascinated computerists for over a decade. Commanding the bridge of your starship, you explore the galaxy, fending off the Klingon invasion with your phasers and photon torpedoes, at the same time conserving your limited time and energy. A real bargain at **$14.95.**

Available for VIC-20™ and COMMODORE 64™

PLAYED WITH KEYBOARD

$14.95

SPACE MAZE

The advanced civilization of BRAL has decided to test the intelligence of the primitive beings who inhabit the planet Earth. You have been chosen as the human rat, to find your way out of this maze in space they have constructed just for you. You've never been in a maze like this before. As you walk through the hallways you see a 3-dimensional view of the corridors leading forward, right and left, as well as up and down! It is up to you to prove the worthiness of the human race. Fast machine action.

Available for VIC-20™

PLAYED WITH JOYSTICK

$14.95

KONGO KONG

Climb ladders, avoid the barrels the crazy ape is rolling at you, and rescue the damsel. Fast machine code action.

Available for VIC-20™ and COMMODORE 64™

PLAYED WITH JOYSTICK OR KEYBOARD

$19.95

POLLYWOG

Hop your tadpole across danger to safety of home. Machine code for fast action.

Available for VIC-20™

PLAYED WITH JOYSTICK

$19.95

GRAVE ROBBERS

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

Available for VIC-20™ and COMMODORE 64™

PLAYED WITH KEYBOARD

$14.95

ANNIHILATOR

Defend your planet against the hostile aliens. All machine code makes this program one of our best arcade games.

Available for VIC-20™ and COMMODORE 64™

PLAYED WITH JOYSTICK

$19.95

ADVENTURE PACK I

(3 programs)

Jack and the Beanstalk Defeat the Giant in your quest for his Golden Hen.

Computer Adventure Re-live the "excitement" of getting your computer. An adventure with a very different flavor.

Moon Base Alpha You must find a way to destroy the meteor that is racing towards your base, or else all moon colonies will be demolished!

Available for VIC-20™ and COMMODORE 64™

PLAYED WITH KEYBOARD

$19.95

ADVENTURE PACK II

(3 programs)

African Escape As the sole survivor of a plane crash, you must find your way out of the dark continent.

Hospital Adventure You are a spy whose mission is to complete the bungled assassination attempt on the evil dictator, who is recuperating in the hospital under heavy guard.

Bomb Threat Get back to town to warn the authorities of the bomb planted by the terrorists who left you prisoner at their hideout.

Available for VIC 20™ and COMMODORE 64™

PLAYED WITH KEYBOARD

$19.95

VIC 20 and **Commodore 64** are trademarks of Commodore Electronics Ltd.

VICTORY SOFTWARE®

VICTORY SOFTWARE INC.
7 Valley Brook Road, Paoli, PA 19301
(215) 296-3787

Programs for the VIC-20™ and the COMMODORE 64™. All programs fit in standard memory. All programs are on cassette tape.

Check your local dealer or order directly. **ORDERING** We accept personal checks, money orders, VISA, and MASTER-CHARGE. Charge orders please include number and expiration date. Overseas orders please use charge or have check payable through a U.S. bank. Add $1.50 postage and handling per order. PA residents please add 6% sales tax.

VISA

master charge

values. Thus we can set things like B=balance over $1000, V=high volume customer, P=payment made within the last 60 days. Now we've done our tests, we can efficiently code things like:

IF NOT B AND NOT V AND NOT P THEN PRINT "DEADBEAT!"
IF B AND V AND NOT P THEN PRINT "HURRY IT UP!"
IF NOT B AND V AND P THEN PRINT "WHERE ARE YOU?"

The booleans make the combinations much more readable.

IF-THEN-ELSE equivalents. You can make logic-tight alternative decisions using boolean values. Suppose we wished to code: IF the month is less than 12 THEN add one to the month, ELSE set the month to 1 and add 1 to the year. This could get messy, since the month might change while we are still testing it, but it turns into crisp coding with:

```
B=(M<12)
IF B THEN M=M+1
IF NOT B THEN M=1:Y=Y+1
```

An Example: Bagels

Bagels is a simple guessing game. It has been written many times before, but let's do it this time with a boolean slant to the coding. The tough part of a bagels program, by the way, is making sure that letters are not counted twice; so that if the secret code is ABFF and I guess FAAC I won't have the A or F matches counted more than once.

The object is to guess the computer's secret code. Let's make a secret code as an array of four values:

```
100 REM BOOLEAN BAGELS
    - JIM BUTTERFIELD
110 DIM C(4),X(4),M(4)
    ,N(4)
```

Array C will hold the mystery combination. X will be your guess. M and N are "matching" flags—they will be in boolean. Now let's scramble our random number generator, so that each game will be different, and give instructions.

```
120 X=RND(0)
130 PRINT "TRY TO
    GUESS MY CODE"
140 PRINT "OF 4
    CHARACTERS USING"
150 PRINT "LETTERS A
    TO F..."
160 PRINT "WATCH FOR
    MULTIPLES."
```

Here comes the mystery combination, calculated by the computer and placed into array C:

```
200 FOR J=1 TO 4
210 C(J)=INT(RND(1)
    *6+1)
220 NEXT J
```

We will count the guesses as they are made. Here comes the question:

```
230 G=1
240 PRINT"GUESS";G;"? ";
```

The guesses are inputted into array X. We use a GET statement and forbid the user to type any keys other than A to F:

```
250 FOR J=1 TO 4
260 GET X$:IF X$=""GOTO
    260
270 X=ASC(X$)-64
280 IF X<1 OR X>6 GOTO
    260
290 PRINT X$;:X(J)=X:
    NEXT J
```

Here comes the testing for matches. First, we need to test to see how many exact matches we have—letters that match in the same place. Note how we use array M(J) as a boolean; M(J) will be true if a match, and false if no match.

```
300 M=0:N=0
310 FOR J=1 TO 4
320 M(J)=(C(J)=X(J))
330 N(J)=M(J):IF M(J)
    THEN M=M+1
340 NEXT J
```

We mark off both M(J) and N(J) as true if we find a match. This tells us not to try to match these letters again. Now we go for the out-of-position matching:

```
350 FOR J=1 TO 4:IF
    M(J) GOTO 410
```

If M(J) is true, we've already matched this letter, so we skip.

```
350 FOR K=1 TO 4:IF
    N(K) GOTO 400
```

If N(K) is true, we've matched the letter of the guess and we skip. Now the test for a match:

```
370 B=(C(J)=X(K))
380 M(J)=B:N(K)=B
```

Again, if we find a match, we mark off the appropriate position in both the mystery number and the guess so that it won't be used again. We should also count the matches we find, and if there's a match skip to the next letter:

```
390 IF B THEN N=N+1
    :GOTO 410
400 NEXT K
410 NEXT J
```

Now, we've found the number of exact and out-of-place matches. We could just go ahead and print them. On the other hand, we have something special to do if the player has guessed four exact letters: the answer has been found and we may need special action. Let's log this condition into a boolean:

```
420 B=(M=4)
430 PRINT M;"EXACT";N;
    "MATCHES"
440 G=G+1:IF NOT B AND
    G<11 GOTO 240
```

Now we take advantage of our boolean, B:

```
450 IF B THEN PRINT
    "YOU GOT IT"
```

The next four lines are overdoing it, really. But I wanted to show that you can take booleans to the extreme. If you can't fit all your IF action onto one line, just keep repeating your boolean decision:

```
460 IF NOT B THEN PRINT
    "IT WAS ...";
470 IF NOT B THEN FOR
    J=1 TO 4
480 IF NOT B THEN PRINT
    CHR$(C(J)+64);
490 IF NOT B THEN NEXT
    J
```

And to complete the game, we'll ask the user if he or she wants to try again:

```
500 PRINT:INPUT
    "ANOTHER";X$
510 IF ASC(X$)=89 GOTO
    200
```

Summary

Booleans may seem like a trick when you first meet them. But they are sound coding. Understanding them is the key to more efficient programs. C

Congratulations to Commodore Challenge Contest Winners

George Currie of Eugene, Oregon, for *PIO*
and
George Short of Easton, Maryland, for *The Maze*

We received a slew of superb software from our talented readers over the past few months—so much so that we ended up with a tie between these two excellent game programs. The judges were evenly divided between the two, except, that is, for the judge who said we should publish both. So that's what we decided to do.

Thanks to all of you who submitted programs. They were all, as usual, great fun. (We never seem to have any problems getting volunteer judges for this department, for some reason.) And keep up the good work. Maybe next time the winner will be you.

The Commodore Challenge
Prizes • Prizes • Prizes

If you've been playing around at home developing original games and programs for your unexpanded VIC 20, send your best—on cassette or disk, please—to the Commodore Challenge contest. Include a brief description of the program's purpose, including documentation on how to use it. If it's a game, be sure to include instructions.

Programs requiring memory expansion are eligible, too, but will not be published unless space allows.

Winners will receive a VIC 20 8K Memory Expander Cartridge. All entries become the property of Commodore Business Machines, Inc., upon submission. Winning entries published by POWER/PLAY will become public domain software.

Fill out the entry form below, and submit it with your game or program to:

Commodore Business Machines, Inc.
1200 Wilson Drive
West Chester, PA 19380
Attn: POWER/PLAY

Commodore Challenge Contest . . . Entry Form

Name________________ Age________ Phone__________

Address________________ Program Title__________

City________________ State________ Zip________

I understand that my software entry becomes the property of Commodore Business Machines, Inc., upon submission, and that winning entries published by POWER/PLAY become public domain software.

Signature__________________________

Parent's signature, if contestant is minor________________

Void Where Prohibited

PIO

by George Currie
For Unexpanded VIC 20

Guide your centipede-like animal down a course full of trees in this fast-paced game. Uses keyboard control only. Instructions are in the program.

```
5 H1=700:H2=600:H3=500:H1$="HEX":H2$="HEX":H3$="HEX":H4=300:H4$="HEX"
10 LI=0:PRINT"     HIGH SCORES      ":POKE36879,190
11 PRINT"LEVEL1-"INT(H1);H1$
12 PRINT"LEVEL2-"INT(H2);H2$
13 PRINT"LEVEL3-"INT(H3);H3$
14 PRINT"LEVEL4-"INT(H4);H4$
15 PRINT"                 "
16 PRINT"                 "
17 PRINT"                 "
18 PRINT"                 "
19 PRINT"                 "
20 PRINT"                 "
21 PRINT"                 "
22 PRINT"                 "
23 PRINT"                 "
24 PRINT"PIO MCMLXXXII"
25 FORC1=0TO7:FORC2=38835TO38645STEP-1:POKEC2,C1:NEXT:NEXT
30 PRINT"THE OBJECT IS TO GUIDEPIO''DOWN THE COURSE WHILE TRYING TOA
VOID THE ";
31 PRINT" TREES":PRINT"EVERY TIME PIO NAILS  A TREE HE IS DISABLED FOR A SECON
D THEN HE  IS OK"
40 PRINT"PRESS Z TO MOVEPIO TOTHE LEFT":PRINT"PRESS CRSR<> TO MOVE  PIO
 TO THE RIGHT"
41 PRINT"PRESS F1 FOR MORE INFO"
42 GETK$:IFK$=CHR$(133)THEN45
43 GOTO42
45 PRINT"THERE ARE 4 LEVELS OF PLAY:":PRINT"1-2 TREES WIDE(EASY)
46 PRINT"2-3 TREES WIDE(SO-SO)":PRINT"3-4 TREES WIDE(HARD)":PRINT"4-
5 TREES WIDE(HA-HA)"
47 PRINT"PRESS THE LEVEL YOU   WANT TO PLAY AND THE  GAME WILL START        "
48 PRINT" FOR THE HIGH SCORES-         PRESS F7       "
```

```
90 PE=PEEK(197):IFPE=0THENNU=2:GOTO100
91 IFPE=56THENNU=3:GOTO100
92 IFPE=1THENNU=4:GOTO100
93 IFPE=57THENNU=5:GOTO100
94 IFPE=63THEN10
99 GOTO90
100 OF=30720:S1=36877:POKE36878,15:POKE36879,29:A=7911:XC=0:SC=0:LA=0:S3=36875:F
D=0
101 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒GET READY!":FORT=1TO2000:NEXT
111 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
112 PRINT"▒▒XXXXXXXSTARTXXXXXXXX▒";:SC$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒SCORE"
120 TR$="♠♠♠♠♠"
150 X=(INT(RND(1)*16)+3)-INT(NU/2):IFXC=0THENPRINTTAB(X);RIGHT$(TR$,NU)
153 IFXC=1THENPRINT
155 SC=SC+1.5:IFPEEK(A)=88THENQ=11:B=1:GOTO200
156 IFPEEK(A)=230THENPRINTSC$;INT(SC):FORT=1TO5000:NEXT:GOTO300
160 IFPEEK(197)=33ANDA>7902THENA=A-1
165 IFPEEK(197)=23ANDA<7918THENA=A+1
176 POKEA,90:POKEA+OF,6:LA=LA+1:IFLA=500THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";:XC=
1
185 GOTO150
200 POKES1,185:FORT=1TO400:NEXT:POKES1,0:SC=SC-9:GOTO176
300 IFNU=3ANDSC>H2THENH2=SC:GOTO334
305 IFNU=2ANDSC>H1THENH1=SC:GOTO324
310 IFNU=4ANDSC>H3THENH3=SC:GOTO344
311 IFNU=5ANDSC>H4THENH4=SC:GOTO354
315 GOTO10
324 GOSUB500:POKES3,0:PRINT"ENTER INITIALS":INPUTR$:H1$=LEFT$(R$,3):GOTO10
334 GOSUB500:POKES3,0:PRINT"ENTER INITIALS":INPUTY$:H2$=LEFT$(Y$,3):GOTO10
344 GOSUB500:POKES3,0:PRINT"ENTER INITIALS":INPUTZ$:H3$=LEFT$(Z$,3):GOTO10
354 GOSUB500:POKES3,0:PRINT"ENTER INTIALS":INPUTU$:H4$=LEFT$(U$,3):GOTO10
500 FORFD=1TO20:PRINT"▒▒▒▒▒▒▒YOU HAVE HIGH SCORE!!!":FORT=1TO50:NEXT:POKES3,240
505 PRINT"▒▒▒▒▒▒YOU HAVE HIGH SCORE!!!":FORT=1TO50:NEXT:POKES3,220:NEXTFD:RETURN
```

The Maze

by George Short
For Unexpanded VIC 20

The object of this intriguing game is to collect as many points as possible while traveling through a series of mazes. You control the wizard with the N, W and E keys, which make him move north, west and east, respectively.

The wizard enters the maze from the south and must exit through the north wall within 12 seconds. The clock in the upper right corner of the screen keeps you aware of the passing time.

Scattered through the mazes are bottles of magic potion that allow the wizard to cast a spell. Using the J key to activate the spell, the wizard can cause walls to disappear, and thus escape from deadend situations quickly.

A bonus "flash" is awarded for every five mazes the wizard completes. The wizard can use a flash (the B key) to go directly to the north wall, destroying everything in his path, and escape to safety.

The player has three wizards at the beginning of the game. Wizards are destroyed if they hit a wall or fail to escape a maze in the allotted time. When all three are killed, the game ends. **C**

```
0 N3=30720:ME=3:SC=0:MT=50:T=0:S1=0:W1=0:POKE36878,15:PRINT"ℷ■■■■■■■■■■THE MAZE
":T=1
1 Y=7691:POKE36869,255:POKE52,28:POKE56,28:FORI=7168TO7664:POKEI,PEEK(I+25600):N
EXTI
2 FORCC=7168TO7175:READAA:POKECC,AA:NEXTCC:DATA26,26,126,26,24,60,126,0
3 FORCC=7384TO7391:READAA:POKECC,AA:NEXTCC:DATA24,24,24,24,24,60,60,24
4 FORCC=7399TO7406:READAA:POKECC,AA:NEXTCC:DATA 254,254,254,254,254,254,254,254
5 PRINT"ℷ";:Z=0
6 X=7679:FORA1=1TOMT:A=INT(RND(1)*475)+1:POKEX+A,29:POKEX+N3+A,0:NEXTA1
7 X1=7680:B=INT(RND(1)*6)+1:FORC=1TOB:B1=INT(RND(1)*200)+1:B2=INT(RND(1)*6)+2
8 POKEX1+B1,27:POKEX1+B1+N3,B2:NEXTC
9 Y=Y+484:TI$="000000":POKE198,0
10 POKEY,0:POKEY+N3,5:IFY<7702THEN35
11 GETD$:PRINTRIGHT$(TI$,2):PRINT"ℷ";:IFTI$>"000011"THEN21
12 IFD$=""THEN11
13 POKEY,32:IFD$="N"THENY=Y-22:GOTO18
14 IFD$="E"THENY=Y+1:GOTO18
15 IFD$="W"THENY=Y-1:GOTO18
16 IFD$="J"THEN27
17 IFD$="B"THEN22
18 IFPEEK(Y)=32THENPOKE36874,200:POKE36874,0:GOTO10
19 IFPEEK(Y)=29THEN21
20 IFPEEK(Y)=27THENS1=S1+1:Z=Z+1:POKE36874,250:POKE36874,0:GOTO10
21 ME=ME-1:POKEY,42:POKE36876,200:FORE=1TO300:NEXTE:POKE36876,140:GOTO34
22 IFW1>0THENW1=W1-1:GOTO24
23 GOTO18
```

```
24 FORX5=1TO24:POKEY,32:Y=Y-22:POKEY,0:POKEY+N3,2:FORLL=1TO5:NEXTLL:IFY<7702THEN
35
26 POKE36874,200:POKE36874,0:NEXTX5
27 IFS1=0THEN10
28 POKE36876,250:POKE36876,0
29 POKEY-22,32:POKEY-1,32:POKEY+1,32:S1=S1-1:GOTO10
30 IFW1>0THENW1=W1-1:GOTO24
31 GOTO18
32 FORX5=1TO24:POKEY,32:Y=Y-22:POKEY,0:POKEY+N3,2:FORLL=1TO5:NEXTLL:IFY<7702THEN
35
33 POKE36874,200:POKE36874,0:NEXTX5
34 FORPP=1TO500:NEXTPP:PRINT"♥":PRINT"YOU BLEW IT!!!":Y=7691:GOTO37
35 FOROI=1TO10:NEXTOI:PRINT"♥":PRINT"YOU MADE IT!!!!!!":IFT/5=INT(T/5)THENW1=W1+
1
36 GOSUB42:SC=SC+(Z*5)+T:T=T+1
37 PRINT:PRINT"SCORE-"SC:PRINT"MEN-"ME
38 PRINT:PRINT:PRINT"SPELLS-"S1:PRINT"FLASHS-"W1
39 FORFF=1TO500:NEXTFF:IFME=0THEN41
40 POKE36876,0:PRINT"QQQQ]]]ENTERING MAZE-"T:FORU=1TO2500:NEXTU:MT=MT+20:GOTO5
41 PRINT"QQQQ]]]]]]GAME OVER":POKE36876,0:END
42 POKE36878,15:FORL=228TO240:POKE36876,L:FORG=1TO50:NEXTG,L:POKE36876,0:RETURN
```

high scores

Up until last issue we had been accepting scores on the honor system. But we thought it was time to start getting more official. So, from now on, in order to have your high score published, you'll have to send in a photo of the screen showing the score. Don't forget—a simple polaroid will do it.

BLUE MEANIES	1,260 Alan S. Newman, Fairfield, CT
CAR CHASE	75,865 Zach Coleman, Charlotte, NC
CLOWNS	
COSMIC CRUNCHER	215,000 Barbara Schreiber, New York, NY
DRAW POKER	17,410 R. Callia, Torrance, CA
JUPITER LANDER	207,400 Christopher Champlain, St. Petersburg, FL
GORF	110,040 Robert Devantier, Davison, MI
MIDNIGHT DRIVE	14.11 km Nathan Mehl, Newark, DE
MOLE ATTACK	331 Heda Takaya, Saskatoon, Sasketchewan
OMEGA RACE	260,050—5 ships Ben Piper, Chico, CA
PINBALL	1,500,000 Joe Ferrari, Commodore, Toronto
RADAR RAT RACE	122,240 John Higginson, South Holland, IL
RAID ON FORT KNOX	
SEA WOLF	10,080 Jimmy Kuhn, Norfolk, VA
SKY IS FALLING	13,810 Rachel Koons, Drexel Hill, PA
SLITHER	261 Amy Miles, Mt. Pleasant, MI
SUPER ALIEN	45,700 Robert Schaeffer, Brookline, MA
SUPER SLITHER	167 David Goldberg, Richardson, TX
SUPER SLOT	7,306 coins Jerry Krueger, Cary, IL
SUPER SMASH	
VIC AVENGER	10,190 Chad McCubbins, Coatesville, IN

If your score didn't set a record this time, keep playing! Maybe you'll topple these champion gamesters next time!

Remote Terminal

by John O'Brien

Our user group profiles usually spotlight the large and well-established groups. But this time we'd like to tell the story of one of the smallest and newest groups from whom we recently received two interesting letters. The first letter was signed by all five members.

The group is in the town of Old Harbor, an Alaskan Aleut village of about 350 residents. The Aleuts are two tribes related to the Eskimos who live on the Aleutian Islands off southwest Alaska. Old Harbor is on Kodiak Island and can be reached, according to the first letter "by a ten hour boat trip, through some of the world's most treacherous waters, or by small plane, through equally treacherous airspace."

The group's name is COMPOOH-T, which stands for Commodore Programmers of Old Harbor. The T comes from the fact that they are all teachers. The group proudly proclaims that they comprise 42% of the teaching and administrative staff of Old Harbor School. It's not a large group—it's a small school—120 students (K-12) and 12 staff members.

Each teaches at least two or three different grade levels. They all migrated to Alaska to teach the natives, and, as group member Paul Mercer put it, "to get away from the hustle and bustle of the lower 48," which is what they call the rest of the United States. Using computers in the classroom sparked an interest in the teachers that seems insatiable.

"Meetings are held each morning at 8:00 a.m. in the hallway of the elementary school, right in front of room 2 (the rooms are not numbered but no matter which way you start counting from it's room 2 because it's in the middle). Meetings last until our principal, not a member, breaks up the meeting, usually after school has started," according to Mercer.

If you think you have a little trouble getting your dealer to carry the peripherals and software you want, then imagine the hardship of buying *everything* by mail. If that isn't bad enough, consider the fact that even then they must often wait until the weather clears so the mail plane can get through (fog is a big problem).

Consequently, the group has taken to writing many of their own programs. They boast of a software library that is almost 100% written by members. Mercer's advice to new members is "after you've written two lines of a program—SAVE IT!!!" Power fluctuations and outages are an hourly occurrence.

Their first letter stated that the group had five members, one VIC 20, two Commodore 64's, three datassettes, two disk drives and "lots of joysticks." When the second letter arrived, things were really looking up. The mail had finally arrived with their long awaited software, three more 64's had been ordered for the school and membership was up to eight and climbing.

In fact, COMPOOH-T recently began extending open memberships to anyone interested in joining. For one dollar you can become a member of, according to Mercer, "the most remote Commodore user group." For more information write to COMPOOH-T, C/O Box 118, Old Harbor, AK 99643. See you in front of Room 2, promptly at 8:00. **C**

Commodore Programmers of Old Harbor, Alaska. First row, l to r: Jane Koyuk, Patricia McDonald, Kate Lowen. Second row, l to r: Jim Preston, Paul Mercer, Dave Wilson, Walt Loewen.

access: commodore user groups

User Group Listing

ALABAMA

Huntsville PET Users Club
9002 Berclair Road
Huntsville, AL 35802
Contact: Hal Carey
Meetings: every 2nd Thursday

ALASKA

COMPOOH-T
c/o Box 118
Old Harbor, AK 99643
(907) 286-2213

ARIZONA

VIC Users Group
2612 E. Covina
Mesa, AZ 85203
Contact: Paul Muffuletto

Commodore User Group
Metro Computer Store
4500 E. Speedway,
Suite 13
Tucson, AZ 85712
(602) 323-3116

Central Arizona PET People
842 W. Calle del Norte
Chandler, AZ 85224
(602) 899-3622
Roy Schahrer

ACUG
c/o Home Computer Service
2028 W. Camelback Rd.
Phoenix, AZ 85015
(602) 249-1186
Dan Deacon
First Wed. of month

West Mesa VIC
2351 S. Standage
Mesa, AZ 85202
Kenneth S. Epstein

ARKANSAS

Commodore/PET Users Club
Conway Middle School
Davis Street
Conway, AR 72032
Contact: Geneva Bowlin

Booneville 64 Club
c/o A. R. Hederich
Elementary School
401 W. 5th St.
Booneville, AR 72927
Mary Taff

CALIFORNIA

SCPUG Southern California PET Users Group
c/o Data Equipment Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: First Tuesday of each month

California VIC Users Group
c/o Data Equipment Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: Second Tues. of each month

Commodore Users Club
1041 Foxenwoods Drive
Santa Maria, CA 93455
(805) 937-4106
Contact: Greg Johnson

Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA
(213) 849-4094
1st Wed. 6 p.m.

Valley Computer Club
1913 Booth Road
Ceres, CA 95307

PUG of Silicon Valley
22355 Rancho Ventura Road
Cupertino, CA 95014

Lincoln Computer Club
750 E. Yosemite
Manteca, CA 95336
John Fung, Advisor

PET on the Air
525 Crestlake Drive
San Francisco, CA 94132
Max J. Babin, Secretary

PALS (Pets Around)
Livermore Society
886 South K
Livermore, CA 94550
(415) 449-1084
Every third Wednesday
7:30 p.m.
Contact: J. Johnson

SPHINX
7615 Leviston Ave.
El Cerrito, CA 94530
(415) 527-9286
Bill MacCracken

San Diego PUG
c/o D. Costarakis
3562 Union Street
(714) 235-7626
7 a.m.-4 p.m.

Walnut Creek PET Users Club
1815 Ygnacio Valley Road
Walnut Creek, CA 94596

Jurupa Wizards
4526 Kingsbury Pl.
Riverside, CA 92503
Contact: Walter J. Scott

The Commodore Connection
2301 Mission St.
Santa Cruz, CA 95060
(408) 425-8054
Bud Massey

San Fernando Valley Commodore Users Group
21208 Nashville
Chatsworth, CA 91311
(213) 709-4736
Tom Lynch
2nd Wed. 7:30

VACUUM
277 E. 10th Ave.
Chico, CA 95926
(916) 891-8085
Mike Casella
2nd Monday of month

VIC 20 Users Group
2791 McBride Ln. #121
Santa Rosa, CA
(707) 575-9836
Tyson Verse

South Bay Commodore Users Group
1402 W. 218th St.
Torrance, CA 90501
Contact: Earl Evans

Slo VIC 20/64 Computer Club
1766 9th St.
Los Osos, CA

The Diamond Bar R.O.P. Users Club
c/o Rincom School
2800 Hollingworth
West Covina, CA 91792
(213) 965-1696
Don McIntosh

Commodore Interest Association
c/o Computer Data
14660 La Paz Dr.
Victorville, CA 92392
Mark Finley

Fairfield VIC 20 Club
1336 McKinley St.
Fairfield, CA 94533
(707) 427-0143
Al Brewer
1st & 3rd Tues. at 7 p.m.

Computer Barn Computer Club
319 Main St.
Suite #2
Salinas, CA 93901
757-0788
S. Mark Vanderbilt

COLORADO

VICKIMPET Users Group
4 Waring Lane, Greenwood Village
Littleton, CO 80121
Contact: Louis Roehrs

Colorado Commodore Computer Club
2187 S. Golden Ct.
Denver, CO 80227
986-0577
Jack Moss
Meet: 2nd Wed.

CONNECTICUT

John F. Garbarino
Skiff Lane Masons Island
Mystic, CT 06355
(203) 536-9789

Commodore User Club
Wethersfield High School
411 Wolcott Hill Road
Wethersfield, CT 06109
Contact: Daniel G. Spaneas

VIC Users Club
c/o Edward Barszczewski
22 Tunxis Road
West Hartford, CT 06107

New London County Commodore Club
Doolittle Road
Preston, CT 06360
Contact: Dr. Walter Doolittle

FLORIDA

Jacksonville Area PET Society
401 Monument Road, #177
Jacksonville, FL 32211

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

South Florida PET Users Group
Dave Young
7170 S.W. 11th
West Hollywood, FL 33023
(305) 987-6982

VIC Users Club
c/o Ray Thigpen
4071 Edgewater Drive
Orlando, FL 32804

PETs and Friends
129 NE 44 St.
Miami, FL 33137
Richard Plumer

Sun Coast VICs
P.O. Box 1042
Indian Rocks Beach, FL 33535
Mark Weddell

Bay Commodore Users Group
c/o Gulf Coast Computer Exchange
241 N. Tyndall Pkwy.
P.O. Box 6215
Panama City, FL 32401
(904) 785-6441
Richard Scofield

Gainesville Commodore Users Club
3604-20A SW 31st Dr.
Gainesville, FL 32608
Louis Wallace

64 Users Group
P.O. Box 561689
Miami, FL 33156
(305) 274-3501
Eydie Sloane

Brandon Users Group
108 Anglewood Dr.
Brandon, FL 33511
(813) 685-5138
Paul Daugherty

Commodore 64/VIC 20 User Group
Martin Marietta Aerospace
P.O. Box 5837, MP 142
Orlando, FL 32855
(305) 352-3252/2266
Mr. Earl Preston

Brandon Commodore Users Group
414 E. Lumsden Rd.
Brandon, FL 33511

Gainesville Commodore Users Group
Santa Fe Community College
Gainesville, FL 32602
James E. Birdsell

GEORGIA

VIC Educators Users Group
Cherokee County Schools
110 Academy St.
Canton, GA 30114
Dr. Al Evans

Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

IDAHO

GHS Computer Club
c/o Grangeville High School
910 S. D St.
Grangeville, ID 83530
Don Kissinger

S.R.H.S. Computer Club
c/o Salmon River H.S.
Riggins, ID 83549
Barney Foster

Commodore Users
548 E. Center
Pocatello, ID 83201
(208) 233-0670
Leroy Jones

Eagle Rock Commodore Users Group
900 S. Emerson
Idaho Falls, ID 83401
Nancy J. Picker

ILLINOIS

Shelly Wernikoff
2731 N. Milwaukee
Avenue
Chicago, IL 60647

VIC 20/64 Users Support
Group
c/o David R. Tarvin
114 S. Clark Street
Pana, IL 62557
(217) 562-4568

Central Illinois PET User
Group
635 Maple
Mt. Zion, IL 62549
(217) 864-5320
Contact: Jim Oldfield

ASM/TED User Group
200 S. Century
Rantoul, IL 61866
(217) 893-4577
Contact: Brant Anderson

PET VIC Club (PVC)
40 S. Lincoln
Mundelein, IL 60060
Contact: Paul Schmidt,
President

Rockford Area PET Users
Group
1608 Benton Street
Rockford, IL 61107

Commodore Users Club
1707 East Main St.
Olney, IL 62450
Contact: David E. Lawless

VIC Chicago Club
3822 N. Bell Ave.
Chicago, IL 60618
John L. Rosengarten

Chicago Commodore 64
Users & Exchange Group
P.O. Box 14233
Chicago, IL 60614
Jim Robinson

Fox Valley PET Users
Group
833 Willow St.
Lake in the Hills, IL 60102
(312) 658-7321
Art DeKneef

The Commodore 64 Users
Group
4200 Commerce Ct., Suite 100
Lisle, IL 60532
(312) 369-6525
Gus Pagnotta

Oak Lawn Commodore Users Group
The Computer Store
11004 S. Cicero Ave.
Oak Lawn, IL 60453
(312) 499-1300
Bob Hughes

INDIANA

PET/64 Users
10136 E. 96th St.
Indianapolis, IN 46256
(317) 842-6353
Jerry Brinson

Cardinal Sales
6225 Coffman Road
Indianapolis, IN 46268
(317) 298-9650
Contact: Carol Wheeler

CHUG (Commodore
Hardware Users Group)
12104 Meadow Lane
Oaklandon, IN 46236
Contact: Ted Powell

VIC Indy Club
P.O. Box 11543
Indianapolis, IN 46201
(317) 898-8023
Ken Ralston

Northern Indiana
Commodore Enthusiasts
927 S. 26th St.
South Bend, IN 46615
Eric R. Bean

Commodore Users Group
1020 Michigan Ave.
Logansport, IN 46947
(219) 722-5205
Mark Bender

Computer Workshop VIC 20/64 Club
282 S. 600 W.
Hebron, IN 46341
(219) 988-4535
Mary O'Bringer

The National Science Clubs of America
Commodore Users Division
7704 Taft St.
Merrillville, IN 46410
Brian Lapley or Tom Vlasic

East Central Indiana VIC User Group
Rural Route #2
Portland, IN 47371
Stephen Erwin

National VIC 20 Program Exchange
102 Hickory Court
Portland, IN 47371
(219) 726-4202
Stephen Erwin

IOWA

Commodore User Group
114 8th St.
Ames, IA 50010

Quad City Commodore Club
1721 Grant St.
Bettendorf, IA 52722
(319) 355-2641
John Yigas

Commodore Users Group
965 2nd St.
Marion, IA 52302
(319) 377-5506
Vern Rotert
3rd Sun. of month

Siouxland Commodore Club
2700 Sheridan St.
Sioux City, IA 51104
(712) 258-7903
Gary Johnson
1st & 3rd Monday of month

421 W. 6th St.
Waterloo, IA 50702
(319) 232-1062
Frederick Volker

KANSAS

Wichita Area PET
Users Group
2231 Bullinger
Wichita, KS 67204
(316) 838-0518
Contact: Mel Zandler

Kansas Commodore
Computer Club
101 S. Burch
Olathe, KS 66061
Contact: Paul B. Howard

Commodore Users Group
6050 S. 183 St. West
Viola, KS 67149
Walter Lounsbery

KENTUCKY

VIC Connection
1010 S. Elm
Henderson, KY 42420
Jim Kemp

LOUISIANA

Franklin Parish Computer
Club
#3 Fair Ave.
Winnisboro, LA 71295
James D. Mays, Sr.

NOVA
917 Gordon St.
New Orleans, LA 70117
(504) 948-7643
Kenneth McGruder, Sr.

VIC 20 Users Group
5064 Bowdon St.
Marrero, LA 70072
(504) 341-5305
Wayne D. Lowery, R.N.

MARYLAND

Assoc. of Personal
Computer Users
5014 Rodman Road
Bethesda, MD 20016

Blue TUSK
700 East Joppa Road
Baltimore, MD 21204
Contact: Jim Hauff

House of Commodore
8835 Satyr Hill Road
Baltimore, MD 21234
Contact: Ernest J. Fischer

Long Lines Computer Club
323 N. Charles St., Rm. 201
Baltimore, MD 21201
Gene Moff

VIC & 64 Users Group
The Boyds Connection
21000 Clarksburg Rd.
Boyds, MD 20841
(301) 428-3174
Tom DeReggi

VIC 20 Users Group
23 Coventry Lane
Hagerstown, MD 21740
Joseph Rutkowski

Hagerstown Users Group
1201-B Marshall St.
Hagerstown, MD 21740
(301) 790-0968
Greg Stewart
1st & 3rd Friday of month 6:30 p.m.

Rockville VIC/64 Users Group
13013 Evanstown St.
Rockville, MD 20853
(301) 946-1564
Meryle or Tom Pounds

MASSACHUSETTS

Eastern Massachusetts
VIC Users Group
c/o Frank Ordway
7 Flagg Road
Marlboro, MA 02173

VIC Users Group
c/o Ilene Hoffman-Sholar
193 Garden St.
Needham, MA 02192

Commodore Users Club
Stoughton High School
Stoughton, MA 02072
Contact: Mike Lennon

Berkshire PET Lovers
CBM Users Group
Taconic High
Pittsfield, MA 01201

The Boston Computer
Society
Three Center Plaza
Boston, MA 02108
(617) 367-8080
Mary E. McCann

VIC Interface Club
c/o Procter & Gamble Inst.
Shop
780 Washington St.
Quincy, MA 02169
C. Gary Hall

Masspet Commodore Users Group
P.O. Box 307
East Taunton, MA 02718
David Rogers

Raytheon Commodore Users Group
Raytheon Company
Hartwell Rd. GRA-6
Bedford, MA 01730
John Rudy

Commodore 64 Users
Group of The Berkshires
184 Highland Ave.
Pittsfield, MA 01201
Ed Rucinski

access: commodore user groups

MICHIGAN

David Liem
14361 Warwick Street
Detroit, MI 48223

VIC Users Club
University of Michigan
School of Public Health
Ann Arbor, MI 48109
Contact: John Gannon

Commodore User Club
32303 Columbus Drive
Warren, MI 48093
Contact: Robert Steinbrecher

Commodore Users Group
c/o Family Computer
3947 W. 12 Mile Rd.
Berkley, MI 48072

W. Michigan VIC 20-64 Users
1311 Portland NE
Grand Rapids, MI 49505
(616) 459-7578
Jim D'Haem

VIC for Business
6027 Orchard Ct.
Lansing, MI 48910
Mike Marotta

South Computer Club
South Jr. High School
45201 Owen
Belleville, MI 48111
Ronald Ruppert

Commodore Users Group
c/o Eaton Rapids Medical Clinic
101 Spicerville Hwy.
Eaton Rapids, MI 48827
Albert Meinke III, M.D.

South East Michigan Pet Users Group
Box 214
Farmington, MI 48024
Norm Eisenberg

Commodore Computer Club
H. Dow High School, Rm #226
Midland, MI 48640
(517) 835-5130
John Walley
9:30 p.m. Sept/May

VIC, 64, PET Users Group
8439 Arlis Rd.
Union Lake, MI 48085
363-8539
Bert Searing

MINNESOTA

MUPET (Minnesota Users of PET)
P.O. Box 179
Annandale, MN 55302
c/o Jon T. Minerich

Twin Cities Commodore Computer Club
6623 Ives Lane
Maple Grove, MN 55369
(612) 424-2425
Contact: Rollie Schmidt

MISSOURI

KCPUG
5214 Blue Ridge Boulevard
Kansas City, MO 64133
Contact: Rick West
(816) 356-2382

PET SET Club of St. Louis
633 Bent Oak Drive
Lake St. Louis, MO 63367
(314) 625-2701 or 625-4576
Tony Ott

VIC INFONET
P.O. Box 1069
Branson, MO 65616
(417) 334-6099
Jory Sherman

Worth County PET Users Group
Grant City, MO
(816) 564-3551
David Hardy

Mid-Missouri Commodore Club
1804 Vandiver Dr.
Columbia, MO 65201
(314) 474-4511
Phil Bishop

MONTANA

Powder River Computer Club
Powder River County High School
Broadus, MT 59317
Contact: Jim Sampson

Commodore User Club
1109 West Broadway
Butte, MT 59701
Contact: Mike McCarthy

NEVADA

Las Vegas PET Users
4884 Iron Avenue
Las Vegas, NV 89110

NEW JERSEY

Amateur Computer Group
18 Alpine Drive
Wayne, NJ 07470

Somerset Users Club
49 Marcy Street
Somerset, NJ 08873
Contact: Robert Holzer

Educators Advisory
P.O. Box 186
Medford, NJ 08055
(609) 953-1200
John Handfield

VIC-TIMES
46 Wayne Street
Edison, NJ 08817
Thomas R. Molnar

VIC 20 User Group
67 Distler Ave.
W. Caldwell, NJ 07006
(201) 284-2281
G. M. Amin

VIC Software Development Club
77 Fomalhaut Ave.
Sewell, NJ 08080
H. P. Rosenberg

ACGNJ PET/VIC/CBM User Group
30 Riverview Terr.
Belle Mead, NJ 08502
(201) 359-3862
J. M. Pylka

NEW HAMPSHIRE

Northern New England Computer Society
P.O. Box 69
Berlin, NH 03570

TBH VIC-NICs
P.O. Box 981
Salem, NH 03079

NEW MEXICO

Commodore Users Group
6212 Karlson, NE
Albuquerque, NM 87113
(505) 821-5812
Danny Byrne

NEW YORK

Capital District PET Users
Ben Green
Albany Area, NY
(518) 370-1820

Long Island PET Society
Ralph Bressler
Harborfields HS
Taylor Avenue
Greenlawn, NY 11740

PET User Club of Westchester
P.O. Box 1280
White Plains, NY 10602
Contact: Ben Meyer

LIVE (Long Island VIC Enthusiasts)
17 Picadilly Road
Great Neck, NY 11023
Contact: Arnold Friedman

Commodore Masters
25 Croton Ave.
Staten Island, NY 10301
Contact: Stephen Farkouh

VIC Users Club
76 Radford St.
Staten Island, NY 10314
Contact: Michael Frantz

Rockland County Commodore Users Group
c/o Ross Garber
14 Hillside Court
Suffern, NY 10901
(914) 354-7439

West Chester County VIC Users Group
P.O. Box 146
Pelham, NY 10552
Joe Brown

SPUG
4782 Boston Post Rd.
Pelham, NY 10803
Paul Skipski

VIC 20 User Club
151-28 22nd Ave.
Whitestone, NY 11357
Jean F. Coppola

VIC 20 User Club
339 Park Ave.
Babylon, NY 11702
(516) 669-9126
Gary Overman

VIC User Group
1250 Ocean Ave.
Brooklyn, NY 11230
(212) 859-3030
Dr. Levitt

L&M Computer Club
VIC 20 & 64
4 Clinton St.
Tully, NY 13159
(315) 696-8904
Dick Mickelson

Commodore Users Group
1 Corwin Pl.
Lake Katrine, NY 12449
J. Richard Wright

8*8 Enthusiasts
P.O. Box 28 Rhodes Rd.
Apalachin, NY 13732
Keith Merrill

VIC 20/Commodore 64 Users Group
31 Maple Dr.
Lindenhurst, NY 11757
(516) 957-1512
Pete Lobol

VIC Information Exchange Club
336 W. 23 St.
Deer Park, NY 11729
Tom Schlegel
SASE & phone please

New York Commodore Users Group
380 Riverside Dr., 7Q
New York, NY 10025
(212) 566-6250
Ben Tunkelang

Parsippany Computer Group
51 Ferncliff Rd.
Morris Plains, NJ 07950
(201) 267-5231
Bob Searing

Capitol Dist. Commodore 64/VIC Users Group
363 Hamilton St.
Albany, NY 12210
(518) 436-1190
Bill Pizer

Hudson Valley Commodore Club
1 Manor Dr.
Woodstock, NY 12498
F.S. Goh
1st Wednesday of month

LIVICS (Long Island VIC Society)
20 Spyglass Lane
East Setauket, NY 11733
(516) 751-7844
Lawrence Stefani

VIC Users Group
c/o Stoney Brook Learning Center
1424 Stoney Brook Rd.
Stoney Brook, NY 11790
(516) 751-1719
Robert Wurtzel

NORTH CAROLINA

Amateur Radio PET Users Group
P.O. Box 30694
Raleigh, NC 27622
Contact: Hank Roth

VIC Users Club
c/o David C. Fonenberry
Route 3, Box 351
Lincolnton, NC 28092

Microcomputer Users Club
Box 17142 Bethabara Sta.
Winston-Salem, NC 27116
Joel D. Brown

VIC Users Club
Rt. 11, Box 686
Hickory, NC 28601
Tim Gromlovits

OHIO

Dayton Area PET
User Group
933 Livingston Drive
Xenia, OH 45385
B. Worby, President
(513) 848-2065
J. Watson, Secretary
(513) 372-2052

Central Ohio PET
Users Group
107 S. Westmoor Avenue
Columbus, OH 43204
(614) 274-6451
Contact: Philip H. Lynch

Toledo PETS
734 Donna Drive
Temperance, MI 48182
Contact: Gerald Carter,
President

Chillicothe Commodore
Users Group
P.O. Box 211
Chillicothe, OH 45601
William A. Chaney

Licking County 64 Users Group
323 Schuler St.
Newark, OH 43055
(614) 345-1327

11433 Pearl Rd.
Strongsville, OH 44136
Paul M. Warner

OKLAHOMA

Southwest Oklahoma
Computer Club
P.O. Box 6646
Lawton, OK 73504
Garry Lee Crowell
1:30 1st Sunday at
Lawton City Library

Tulsa Area Commodore Users Group
Tulsa Computer Society
P.O. Box 15238
Tulsa, OK 74112
Annette Hinshaw

OREGON

NW PET Users Group
John F. Jones
2134 N.E. 45th Avenue
Portland, OR 97213

PENNSYLVANIA

PET User Group
Gene Beals
P.O. Box 371
Montgomeryville, PA 18936

Penn Conference
Computer Club
c/o Penn Conference of SDA
720 Museum Road
Reading, PA 19611
Contact: Dan R. Knepp

PACS PET Users Group
20th & Olney Streets
Philadelphia, PA 19120

Glen Schwartz
807 Avon
Philadelphia, PA 19116

Gene Planchak
4820 Anne Lane
Sharpsville, PA 15150
(412) 962-9682

PPG (Pittsburgh PET Group)
c/o Joel A. Casar, DMD
2015 Garrick Drive
Pittsburgh, PA 15235
(412) 371-2882

Westmoreland Commodore
Users Club
c/o DJ & Son Electronics
Colonial Plaza
Latrobe, PA 15650
Jim Mathers

COMPSTARS
440 Manatawny St.
Pottstown, PA 19464
Larry Shupinski, Jr.
Meet at Audio Video
Junction

Commodore Users Club
3021 Ben Venue Dr.
Greensburg, PA 15601
(412) 836-2224
Jim Mathers

VIC 20 Programmers, Inc.
c/o Watson Woods
115 Old Spring Rd.
Coatesville, PA 19320
Robert Gougher

G.R.C. User Club
300 Whitten Hollow Rd.
New Kensington, PA 15068
Bill Bolt

NADC Commodore Users Club
248 Oakdale Ave.
Horsham, PA 19044
Norman McCrary

CACC (Capitol Area Commodore Club)
134 College Hill Rd.
Enola, PA 17025
(717) 732-2123
Lewis Buttery
Union Deposit Mall at 7 p.m.

PUERTO RICO

CUG of Puerto Rico
RFD #1, Box 13
San Juan, PR 00914
Ken Burch

VIC 20 User Group
655 Hernandez St.
Miramar, PR 00907
Robert Morales, Jr.

RHODE ISLAND

Irving B. Silverman, CPA
160 Taunton Ave.
E. Providence, RI 02914
Contact: Michelle Chavanne

Newport VIC/64 Users
10 Maitland Ct.
Newport, RI 02840
(401) 849-2684
Dr. Matt McConeghy

SOUTH CAROLINA

Beauford Technical College
100 S. Ribaut Rd.
Beauford, SC 29902
Dean of Instruction

SOUTH DAKOTA

PET User Group
515 South Duff
Mitchell, SD 57301
(605) 996-8277
Contact: Jim Dallas

VIC/64 Users Club
203 E. Sioux Ave.
Pierre, SD 57501
(605) 224-4863
Larry Lundeen

TENNESSEE

River City Computer
Hobbyists
Memphis, TN
1st Mon. at Main Library

Nashville VIC Users Group
American Computer Store
1004 8th Ave., S.
Nashville, TN 37203
(615) 242-8592
Jane Maggard
1st Thurs. of month, 6 pm

Commodore User Club
Metro Computer Center
1800 Dayton Blvd.
Chattanooga, TN 37405
Mondays 7:30 pm

Metro-Knoxville 64 Users Club
7405 Oxmoor Rd., Rt. #20
Knoxville, TN 37921
(615) 938-3773
Ed Pritchard

TEXAS

SCOPE
1020 Summit Circle
Carrolton, TX 75006

PET Users
2001 Bryan Tower
Suite 3800
Dallas, TX 75201

Larry Williams
P.O. Box 652
San Antonio, TX 78293

PET User Group
John Bowen
Texas A & M
Microcomputer Club
Texas A & M, TX

CHUG (Commodore Houston
Users Group)
8738 Wildforest
Houston, TX 77088
(713) 999-3650
Contact: John Walker

Corpus Christi Commodores
3650 Topeka St.
Corpus Christi, TX 78411
(512) 852-7665
Bob McKelvy

Commodore Users Group
5326 Cameron Rd.
Austin, TX 78723
(512) 459-1220
Dr. Jerry D. Frazee

VIC Users Group
3817 64th Dr.
Lubbock, TX 79413

Southeast Houston VIC
Users Group
11423 Kirk Valley Dr.
Houston, TX 77089
(713) 481-6653

64 Users Group
2421 Midnight Circle
Plano, TX 75075
S.G. Grodin

UTAH

Utah PUG
Jack Fleck
2236 Washington Blvd.
Ogden, UT 84401

The Commodore Users
Club
742 Taylor Avenue
Ogden, UT 84404
Contact: Todd Woods Kap,
President
David J. Shreeve,
Vice President

The VIClic
799 Ponderosa Drive
Sandy, UT 84070
Contact: Steve Graham

VIC 20 Users
324 N. 300 W.
Smithfield, UT 84335
Dave DeCorso

Northern Utah VIC & 64
Users Group
P.O. Box 533
Garland, UT 84312
David Sanders

The Utah Commodore Users Group
652 West 700 North
Clearfield, UT 84015
(801) 776-3950
Rodney Keller or Richard Brenchly

VIRGINIA

Northern VA PET Users
Bob Karpen
2045 Eakins Court
Reston, VA 22091
(803) 860-9116

VIC Users Group
Rt. 2, Box 180
Lynchburg, VA 24501
Contact: Dick Rossignol

VIC Users Group
c/o Donnie L. Thompson
1502 Harvard Rd.
Richmond, VA 23226

Dale City Commodore
User Group
P.O. Box 2004
Dale City, VA 22193
(703) 680-2270
James Hogler

access: commodore user groups

Tidewater Commodore
Users Group
4917 Westgrove Rd.
Virginia Beach, VA 23455
Fred Monson

Fredericksburg Area
Computer Enthusiasts
P.O. Box 324
Locust Grove, VA 22508
(703) 972-7195
Michael Parker

Commonwealth 20/64
Users Group
1773 Wainwright Dr.
Reston, VA 22090
(703) 471-6325
Tal Carawan, Jr.

VIC 20 Victims
4301 Columbia Pike #410
Arlington, VA 22204
(703) 920-0513
Mike Spengel

Peninsula Commodore 64 Users Group
124 Burnham Place
Newport News, VA 23606
(804) 595-7315
Richard G. Wilmoth

WASHINGTON

NW PET Users Group
2565 Dexter N. 3203
Seattle, WA 98109
Contact: Richard Ball

PET Users Group
c/o Kenneth Tong
1800 Taylor Ave. N102
Seattle, WA 98102

VIC 20 Computer Club
947 N. Burroughs Ave.
Oak Harbor, WA 98277
Michael D. Clark

Central Washington
Commodore Users Group
1222 S. 1st St.
Yakima, WA 98902
Tim McElroy

Blue Mountain Commodore
Users Club
667 Canary Dr.
Walla Walla, WA 99362
(509) 525-5452
Keith Rodue

WEST VIRGINIA

Personal Computer Club
P.O. Box 1301
Charleston, WV 25325
Cam Cravens

WISCONSIN

Sewpus
c/o Theodore J. Polozynski
P.O. Box 21851
Milwaukee, WI 53221

Waukesha Area Commodore
User Group (WACUG)
256½ W. Broadway
Waukesha, WI 53186
Contact: Walter Sadler
(414) 547-9391

Commodore User Group
1130 Elm Grove St.
Elm Grove, WI 53122
Tony Hunter

Commodore 64 Software
Exchange Group
P.O. Box 224
Oregon, WI 53575
E. J. Rosenberg

C.L.U.B. 84
6156 Douglas Ave.
Caledonia, WI 53108
(414) 835-4645 pm
Jack White
2nd Sat every month 10:00 am

CANADA

Toronto PET
Users Group
381 Lawrence Ave. West
Toronto, Ontario, Canada
M5M 1B9
(416) 782-9252
Contact: Chris Bennett

PET Users Club
c/o Mr. Brown
Valley Heights Secondary School
Box 159
Langton, Ont. N0E 1G0

Vancouver PET Users Group
P.O. Box 91164
West Vancouver, British
Columbia
Canada V7V 3N6

CCCC (Canadian
Commodore Computer Club)
c/o Strictly Commodore
47 Coachwood Place
Calgary, Alberta, Canada
T3H 1E1
Contact: Roger Olanson

W.P.U.G.
9-300 Enniskillen Ave.
Winnipeg, Manitoba R2V 0H9
Larry Neufeld

VIC-TIMS
2-830 Helena St.
Trail, British Columbia
V1R 3X2
(604) 368-9970
Greg Goss

Arva Hackers
Medway High School
Arva, Ontario N0M 1C0
D. Lerch

Nova Scotia Commodore
Computer Users Group
66 Landrace Cres.
Dartmouth, N.S. B2W 2P9
Andrew Cornwall

Bonnyville VIC Cursors
Box 2100
Bonnyville, Alberta T0A 0L0
(403) 826-3992
Ed Wittchen

FINLAND

VIC-Club in Helsinki
c/o Matti Aarnio
Linnustajankj 2B7
SF-02940 ESP00 94
Finland

KOREA

Commodore Users Club
K.P.O. Box 1437
Seoul, Korea
Contact: S. K. Cha

MEXICO

Asociacion De Usarios
Commodore
c/o Alejandro Lopez
Arechiga
Holbein 174-6° Piso
Mexico 18, D.F.

Club de Usarios Commodore
Sigma del Norte
Mol del Valle, Local 44
Garza Garcia, N.L. 66220

NEW ZEALAND

Commodore Users Group
Meet at VHF Clubrooms
Hazel Ave.
Mount Roskill
3rd Wed. of month, 7:30 pm
Roger Altena 278-5262

Nelson VIC Users Group
c/o P.O. Box 860
Nelson, New Zealand
Peter Archer

E.R. Kennedy
c/o New Zealand Synthetic
Fuels Corp. Ltd.
Private Bag
New Plymouth

NORWAY

VIC Club of Norway
Nedre Bankegt 10,
1750 Halden
Norway

UNITED KINGDOM

North London Hobby
Computer Club
Dept. of Electronics &
Communications
Engineering
The Polytechnic of North
London
Holloway Rd.
London N7 8DB

Croydon Microcomputer Club
111 Selhurst R.
Selhurst, London SE25 6LH
01-653-3207
Vernon Gifford

User Bulletin Board

User Groups Forming:

HAWAII

Commodore User Group
of Honolulu
c/o PSH
824 Bannister St.
Honolulu, HI
(808) 848-2088
3rd Fri. Kaliho Library

MICHIGAN

VIC Commodore User Club
486 Michigan Ave.
Mariesville, MI 48040
(313) 364-6804
M. Gauthier

Classified

...AND EVEN MORE NEW GAME CARTRIDGES FOR THE COMMODORE 64

by Diane LeBold

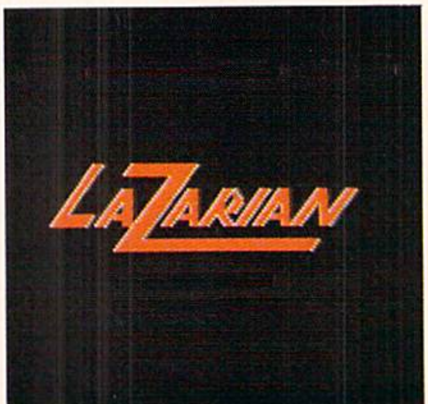
Two very hot Bally/Midway arcarde games and a brand new Commodore-designed race car game top the list of new releases for the Commodore 64.

One dreary morning not too long ago I was summoned to the locked (not to mention security- guarded) chambers where they hide our game programmers. I got my usual warm welcome —nobody even looked up when I walked in. (They're actually a great bunch of folks. Just—you know—*focused* on what they're doing.) After I bounced off the ceiling a few times, stood on my head and sang the complete Hallelujah Chorus in four-part harmony, somebody finally noticed I was there and things got a little more cordial. In fact, by lunchtime I walked out with the full scoop on all the great new games you'll be seeing soon for the Commodore 64.

LaZarian (for one player with joystick)

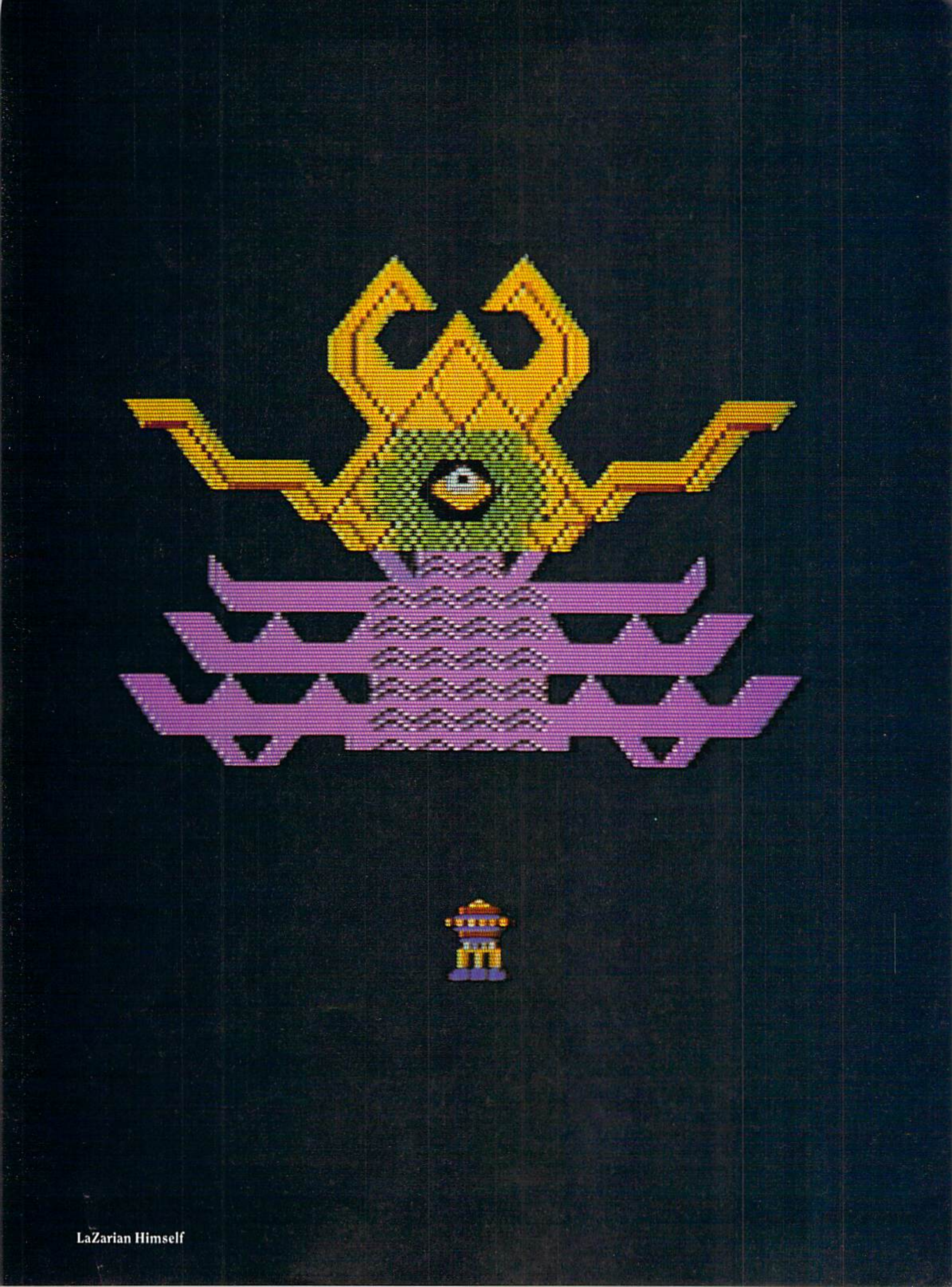
Those of you who have played this game in the arcades will not be disappointed with this Bally/Midway favorite. All the wild graphics (the bouncing eye is just as fearsome as it is in the arcades), great music, and exciting play action are right there—with a couple of added advantages. First, if you're a novice player and you get wiped out in the first level of play you can reset the game and continue from where you got killed instead of having to start all over. (Your score, however, is set back to zero.) Second, a "pause" feature lets you freeze the game, go off and eat dinner or answer the phone, then come back and pick up again exactly where you left off.

Those who aren't familiar with the game have a great experience in store. In the first round you have to shoot all the meteorites and rescue your lander (inspired to action by Beethoven's ninth symphony). Once you get through that round, you must navigate the "Tunnel of Fear," warding off aliens, killer fish and gun emplacements, to rescue your ship —this time heartened by the William Tell Overture.

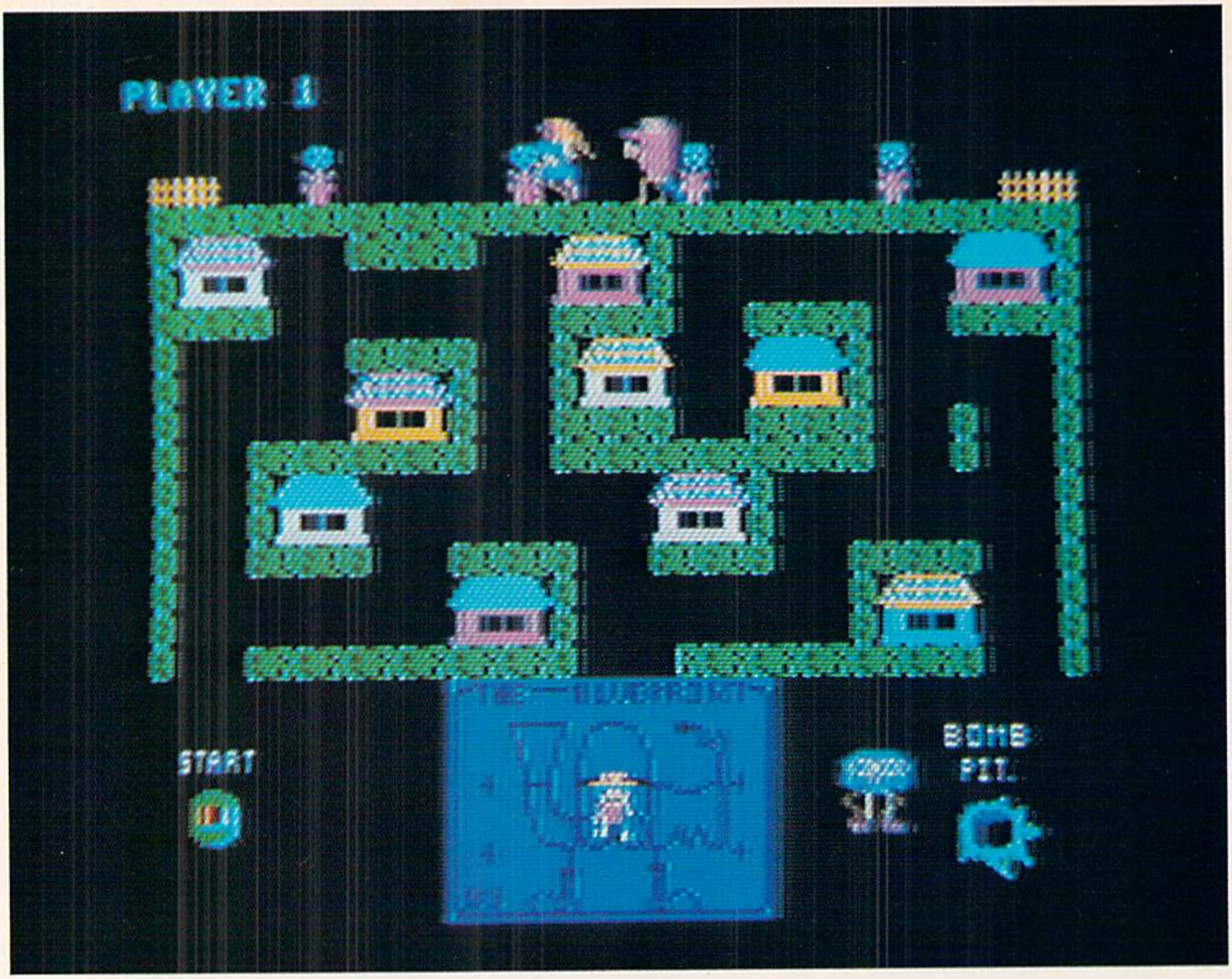


Then you get to the final round—LaZarian himself (or, rather itself, since the awesome creature is a kind of combination pyramid-octopus-cyclops with a single evil-looking eye in the middle of what might be the pyramid's forehead). If you manage to burrow through this monstrous structure, you free the eye, which then bounces around trying to get you. You've got to hit it four times before it will die. Then you move on to increasingly difficult levels of play, where meteors shoot more often, aliens and fish are more vicious and LaZarian is more dangerous.

The programmer who's been working on *LaZarian* says his high score is around 18,000 (you get a bonus ship at 14,000 points).



LaZarian Himself



Blueprint

Blueprint (for one player with joystick)

The second Bally/Midway hit you'll be seeing very soon is *Blueprint*—a very funny, not to mention complex and challenging—memory-maze-monster game. Once again, the graphics, sound effects and play action of the original arcade game have, in our programmer's words, "been copied to a 'T.'"

There's a lot going on in this wacky game, so it's a little hard to describe. I'll bet the first thing most people would notice, however, is the buxom beauty who keeps running across the top of the screen, pursued by a hairy ogre.

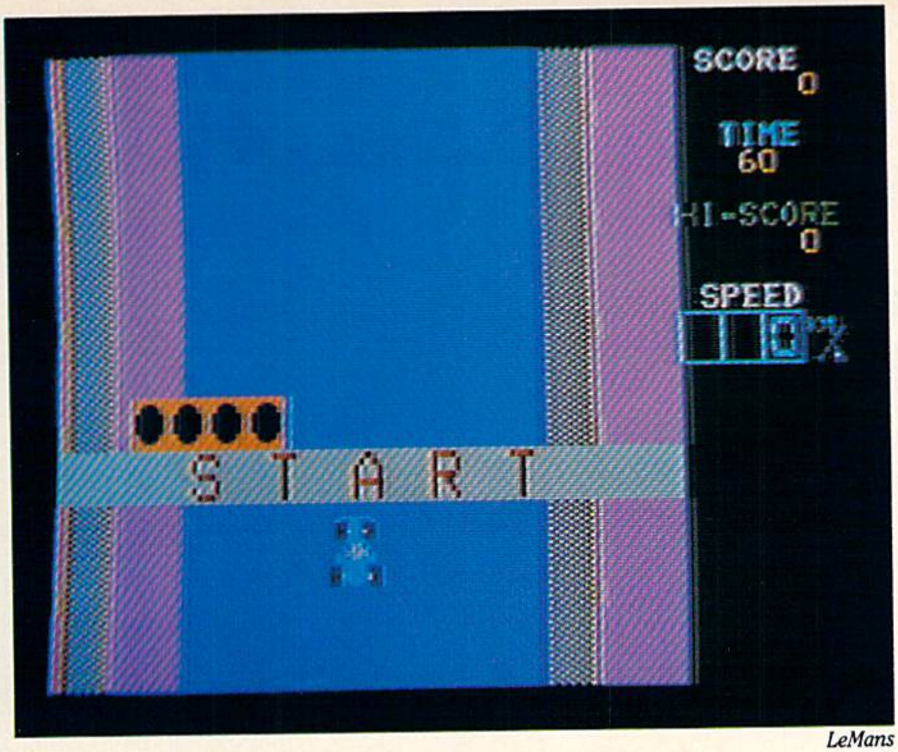
This unlikely couple actually composes the timing mechanism for what's going on in the rest of the game. Each time they race across the screen the ogre is a little closer to the distressed damsel. If he catches her, he throws her into a "swag bag" (you even see her squirming around inside) and the hero—who meanwhile has been frantically trying to build a machine to bonk the ogre with big beach balls—dies of a broken heart. So the trick is for the hero to get the machine finished before the ogre catches the lady.

But getting the machine finished is not so easy for the little guy—although he does somehow always seem to be smiling. He's got a blueprint of what he's supposed to build (would you believe the machine wears shoes?) and ten houses where pieces of the machine are hidden. When the game starts, eight of the houses contain the pieces he needs to build the machine and two contain—uh oh—bombs. His job is to go into a house and remove its contents. If it's a piece of the machine, he runs down and sticks it on the blueprint. But if it's a bomb, he has to run to the "bomb pit" and throw it in before it blows him to smithereens. To complicate his life, when he removes a piece of the machine from a house, the next time he goes into that house he'll find—a bomb. With a short fuse. So he should definitely avoid wearing out his welcome at any of the houses.

As if that weren't enough, the poor guy has to avoid killer flower pots the ogre keeps knocking off the ledge above; control a perverse monster named Willy, who keeps trying to jump on the machine's "start" button (the machine will fall apart if you start it before all the pieces are in place); and keep away from a hairy-faced "maze monster" who wants to eat him.

If by some miracle our hero does complete the machine, he can then use it to shoot at the lecherous old ogre and save the lovely lady. But once he konks the ogre, the whole thing starts all over again—only worse. The ogre runs faster this time, flower pots fall more often, bombs have shorter fuses . . . the poor guy just can't win. (Is this starting to sound like a version of your own life?)

Actually, he can win—sort of. According to the programmer who's been working on the game, things at least won't get any worse after about five or six levels. That programmer, by the way, says his high score on this version of *Blueprint* is about 50,000.



LeMans

LeMans (for one player with paddle)

This Commodore original, available right now, is a classic race car game that uses the paddle like a steering wheel and the fire button like a gas pedal. You have an aerial view of the track in all its high-res glory, with curves, ice, sudden two-way traffic and a very exciting (namely dark) tunnel, all of which come up unpredictably to challenge your reflexes. The interesting thing about this game is that if you get really good and don't mind taking your meals intravenously, you could go on playing forever, because there's almost no end to how long you can keep going.

The crucial factors in this game are the distance you travel, the number of cars you pass and the time it takes you to do it. You start out with 60 fairly slow "units" of time. (You can watch the countdown on your screen as you use up these precious units.) As you race along you get points for distance you cover and the number of cars you pass (every time you pass ten cars you get a 1000-point bonus).

If you manage to score at least 20,000 points before your 60 units of time are up, you get another 60 units, and can continue the race. Only this time around each unit is a little shorter—which means you have less real time to score the 20,000 points you need to get another 60-unit extension.

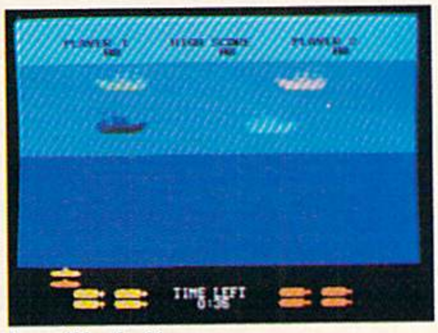
You'll be amazed at how realistically your car responds to the changing road conditions in this game. When you hit the ice, for instance, you don't have as much control, and the car slides instead of turning when you try to steer out of the skid. And the dark tunnel effects are so realistic it's almost like being there. You see only the taillights of the cars ahead, and then their silhouettes when they come into the range of your headlights. Actually, that tunnel is pretty spooky, now that I think of it.

If you crash you're forced to go to the pits, and if you ride the shoulder, even though it might prevent a crash, your car slows down. Either way you lose time—the deciding element in this game. So the obvious way to stay in the race and keep racking up points is to be a quick, alert and safe driver. Just like in the real world.

The scores in this game are, as you might guess, VERY high. The programmer who's responsible for this one wouldn't even hazard a guess on how high you might expect to go, although there probably is some kind of limit. We'll just have to wait and see how you all make out, now that we're shipping this one.

VIC 20 Conversions

In addition to these three exciting new cartridge games designed specifically for the 64, you'll soon be seeing (if you haven't already that is) 64 versions of some of our most popular VIC games. These releases include *Omega Race, GORF, Clowns* and *Sea Wolf,* all of which happen to be Bally/Midway games and all of which play just like the original arcade versions.



Seawolf for the 64

All four, naturally, boast colorful high-resolution graphics that are truer to the original arcade graphics than ever before. In the case of *Sea Wolf,* three stages of water make it even more challenging than the VIC version, according to an inside source.

As time goes on you can look forward to even more great new games for both the Commodore 64 and the VIC 20 emerging from those closely guarded halls where our game programmers lurk long into the night. Keep watching your dealer's shelves for new releases. **C**

OOPS

by Jeff Bruette

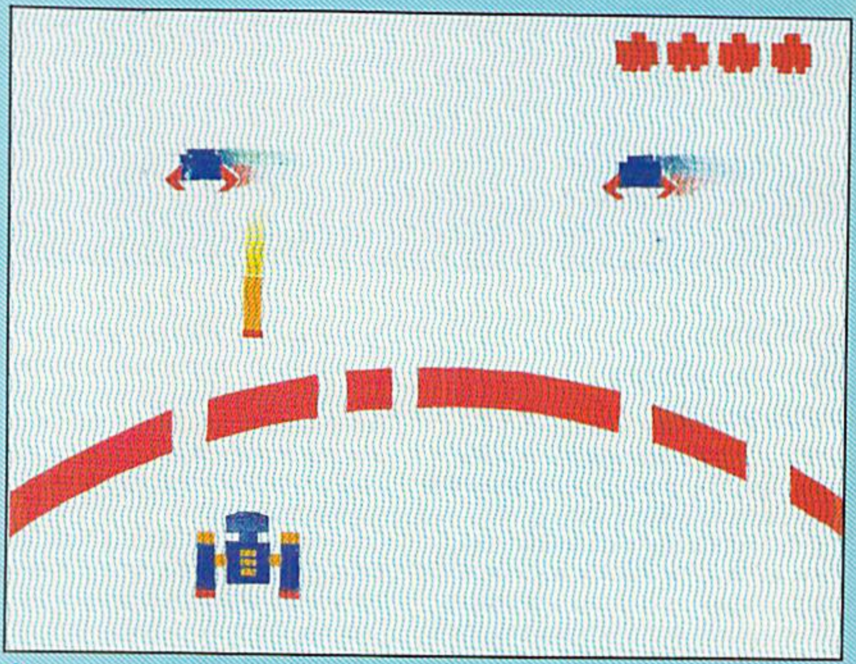
Score higher by taking advantage of a few programming "bugs" in the *Omega Race* and *GORF* cartridges for the VIC 20.

Evin programors make a misteak onse in a wile. With all our efforts to make a game play well and fit into a limited memory, we sometimes overlook a potential "bug." And sometimes these bugs work to the user's advantage. Let me tell you about a few of them in Commodore's *Omega Race* and *GORF* cartridges that will help you rack up higher scores than you may have been getting.

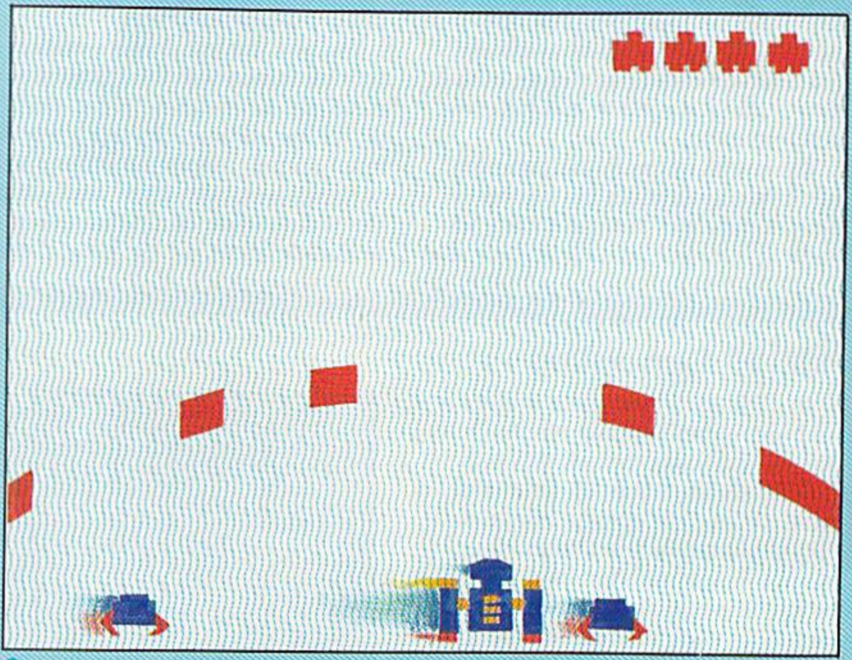
Possibly the most useful bugs are in the VIC 20 version of *Omega Race*. The first one I'm going to talk about is not really a bug in the true sense, so let's call it an "undocumented feature." Normally in this game you get three ships when you start. However, if you hold down the SHIFT key while pressing either F1 (for joystick) or F3 (for paddle) at the beginning of the game, you will get—count 'em—five ships! Let's see if that improves your score!

Another interesting quirk in *Omega Race* that you probably don't know about (although some of our readers have discovered this one by accident) is the ability to replenish your ships if you get down to one or two. Keep count of the number of screens you've cleared. After you clear the fourth screen, when the music starts to play, hold down the function key you would use to start the game. Continue to hold the key down until after the "droid force eliminated" message appears on the screen. (Sometimes just part of the message will appear, but that doesn't matter.) When the fifth screen appears you should now have three ships. This procedure sometimes works using shifted function keys, but the results in that case are inconsistent.

Keep in mind that if you have not lost any ships up to this point and have been awarded a bonus ship, it would be a disadvantage to use this method, because the ship count goes to three regardless of the number of ships you had. You can continue doing this every



A



C

fourth screen, just before the "droid force eliminated" message. Try it. I think this also might improve your score!

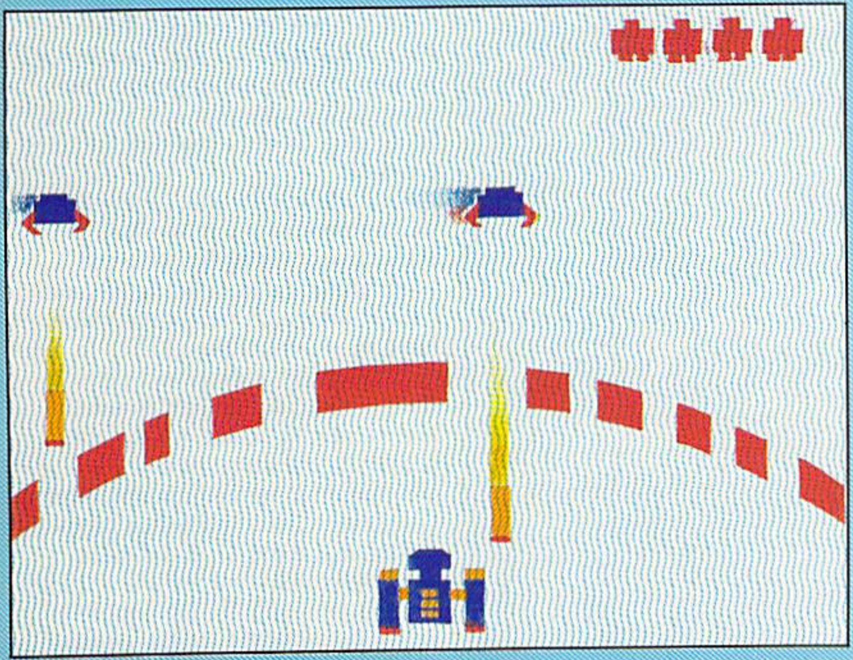
The next bug I'd like to tell you about is in *GORF* for the VIC 20. This one allows you to play in the "invaders" round for as long as you want, racking up points almost indefinitely. The procedure is rather involved, so I've included some pictures to help explain.

Here goes . . . Shoot all the invaders except the one in the upper left corner and the one in the upper right corner (Figure A). Then sit in the lower center of the screen to avoid the invaders and their missiles (Figure B). When the invaders are on the bottom row of the screen, move with them to get your ship out of the center of the screen. (Figure C).

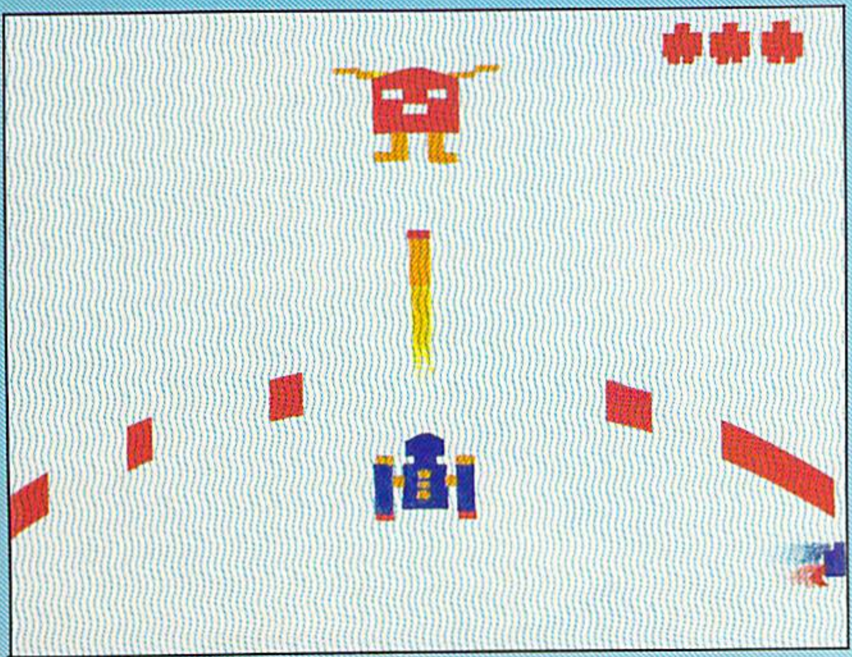
When one of the invaders drops off the screen your ship will be destroyed. Immediately push forward on the joystick and move the next ship to the center of the screen. An invader will keep running from one side of the screen and off the other side. This invader will never hit you if you stay as high on the screen as you can. Also, the invader will never drop off the bottom of the screen. You can now just sit there and shoot the GORF and two types of flying saucers for bonus points (Figure D). When you get bored you can go to the next round by crashing your ship into the remaining invader.

These are the only useful bugs I am aware of, right now, Hopefully they will help you get a higher score on *Omega Race* and *GORF*. But if you find anything else you think is strange when you're playing one of our games, please drop me a line either through the regular mail or through EMAIL on CompuServe (PPN 70001,1153). That way, when I hear something interesting I can pass it on to the rest of you.

Well, better get back to *Wizard of Wor* . . . C



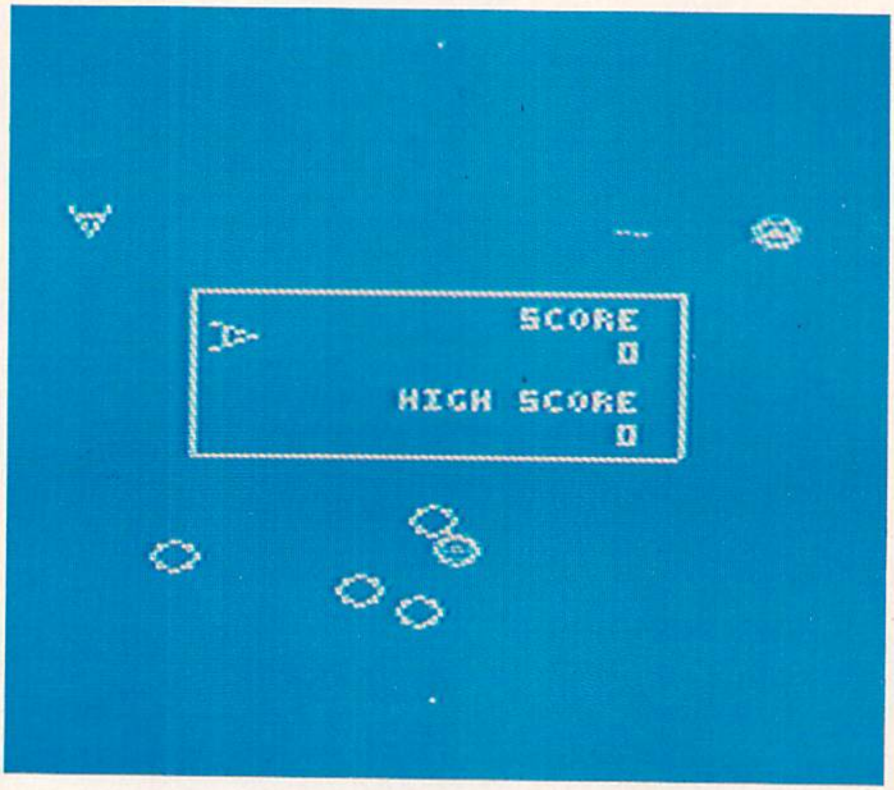
B



D

OMEGA RACE

In response to David Berezowski's article in the the notes from the champ of *Omega Race* who those of us who use joysticks—especially since are David's hints for "space jockeys" who prefer



An article in the Winter, 1982, *Power/Play* gives the reader some hints on how to score well at *Omega Race* using game paddles, but it is also possible to score as well or even higher using a joystick.

First you need to learn to use the joystick to make your ship go where you want it to go. Unlike the game paddle, which allows you to make a quick and controlled rotation, the joystick makes a rapid rotation that takes practice to control. If you also apply forward thrust while turning you are likely to careen wildly about the screen until you encounter a destructive object.

It is wisest to apply thrust and rotation in separate actions to maintain good control and, above all, do not panic! Controlled rotation can be had by gently nudging or tapping the joystick left or right. Controlling the rotation so you can get the ship pointed in the direction you want is the hardest part of mastering joystick control, but once the joystick is mastered the droid forces will be at your mercy.

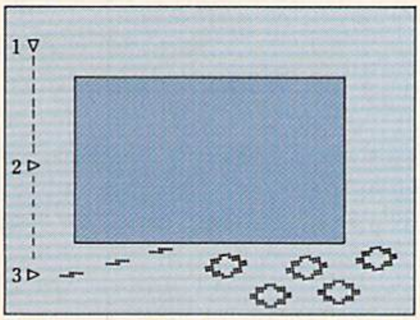
by David Owens

A JOYSTICK WORKS, TOO!

Winter, 1982, *Power/Play*, David Owens writes: "When I read only used game paddles I decided someone should defend my highest score, using five ships, was 244,100." So here sticks to paddles.

Thrust-Rotate-Fire

After you've spent a few games careening around the screen towards obliteration, you are ready for a more organized approach to the game. The most successful strategy to help a beginner clear their first screen is as follows:



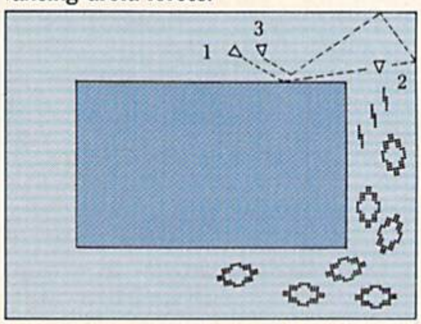
1. As soon as the game starts, thrust forward. This heads you straight down.
2. Rotate the ship until it is exactly horizontal, thus making it possible to take aim at a maximum of ships.
3. Fire on the ships. You should be able to clear the screen in a few passes up and down.

Remember that you can only fire four times in rapid succession before you need a brief intermission to reload. Spreading out your shots to hit several ships is better than shooting your entire load at once, possibly striking only one ship with all four shots. Also note that missiles are released when you lift the button, a crucial point for good timing.

Using the thrust-rotate-fire technique you can clear many screens, but you'll notice that most ships are struck as they recede. Thus few hits are made on higher scoring command ships and death ships. Another technique is needed.

The Head-on Approach

The key to staying in the game with a joystick is the same as with paddles: get the command ships to fire harmlessly into the side of the screen, then move out and destroy them. But rather than traveling exactly on a horizontal line, try traveling in a repeating diagonal pattern, which allows more firing time on the advancing droid forces.



1. Thrust at an angle to make a diagonal pass across the droid forces.
2. Rotate and fire.
3. As ship returns to original position, rotate to diagonal and repeat cycle until all droids are destroyed.

By using these two basic techniques separately or jointly you can find success. Half the fun is developing your own variations that work for different situations.

The Kamikaze Secret

One would think that destroying as many droid forces as rapidly as possible using the fewest ships would be the best way to score well. But there is a little known secret: Sometimes it's better to lose ships for a better score!

After clearing four screens you advance to the next droid force level, more difficult than the last. It makes sense to score as many points as possible before moving to the next level, where you are likely to lose your ships quicker and score less. But try this, instead. Clear three screens, then sacrifice your next ships at the last second until you have only one ship left. Use it to clear the fourth screen and advance to the next level.

But how can you continue with only one ship? Here comes the secret. If you hold down the fire button just after you clear the fourth screen and keep it down until the next level appears, you will get two more ships while maintaining your score! But be forewarned! If you have six ships and hold down the fire button after you clear the fourth screen your force will be depleted to three ships!

Using your ships to best advantage you should score over 100,000 against the first force and over 200,000 against the second force. Note that survival against the second force is most dependent upon destroying the first death ship, which comes almost immediately. Use the first technique described here to get quickly to the bottom of the screen and fire on it. Diagonal bursts often strike death ships better than horizontal or vertical shots, because they do not leave as many holes for them to sneak through. Note that survival against the third force requires impeccable technique and lots of luck!

Ready? Space jockeys, man your ships! Take hold of your joystick and show those Omegan paddle warriors how space was meant to be conquered! **C**

PART 3

The Commodore 64 Piano

Finally, you're ready to make some music with the Sound Interface Device (SID) in your Commodore 64! (This will undoubtedly be a great relief to those who have been following this series from the start.) Beginners may simply want to type in the program listing, following Paul's instructions to turn their 64 into a piano keyboard that produces a real piano sound. However, the more intrepid among you can go through Paul's line-by-line explanation—you'll learn a great deal, regardless of your computer experience, if you do. (Parts 1 and 2 appeared in the Winter, 1982 and Spring, 1983, issues of *Power/Play.)*

Yes, folks, it's that time again. Hopefully by now you've begun to see the various interactions among envelopes, frequencies and waveforms. These are our basic tools for music (or, indeed, any sound) synthesis on the Commodore 64.

To recap, the program in our last installment calculated a frequency array for seven octaves of notes. We will use that program as a base for this article, as we create a program that will allow us to play the keyboard like a piano. If you missed that issue, you'll find the original program on page 45.

Before we begin adding lines to the program, you will need to delete all the lines from 450 onward (simply enter the line numbers by themselves and press RETURN). If you remember, these lines let us hear the frequency table being created each time we ran the program, and we certainly don't want that to happen now.

by Paul Higginbottom

MAKING FRIENDS

SI

To allow us to play the keyboard, our program must first see if a key is pressed. If it is, the program must evaluate which note that key corresponds to and gate a voice on with that particular frequency. To sound piano-like, when a second key is hit (after the first is released) the first sound should still be able to be heard. In order to do that, our program must "cycle" through the voices. To put this another way, the first key pressed will play the first voice, the second key will play the second voice, the third the third voice—and the fourth will go back to the first voice. A statement such as:

VOICE=VOICE + 1: IF VOICE>2 THEN VOICE = 0

will cycle the voice counter. The IF statement ensures the variable VOICE doesn't go out of the range of 0 to 2.

Load the program from our last installment (refer to page ??), delete lines 450 onward and enter the following:

WITH

D

```
500 SID=54272
510 VOICE=0:OCT=4:WAVE=32
520 VM=7:HI=256
530 FORI=0TO23:POKESID+I,0:NEXT
540 POKESID+24,15
550 READA,D,S,R,PW
560 FORI=0TO2:INDEX=SID+I*VM
570 POKE INDEX+5,A*16+D
580 POKEINDEX+6,S*16+R
585 POKEINDEX+2,PWAND255
590 NEXT
600 DATA5,11,10,0,800
686 POKEINDEX+3,PW/HI
700 K$="Q2W3ER5T6Y7UI9O0P@-*£↑"
710 DIMK(255)
720 FORI=1TOLEN(K$)
730 K(ASC(MID$(K$,I,1)))=I
740 NEXT
750 PRINT:PRINT" 2 3   5 6 7   9 0   - £"
760 PRINT"Q W E R T Y U I O P @ * ↑"
800 GETA$:IFA$=""GOTO800
810 KEY=K(ASC(A$))-1:IFKEY<0GOTO800
820 IFKEY>11THENFRQ=F(OCT+1,KEY-12):GOTO835
830 FRQ=F(OCT,KEY)
835 FH=INT(FRQ/HI):FL=FRQ-FH*HI
840 INDEX=SID+VOICE*VM
850 POKEINDEX,FL:POKEINDEX+1,FH
860 POKEINDEX+4,WAVE:POKEINDEX+4,WAVE+1
870 VOICE=VOICE+1:IFVOICE>2THENVOICE=0
880 GOTO800
```

Line-by-line explanation of the program:

500 Assigns the address of the SID chip to the variable "SID."

510 Initializes variables "VOICE" (current voice being played), "OCT" (current octave the keyboard notes begin at) and "WAVE" (the current waveform to be POKEd into the control register of a voice).

520 Sets two constants. VM (Voice Multiplier) is set to 7 because there are 6 bytes to control each voice and the memory address of the start of each voice is computed by VOICE*VM. HI is set to 256, which is the high order value divisor. See Part 1 in this series for an explanation of how certain values are represented by two values—"lots" of 256 plus a remainder.

530 This FOR . . . NEXT loop initializes the SID chip register to zero to ensure that no previously run program will effect this one.

540 Sets SID register 24 to 15. 24 is the volume register, and we are setting the volume to 15, which is the maximum value.

SID

550 Reads in the envelope and pulse width parameters we wish to test from the DATA statement at line 600. The variables A, D, S and R represent the four parts of the envelope: Attack, Decay, Sustain and Release. PW is Pulse Width.

560 Start of a FOR . . . NEXT loop that will set the three voices in the SID to the parameters just READ. INDEX is set to the start address of the block of seven bytes for the current voice.

570 Puts the attack and decay values into register 5 of the current voice.

580 Puts the sustain and release values into register 6 of the current voice.

585 Puts the low value of the pulse width into register 2 of the current voice.

586 Puts the high value of the pulse width into register 3 of the current voice.

590 Continues looping until done with the NEXT command.

600 DATA to be tested. This consists of Attack, Decay, Sustain, Release and Pulse Width (if that waveform is used).

700 K$ is set to hold all the keys that can be played. You should note that these keys represent a piano layout on the keyboard.

710 Dimensions the K< array. This will be an array of the key positions subscripted by the ASCII value of the keypress. This is necessary because we need to know the key position along the piano layout to calculate the frequency that should be used.

720 Start of a FOR . . . NEXT loop to assign the K< array. The loop goes from 1 to the number of characters in K$, which is calculated using the LEN< function.

730 This gives the ASCII value of the i'th character in K$.

740 Continues the loop.

750-760 Prints the keyboard on the screen.

800 Main loop. Checks the keyboard. If nothing is pressed, it goes back to this line again and continues checking.

810 Assigns KEY to the position of the keypress on the keyboard minus one. Since elements of the array for keys not included in the "piano" will be equal to zero, this expression will equal minus one if an invalid key is pressed. The IF statement checks for invalid keys by seeing if KEY<0. If it is, the program goes back to the keyboard checking line again.

820 There are 12 notes in an octave. Since there are more than 12 keys in our piano layout, the program must check for this, and if the key position is greater than 11, it must subtract 12 from the key and add 1 to the octave. Instead of *actually* adding 1 to the octave and subtracting 12 from the key, this line evalues the frequency right there with the subscripts OCT+1 and KEY−12 and skips the next line, which would calculate the frequency normally.

830 Gets frequency from the array.

835 Evaluates the low and high values of the given frequency in variables FL and FH.

840 INDEX is set to the start of the voice block of registers.

850 The low and high frequency values are now POKEd into the SID. INDEX is equal to the position of the low byte of the frequency and INDEX+1 to the high byte.

860 Because we cannot detect the duration of the keypress, the voice is gated off and then on, so the A-D-S cycle is executed when a key is depressed. The voice is gated off first, so the release will have been completed before we gate the voice on. To release, register 4 (the control register) is POKEd with WAVE (the value of the current waveform) which is either 16, 32 or 64 (or 128 for noise, but this isn't particularly musical.) To start the cycle it is POKEd with WAVE+1.

870 This cycles the voice, as was explained earlier, so that each successive keypress uses the next available voice. This gives the user the ability to more or less play chords by playing keys in rapid succession.

880 This loops the program back to the start of the main loop to check the keyboard again.

Read the description of the program

carefully, until you understand how it works. Check for any mistakes and SAVE the program *before* entering the RUN command, because if you've made a mistake you might cause an unrecoverable crash and have to do all that typing again!

When you've SAVEd the program RUN it and you should (famous last words), after a second or two, see the layout of the keys to play appear on the screen. You can then play the keys and you'll hear the envelope that the program is using.

OK, maestro, the playing's over; it's back to work. Now it's time to experiment! The key line to change is the DATA statement in line 600. Press the STOP key and enter:

600 DATA 3,9,0,0,800

This will make the attack and decay shorter than before. You may note we have a sustain level of zero, which means the decay will fade to NO volume (or very close).

RUN the program again and hear the difference. Experiment with other numbers. For example, if you have the sustain number (the third one) greater than zero, the notes will stay on until the VOICE variable has cycled around to the same voice again and it is changed to a new note.

The VOICE cycling line (870) could also be changed so the piano becomes monophonic (one voice) by changing VOICE>2 to VOICE>0. Try that. Also, the octave and waveform could be changed. Try entering:

510 VOICE=0:OCT=2:WAVE=16

This will produce notes two octaves lower than before, and is now using the "mellow" pulse wave.

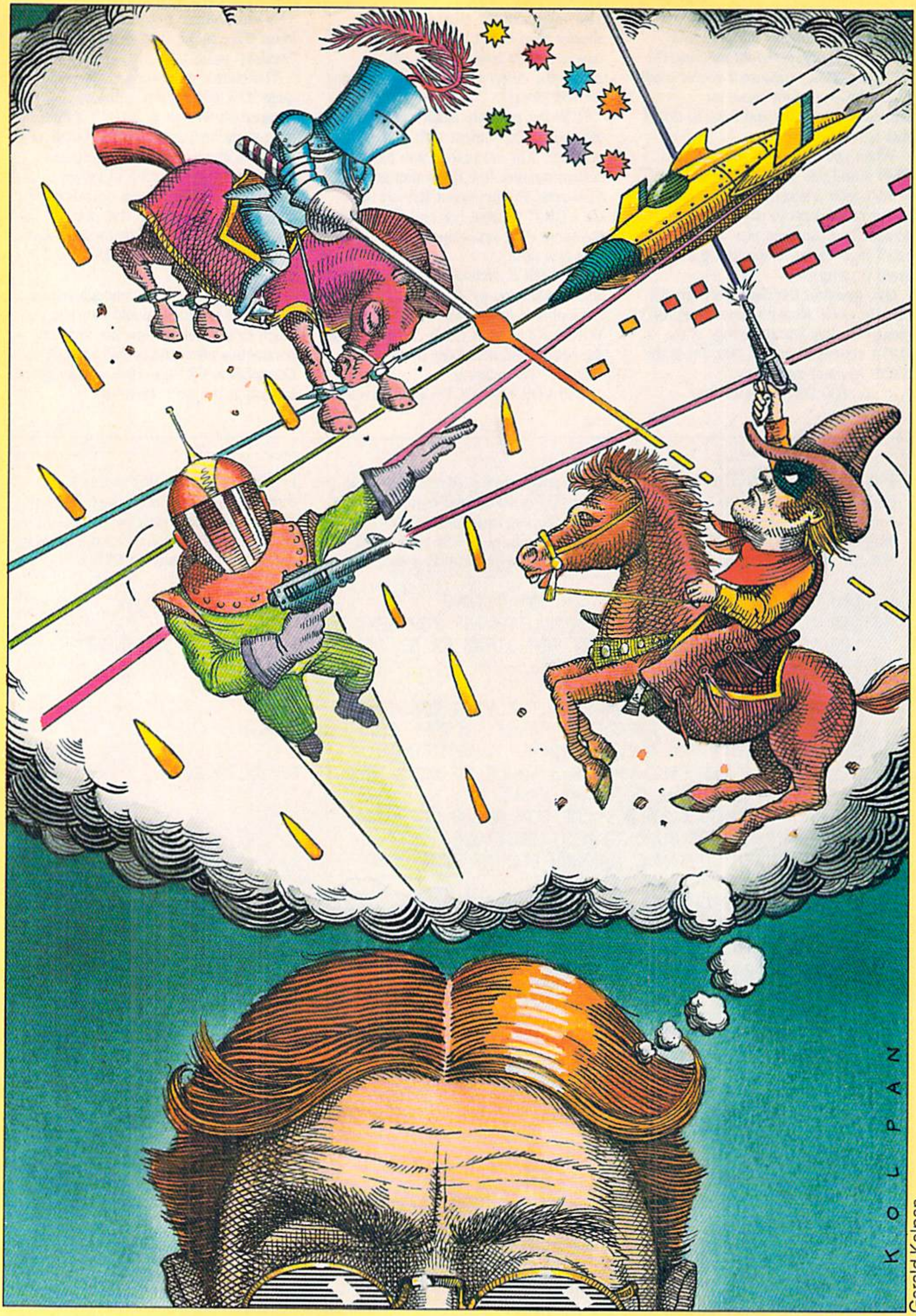
This simple program actually provides quite a bit of flexibility, although, because it's written in BASIC, it does have some limitations. For instance, no real-time special effects can be performed and the duration of keypress cannot be checked. Look out for a cartridge coming out soon that does all this and more! Also, a real synthesizer keyboard add-on with more SID chips is coming soon!

I would expect this is enough for this time. This program should give you a much better feel for how the various parameters affect the sound on your Commodore 64. Next time we can go on to look at the filter. Have fun! C

Program from Part 2

For those who missed the last issue of *Power/Play*, this is the (corrected) base program from Part 2 of "Making Friends with SID." You need it to start off on your programming adventure here in Part 3. For those who tried entering it from the last issue, please note the corrections in lines 110 (the ↑ was left out) and 485 (the first POKE is supposed to be s+6). Also note that in that issue there was an error in the text on page 43, second full paragraph. The last lines in that paragraph should read "the ratio between semitones is 2↑(1/12):1".

```
100 FR=3520 :REM NOTE 'A' IN TOP OCTAVE
110 CO=2↑(1/12):REM CONSTANTMULTIPLIER FOR NEXT SEMITONE
120 FORI=1TO9:FR=FR/CO:NEXT:REM START FR AT 'C' BY GOING BACK 9 SEMITONES
130 SS=16777216:REM SID CLOCK
140 CS=1022730:REM CPU CLOCK
150 FC=SS/CS:REM FREQUENCY MULTIPLYING CONSTANT
200 DIMF(7,11):REM FREQUENCY ARRAY (OCTAVE,SEMITONE)
300 FORI=0TO11:REM CYCLE THROUGH 12 SEMITONES
310 S=FR*FC:REM CALCULATE SID VALUE OF SEMI TONE IN TOP OCTAVE
400 FORJ=7TO0STEP-1:F(J,I)=S:S=S/2
410 NEXT:REM CALCULATE VALUE FOR ALL 8 OCTAVES
420 FR=FR*CO:REM GO ONTO NEXT SEMITONE
430 NEXT:REM CONTINUE THROUGH ALL 12 SEMITONES
450 REM
460 REM PRINT OUT ALL THE FREQUENCIES
470 S=54272:REM START ADDRESS OF SID CHIP
475 FORI=0TO24:POKES+1,0:NEXT:REM INITIALIZE SID CHIP
480 POKES+24,15:REM SET VOLUME
485 POKES+6,11:REM ATTACK=0:DECAY=0:SUSTAIN=0:RELEASE=11
500 PRINT"FREQUENCY TABLE"
510 PRINT"---------------"
520 PRINT"OCT SEM FREQUENCY"
600 FORI=0TO7
610 FORJ=0TO11
620 PRINTI;TAB(4);J,INT(F(I,J))
624 POKES+4,32:REM GATE OFF THE VOICE FIRST
625 H=INT(F(I,J)/256):REM CALCULATE HIGH BYTE OF FREQUENCY
626 L=F(I,J)-H*256:REM CALCULATE LOW BYTE
627 POKES,L:POKES+1,H:REM PUT IN FREQUENCY
628 POKES+4,33:REM NOW GATE IT ON
629 FORK=1TO100:NEXT:REM WAIT A BIT
630 NEXTJ,I
```



Gerald Kolpan

BIT-MAPPED GRAPHICS on the Commodore 64

by Jim Gracely

One of the most exciting, but sometimes frustrating features of the Commodore 64 is bit-mapped (or high resolution) graphics. The idea of individually controlling 64,000 dots on the screen just cries out to be played with. Visions of cartoons and wild animation dance in our heads. However, once the subject is tackled, many people turn away in frustration. After all, they say, there are all kinds of POKEs, PEEKs and special rules that even the *Commodore 64 Programmer's Reference Guide* doesn't clear up. They can't seem to find the hidden key that makes the screen burst into designs. Well, it's not quite that simple. In this article we'll look at one way to approach bit-mapped graphics, and hopefully remove some of the frustration. Along the way we'll present some utility programs and some high resolution graphics.

Starting Out

The first step in working with bit-mapped graphics is to get a feel for how it works. It's not necessary to completely understand how it works to use it! The *Programmer's Reference Guide* provides a good start on explaining how it works (pages 121 through 127), and if you sit down with your 64 and some graph paper you would eventually figure out all the details. But we're not going to do that. There are just too many exciting things that can be done with BMG (Bit-Mapped Graphics) to bother using up a couple of pages explaining it in detail. So, here's just a brief explanation.

The screen that the 64 normally displays consists of 1000 character blocks. This means it takes 1000 POKEs to fill the whole screen with any character. Even the period at the end of a sentence uses up one whole character block. There's no way to get any more characters on the screen than this . . . unless . . . we switch to BMG mode.

There is one bit in register 53265 that tells the 64 if it is in BMG mode. By setting this bit to a 1, we've entered the world of BMG. Bit 5 in register 53265 is the magical bit and using POKE53265,PEEK(53265)OR32 will set this bit to a 1.

I know what you're saying: fine, wonderful, so now what's going on? How come my screen just went KAPLOOOEE when I typed that? The screen that the 64 displays in BMG code is NOT made up of 1000 character blocks. Instead, the screen is displaying 8000 bytes of program memory! All of the normal character set is inaccessible. All we can do now is turn bits on and off. The 64 automatically puts the beginning of the 8000 byte screen at 4096, but that's the same place BASIC starts! So, to keep things as simple as possible, we can move this starting position to 8192 by using POKE53272,PEEK(53272)OR8. This allows us 8K of BASIC for our programs before we start writing onto our screen. For those of you who are compiling huge programs using BMG, the trick is to move the beginning of BASIC.

Let's review. After entering the two POKEs mentioned above, we are in BMG mode. The first location on the screen is byte 8192 and the last is 8192+7999 or 16191. Now we're going to change the screen into graph paper. POOF! At the upper left corner, X=0 and Y=0 and at the bottom right corner X=319 and Y=199. The X values are 0 to 319 across the top of the screen, and the Y values are 0 to 199 along the side of the screen. We really haven't done anything other than to change our point of view. (Get out your calculator: does 8000 bytes * 8bits/byte = 320* 200 bits?) We have come to the reason BMG seems so hard: how do we put a dot (one bit on) in the middle of the screen (X=160, Y=100)?

Still thinking about that one? You can stop now. The way to put a dot at X=160,Y=100 is to use the formulas given in the *Programmer's Reference Guide!* That's that. We can now create what I call the base graph program (BGP if you're into mneumonics):

```
1 POKE53265,PEEK(53265)OR32
2 POKE53272,PEEK(53272)OR8
3 FORX=8192TO16191:POKEX,
  0:NEXT
4 FORI=1024TO2023:POKEI,
  3:NEXT
190 GOTO240
200 CH=INT(X/8):RO=INT(Y/8)
    :LN=YAND7
210 BY=8192+RO*320+8*CH+LN
    :BI=7-(XAND7)
220 POKEBY,PEEK(BY)OR(2↑BI)
230 RETURN
240 POKE1024,16
250 GETA$:IFA$<>" "THEN250
260 POKE53265,PEEK(53265)
    AND223
270 POKE53272,PEEK(53272)
    AND247
275 CLR:PRINT"♥":END
```

Line 1 puts us into BMG mode

Line 2 puts the beginning of the screen at 8192

Line 3 clears the bit map by POKing zeros to all 8000 memory locations.

Line 4 sets the colors to cyan and black (more about colors later)

Lines 200-270 plot any X,Y point onto the screen (pages 125 and 126 of the *Programmer's Reference Guide)*

This program is the base of all the following graphic routines. Enter this program and save it. When you write a graphics routine, number the lines from 10 to 180. Every time your routine generates an X,Y point, use a GOSUB 200 to plot it. Save your program. To combine the two programs, use this trick: load the base program and list it. Load your program. Before you list your program, move the cursor to line number 1 of the base program and hit RETURN. Continue hitting RETURN on every line including line 275. Now list your program, and the base program is part of it! You can save your program again, and this time it includes the base program. Using this method will save you a lot of time and mistakes, and you don't even have to think about what the base program is doing.

Now let's get into some GRAPHICS! Keep in mind that these routines need the base program added to them, and watch the various ways that the X,Y points are generated. You're going to notice that BMG is not very fast. That is one of the limitations of BASIC.

Lines

Look at the following program:

```
10 Y=100
20 FORX=0TO319
30 GOSUB200
40 NEXT
```

The value of Y remains equal to 100 (halfway down screen) and X increases from 0 to 319. Each time X increases, a GOSUB 200 is performed. Enter this program and add the base program to it as described above. Now run it. A line! It's not very fast, but it's a straight line and it's in *HIGH RESOLUTION!*

Add the following lines to the program and run it again:

```
50 X=160
60 FORY=0TO199
70 GOSUB200
80 NEXT
```

A vertical line, too! Things are really getting exciting now!! To get lines that aren't just horizontal or vertical takes a couple of extra lines of programming. Save the previous program if you want, and enter this one:

```
10 X1=0:Y1=0:X2=319:Y2=199
20 DX=X2-X1:DY=Y2-Y1:X=X1
   :Y=Y1:L=SQR(DX*DX+DY*DY)
30 XI=DX/L:YI=DY/L
40 FORZ=1TOL
50 GOSUB200
60 X=X+XI:Y=Y+YI
70 NEXT
```

Add the base program and run it. This program uses some basic geometry to draw a line from point X1,Y1 to X2,Y2. Let's take a quick look at it.

Line 20 finds the distance between X1,Y1 and X2,Y2 using a distance formula (an interesting side note is that BASIC takes about 20 times longer to perform DX.2 than DX*DX!).

Line 30 uses DX and L to compute the X increment (XI) or the amount that X changes per block. The same thing is also done for the Y number.

Lines 40-70 set up a FOR-NEXT loop from 1 to L. Each time through the loop, XI is added to X1 (X) and YI is added to Y1 (Y). By computing XI and YI in line 30, X will equal X2 and Y will equal Y2 when the loop is done.

This line program is another good utility program to use inside of other programs. Once we find some points, we draw lines between them!

Colors

Now let's take a short break from geometry and talk about colors. Locations 53280 and 53281 won't change the screen colors when in BMG mode, and the screen color locations (55296 through 56295) aren't used for anything. This is because of the way memory is shuffled around in BMG. However the screen colors can still be changed. What used to be the screen memory locations (1024-2023) are now the color memory locations for the screen! To change the color of the whole screen or any part of it you have to POKE the colors you want. To make life harder still, the number you have to POKE is a little different than usual.

The locations 1024-2023 are the memory locations for both the background color and the character colors. The eight bits in each location are split up into four upper bits and four lower bits. The upper bits are used for the character (or dot) color and the lower bits the background color. Page 61 of the *Programmer's Reference Guide* has the POKE numbers for each of the colors available. To get the combination you want, multiply your character color number by sixteen and add it to your background color number. The result is the number to POKE.

An example to the rescue. Let's use white characters and a black background. The chart on page 61 says that white is the number 1 and black is 0, so the number we want to POKE is 1*16+0 or 16. To set this for the whole screen change line 4 of the base program to:

FOR I = 1024 TO 2023 : POKE I,16 : NEXT

Circles are the last geometric plot we are going to consider. They are good for making some designs but I have found them most useful for defining points to draw lines between. There are two examples of that later. For now, look at the following program:

```
10 R=10:H=160:K=100
20 FORI=1TO360
30 T=I*π/180
40 X=R*COS(T)+H
50 Y=R*SIN(T)+K
60 GOSUB200
70 NEXT
```

This routine will plot a circle with a radius of R and with its center at X=H and Y=K. The circle is defined by rotating one point around the center.

Lines 40 and 50 compute the X and Y values of this point as it is rotated through 360 degrees. In this example the radius is 10 and the center is at 160,100 (the center of the screen). The program is simple but the concept is a little harder, so we'll just take a look at what we've done.

Lines 20-30 set up a FOR-NEXT loop from 1 to 360 (the number of degrees in a circle). Line 30 changes degrees to radians (which BASIC uses).

Lines 40-50 compute what X and Y will be as the angle increases from 1 to 360.

Lines 60-70 jump to our plot routine and keep the loop going.

This program is also slow. After all, it's plotting 360 points! One quick and dirty way to speed it up is to add a STEP to the loop. If line 20 were:

FOR I = 1 TO 360 STEP 3

then only 120 points would be plotted (360/3=120) and the program would be three times as fast. Of course you can use any step, but the bigger the step, the dottier the circle will look. Just think, if you said STEP 120 it would only plot 3 points which would certainly not look much like a circle!

All right, let's combine some of these plots and ideas to create some designs. The first listing is called GEOMETRIC and it needs the base graph added to it. This program uses part of the circle routine to define some points and then uses the line routine to draw lines between the points.

```
10 N=0:R=100:H=160:K=100
   :DIMX(20),Y(20)
20 FORI=0TO330STEP30
25 T=I*π/180
30 N=N+1
35 X(N)=COS(T)*R+H
40 Y(N)=SIN(T)*R+K
45 NEXT
50 FORT=1TON-1
60 FORQ=T+1TON
70 X1=X(T):Y1=Y(T)
80 X2=X(Q):Y2=Y(Q)
90 DX=X2-X1:DY=Y2-Y1:X=X1:
   Y=Y1:L=SQR(DX*DX+DY*DY)
95 XI=DX/L:YI=DY/L
100 FORZ=1TOL
110 GOSUB200
120 X=X+XI:Y=Y+YI
130 NEXT:NEXT:NEXT
```

This program first defines 12 points of a circle and stores them in X(N) and Y(N) arrays. It then sets up two FOR-NEXT loops. The outer loop sets the first point of the circle to X1,Y1. The inner loop then sets the next point to X2,Y2 and draws a line to it. In the same way, the inner loop draws a line to each point higher in the array than X1,Y1. Also in the same way, the outer loop sets each point of the array to X1,Y1.

Lines 20-40 define 12 points of a circle (360/30=12). Each point is placed into the X(N) and Y(N) array. The N pointer is incremented after each loop.

Lines 50-60 set up the outer FOR-NEXT loop and define X1 and Y1. The values for each X1,Y1 set are taken from the array elements 1 through N-1.

Lines 70-80 set up the inner FOR-NEXT loop and define X2 and Y2. This loop is bounded from one greater than the outer loop (T+1) to N−1.

Lines 90-130 are the line drawing routine presented earlier.

The result of all this work is an interesting blend of lines and curves. By the way, don't expect this program to complete in a couple of minutes. It has to compute and draw 66 lines, and you remember how long it took to draw one line!

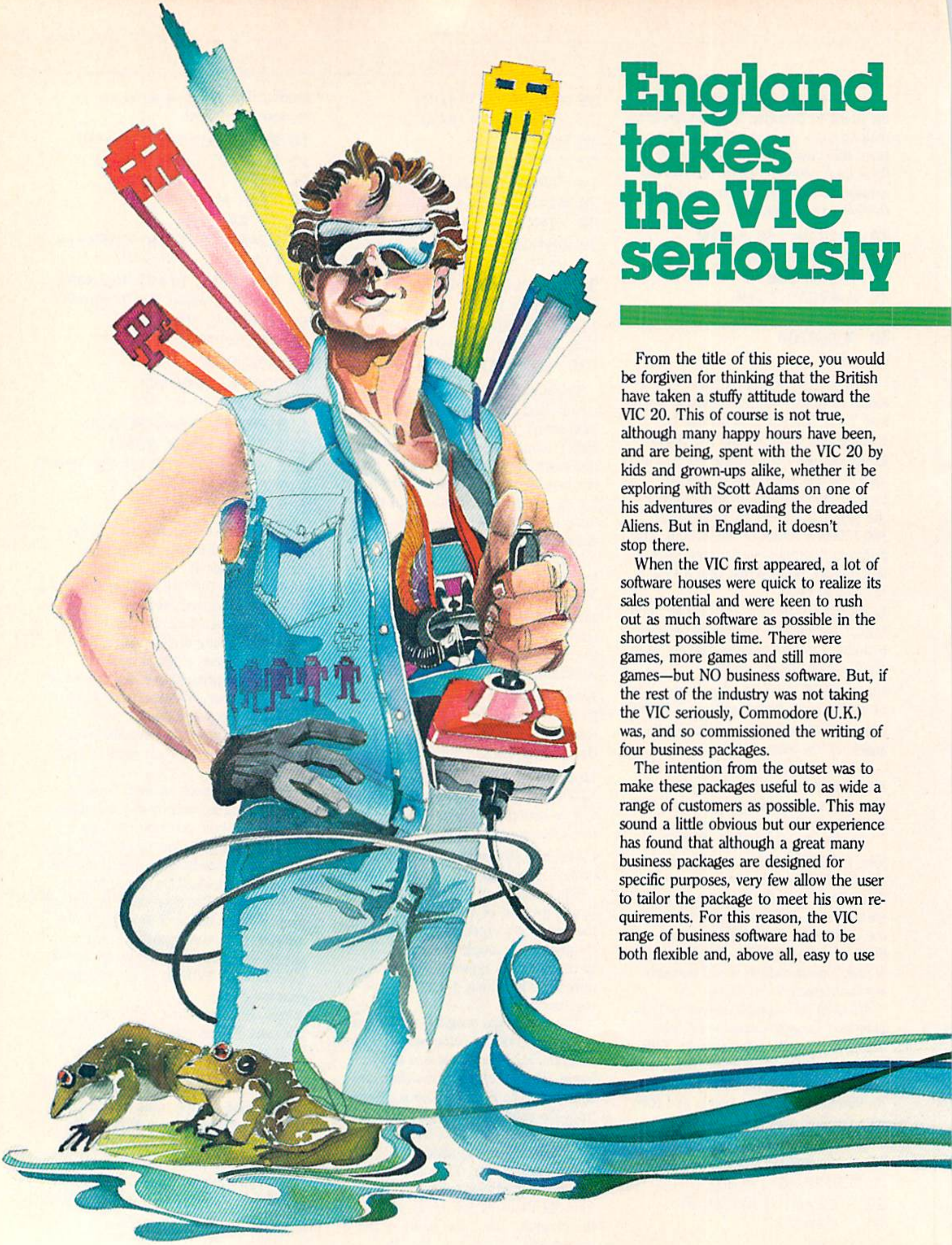
The last design program we'll discuss starts and ends the same as the previous program, but the middle is a little different. This time after you define and store 12 circle points, a line is drawn from the center of the screen to each point. Then the value of DX for each point is doubled and added to X1 to make a new X2,Y2. The old X2,Y2 becomes X1,Y1 and a line is then drawn from X1,Y1 to X2,Y2. Lines 110-140 of the program switch the values of the points, and loop back to the line routine. This design is surely one to impress friends with.

```
10 R=80:H=160:K=100:P=0
15 FORI=0TO354STEP6
16 PRINTI:NEXT:END
20 X1=H:Y1=K
25 T=I*π/180
30 X2=R*COS(T)+H:Y2=R*SIN
   (T)+K
50 DX=X2-X1:DY=Y2-Y1:X=X1
   :Y=Y1:L=SQR(DX*DX+DY*DY)
60 XI=DX/L:YI=DY/L
70 FORZ=1TOL
80 GOSUB200
90 X=X+XI:Y=Y+YI
100 NEXT
110 P=1-P:IFP=0THEN150
120 X3=X1+2*DX:Y3=Y1
130 X1=X2:Y1=Y2:X2=X3:Y2=Y3
140 GOTO50
150 NEXT
```

Speaking of impressing friends, be careful! I had a friend stop by after lunch the other day and I wanted to show him the GEOMETRIC design, but before it had drawn 20 lines, he was ready for dinner! I knew I had made a mistake. Drawing designs takes time. If you want to save a design, you can dump it to a printer (look at page 88 of the March issue of *Commodore* magazine). Even experimenting is hard because once something starts drawing it takes a while to see the result, or even a mistake!

Here are a couple of hints to keep you from pulling your hair out: before letting the whole program run, use a RUN10. This will allow you to spot any problems (divide by zero, illegal quantity . . .) more quickly. Also, watch your X and Y values! The subroutine at 200 will happily plot a dot at X = −2000! Unfortunately, it won't appear on the screen, and just might freeze up the 64 (you could be POKing in the middle of your program!).

Well, that's our tour of bit-mapped graphics. The sample programs should keep you glued to your monitors for a while, and maybe you even picked up some good ideas for your own designs. If nothing else, I hope some of your frustrations have floated off to bother someone else. By the way, watch next issue for some screen-to-tape and disk-dump routines. They can really save time! C



England takes the VIC seriously

From the title of this piece, you would be forgiven for thinking that the British have taken a stuffy attitude toward the VIC 20. This of course is not true, although many happy hours have been, and are being, spent with the VIC 20 by kids and grown-ups alike, whether it be exploring with Scott Adams on one of his adventures or evading the dreaded Aliens. But in England, it doesn't stop there.

When the VIC first appeared, a lot of software houses were quick to realize its sales potential and were keen to rush out as much software as possible in the shortest possible time. There were games, more games and still more games—but NO business software. But, if the rest of the industry was not taking the VIC seriously, Commodore (U.K.) was, and so commissioned the writing of four business packages.

The intention from the outset was to make these packages useful to as wide a range of customers as possible. This may sound a little obvious but our experience has found that although a great many business packages are designed for specific purposes, very few allow the user to tailor the package to meet his own requirements. For this reason, the VIC range of business software had to be both flexible and, above all, easy to use

Four business programs for the VIC 20, created in the United Kingdom, will be available in the U.S. very soon.

By Mike Smith

even for the person who had never touched a computer before.

The type of business software was another important consideration. From enquiries received from end users, it was evident that inventory control, information handling and word processing were the main areas of interest. We decided therefore to commission programs for all three applications and an additional 'spread sheet' package. All four packages have been tremendously well received in the U.K. and are being used in a wide variety of ways in business and the home.

For example, the University Hospital in Cardiff, Wales, has been using *VICFILE,* an information-handling and record-manipulation program, for some time in the diagnosis and treatment of diabetics. Patients are hooked up to a machine that monitors their blood-sugar level. If this falls below a certain figure, the shortfall is corrected by means of a quantity of insulin automatically administered intravenously.

Before *VICFILE* was purchased, the readings from the machine for each patient were written on paper. This not only used up a great deal of time and paper but was also prone to errors. The introduction of *VICFILE* also has allowed the readings for each patient to

Robert Neumann

be stored on diskette. Therefore the physician can print lists of readings for each patient whenever they are needed, and can quickly assimilate the information and prescribe the proper insulin dosage.

In a lighter vein (no pun intended—honestly!!), *SIMPLICALC*, a 'what if' spread sheet package, is at present being used by a small business manufacturing chocolate Easter eggs and candy bars. The owner uses the package to calculate the prices to charge for each item based on the quantity of ingredients used, labour costs, overheads, etc. It has proved to be a great success. So much so, that he presented me with an Easter egg for helping him design his application, with the assurance that I would not be eating into any of his profits!

VICWRITER, a simple-to-use, low-cost word processor for the VIC, has found great favor, especially among students and journalists. The features editor of the London Times was particularly impressed. We have also received "divine approval" for this package from a minister in the north of England who uses it to prepare his sermons and produce a weekly parish newsletter.

VIC STOCK CONTROL has proved to be very popular both in business and the home as an inexpensive way of controlling and reporting inventory holdings. One of my colleagues at Commodore (U.K.) is something of a wine connoisseur and uses the package to check the state of his cellar without having to fumble around among the cobwebs. He keeps records of vintage, type, age and can even produce a value report for insurance purposes.

Of prime importance for all the VIC business software is the quality of user documentation. All too often the advantages of a package have become negated by a manual incomprehensible except, that is, to the software author, who probably wrote the manual anyway. Moral: the less a writer knows about the software at the start, the better the manual will probably turn out.

Commodore (U.K.) is proud of the standard of user documentation in these four packages. (As a technical writer myself, this view may appear to be somewhat biased but it has been proved by the fact that we get very few enquiries from end users about how to operate the software.) The approach we take in all manuals is to show what CAN be achieved and then furnish sufficient information so the user can design his own application. The success of this approach can be judged not only by the volume of sales of VIC business software but also by the uses to which it is being put.

The price of VIC business-oriented software has purposely been kept low to encourage as many VIC owners as possible to try these products for themselves for use both in the office and in the home. Inevitably, this has caused a few domestic arguments, with the kids wanting to Crunch the Cosmos while Dad is trying to work out how far this month's pay check will go. Most fathers, for the sake of a little peace, resort to the only real solution—another VIC and a portable TV.

The success of the VIC range of business software has finally persuaded even the most skeptical VIC user that this computer is simply MORE than 'fun and games.' C no more pencils, no more books

A First Look at LOGO

by David Malmberg

We are delighted to have David Malmberg take over our learning-at-home department. Beginning with this issue, Dave will become a regular contributor to Power/Play, *covering the many facets of education in the home. But, since he happens to be an expert in (among other things) LOGO and Turtle Graphics, you can bet you'll be learning a lot about using these exciting programming languages. Coincidentally, that's just how he's starting off...*

LOGO on the Commodore 64 is truly a phenomenon! As a computer language it is both simple to learn and extremely powerful to use. A cousin of the language LISP, LOGO is a frequently used tool in high-level original research in the field of Artificial Intelligence at major universities. At the same time, LOGO is being increasingly used to teach children as young as four computer programming and problem solving skills. Yet LOGO is considered by many educators to be much more than just a computer language. It has been called an "environment for learning" and a "collection of the tools for thought" by some of its disciples.

LOGO was first developed by Seymour Papert and others at MIT's Artificial Intelligence Laboratory in the late 60's. Papert had worked with Jean Piaget, the famous child psychologist, studying how children think and learn. Piaget felt that children learn best by self-discovery and by trial-and-error, and that the real challenge to educators is to provide both the environment and the tools to nourish this discovery process. LOGO was designed with these ideas in mind. The Commodore 64 implementation of LOGO succeeds extremely well in providing both the tools and the environment for self-discovery.

The principle discovery tool in LOGO for children and other beginning programmers is Turtle Graphics. LOGO's Turtle Graphics capability allows the programmer to instruct an imaginary Turtle (represented by a small triangle) how to roam over the surface of the Commodore 64's screen. As this Turtle moves, it can leave behind very interesting and artistic pictures. Using the Turtle, it is easy for you to try various ideas and to get immediate feedback (in the form of a drawing on the screen) on whether the ideas work as you think they should. If they don't work quite right, LOGO easily enables you to try something else or to explore your "mistakes" if you find them interesting. This ability to "debug" your ideas and to gradually work towards a solution to a problem is the cornerstone of LOGO's implementation of the "Piagetian" view of learning.

Let's look at Commodore 64 LOGO more closely. Consider the following short LOGO program to draw a square on the screen:

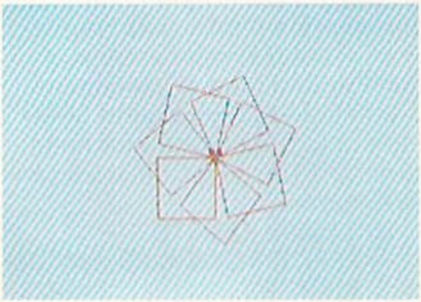
```
TO SQUARE :LENGTH
    REPEAT 4 [FORWARD :LENGTH
    RIGHT 90]
END
```

LOGO is a *procedural* language in that programs are broken up into smaller, separate tasks or procedures. In the case of SQUARE, the entire program is a single procedure. The first line of a procedure (beginning with TO) gives the title and lists the inputs to the procedure, if any. The last line of each procedure is always END. The middle lines, called the body, give the commands telling what the procedure does. In SQUARE, the body instructs the Turtle to REPEAT four times the sequence of commands enclosed in the brackets, namely to move FORWARD a distance of LENGTH (which is an input) and then turn RIGHT 90 degrees. The colon (pronounced "dots" by LOGO-philes) in front of the name LENGTH means to use the current value associated with LENGTH.

Once a procedure is defined in LOGO it becomes another term the computer *knows* and it may be used just as if it were part of the original LOGO language. For example, once SQUARE is defined you can use it in another procedure, such as:

```
TO PINWHEEL :TIMES :LENGTH
    REPEAT :TIMES [SQUARE
    :LENGTH RIGHT 360/:TIMES]
END
```

Once PINWHEEL is defined in LOGO as a procedure, you may use it in direct mode by entering PINWHEEL 7 50 to draw seven separate squares (with sides of LENGTH 50 pixels) all pivoted around a common point in the center of the screen (the Turtle's HOME position.) See Figure 1.

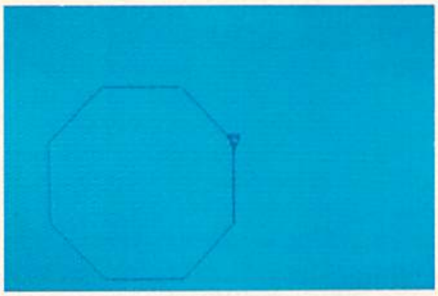

1

This ability to expand the vocabulary of the LOGO language (as was done with SQUARE) is called *extensibility*. Very few computer languages have this very powerful capability to grow as the programmer directs. The languages LOGO, LISP, APL and FORTH are extensible. BASIC, the "standard" for personal computers, and most other programming languages are not extensible.

Let's explore some other examples of LOGO's Turtle Graphics. Consider the following procedure:

```
TO POLYGON :SIDES :LENGTH
    REPEAT :SIDES [FORWARD
    :LENGTH RIGHT 360/:SIDES]
END
```

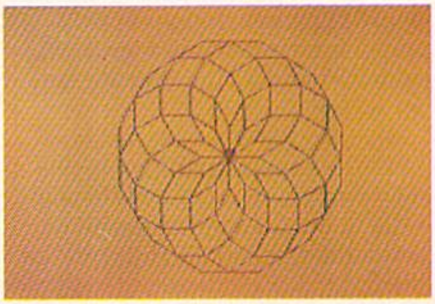
This procedure causes the Turtle to draw a regular polygon. Notice that if SIDES has a value of four, POLYGON will cause a square to be drawn. If you enter POLYGON 8 65 the procedure would draw the octagon shown in Figure 2. The small triangle-shaped object on the left side of the octagon in Figure 2 is LOGO's representation of the Turtle. If you want it to disappear so it will not detract from your pictures, you can give the command HIDETURTLE.

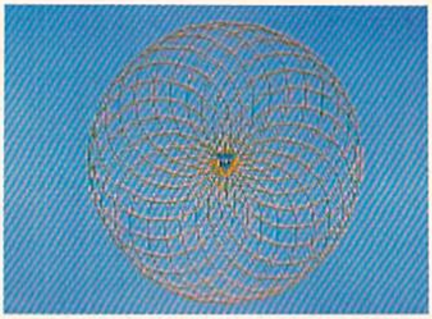

2

To further generalize PINWHEEL you can define a new procedure:

```
TO PINWHEEL.POLY :TIMES
    :SIDES :LENGTH
    REPEAT :TIMES [POLYGON
    :SIDES :LENGTH RIGHT
    360/:TIMES]
END
```

As examples of the types of fantastic designs that can be drawn by LOGO Figure 3 shows PINWHEEL.POLY with TIMES=12, SIDES=12 and LENGTH=25 and Figure 4 has TIMES=24, SIDE=36 and LENGTH=10.


3


4

Probably the most famous Turtle Graphics procedure in LOGO is:

```
TO POLYSPI :LENGTH :ANGLE
    :CHANGE
    IF :LENGTH > 200 THEN STOP
    FORWARD :LENGTH
    RIGHT :ANGLE
    MAKE "LENGTH :LENGTH +
    :CHANGE
    POLYSPI :LENGTH :ANGLE
    :CHANGE
END
```

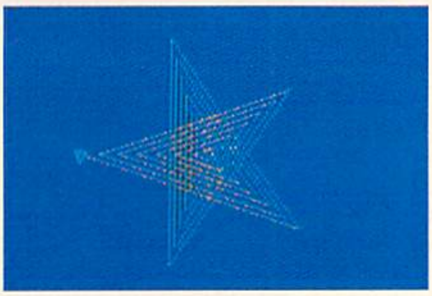
This procedure has several new LOGO commands. The second command tests if the current value of LENGTH is greater than 200 and STOPs drawing if it is. This command is similar to its BASIC language counterpart. The MAKE command is used to assign values to variables. In this case LENGTH is being assigned a value equal to its current value plus the current value of CHANGE. The MAKE command is similar to the LET or just "=" commands in BASIC.

The POLYSPI procedure also illustrates another special capability of LOGO: it is *recursive* in that LOGO allows something to be defined in terms of itself. Notice that POLYSPI contains a reference to itself in the next-to-the-last line. Recursion is a very powerful property for a computer language because it often enables extremely complex and difficult problems to be formulated and solved quite easily. Other languages that are recursive are LISP, APL, C, and Pascal. BASIC and most other languages are not recursive.

Figure 5 shows POLYSPI 1 123 2 and

Figure 6 shows POLYSPI 5 144 6.

5



6

Turtle Graphics is just one of LOGO's strengths. LOGO is also a *list processor* and has tremendous capabilities in the manipulation of text. As a result LOGO is one of the key tools being used by researchers in Artificial Intelligence who are working on getting computers to "understand" the English language rather than needing to always communicate via a programming language.

As an example of LOGO's capabilities in these areas, let's write a program to translate English into Pig Latin. In case you have forgotten, the rules for this translation are: (1) strip off the leading consonants (if any) and append them to the end of the word, then (2) add "AY" to the end of the word. For example, "pig latin is fun" would be translated as "igpay atinlay isay unfay." This sounds simple enough but this translation process is a relatively difficult task for most computer programming languages. If you are skeptical, before reading any further try to write a general BASIC program to translate any sentence or phrase.

In LOGO you must program in a "structured" style. This means break-

no more pencils, no more books

ing the programming job into small meaningful pieces and making a separate procedure out of each task. For your Pig Latin program, there are three separate tasks needed:

1—Determining if a word begins with a vowel

2—Translating a single word

3—Translating a phrase with one or more words

Let's see how LOGO can handle each of these tasks. First, the following procedure would check whether the first letter of a word is a vowel or a consonant:

```
TO VOWEL? :LETTER
    OUTPUT MEMBER? :LETTER
    [A E I O U]
END
```

The term MEMBER? is a standard LOGO procedure that tests whether something is an element of a list (or set) of items. In this instance MEMBER? tests if the current value of LETTER is a member of the list A, E, I, O, or U and returns a TRUE value if it is and a FALSE otherwise. In LOGO lists can be any size and can contain any combinations of numbers, letters, words, LOGO commands, or even other lists. Lists are always shown in square brackets. LOGO's ability to manipulate lists makes the language extremely powerful for solving certain types of problems.

Because LOGO is programmed one procedure at a time, the process of testing and debugging is greatly simplified. For example, before writing the rest of the Pig Latin program, you can test this procedure by itself. Specifically, if you enter VOWEL? "Q, the Commodore 64 will respond with RESULT: FALSE and if you enter VOWEL? "E, you will get RESULT: TRUE. After doing these tests you know this part of the program works as it should.

The next procedure you need will translate a single word:

```
TO PIG.WORD :X
    TEST VOWEL? (FIRST :X)
    IFTRUE OUTPUT WORD :X "AY
    IFFALSE OUTPUT PIG.WORD
    WORD (BUTFIRST :X) (FIRST :X)
END
```

This procedure uses several LOGO terms that need explanation:

FIRST :X outputs the first character of X if X is a string of characters or the first word if X is a list of words. For example:

PRINT FIRST "ABCD will print A

PRINT FIRST [NOW IS THE TIME] prints

NOW

BUTFIRST :X outputs everything except the first character if X is a string of characters and all but the first word if X is a list of words, i.e.,

PRINT BUTFIRST "ABCD will print BCD

PRINT BUTFIRST [NOW IS THE TIME]

prints IS THE TIME

WORD takes two or more character strings and concatenates them. For example, WORD :X "AY (in the third line) appends AY to the current characters of X. In the next-to-the-last line the terms WORD (BUTFIRST :X) (FIRST :X) create a new character string, which has the first letter of X shifted to the last letter.

TEST VOWEL? (FIRST :X) in the second line of the procedure checks whether the first character of the current string X is a vowel. If it is, the next line (IFTRUE, etc.) adds AY to the end of X and outputs the result. If it is not a vowel the line beginning with IFFALSE shifts the leading character of X (a consonant) to the end of X and tries again with the new X string by using the *recursive* reference to PIG.WORD. This procedure will keep calling itself recursively until it finds the first vowel and outputs the correct Pig Latin translation of the word.

Once again you would test this procedure by itself before going on to the next part of the program. For example:

PRINT PIG.WORD "LOGO prints OGOLAY

PRINT PIG.WORD "ANSWER prints ANSWERAY

Your third and last procedure handles the translation for more than one word:

```
TO PIG.LATIN :PHRASE
    IF :PHRASE = [] THEN OUTPUT []
    OUTPUT SENTENCE
    (PIG.WORD (FIRST :PHRASE))
    (PIG.LATIN (BUTFIRST :PHRASE))
END
```

This procedure also has a new LOGO term. SENTENCE combines individual words and/or lists of words into a single long list of words. For example, SENTENCE "ONE [TWO 3 4] [FIVE 6] creates the list [ONE TWO 3 4 FIVE 6]. In the above procedure, the use of SENTENCE creates a new list composed of the first word of the phrase translated into Pig Latin (via the PIG.WORD reference) plus the remaining words of the phrase. By recursively calling PIG.LATIN with the remaining untranslated words each word in the phrase is translated in turn. The second line of the procedure tests for the condition that the current list of words in the phrase is empty (i.e., all the words have been translated) and outputs a "null" list to stop the recursive calls.

Now you can test the entire program. For example:

PRINT PIG.LATIN [COMMODORE LOGO IS FUN]

Prints: OMMODORECAY OGOLAY ISAY UNFAY

If you count the lines in these three procedures you will see that your general Pig Latin program only took twelve lines and that included three END lines. A BASIC program to do this same translation would have taken much, much longer to write and would have been much more complicated. In addition the BASIC version would not have been half as easy to test and debug.

In summary, Commodore's new LOGO for the 64 is a very exciting language. It has broad application and offers something of value to beginner and expert alike. Its Turtle Graphics features offer a fun and effective way to learn programming and problem-solving skills. At the same time Commodore 64 LOGO is a very powerful language and has many capabilities not found in BASIC, such as extensibility, recursion and list processing. LOGO is a language for everyone! C

Let your VIC-20 TALK TO YOU with our Votrax Based "VOICE SYNTHESIZER"

$79.00

$79.00

Real Sound
Adventures
Talking
Action
Games

Sale $79.00

+

Sale $14.95
4" Speaker

You can easily program your voice synthesizer to give you real sound adventures and talking action games. With our speech editor you can create words and sentences using easy to read symbols that translate into data statements to be used in your own custom programs.

* FREE — Your choice of $19.95 Speaker and Cabinet 4" or $14.95 Voice Editor.

Voice Synthesizer	$109.00
	Sale $79.00
Special Synthesizer Speaker *	List $19.95
	Sale $14.95
Word Creator Editor Tape and manual *	List $14.95
	Sale $9.95

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

kids' corner

Our New Department: **For Kids Only**

by Betsy Byrne

To all young Commodore users: Here it is, at last—a section of *Power/Play* all your own. I have noticed that magazines sometimes have programs of interest to kids for *parents* to read or type in, but not much is actually included for you to do (or read) on your own. The editors of *Power/Play* have asked me to do something about this situation, so I'm going to do my level best to put together a section where kids can read about the kinds of computer things *kids* are interested in!

All the kids whose programs are in this issue live in New Mexico. That is because *I* happen to live in New Mexico, and at the moment I don't personally *know* any kids with Commodore computers that live anywhere else! It's up to all of you who are reading this to change that, and I have a whole list of ways that you can use:

1. Send me programs that you have written, or that your parents have written for you.
2. Send me a letter telling about you, your friends, and your family. Let us know what you have been doing with your Commodore computer.
3. If you especially like (or dislike!) a particular game, program, or computer book, write a review!
4. Do you like to draw? Cartoons or drawings of computers and the people that use them would be great.
5. Do you have questions about programming or using your computer that you haven't been able to find answers to? Send 'em in!
6. How about stories (fact or fiction) and poems about computers? These can also be a lot of fun to write (and read).

By the time the *next* issue comes out, we want to be sure to have articles from kids all over the country. So come on, get out your computer, your pencil and your stamps. And when you send in your contribution, be sure to include a picture of yourself or one of you and your computer. Don't forget to include your area code and phone number so I can call you if I need more information. Here's HOW and WHERE to send stuff:

Programs: should be on tape, plainly marked with the program name, your name, and the type of computer it was made on. Be sure to include directions, and check to see that the copy *works* before you mail it. Tape mailers and protective envelopes are found in most office supply stores.

Articles and other contributions: Type or print them double-spaced on a printer, if you can. Otherwise, hand print, using every other line. Be sure your name, phone number, and address are on every page. Number the pages at the top or the bottom. Then staple, or otherwise attach pages to each other.

Don't forget to include a few paragraphs telling about yourself and everything you can think of to say about whatever it is you are sending in. Remember—we want a picture of you!

The address for programs and articles is:

Commodore Kids
c/o Betsy Byrne
6212 Karlson N.E.
Albuquerque, N.M. 87113

Unfortunately, we can't send your articles, tapes, or pictures back to you, so be sure you keep a copy of anything you send. And *the next time* you open *Power/Play* magazine, it may be your picture you see, or your program that is listed for other kids to try out. Believe me, it really feels good! C

Kids and Computers

by Betsy Byrne

Kids and Computers—are they really made for each other? From all indications, kids seem to think so. Children and teens are taking to computers like ducks to water. From twelve month-old Amity Palmer, pushing random keys to display letters on a brightly colored screen, to young Rich Kotomori, who uses a sophisticated word processor to turn out his homework.

What do fifteen year-old Rich and baby Amity have in common with each other and with possibly a million other young computer users? The Commodore computers they are using to carve out a niche in this computer generation.

These young people don't think that they are doing anything remarkable. By their standards, or by the standards of youthful programmers like Diana Kaupilla and Chris Harris they are not. But to parents whose generation was amazed at the introduction of hand-held calculators, their achievements are awesome. And to grandparents, many of whom did not see a television set until their teens, they're nothing short of miraculous!

When kids get together to talk about computers and to share ideas or comment on the latest available software, a kind of magic sometimes seems to develop, an excitement that leaps the normal barriers of age and gender that they once thought so important. Suddenly the important questions are "Do you have a VIC 20 or 64?" or "How long have you had yours? Did you figure out that new Adventure game?" And best of all is the one that begins with "Watch this! I figured out a BETTER way to . . .". This might not sound like a dialogue that will change the future, but I submit to you that

Amity Palmer with her father David and "her" Commodore 64.

it surely will.

This new phenomenon, these "Commodore Kids", are the reason that we are setting aside these pages of *Power/Play* exclusively for articles and programs by, for, and about young Commodore users. Material written by kids, their parents, and their teachers will now be a regular feature in each issue. We hope to provide them with a forum for ideas and techniques, a place of their own where they can meet others like themselves and better learn to use this wonderful tool, this magic toy, that their generation has claimed for its own. **C**

Play Tone-Match

A Game by Christopher Harris

Christopher Harris (he likes Chris) is twelve years old, and a seventh grader at Jefferson Middle School in Albuquerque, New Mexico. When the Harris family bought a Commodore 64 last fall, Chris started right in to teach himself programming, with a little help from his father, Dick. By December, Chris was well on the way to successfully converting his favorite arcade game to the 64. On Christmas he received a Coleco game machine, but Chris, being a true computer enthusiast, returned the present. He came home instead with a VIC 20, a "real computer". (No I am NOT making this up!)

I wanted to know how Chris happened to write the VIC program I have reprinted here. When I asked, he told me: "I was just fooling around with the VIC 20 sounds and up came Tone-Match." And on the subject of using entire words for variables: "Instead of REMark statements, I use variables that are very obvious, so that later on when I go back and look at the program I remember exactly what I did and why." I called Chris to ask him how to work Tone-Match and he laughed "Oh, no! I knew I left something out! DIRECTIONS!" But we have them now, and I will pass them on to all of you who will want to type in Chris' program and have fun improving your "ear" for music:

1. When the computer asks for a level, answer with a number, remembering that 1 is the hardest.
2. You will hear a tone—listen carefully! How long it plays depends on the skill level. When you hear the second tone, try to make it match the first, using the + and − keys to go up and down the scale.
3. When you think you have matched the first tone, press F1. The computer will tell you how close you came to matching it.

To play again, press any key. Your new level will match your last score. For example, if you were 33 points off, last try—your new level will be 33. **C**

—Betsy Byrne

Christopher Harris, age 12.

```
Listing Codes Used

<CU>     Cursor Up (shifted cursor)
<CD>     Cursor Down (unshifted cursor)
<CL>     Cursor Left (shifted cursor)
<CR>     Cursor Right (unshifted cursor)
<HM>     Cursor Home (unshifted CLR/Home)
<SC>     Clear Screen (shifted CLR/Home)
<RV>     Reverse On (control and 9)
<RO>     Clear Screen (control and 0 )
<ID>     Insert Key or Delete Key
<CC>     Color Change (you will have to decide
          what color unless specified)
```

```
Lines  30-34  ** Initialize program.
Lines  40-50  ** Checks to see if you pressed a number or letter.
Lines 100-110 ** Checks for + or - key pressed (raises or lowers tone)
Lines 120-130 ** Checks tone for too high or too low.
Line  150     ** Checks for F1 pressed: If so goto 500.
Lines 500-502 ** Makes the program wait after it tells you the score.
lines 510-810 ** Does a Scale and tells you your new score.
Note:In line 33 the W's are typed as shifted W's; they are graphics.
30 POKE36879,8:PRINT"{CC}"
31 S1=36874:S2=36875:S3=36876:S4=36877:V=36878
32 PRINT"{SC}{CD}{CD}{CD}{CD}  ***{RV}WELCOME TO{RO}***"
33 PRINT"   {RV}WWW{CC}TONE-MATCH{CC}WWW"
34 PRINT"{CD}{CD}{CD}{CD}WHAT LEVEL WOULD YOU   LIKE TO START AT?":
40 INPUTLV$:LV=VAL(LV$)
50 IFLV=0THENPRINT"BETWEEN 1AND INFINITY,PLEASE{CU}{CU}{CU}{CU}":GOTO30
60 REAL=INT(RND(1)*128)+128
70 FAKE=INT(RND(1)*128)+128
80 POKEV,15:POKES3,REAL
90 FORI=1TO(LV*100):NEXT:POKES3,FAKE
100 Q=PEEK(197):IFQ=5THENFAKE=FAKE+1
110 IFQ=61THENFAKE=FAKE-1
120 IFFAKE>255THENFAKE=255
130 IFFAKE<128THENFAKE=128
140 POKES3,FAKE
150 IFQ=39THEN500
160 GOTO100
500 FOR I=1TO10
501 GETJ$:IFJ$<>""THEN500
502 NEXT
503 IFREAL>FAKETHEN600
510 IFREAL<FAKETHEN700
520 FORI=128TO255:POKES3,I:FORB=1TO50:NEXT:NEXT
530 PRINT"{SC}{CD}{CD}{CD}{CD}!!!!!{RV}{CC}YOU DID IT{CC}{RO}!!!!!"
540 LV=1
550 GOTO800
600 FORI=128TO255STEP(REAL-FAKE):POKES3,I:FORB=1TO50:NEXT:NEXT
610 LV=REAL-FAKE:PRINT"{SC}{CD}{CD}{CD}{CD}YOU WERE "LV" OFF"
620 GOTO800
700 FORI=128TO255STEP(FAKE-REAL):POKES3,I:FORB=1TO50:NEXT:NEXT
710 LV=FAKE-REAL:PRINT"{SC}{CD}{CD}{CD}{CD}YOU WERE "LV" OFF"
800 POKEV,0
801 GETB$:IFB$=""THEN801
810 GOTO60
```

Friends Of The VIC Turtle

By Albert Rizzoli, M.D.

Most of us with children have purchased a toy called Etch-A-Sketch. This clever toy lets you draw pictures by turning knobs. One knob is for the X direction and the other for the Y direction. I have written a short program to simulate this toy. It requires the Super Expander and a set of paddles. The button on one paddle clears the screen and the other changes the color of the line.

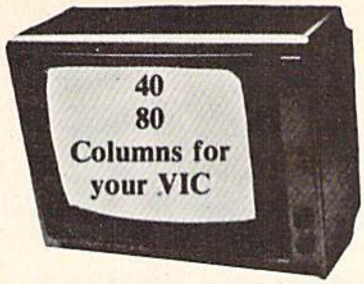
If you have several lines on the screen of different colors, you begin to notice a limitation of the Super Expander. The color in a given region can only be one color. These regions correspond to the size of one VIC alphanumeric character. This can be frustrating if you draw two different color lines close together.

There is another idiosyncracy in this program. My paddles are electronically noisy, particularly at the extremes of rotation or if turned rapidly. The moving X or Y line is usually smooth, but the non-moving paddle may give some jitter to the baseline. Lines 115 and 117 smooth this out some, but a better hysteresis algorithm should be written. Please send your ideas to Betsy for inclusion in this column. In defense of the program, it is still a lot of fun and can draw some very colorful pictures.

A similar program can be written using a joystick, see "VIC Make-A-Sketch" in *COMPUTE!'S FIRST BOOK OF VIC,* page 31. Danny Byrne of Albuquerque has also written a sophisticated graphics package using a joystick. **C**

```
1 REM ETCH-A-SKETCH
2 REM VIC-20 VERSION BY ALBERT RIZZOLI
3 REM REQUIRES SUPER EXPANDER AND PADDLES
4 REM FIRE BUTTONS CHANGE COLOR AND CLEAR SCREEN
10 GRAPHIC2:COLOR1,3,0,0
20 S=256*PEEK(648):A=30270:LL=S
30 IFPEEK(648)=16THENA=33792
40 DD=37154:P1=37151:P2=37152
50 PX=36872:PY=36873
60 POKEDD,127:P=PEEK(P2)AND128
70 FR=-(P=0):POKEDD,255
80 P=PEEK(P1):FL=-((PAND16)=0)
90 VL=PEEK(PX):VR=PEEK(PY)
100 XO=X:YO=Y
110 X=INT(VL*1023/255):Y=INT(VR*1023/255)
115 IFABS(XO-X)<5THENX=XO
117 IFABS(YO-Y)<5THENY=YO
120 DRAW3TOX,Y
130 IFFRTHEN:SCNCLR
140 IFFLTHEN:C=C+1
150 IFC=1THENC=C+1
160 IFC>7THENC=0
170 COLOR1,3,C,C
180 GOTO60
```

SALE! $99



SALE! $99

VIC-20 COMPUTER
40-80 COLUMN BOARD

WORD PROCESSOR

ALL FOR ONLY $99

MAILMERGE DATABASE

PLAN MANAGER

ALL IN 40 AND 80 COLUMN FORMAT

TERMINAL EMULATOR

ELECTRONIC SPREADSHEET LIKE VISICALC

Now you can get 40 or 80 Columns on your T.V. or monitor at one time! No more running out of line space for programming and making columns. Just plug in this board and you immediately convert your VIC-20 computer to 40 or 80 columns! PLUS, you get a Word Processor, Mail Merge program, Electronic Spreadsheet (like VISICALC) and Terminal Emulator! These PLUS programs require only 8K RAM memory and comes in an attractive plastic case with instructions. List $149 Sale $99

• COMMODORE 64 COMPUTER — "80 COLUMN BOARD" LIST $275 SALE $179

(Less $20 Accessory Purchase Discount)

"15 DAY FREE TRIAL"

- **We have the lowest VIC-20 prices**
- **We have over 500 programs**
- **Visa — Mastercharge — C.O.D.**
- **We love our customers!**

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Bank Sprites: **Gain Interest**

Using Sprites in Different Video Banks
By Tim Villanueva

This article is written for users who understand the use of sprites on the Commodore 64. For information regarding memory blocks, sprite pointers, and sprite registers, refer to pages 69-71 in the *Commodore 64 User's Guide* or pages 131-181 in the *Commodore 64 Programmer's Reference Manual*.

A sprite's shape is defined by a 64-byte block of memory. The starting address (memory location) is equal to the block number times 64. For example, memory block 13 starts at 13*64 or 832. The sprite only uses 63 bytes of the block, with the last byte being left empty. If you have made sprites using the *User's Guide,* you are probably putting them (the data for them) in the memory blocks that are mentioned—13, 14, and 15. This may be enough for the experimenter, but someone who wants to implement sprites in a large program may need more than three defined shapes.

Another place you can put your sprites is in BASIC RAM memory. You can place the data for your sprites in any free RAM available. The sprite pointers tell the computer exactly where your sprite is. For instance, when working with sprite #2, the sprite pointer is memory location 2042. The number POKEd into location 2042 tells the computer the block of memory your sprite is in. So, if you POKE 2042, 192, it tells the computer to set the data for your sprite in memory block 192 (starting address = 192*64—or 12288). This location is in the BASIC RAM. The problem you face when placing your sprites in BASIC RAM is that you lose a horrendous amount of BASIC memory if you don't want your BASIC program to override the sprite. Thus, you may want to put the sprite in as high memory as possible. Because of this, you may recall, the largest number a memory location can hold is 255. So, if you POKE the sprite pointer with 255—POKE 2042, 255—this should put you into as high memory as possible. But 255*64 is only equal to 16320. How do you set your sprites into even higher memory??????

Now let's talk about video banks. The Video Interface Chip (VIC II chip) inside the computer (the part that controls any display operations of the computer) uses the

Tim Villanueva showing sprite manipulation on the Commodore 64.

RAM memory for some of its information. For example, locations 1024 through 2023 control the screen memory and 2040-2047 control sprite pointers. The video chip can access ("see") 16K of memory at a time. Since the Commodore 64 has 64K of memory, the memory has been divided into four 16K video banks—16K sections of memory where the VIC II chip sets its information. When in normal operation, the VIC II chip is operating in bank 0 (banks are numbered 0-3); thus, all the display information is contained in the first 16K of memory. That's why the screen is 1024-2023. By changing the video banks, you can set your sprites into higher memory. If you place yourself in bank 1, your sprite pointers now point to bank 1. Thus, memory block 0 now starts at the beginning of the second 16K bank of memory—or location 16384. If you were in bank 2, memory block 0 would start at the beginning of the third 16K bank of memory—or location 32768. But changing video banks affects several things. First, here are the starting locations for the four video banks:

Bank Number	Starting Location
0	0
1	16384
2	32768
3	49152

For instance, when you are in bank 2, all the display information the VIC II chip needs comes from that bank. See the problem? If you remember, the screen memory was at locations 1024-2023. But if now you are in bank 2, that cannot be possible. This is so true! But don't fret, the screen stays relative to the video bank. What???? That means the starting address for the screen is still 1024, but you must add in the starting address for the bank. For example, if you are in bank 2, the screen starts at 1024+32768. Any more problems? What about the sprite pointers? Again, these stay relative. They move just as the screen does. Thus, if you are in bank 2, the sprite pointer for sprite 2 (2042) is equal to 2042+32768.

Now, one more problem you probably cannot recognize yet. All of the above sounds just fine and dandy, but it does not take care of EVERYTHING. With the proper POKEs, you can change video banks and know where you are. You know that if you are in bank 2, the screen starts at 1024+32768. Thus, if you want to POKE a character on the screen in the first location—POKE 33792, # (33792 =32768 + 1024). Unfortunately, the computer operating system doesn't know this much. When you type on the keyboard, you will NOT see anything. This is because the computer is still putting the characters you type at the old screen locations 1024-2023. Where the computer puts its characters is controlled by the operating system. This is not part of the VIC II chip and does not change when you move video banks. But have no fear, you can tell the operating system where to stick its characters!! Memory location 648 (part of the operating system) controls this. The number POKEd into this location represents where the computer puts its characters. The formula for figuring this "mysterious number" is the following:

mysterious number = starting address/256

Example: If you are in bank 2, the screen starts at 33792 (32768+1024). Thus, the mysterious number = 33792/256, which is 132. So, to move the cursor where it belongs, POKE 648,132. That's all there is to it.

Now, how to change video banks (finally). The following POKEs will change banks:

1. POKE 56578, PEEK (56578) or 3

   I'd rather not explain this POKE. Let's just say it warns the computer that you are about to change banks. It must be done before step 2.

2. POKE 56576, (PEEK(56576) AND 252) or A

   "A" can either be a 0, 1, 2, or 3. Good guess! Yes, the 0, 1, 2, and 3 stand for the bank you want to enter—EXCEPT IN REVERSE. So, if you want bank 3, make A=0. The following chart explains:

Value of A	Bank Number	Starting Location
0	3	49152
1	2	32768
2	1	16384
3	0	0

Now the question is "Which bank should I go to?" Well, I suggest bank 2—that's my bank!! But seriously, if you wish to get into high memory, bank 2 is the best choice. Why? Well, one of the best reasons is that in banks 1 and 3 the character set is not available. And I hope you don't go through all the above trouble to get into bank 0!! Very few BASIC programs are long enough to bother anything you put in bank 2.

The following program will put you in bank 2, and set things up for you. (By the way, when you first change banks, the screen will be full of garbage. This is because the screen moved up into high RAM, where there is only a bunch of random patterns of numbers. Line 30 in this program will take out the garbage!)

```
10 REM    CHANGE TO VIDEO BANK 2
20 POKE   56578, PEEK (56578) OR 3
30 POKE   56576, (PEEK (56576) AND 252) OR 1
40 POKE   648,132
50 PRINT CHR$(147): REM CLR/HOME
60 PRINT   "HERE WE ARE IN BANK 2"
70 END
```

Whew, made it (I hope). Now what? Well, back to sprites.... How do we tie all this in to sprites? Now we can create several different shapes for our sprites. Since BASIC probably won't bother us (as in bank 0) and we don't have to worry about killing the operating system (as in bank 0), we can put our data in any of the memory blocks available (0-255) except for where the screen and sprite pointers are. (Remember, they changed banks also). So, all memory blocks are fair game except blocks 16-31. Now, how do we deposit our sprites? Exactly the same as before. (How's that?) We need to POKE the 63 data values which define our sprite into a memory block. To figure the starting address for this, just multiply the block by 64, and add in the bank starting address. Example (only God knows where we would be without examples!): If you want your sprite in block 50, and you're in bank 2 (the only bank to be in), the starting address will be: bank starting address + (memory block*64), Or ---> 32768+(50*64) = 35968. So, you need to start entering the 63 data values at memory location 35968. You must also remember to change your sprite pointer. The sprite pointer will only be the original value + bank starting address. So, for sprite 2, the pointer will be 2042+32768 = 34810. So, POKE 34810,50 (which matches with the 50 above).

Just to test this, create a sprite that is just lines. Type the following once you have changed video banks:

POKE 53269,4	REM ENABLES SPRITE 2
POKE 53252,255	REM SETS X COORDINATE
POKE 53253,200	REM SETS Y COORDINATE (SPRITE WILL BE IN LOWER RIGHT)
POKE 53289,1	REM SETS COLOR TO WHITE
FOR T=0 TO 62: POKE 35968+ T,15: NEXT	REM POKES DATA FOR SPRITE INTO REM MEMORY BLOCK 50
POKE 34810,50	REM SETS SPRITE POINTER TO MEMORY BLOCK 50

The sprite should now be in the lower right-hand corner of the screen. It should look like three vertical lines.

Now that we have explained the basics, we can discuss the real stuff... cough cough cough (Just kidding!) Good luck.

Editor's Note: Tim Villanueva is the author of several exciting games for the Commodore 64, using great sprite graphics. **C**

How to Create Your Own Sprite Creator

On The Commodore 64

by Tim Villanueva

Sprites are one of the easiest forms of high resolution graphics available. And of course the 64 has them. But making a sprite from scratch is not all that easy. It requires lots of work to figure and calculate the data. The following program will take away the pain of creating sprites by hand.

If you haven't worked with sprites yet, I suggest first reading the short explanation in the *Commodore 64 User's Guide,* pages 69-71. What the following program does is print a grid on the screen that represents your sprite in large form. You fill in the grid with asterisks in the pattern you want your sprite to form. Then the program reads the grid and calculates data for the sprite. It automatically enters the data into memory, and the sprite will form before your eyes. First, here is the program:

```
10 PRINTCHR$(147);
20 FORT=1TO21:PRINT"..,,..,,..,,..,,..
   ,,..,,":NEXT:REM 24 DOTS
40 V=53248:POKEV+21,4:POKEV+41,1
50 POKEV+4,255:POKEV+5,200
60 POKE2042,13
70 END
100 C=8
105 FORY=0TO20
110 FORX=1TO24
120 P=PEEK(1023+X+40*Y)
130 C=C-1:IFP=42THENQ=Q+2↑C
140 IFC=0THENC=8:G=G+1:POKE831+G,Q:Q=0
150 NEXTX
160 NEXTY
170 END
200 PRINTCHR$(147):PRINT:PRINT
210 FORT=0TO62
220 PRINTPEEK(832+T),
230 K=K+1:IFK=3THENK=0:PRINT
240 NEXT
250 END
```

Now, here is an explanation of the program:

10-20	Prints the grid on the screen
40-50	Turns sprite on in lower right of the screen, color white.
60	Sets sprite pointer to memory block 13 (starts at location 832)
100	Start of read section. Sets C=8
105	Loop for each of the 21 rows
110	Loop for each row, 24 dots across
120	PEEK the location of each dot, one at a time through the loop.
130	Decrement exponent value— If current dot location contains an asterisk, increment data value by adding to it 2↑C, where C represents the actual dot number of the set of 8. (3 sets in each row of 24)
140	When C=0, set of 8 finished. POKE data value into proper location and reset the value. Reset value of C.
150-160	Return through each loop until finished
200	Start of data section—prints data values in rows of 3.
210	Read 63 memory locations (63 data values to comprise a sprite)
220	Print out the value
230	Format output into rows of three (21 rows with 3 values—each value represents 8 dots or pixels, 8*3=24 across)
240	Return loop
250	End.

How to Use the Sprite Creator

Once the above program is typed in correctly, you are ready to create. The first step is to type RUN. The grid will form and the word READY will appear below the grid. Using the cursor keys, move the cursor UP into the grid. Be sure not to move the cursor down or you may cause the grid to be pushed off the screen. Fill the grid with asterisks in the desired shape of your sprite. When you have achieved the desired shape, move the cursor back to the bottom of the grid. NOTE: Now place the cursor directly on the line below the last row of dots on the grid. Now type in RUN 100. Your sprite should form before your very eyes in the (lower right-hand corner of the screen.) When your sprite is done, the computer will signal READY. You can now move back up into the grid to make any corrections you would like. But be sure not to move the cursor down too far or you will lose part of your grid! When you are done making corrections, move the cursor once again directly below the grid and type RUN 100. You may continue to do this over and over until you achieve the desired shape.

Once you have achieved the correct shape, you are ready for the data. To set the data for the sprite, type RUN 200. The data should print out 21 rows of 3 values each. This makes it easy to copy down the values on paper (or something) to be used in future programs.

Using This Sprite Creator to Make Multicolor Sprites

In the grid that this program makes, each period or comma represents an actual bit in memory (which represent the pixels of your sprite). The usual sprite is 24 pixels across, 21 pixels down. You can create your sprites in three colors. But how does the computer know what color a certain bit should be? Well, when you put a sprite in multicolor mode, it changes the way the computer deals with sprites. Instead of looking at the rows with 24 dots, it examines them as 12 pairs of dots. Each pair of dots now represents one pixel, so instead of 24 pixels across, you have 12 pixels across. The sprite stays the same size, but you lose half of your horizontal resolution. Now the computer can put color into your sprite. Since two bits now represent one pixel horizontally, the computer can use the pattern contained in the two bits for color information. There are 4 possible patterns with the two bits:

1.	00	2.	10
3.	01	4.	11

This can be represented on the grid with an asterisk representing a 1, and a period or comma representing a zero. Now you should understand why the grid is made of periods and commas. If creating a sprite in multicolor, you need to look at each pair of either periods or commas as one pixel. The pattern in that pair will represent the color. If there is no pattern in that pair (00), that will represent the background color.

These are the registers for each sprite:

Sprite Number	Memory Location
0	53287
1	53288
2	53289
3	53290
4	53291
5	53292
6	53293
7	53294

Now, how to switch sprites to multicolor. Memory location 53276 controls multicolor mode for sprites. You can switch individual sprites on to multicolor by POKing this location. You may have both multicolor sprites and "normal" sprites on the screen simultaneously. The following POKE will switch a sprite to multicolor.

POKE 53276, PEEK (53276) OR (2↑SN)

Where SN is the sprite number (0-7).

To switch a sprite off multicolor, you must do the following POKE:

POKE 53276, PEEK (53276) OR (255-2↑SN)

Where SN is the sprite number (0-7).

If you want to switch on all sprites to multicolor, the following will do it:

POKE 53276,255

If you want to switch all sprites off multicolor,

POKE 53276,0

How do you control the color for each different bit pattern? The computer has reserved three places for you to enter (or POKE) the color codes which will represent the bit patterns. They are the following:

Bit Pattern	
00	Transparent, Screen color shows through
01	Sprite Multicolor register 0 (53285)
10	Sprite Color Register (depends on which sprite)
11	Sprite Multicolor register 1 (53286)

So, if on the grid you fill a bit pair (either a pair of commas or a pair of periods) with a 01 bit pattern, that pixel (which is actually the size of two pixels) will be the color represented by memory location 53285. If the bit pattern is an 11, the color will be represented by memory location 53286. If the pattern is a 10, then the color will depend on that individual sprite's color register. C

Night of Death

A Game for the VIC 20
by Diana Kaupilla

Diana Kaupilla, age 10.

Diana Kaupilla is ten years old and goes to Chelwood Elementary School in Albuquerque, where she is in the fifth grade gifted program. She has had her computer since Christmas, and is the main user of the computer at her house. She has had some computer experience at school, but she says it was so different from the VIC 20 that it was "no help at all."

When asked where she got the idea for this program she said, "I first got the idea when I was reading a book about choosing your own adventure, and I wondered if it could be put onto the computer. I started fiddling with it and I figured out how to make one. My first story was called *Night of Death*."

So, here is *Night of Death*, just the way Diana wrote it. **C**

```
1 PRINT"♥"
10 FORX=1TO1000:NEXTX
14 FORX=1TO1000
16 NEXTX
20 PRINT"------------"
30 PRINT"E S C A P E "
40 PRINT"------------"
45 FORX=1TO1000:NEXTX
50 PRINT"F R O M "
60 FORX=1TO1000
65 NEXTX
70 PRINT"H A D E S "
71 FORX=1TO1000
72 NEXTX
76 PRINT"♥"
80 PRINT"YOU KICK A ROCK AND A HOLE OPENS IN THE GROUND IN FRONT OF YOU."
90 PRINT"DO YOU 1:JUMP OR 2:RUN"
100 INPUTX
110 IFX=1GOTO130
120 IFX=2GOTO140
130 PRINT"♥"
135 PRINT"YOU ARE IN A CAGE.THE GUARDS ARE ASLEEP"
136 PRINT"DO YOU 1:ESCAPE OR 2:STAY FOR A BETTER TIME"
137 INPUTX
138 IFX=1THEN160
139 IFX=2THEN180
```

```
140 PRINT"♥"
141 PRINT"THE LAST THING YOU FEEL IS A KNIFE IN YOUR NECK"
142 GOTO10000
160 PRINT"♥"
161 PRINT"YOU ARE IN A DARK CORRIDOR"
162 PRINT"DO YOU 1:GO BACK OR 2:KEEP GOING"
163 INPUTX
164 IFX=1THENGOTO190
165 IFX=2THENGOTO210
180 PRINT"ITS TOO LATE. THEY WAKE UP AND CHOP OFF YOUR HEAD!"
181 GOTO10000
190 PRINT"YOU SEE THE GUARDS AGAIN. ONE OF THEM STIRS IN HIS SLEEP"
191 GOTO180
210 PRINT"YOU SEE AN OLD WOMEN, AND SHE WALKS UP TO YOU."
220 PRINT"RUN SHE SAYS, RUN FROM THIS GOD FORSAKEN PLACE."
230 PRINT"WHY DON'T YOU RUN, YOU ASK."
240 PRINT"I CAN'T. THE CRYSTAL HAS ME."
250 PRINT"I DON'T KNOW WHAT YOU'RE TALKING ABOUT, YOU SAY."
260 PRINT"LISTEN THEN SHE SAYS."
270 PRINT"A LONG TIME AGO,"
280 PRINT"AN EVIL MAGICIAN"
290 PRINT"FOUND A CRYSTAL"
300 PRINT"THAT GAVE HIM MAGICAL POWERS BEYOND IMAGINATION."
310 FORX=1TO15000
320 NEXTX
325 PRINT"♥"
330 PRINT"I CAN HELP YOU, YOU SAY."
340 PRINT"THEN STEAL THE CRYSTAL FROM THE THRONE ROOM!"
350 PRINT"THEN SHE WALKS AWAY."
360 PRINT"DO YOU 1:TRY TO STEAL THE CRYSTAL OR 2:TRY TO GET HOME"
364 INPUTX
365 IFX=1GOTO380
370 IFX=2GOTO190
380 PRINT"YOU SEARCH AND SOON SEE A HUGE GIANT SLEEPING."
385 PRINT"YOU SEE A RED, GLOWING CRYSTAL NEAR HIM."
390 PRINT"YOU GRAB IT AND RUN."
400 PRINT"YOU SEE SOME STRANGE MONSTERS CHASING YOU."
410 PRINT"YOU RAISE THE CRYSTAL AND THEY TURN AND FLEE."
420 PRINT"CONGRATULATIONS. YOU WON!!!!!!!!!!!!!"
1005 END
10000 PRINT"DO YOU WANT TO TRY AGAIN 1:YES 2:NO"
10001 INPUTX
10002 IFX=2GOTO10004
10003 IFX=1GOTO1
10004 PRINT"♥"
10005 PRINT"WHY NOT YOU #$(!$#(!%%#$(&!!??"
10006 END
```

The Octopus Maze

by Gerald and Betty Schueler

An octopus is on the loose. He escaped from his lair and only you can send him back. While this may be slightly melodramatic, the following program does involve moving an octopus-like figure through a maze and back to its lair. The program is both an exercise in joystick dexterity and a demonstration of the sprite graphics available on the Commodore 64.

A sprite is a high-resolution, programmable object. The Commodore 64 can make and control up to eight sprites with standard programming. (See the *Commodore 64 User's Guide* or the *Commodore 64 Programmer's Reference Guide*). Additional sprites are possible with a little more complicated program. The sprites are controlled by a separate picture-maker inside the computer, which handles the video display. The movement of the sprite can be controlled by pre-programming mathematical equations or by programming movement through the use of a joystick or the keyboard.

In our program "The Octopus Maze" movement of the octopus sprite is controlled by use of a joystick, which should be plugged into port #2. Pushing forward on the joystick makes the sprite move up; pulling back on the joystick moves the sprite down and a left or right movement of the joystick makes the sprite move left or right, respectively. If any wall of the maze is touched by the sprite, it will escape your control and the sequence must be repeated from the beginning (Line 240, which asks the computer to detect any sprite-character collisions. Lines 245-260 check to see that the sprite remains inside the screen border).

The game is won by moving the octopus-sprite into contact with the stationary skull-like sprite which is the octopus' lair. Line 230 checks to see when this sprite-to-sprite collision occurs. A successful game can be replayed by simply pushing the firing button on the joystick. This feature is easily attained by the WAIT command in line 320.

The Commodore 64 allows for up to 15 different colors as follows:

COLOR	POKE NO.	COLOR	POKE NO.
Black	0	Orange	8
White	1	Brown	9
Red	2	Light Red	10
Cyan	3	Grey 1	11
Purple	4	Grey 2	12
Green	5	Light Green	13
Blue	6	Light Blue	14
Yellow	7	Grey 3	15

In our version of "The Octopus Maze", the lair-sprite will be light red because line 120 contains the statement, POKE V+40,10. This POKEs light red (code 10) as the lair color. Similarly, the octopus-sprite will be yellow because line 155 says, POKE V+37,7. This POKEs yellow (code 7) as the octopus' color. Changing these numbers in the program can provide a wide variety of colors; however, some color combinations work better than others. For example, cyan, grey and black are good alternate octopus colors but red and green are poor because most of the details are washed out.

If the octopus-sprite moves too fast, or if you simply prefer to start out at an easier pace and work up with experience, the movement of the sprite can be cut back by changing the horizontal and vertical position increments. Use the following modifications for lines 260-320.

```
260  IFS=1THENDY=-1
265  IFS=2THENDY=1
270  IFS=4THENDX=-1
275  IFS=8THENDX=1
280  IFX=255ANDDX=1THENX=0:POKEV+16,1
285  IFX=0ANDDX=-1AND(PEEK(V+16)AND1)=1THENX=255:POKEV+16,0
290  REM--COLLISION INDICATORS
295  IF(PEEK(V+30)AND1)=1THEN300
300  IFX>30ANDY>60AND(PEEK(V+31)AND1)=1THEN160
```

```
305  IFX<25ANDDX=-1ANDPEEK(V+16)=0THEN160
310  IFX>65ANDDX=1ANDPEEK(V+16)=1THEN160
315  IFY<51ANDDY=-1THEN160
320  IFY>229ANDDY=1THEN160
```

Experienced game players may want to move the X values up to 4 or 5 and the Y value up to 3 for faster game play. Unfortunately, the movement of the octopus-sprite will lose much of its smooth flow at higher X and Y values, but the increased skill needed to win the game may make the loss worthwhile.

Pre-programmed cassette tapes of this game are available for $5 each from: The Copy Cats 680 W. Bel Air Ave. Aberdeen, MD 21001 301-272-0472

```
5 PRINT"▯"
10 Z=1024:V=53248:C=55296:B=53280:K=53281:SS=54272
15 POKEB,14:POKEK,6
20 PRINT"▮MOVE THE MONSTER INTO HIS LAIR WITHOUT
   TOUCHING THE MAGIC WALLS"
25 PRINT"▮IF A WALL IS TOUCHED, THE MONSTER WILL
   ESCAPE AND YOU MUST START AGAIN
30 PRINT"▮PUSH THE FIRE BUTTON TO START"
35 WAIT56464,16,16
40 PRINT"▯"
45 REM--THE MAZE
50 FORA1=Z+6TOZ+726STEP40:POKEA1,102:NEXT
55 FORD1=C+6TOC+726:POKED1,3:NEXT
60 FORA2=Z+253TOZ+973STEP40:POKEA2,102:NEXT
65 FORD2=C+253TOC+973STEP40:POKED2,3:NEXT
70 FORA3=Z+254TOZ+273:POKEA3,102:NEXT
75 FORD3=C+254TOC+273:POKED3,3:NEXT
80 FORA4=Z+306TOZ+706STEP40:POKEA4,102:NEXT
85 FORD4=C+306TOC+706STEP40:POKED4,3:NEXT
90 FORA5=Z+740TOZ+751:POKEA5,102:NEXT
95 FORD5=C+740TOC+751:POKED5,3:NEXT
100 FORA6=Z+553TOZ+559:POKEA6,102:NEXT
105 FORD6=C+553TOC+559:POKED6,3:NEXT
110 POKEV+30,0:POKEV+31,0
115 REM--THE SPRITE LAIR
120 POKE2041,193:POKEV+40,10:POKEV+23,2
125 F=12352
130 READQ%:IFQ%=-1THEN145
135 POKEF,Q%
140 F=F+1:GOTO130
145 POKEV+29,2:POKEV+2,175:POKEV+3,135
150 REM--THE SPRITE MONSTER
155 POKE2040,192:POKEV+39,7
160 X=30:Y=60:DX=0:DY=0
165 P=12288
170 READE%:IFE%=-1THEN180
175 POKEP,E%:P=P+1:GOTO170
180 POKEV+21,3:GOTO235
```

```
185 REM--ROUTINE FOR TOUGHING WALLS
190 POKEV+21,0:FORL=SSTOSS+24:POKEL,0:NEXT
195 IF(PEEK(V+16)AND1)=1THENPOKEV+16,0
200 POKESS+24,143:POKESS+6,240:POKESS+4,33
205 FORT=1TO50:POKESS,223:POKESS+1,39:NEXT
210 FORT=1TO75:POKESS,71:POKESS+1,5:NEXT
215 FORL=SSTOSS+24:POKEL,0:NEXT
220 PRINT"♥"SPC(122)"▮SORRY BUT YOU TOUCHED THE WALL"
225 PRINTSPC(202)"▮TRY AGAIN? PUSH FIRE BUTTON"
230 WAIT56464,16,16:RESTORE:GOTO40
235 REM--POSITION INDICATORS
240 POKESS,16:POKESS+1,39:POKESS+4,129:POKESS+6,240:POKESS+24,95
245 FORT=1TO5:FORM=95TO0STEP-5:POKESS+24,M:NEXT:NEXT
250 S=15-(PEEK(56320)AND15)
255 IFS=0THENDX=0ANDDY=0
260 IFS=1THENDY=-2
265 IFS=2THENDY=2
270 IFS=4THENDX=-3
275 IFS=8THENDX=3
280 IFX=255ANDDX=3THENX=0:POKEV+16,1
285 IFX=0ANDDX=-3AND(PEEK(V+16)AND1)=1THENX=255:POKEV+16,0
290 REM--COLLISION INDICATORS
295 IF(PEEK(V+30)AND1)=1THEN350
300 IFX>30ANDY>60AND(PEEK(V+31)AND1)=1THEN190
305 IFX<25ANDDX=-3ANDPEEK(V+16)=0THEN190
310 IFX>65ANDDX=3ANDPEEK(V+16)=1THEN190
315 IFY<51ANDDY=-2THEN190
320 IFY>229ANDDY=2THEN190
325 REM--SPRITE MOVEMENT
330 X=X+DX:Y=Y+DY
335 POKEV,X:POKEV+1,Y
340 GOTO250
350 REM--ENDING FOR WINNING GAME
355 FORL=SSTOSS+24:POKEL,0:NEXT
360 POKESS+4,33:POKESS+24,143:POKESS+6,240:POKESS+5,190
365 FORT=1TO30:POKESS,162:POKESS+1,37:NEXT
370 FORT=1TO40:POKESS,60:POKESS+1,50:NEXT
375 FORT=1TO30:POKESS,162:POKESS+1,37:NEXT
380 FORT=1TO100:POKESS,15:POKESS+1,67:NEXT
385 FORL=SSTOSS+24:POKEL,0:NEXT:POKEV+21,0
390 PRINT"♥"SPC(90)"▮▮CONGRATULATIONS"
395 PRINTSPC(175)"▮YOU DID IT"
400 PRINTSPC(205)"▮PLAY AGAIN? PUSH THE FIRE BUTTON"
405 WAIT56464,16,16:RESTORE:POKEV,30:POKEV+1,60:GOTO40
1000 DATA0,124,0,1,254,0,7,255,128,31,255,224,63,255,240,56,120,120,120,120
1100 DATA120,120,120,124,255,75,252,255,207,254,255,255,254,250,202,175
1200 DATA250,202,167,224,136,39,224,0,39,100,75,135,100,170,134,116
1300 DATA235,142,118,170,95,-1
1400 DATA0,63,128,0,127,192,0,255,224,48,255,240,57,255,240,227,196,120
1500 DATA131,206,120,135,255,252,131,255,248,227,255,240,31,213,112,1,192,120
1600 DATA1,241,252,3,191,196,12,47,199,8,98,103,112,198,32,96,132,48,1
1700 DATA132,28,7,31,7,7,27,7,-1
```

"Look what my new lets me do with

Here's what you get with the complete Alphacom printer project set.

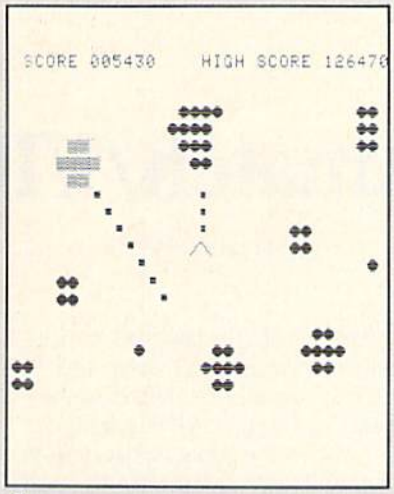
- ■ Fast, quiet 40-column Alphacom VP42 printer, which includes the full Commodore graphics set.
- ■ 5 great software programs. A real 40-column *Word Processor*, graphics *Sketch Pad*, beautiful *Kaleidoscope* program, useful *Record Keeper*, and a general purpose *Screen Printer*.
- ■ Compute Magazine's *First Book of VIC*, full of fascinating step-by-step VIC 20 computer adventures.
- ■ All the cables, printer paper, and easy-to-follow instructions you need to begin enjoying your Alphacom VP42 as soon as you open the box.

Get the complete Alphacom Printer Set for just $209.95 (suggested list). Call toll free for the name of your nearest dealer: 800/538-7047 anywhere in the USA (except California: 408/559-8000). If there is no convenient dealer, you may order direct from Alphacom. Same day shipment with MasterCard, VISA, or American Express card.

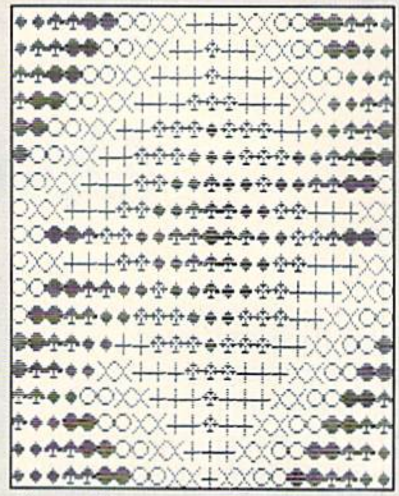
Alphacom™ printer set my VIC 20™..."

```
     History Book Report
       by Joe Grimes

Book  : I, Claudius
Author: Robert Graves

   Back in the old days of
Empire the cruel Emperor Ca
murdered and a new leader,
was chosen by the Praetoria
(The Romans had rather odd
order in those days.) The P
Guard thougt they had chose
simpleton and weakling who
easily manipulated to the G
purposes.
      Much to everyone's su
Claudius lasted for over a
and turned out to be a wise
moderate leader. Robert Gra
I, Claudius describes the E
life before he ascended the
```

Write and print my letters, homework—just about anything. With *Word Processor* software that comes with the printer.



Create my own custom computer games. The printer set includes the new *First Book of VIC:* it's full of great game ideas.



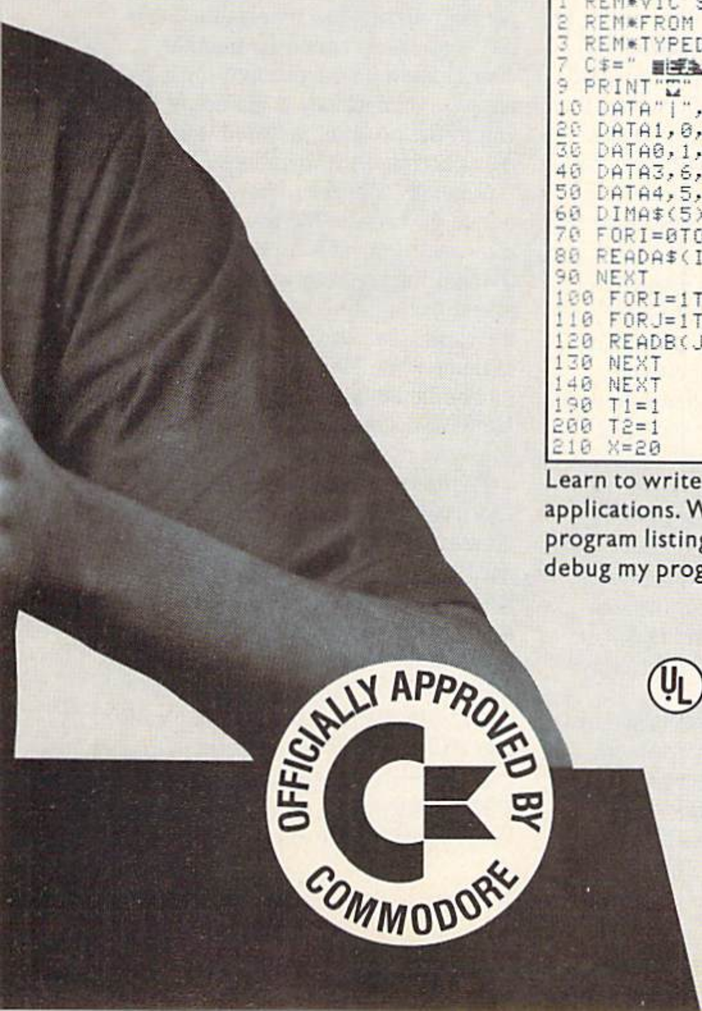
Show off my computer graphics creations. Software includes *Kaleidoscope* and *Sketch Pad* graphics programs.

```
1 REM*VIC SQUIGGLE*
2 REM*FROM VIC 1001 USER'S
3 REM*TYPED AND DEBUGGED BY
7 C$="         "
9 PRINT"  "
10 DATA"|","—","J","L","┌",
20 DATA1,0,5,6
30 DATA0,1,4,3
40 DATA3,6,2,0
50 DATA4,5,0,2
60 DIMA$(5),B(5,5)
70 FORI=0TO5
80 READA$(I)
90 NEXT
100 FORI=1TO4
110 FORJ=1TO4
120 READB(J,I)
130 NEXT
140 NEXT
190 T1=1
200 T2=1
210 X=20
```

Learn to write my own VIC 20 applications. With hard copy program listings that help me debug my programs.

```
Recipe 334

CHEESE BERRY PIE

Ingredients

    2 Pks (3 oz. ea.) cream
    1/3 cup powdered sugar
    1/3 cup sour cream
    2 tsp grated orange pee
    Baked 9-inch pie shell
    2 - 3 cups whole fresh
    strawberries/raspberrie
    1/2 cup strawberry/rasp
    preserves, sieved

Procedure

  Soften cheese.  Beat in s
sour cream and orange peel.
in pie shell.  Top with ber
to glaze.  Chill.  If you l
```

Keep the family's favorite recipes on a VIC 20 cassette. Now Mom can't pretend that she lost the recipe for cheeseberry pie.



UL

Alphacom™

2323 South Bascom Avenue
Campbell, CA 95008

Alphamatch/VIC 20

by Barbara J. Moody

Although Alphamatch *was designed for classroom use, we think it is also an excellent resource for young children at home. To get both graphics and sound effects you need Commodore's Super Expander cartridge. If you don't mind leaving out the sound effects, you only need a 3K memory expander.*

After sixteen years as an educational supervisor and consultant, I have returned to the busy peace and joy of a kindergarten classroom. Having become addicted to my VIC 20 about a year ago, I was eager to put it to work for my young students. I purchased the preschool programs from my local Commodore dealer, sent for program descriptions from publishers who advertise learning programs for children, and reached the conclusion that programs suitable for my youngsters are scarce indeed! Here and there, now and then, I found a gem that really worked in my classroom, but more usually the programs I found were too complex and/or too advanced in concepts for use by five and six year-olds.

Disappointing? Yes. But what better reason and incentive to begin my own programming! *ALPHAMATCH* is one of the programs I developed for use in my classroom. The children have used it eagerly and successfully this year. It is often chosen over other available programs.

ALPHAMATCH is designed to help young children learn the upper and lower case forms of the letters in the alphabet and to practice matching them. After some initial guidance from a parent or teacher, most five to seven year-olds should be able to use the program on their own. The interest level appears to be high, even for older children. (I have even found some of my fellow educators enjoying it!)

The program consists of two parts. In the first, the child must match upper case letters by looking at the target letter displayed on the screen, finding that letter on the keyboard, and pressing the letter key. In the second part, the target letter is given in its lower case form. The child must find and press the matching upper case letter key.

Each correct response is rewarded with the appearance of part of a toy on the screen (red wagon, jack-in-the-box, truck, boat, train). Several correct responses are required to complete each of these pictures. When an incorrect response is made, there is no reinforcement of it. The cursor simply returns and flashes on that response until the child locates and presses the correct matching key. Thus, while it is possible to make an incorrect match, the computer will wait until a correct match is made before either rewarding or reinforcing that learning.

Screen and border colors change as the program moves from part one to part two. They change again as the program ends and a "good work" message is flashed among the pictured toys.

Sound is used as an optional reinforcer. If the classroom activity in general requires quiet—or if Mom needs a little peace at home—the program will run nicely and satisfyingly with the T.V. sound turned down. It is more exciting, however, with the sound turned up.

Each correct response is rewarded with a pleasant, light tone as the letter is printed and the graphic reward appears. An incorrect response is indicated by a heavier, buzzing low tone as the incorrect response is erased for another choice. Each time a pictured toy is completed, a mini-melody is played. At the end of the program, a "good work" banner is accompanied by a fanfare.

Generally, I prefer to have the speaker turned up because the sound signals allow me to monitor a child's progress through the program while I continue to attend to the needs of other children in the classroom. This should apply also at home when Mom needs to carry on routine home tasks while Johnny or Jenny works at learning, the computer play way.

The frequency and pattern of light and heavy tones tells me how many guesses and how much difficulty the child is having with the task of matching letters. The frequency of mini-melodies signals the child's pacing through the program, and the fanfare tells me when it is time to set up another game for the child or to give another child a turn to play *ALPHAMATCH.*

Because the sound was typed as PRINT statements, as allowed by the Super Expander, it will not work with ordinary expansion. In any other configuration, these statements will merely be printed, causing considerable confusion in the screen display!

With or without sound, *ALPHAMATCH* provides an interesting and satisfying experience with Computer Assisted Education—the play way!

Program Notes:

ALPHAMATCH requires a VIC 20 with a 3K Super Expander. The sound element can be eliminated and the program will run on an ordinarily expanded VIC 20 by making the changes stated below. This is a long program, however, and it will NOT fit into an unexpanded VIC 20!

To eliminate sound commands:

Delete lines 965,1225,1525,1845,3055,8025, and 9000

Delete the final PRINT statement in lines 3085 and 3090

The program works with two subroutines. The first, at line 8000, provides for input and processing of the child's response for each target letter. It also provides the buzzing tone for error and erase of incorrect responses. The second, line 9000, provides the light tone for each correct response.

Instructions for playing the game are provided in lines 290 through 690. Lines 740 through 780 provide the target letter and response area on the screen. Line 800 sets the display to upper/lower case mode, while line 4000 returns it to the upper case mode and clears the screen for playing the game again.

The main body of the program is contained in lines 810 through 3060, which provide the target letters and reward graphics. All pictured toys are POKEd to the screen. C

To save eye-hand fatigue, you may obtain a copy of this program by sending $3.50 (To cover duplicating and postage), a blank cassette and a mailer to:
Barbara J. Moody
1211 Sabattus Street #17
Lewiston, Maine 04240

```
50 REM ALPHAMATCH
51 REM BARBARA J.MOODY
52 PRINT"A BEEJAEM PROGRAM":PRINT"FOR
   LITTLE LEARNERS"
53 PRINT"PRESS SPACE BAR":PRINT"TO
   CONTINUE ---"
54 GET A$:IF A$=""THEN54
60 PRINT""
70 PRINT""
80 PRINT""
90 PRINT""
100 PRINT""
110 PRINT""
120 PRINT""
130 PRINT" "
140 PRINT" "
150 PRINT" "
160 PRINT" "
170 PRINT" "
180 GETR$:IFR$=""THEN180
290 PRINT"DIRECTIONS:"
300 PRINT""
310 PRINT" FIRST, THE CHILD"
320 PRINT" WILL BE SHOWN AN"
330 PRINT"UPPER CASE(CAPITAL)"
340 PRINT"LETTER.THE CHILD MUST"
350 PRINT"PRESS THAT LETTER ON"
```

```
 360 PRINT"THE KEYBOARD.THIS"
 370 PRINT"WILL SHOW THE LOWER"
 380 PRINT"CASE OF THE TARGET"
 390 PRINT"LETTER. THE CHILD"
 400 PRINT"SHOULD STUDY THESE"
 410 PRINT"MATCHED LETTERS THEN"
 420 PRINT" PRESS RETURN FOR"
 430 PRINT"THE NEXT LETTER."
 460 GETR$:IFR$=""THEN460
 470 PRINT""
 480 PRINT" WHEN ALL OF THE"
 490 PRINT" UPPER CASE AND"
 500 PRINT"LOWER CASE LETTERS"
 510 PRINT"HAVE BEEN MATCHED,"
 520 PRINT"THE CHILD WILL BE"
 530 PRINT"SHOWN A LOWER CASE"
 540 PRINT"LETTER AND MUST"
 550 PRINT"TYPE THE MATCHING"
 560 PRINT"UPPER CASE LETTER."
 580 GETR$:IFR$=""THEN580
 630 PRINT"EACH CORRECT MATCH"
 640 PRINT"WILL BE REWARDED"
 650 PRINT"WITH A PART OF A"
 660 PRINT"PICTURED TOY. EACH"
 670 PRINT"INCORRECT MATCH"
 680 PRINT"WILL BE ERASED TO"
 690 PRINT"ALLOW ANOTHER TRY."
 730 GETR$:IFR$=""THEN730
 735 POKE36879,26
 740 PRINT""
 750 PRINT""
 760 PRINT""
 770 PRINT""
 780 PRINT""
 800 PRINTCHR$(14)
 810 PRINT"":X=67:GOSUB8000
 835 PRINT" "
 840 POKE7738,(96+128):POKE(7738+30720),2:POKE7743,64:POKE(7743+30720),0
 850 PRINT"":X=70:GOSUB8000
 870 POKE7744,115:POKE(7744+30720),0:POKE7739,(96+128):POKE(7739+30720),2
 880 PRINT"":X=81:GOSUB8000
 900 POKE7740,(96+128):POKE(7740+30720),2:POKE7741,(96+128):POKE(7741+30720),2
 910 PRINT"":X=77:GOSUB8000
 930 POKE7742,(96+128):POKE(7742+30720),2:POKE7760,102:POKE(7760+30720),0
 940 PRINT"":X=65:GOSUB8000
 960 POKE7764,102:POKE(7764+30720),0:POKE 7828,(114+128):POKE(7828+30720),0
 965 GOSUB 9000
 970 PRINT"":X=87:GOSUB8000
 990 POKE7849,107:POKE(7849+30720),0:POKE7851,115:POKE(7851+30720),0
1000 PRINT"":X=73:GOSUB8000
1020 POKE7850,127:POKE(7850+30720),4:POKE7872,(127+128):POKE(7872+30720),4
1030 PRINT"":X=75:GOSUB8000
```

```
1050 POKE 7894,127:POKE(7894+30720),4:POKE7871,105:POKE(7871+30720),7
1060 PRINT"████████I":X=84:GOSUB8000
1080 POKE 7873,105:POKE(7873+30720),7:POKE7893,105:POKE(7893+30720),7
1090 PRINT"████████♦":X=90:GOSUB8000
1110 POKE7895,105:POKE(7895+30720),7:POKE 7915,(96+128):POKE(7915+30720),5
1120 PRINT"████████Γ":X=79:GOSUB8000
1130 POKE 7916,(96+128):POKE(7916+30720),5:POKE 7917,(96+128):POKE(7917+30720),5
1140 PRINT"████████|":X=66:GOSUB8000
1160 POKE7937,(96+128):POKE(7937+30720),5:POKE7938,(96+128):POKE(7938+30720),5
1170 PRINT"████████Γ":X=68:GOSUB8000
1190 POKE7939,(96+128):POKE(7939+30720),5:POKE7959,(96+128):POKE(7959+30720),5
1200 PRINT"████████|":X=89:GOSUB8000
1220 POKE7960,(96+128):POKE(7960+30720),5:POKE7961,(96+128):POKE(7961+30720),5
1225 GOSUB 9000
1230 PRINT"████████I":X=71:GOSUB8000
1250 POKE7923,(96+128):POKE(7923+30720),2:POKE7924,(96+128):POKE(7924+30720),2
1260 PRINT"████████L":X=76:GOSUB8000
1280 POKE7925,(96+128):POKE(7925+30720),2:POKE7926,(96+128):POKE(7926+30720),2
1290 PRINT"████████L":X=82:GOSUB8000
1310 POKE7927,111:POKE(7927+30720),2:POKE 7949,118:POKE(7949+30720),2
1320 PRINT"████████Γ":X=69:GOSUB8000
1340 POKE7950,108:POKE(7950+30720),7:POKE7945,(96+128):POKE(7945+30720),2
1350 PRINT"████████X":X=86:GOSUB8000
1370 POKE7946,107:POKE(7946+30720),6:POKE7947,115:POKE(7947+30720),6
1380 PRINT"████████/":X=78:GOSUB8000
1400 POKE7948,(96+128):POKE(7948+30720),2:POKE7967,(96+128):POKE(7967+30720),2
1410 PRINT"████████|":X=72:GOSUB8000
1430 POKE7968,(96+128):POKE(7968+30720),2:POKE7969,(96+128):POKE(7969+30720),2
1440 PRINT"████████/":X=85:GOSUB8000
1460 POKE7970,(96+128):POKE(7970+30720),2:POKE7971,(96+128):POKE(7971+30720),2
1470 PRINT"████████\":X=74:GOSUB8000
1490 POKE7972,(96+128):POKE(7972+30720),2:POKE7989,102:POKE(7989+30720),0
1500 PRINT"████████♥":X=83:GOSUB8000
1520 POKE7992,102:POKE(7992+30720),0:POKE 7994,102:POKE(7994+30720),0
1525 GOSUB 9000
1530 POKE36879,30
1550 PRINT"████████W":X=87:GOSUB8000
1570 POKE 8026,105:POKE(8026+30720),2:POKE 8046,(96+128):POKE(8046+30720),6
1580 PRINT"████████P":X=80:GOSUB8000
1600 POKE8047,127:POKE(8047+30720),6:POKE8048,127:POKE(8048+30720),6
1610 PRINT"████████D":X=68:GOSUB8000
1630 POKE8049,127:POKE(8049+30720),6:POKE 8050,(96+128):POKE(8050+30720),6
1640 PRINT"████████F":X=70:GOSUB8000
1660 POKE8067,(96+128):POKE (8067+30720),6:POKE 8068,(96+128):POKE(8068+30720),6
1670 PRINT"████████Y":X=89:GOSUB8000
1690 POKE8069,(96+128):POKE(8069+30720),6: POKE8070,(96+128):POKE(8070+30720),6
1700 PRINT"████████S":X=83:GOSUB8000
1720 POKE8071,(96+128):POKE(8071+30720),6:POKE8072,(96+128):POKE(8072+30720),6
1730 PRINT"████████K":X=75:GOSUB8000
1750 POKE8073,(96+128):POKE(8073+30720),6:POKE8095,(96+128):POKE(8095+30720),2
1760 PRINT"████████G":X=71:GOSUB8000
1780 POKE8089,(96+128):POKE(8089+30720),2:POKE8090,(96+128):POKE(8090+30720),2
```

```
1790 PRINT"A":X=65:GOSUB8000
1810 POKE8091,(96+128):POKE(8091+30720),2:POKE8092,(96+128):POKE(8092+30720),2
1820 PRINT"U":X=85:GOSUB8000
1840 POKE8093,(96+128):POKE(8093+30720),2:POKE8094,(96+128):POKE(8094+30720),2
1845 GOSUB9000
1850 PRINT"N":X=78:GOSUB8000
1870 POKE8151,102:POKE(8151+30720),0:POKE8148,102:POKE(8148+30720),0
1880 PRINT"Z":X=90:GOSUB8000
1890 POKE8125,122:POKE(8125+30720),0:POKE8083,120:POKE(8083+30720),2
1900 PRINT"B":X=66:GOSUB8000
1910 POKE8108,123:POKE(8108+30720),7:POKE8130,126:POKE(8130+30720),7
1920 PRINT"N":X=78:GOSUB8000
1930 POKE8143,102:POKE(8143+30720),0:POKE8146,102:POKE(8146+30720),0
1940 PRINT"Q":X=81:GOSUB8000
1950 POKE8084,(96+128):POKE(8084+30720),2:POKE8085,(96+128):POKE(8085+30720),2
1960 PRINT"C":X=67:GOSUB8000
1970 POKE8104,(96+128):POKE(8104+30720),2:POKE8105,(96+128):POKE(8105+30720),2
1980 PRINT"X":X=88:GOSUB8000
1990 POKE8106,(96+128):POKE(8106+30720),2:POKE8107,(96+128):POKE(8107+30720),2
2000 PRINT"E":X=69:GOSUB8000
2010 POKE8126,(96+128):POKE(8126+30720),2:POKE8127,(96+128):POKE(8127+30720),2
2020 PRINT"V":X=86:GOSUB8000
2030 POKE8128,(96+128):POKE(8128+30720),2:POKE8129,(96+128):POKE(8129+30720),2
2040 PRINT"I":X=73:GOSUB8000
2050 POKE8077,104:POKE(8077+30720),0:POKE8078,104:POKE(8078+30720),0
2060 PRINT"T":X=84:GOSUB8000
2070 POKE8079,104:POKE(8079+30720),0:POKE8080,104:POKE(8080+30720),0
2080 PRINT"R":X=82:GOSUB8000
2090 POKE8099,105:POKE(8099+30720),5:POKE8100,105:POKE(8100+30720),5
3000 PRINT"J":X=74:GOSUB8000
3010 POKE8101,105:POKE(8101+30720),5:POKE8102,105:POKE(8102+30720),5
3020 PRINT"M":X=77:GOSUB8000
3030 POKE8121,105:POKE(8121+30720),5:POKE8122,105:POKE(8122+30720),5
3040 PRINT"O":X=79:GOSUB8000
3050 POKE8123,105:POKE(8123+30720),5:POKE8124,105:POKE(8124+30720),5
3055 GOSUB 9000
3060 PRINT"L":X=76:GOSUB8000
3065 POKE36879,29
3080 PRINT"***************":PRINT"*             *":PRINT"* GOOD WORK! *"
3085 PRINT"*             *":PRINT"***************":PRINT"T6V803FFA"
3090 PRINT"GOODBY!":PRINT"T6V803CF"
4000 PRINT""CHR$(142):END
8000 A$="": PRINT"":INPUTA$
8010 IF LEFT$(A$,1)<>CHR$(X)THEN PRINT"T4V801C":GOTO 8000
8020 IF LEFT$(A$,1)=CHR$(X)THENPRINT"";LEFT$(A$,1)
8025 PRINT"T402V8F"
8030 FORT=1TO1000:NEXTT
8040 PRINT" ":RETURN
9000 PRINT"T5V803AFA":RETURN
```

Joyrite

by Mike & Annette Hinshaw

This program makes it even easier to draw pictures on your screen with the Commodore Super Expander cartridge.

Have you ever wanted to draw pictures on your computer screen? With the VIC 20 Super Expander cartridge, you can do it with a joystick. A short program demonstrating the RJOY(X) function (one of several graphics statements and functions the cartridge adds) permits drawing lines on the screen by moving the joystick. The program is fun, but limited. It is as if the artist drawing on the screen cannot lift his pencil from the paper, and cannot erase if he makes a mistake, but must start over with a new sheet of paper (blank screen). If he is careless and lets the line go off the page, the program blows up and he has to start over again.

We liked the program so well we set out to fix those inconveniences. The new program, which we call *Joyrite*, lets you draw on the screen and clear the screen as before, but now you can erase if you make a mistake, or "pick up your pencil" when you want to. It's easy to draw complex figures on the screen. We even drew a simple electronic circuit that looked like it had been professionally drafted. It takes practice to move the dot on the screen so it comes out like you want it to, but since you can erase anything that doesn't look right, it doesn't matter.

The RJOY(X) function is the core of the program. It works like a binary counter, with one switch, or bit, for each of the four directions. For example, if you move the joystick up, RJOY(X) equals four.

To make the screen coordinates come out right with the value of the RJOY(0), X and Y are one third of the number used by VIC in the POINT statement to put a dot on the screen. This program uses relational operators to test which of the four joystick positions is "on". The logical AND in the relational statement checks J against the possible joystick values. For instance, if J is equal to 4 (up), line 140 will look to the computer like X=X+(−1)−(0) and line 150 will read Y=Y+(0)−(0).

The POINT statement is the way the dot moves on the screen, and the lines are formed by the dots. The erasing or "not writing" sequence (lines 1000 to 1050) just changes the color of the dot so it's the same as the screen, or background, color. Even though you can't see it, the dot moves the same way it did when you drew the line. If you get lost moving the invisible dot, just hit the fire button to make it change color so you can find it, and then change it back again.

```
1 SCNCLR
2 PRINT"THIS PROG WILL DRAW   IN ACCORDANCE WITH
  THE JOYSTICK"
3 PRINT"HIT THE FIRE BUTTON TO LEAVE NO TRAIL,SPACE
  TO CLEAR THE SCREEN"
4 FORT=1TO5000:NEXT
10 GRAPHIC3
20 COLOR0,7,1,1
125 X=170:Y=180
130 J=RJOY(0)
135 REGION1
140 X=X+((JAND4)=4)-((JAND8)=8)
150 Y=Y+((JAND1)=1)-((JAND2)=2)
152 IFX<0THENX=INT(1023/3):Y=Y-3
153 IFX>INT(1023/3)THENX=0:Y=Y+3
154 IFY<0THENY=INT(1023/3):X=X-3
155 IFY>INT(1023/3)THENY=0:X=X+3
160 POINT2,X*3,Y*3:IFJ=128THENGOTO1000
165 IFPEEK(197)=32THEN:SCNCLR
170 GOTO130
1000 POINT0,X*3,Y*3
1010 J=RJOY(0)
1020 X=X+((JAND4)=4)-((JAND8)=8)
1030 Y=Y+((JAND1)=1)-((JAND2)=2)
1032 IFX<0THENX=INT(1023/3):Y=Y-3
1033 IFX>INT(1023/3)THENX=0:Y=Y+3
1034 IFY<0THENY=INT(1023/3):X=X-3
1035 IFY>INT(1023/3)THENY=0:X=X+3
1040 POINT0,X*3,Y*3:IFJ=128THEN130
1045 IFPEEK(197)=32THEN:SCNCLR
1050 GOTO1010
```

To use the graphics statements added to the VIC by the Super Expander, the program must be initialized, as in line 10, with GRAPHIC plus a mode numbered one, two, or three. This program will work with any of the modes, but each works a little differently. GRAPHIC 1 is multi-color, but not high resolution, so your line is thicker. GRAPHIC 2 will give you high resolution, but not as many colors. GRAPHIC 3, which we like best, permits both high resolution and multi-color.

Another thing that needs to be initialized is color, which we do in line 20 with COLOR 0,7,1,1. These four numbers can be changed. They set, respectively, screen, border, character and auxiliary color. In this program, we specify white (1) for the character, black (0) for the screen and yellow (7) for the border. This combination works well on a black and white screen also. It's fun to experiment with different colors in this program, but remember that some combinations don't show your drawing very clearly.

One color setting you should not change is the one used with the POINT statements. In line 160, the 2 specifies the same color as the character color. In lines 1000 and 1040, the 0 tells the computer to draw with the same color as the screen. If you change it, you won't be able to erase.

We added the routines in lines 152-155 and 1032-1035 so the program wouldn't crash when you move the dot off the screen. This is especially helpful when you can't see the dot. When the dot hits a screen edge, it "wraps around" and appears on the opposite edge. If you draw a single straight line, the wraparound line will not meet the line you started, but will be one or two positions off, depending on the value of X and Y at the time the line comes to an edge.

In lines 165 and 1045, the computer PEEKs at the address where the name of the key being pressed on the keyboard is stored. A 32 means the spacebar. Please note that the colon after the THEN in the IF...THEN statements in these lines is not a misprint. When the Super Expander statements are used with the IF...THEN statement, they won't work unless this colon is present.

This program is a good demonstrator to show your friends how much fun it can be playing with a computer. **C**

Reinventing the Wheel in PET BASIC

by Dean S. Rossa

This interesting little program creates the illusion of a spinning wheel. It will run on the PET, CBM and Commodore 64 computers.

To create the illusion of a spinning circle on the PET is to reinvent the wheel. Two points do not determine a line in standard PET BASIC and a point rotating equidistant from a fixed point cannot define a circle. Lines and shapes displayed by PRINT statements are defined by the number of characters and cursor movements required to flesh them in on the screen. This can make the drawing of separate circle segments pretty tedious.

This program constructs a wheel by defining six strings, each of which represents a pie-shaped wedge of the circle. It also defines the four diameters that can be easily defined by PET graphics: a horizontal, a vertical, and two diagonals. To make the wheel appear to move, we print a string and then overprint it with spaces. A fixed point is defined; a sector in a fixed position relative to the point is printed and is then blanked; then other sectors and lines in differing positions relative to the point are printed and blanked in a clockwise rotation.

The program itself keeps score, prompts players, and tidies the screen in a straightforward game, and in a roundabout way takes a turn demonstrating the potentials of PET/CBM graphics. **C**

```
10 PRINT CHR$(142) "┐■■■                ■SPIN THE WHEEL■"
20 PRINT:PRINT"            A GAME FOR TWO PLAYERS"
30 PRINT:PRINT"■GOAL■ IS TO SCORE CLOSEST TO 5000 POINTS"
40 PRINT "         WITHOUT EXCEEDING 5000 POINTS" :PRINT:PRINT
50 PRINT"SHORT SPIN WILL MOVE WHEEL TO ONE"
60 PRINT" OF NEXT 4  NUMBERS IN SEQUENCE ":PRINT
70 PRINT"LONG SPIN WILL MOVE WHEEL TO NUMBER"
80 PRINT"5 TO 8 AWAY IN CURRENT SEQUENCE"  :PRINT
90 PRINT"SEQUENCE IS: 1,5,10,20,50,100,500,1000":PRINT
100 PRINT"SETTING UP...........PLEASE WAIT":PRINT:PRINT
110 REM ****** BUILD WHEEL; "┐","■","▌","▌"ARE CURSOR UP,DOWN,LEFT,RIGHT *****
120 Q=59464: L$="▌":   H$="▒■■■■■■■■■■■■▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌"
130 J=1: FOR I=1TO7: FOR K=1TOJ: B$(1)=B$(1)+" ":B$(4) =B$(4) +" "
140 NEXT K:    B$(1) =B$(1)+"┐"  :B$(4)=B$(4) +"■"
150 FOR K=1 TO J+1:B$(1)=B$(1)+L$ :B$(4) =B$(4)+L$ :NEXT K:J=J+2: NEXT I
160 J=13:  FORI=1TO 7 : FOR K=1 TO J
170 B$(2)=B$(2)+" " :B$(3)=B$(3)+" ":B$(5)=B$(5)+" ":B$(6)=B$(6) +" " :NEXT K
180 B$(2)=B$(2)+"┐":B$(3)=B$(3)+"■" :B$(5)=B$(5)+"■":B$(6)=B$(6)+"┐"
190 FORK=1TOJ-1:B$(2)=B$(2)+L$:B$(3)=B$(3)+L$:B$(5)=B$(5)+L$:B$(6)=B$(6)+L$
200 NEXTK:J=J-2:NEXTI:B$(5)="▌▌▌▌▌▌▌▌▌▌▌▌▌▌"+B$(5):B$(6)="▌▌▌▌▌▌▌▌▌▌▌▌▌▌"+B$(6)
210 FOR I=1 TO6:L$(I) ="■" +  B$(I)  : NEXT I
220 N$=CHR$(225):FORI=1 TO7:VD$=VD$+N$+"■▌":VU$=VU$+N$+"┐▌":     NEXT
230 S$=CHR$(191):FORI=1TO7:S1$=S1$+S$+"■" :S2$=S2$+S$+"┐" :NEXT:
240 A$=CHR$(192):FOR I=1TO13:H1$=H1$+A$:NEXT
250 INPUT"ENTER NAME OF PLAYER 1";N$(1):N$(1)=LEFT$(N$(1),8)
260 INPUT"ENTER NAME OF PLAYER 2";N$(2):N$(2)=LEFT$(N$(2),8) : PRINT"┐"
```

```
270 DATA"   1","   5","  10","  20","  50"," 100"," 500",1000
280 FORI=1TO8:READR$(I):NEXTI:W$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒":K=0:GOTO 370
290 FOR P=1TO2 : PRINT"▒▒"N$(P)"▒  ENTER ▒L▒ FOR A LONG SPIN,"
300 PRINT  "          ▒P▒ TO PASS, OTHER KEYS FOR SHORT" :POKE158,0
310 GET S$:IFS$="" THEN 310
320 PRINT"▒                                          "
330 PRINT"▒▒                                         "
340 IF S$="P" THEN 520
350 R=INT(RND(0)*4)+1:  IF S$="L"THEN R=R+4
360 FOR I=1 TO R: K=K+1:IFK>8THEN K=1
370 POKE59467,16:POKE59466,3:  PRINTH$L$(1):POKEQ,220: POKEQ,225:POKEQ,250
380 PRINTH$L$(2)            :POKEQ,250: POKEQ,255: POKEQ,250 :REM PRINT WHEEL
390 PRINTH$S2$:PRINT W$R$(K) :POKEQ,250: POKEQ,255: POKEQ,250 :REM POKE SOUNDS
400 PRINTH$L$(3)            :POKEQ,250: POKEQ,255: POKEQ,250
410 PRINTH$"▒"H1$ H$L$(4)    :POKEQ,250: POKEQ,255: POKEQ,250
420 PRINT H$L$(5)  H$VD$     :POKEQ,250: POKEQ,255: POKEQ,250
430 PRINT  H$ L$(6)          :POKEQ,250: POKEQ,255: POKEQ,250
440 PRINT  H$"███████████████" "▒"H1$ H$ VU$ : POKE Q,250: POKE Q,255 : POKEQ,0
450 PRINTH$"▒" S1$"▒"  "████████████████"S2$:PRINT"▒▒▒▒▒▒▒▒▒▒▒"
460 PRINTH$S2$"████████████████""▒▒" S1$
```

```
470 T=T+1: IF T=1 THEN GOTO 290
480 IFI=R THEN 500
490 PRINTH$ B$(6) H$B$(1) H$ B$(2) H$ B$(3) H$ B$(5) H$ B$(4):NEXT I
500 T(P)=T(P)+VAL(R$(K)):IFT(P) >5000THEN530
510 PRINT"SQQQ"N$(1)":"T(1)TAB(22)N$(2)":"T(2)
520 NEXT P : GOTO290
530 PRINT"SQQQ"N$(1)":"T(1)TAB(22)N$(2)":"T(2)"SQQQQQQQQQQQQQQQQQQQQ"N$(P)" LOSES
540 POKE 59468,12:POKE 59467,16 :POKE59466,15
550 FOR Z=0TO100STEP5:POKE59464,Z:FOR W=1TO50:NEXT W:NEXTZ:POKE 59464,0
560 PRINT:PRINT"ENTER Y TO PLAY AGAIN"
570 GET K$:IF K$=""THEN 570
580 IFK$<>"Y"THEN STOP
590 PRINT"                           SQQQQQQQQQQQQQQQQQQQ                 "W$"     "
600 T(1)=0:T(2)=0:PRINT"SQQQ                                             ":GOTO290
65535 REM ********** DEAN S. ROSSA ***** JONESBORO, ARK. ***** 1983**********
```

STAR: An Astronomy Program for the VIC 20

by Alex Bakman

There aren't too many scientific programs for Commodore's VIC 20, especially in astronomy. Well, astronomy is my hobby and I badly needed a program that would save me time on calculations. I knew, for instance, a program that would calculate azimuth and altitude would be very useful.

So, without further delay, I created one. If you're familiar with astronomy you would know that you need two coordinates to focus a telescope on the target of your observation. The first coordinate is the horizontal rotation, called azimuth. It runs from 0 to 360 degrees in a complete circle. The second coordinate is the elevation angle. It is known as altitude, and is between 0 and 90 degrees.

The output of this program will give you the two coordinates you need. So, for instance, if you have an azimuth equal to 90 and an altitude of 45, you would then rotate the telescope to the 90 degree mark and move it up 45 degrees.

The input of the program is as follows:

1. Your local time of observation
2. Date and month
3. Longitude and latitude of your location
4. Right ascension and declination

The program takes this information and starts off by converting your local time to universal time. Hour angle is calculated immediately afterwards.

Most computers operate in radians, so if your input is in degrees, you obviously have to convert that input to radians. The formula for the conversion is:

$$\text{Rad} = \text{Deg} * \text{pi}/180$$

Now the program comes to the final stage and calculates the azimuth and altitude. As it does so, it displays the results on the screen.

So type the program into your VIC and save yourself from boring calculations! C

```
5 REM STAR POSITION
10 X$="*************************":PRINT"
   "X$
11 PRINT"CALCULATION OF AZIMUTH AND
   ALTITUDE":PRINTX$
13 FORA=1TO10↑4:NEXT
15 K=6.62240808
20 PRINT""X$"TIME OF OBSERVATION(IN
   HOURS)":INPUTT:IFT>24ORT<0THEN20
21 PRINT"MONTH":INPUTA$
22 IFA$="JAN"THENA=0
23 IFA$="FEB"THENA=31
24 IFA$="MAR"THENA=59
25 IFA$="APR"THENA=90
26 IFA$="MAY"THENA=120
27 IFA$="JUN"THENA=151
28 IFA$="JUL"THENA=181
29 IFA$="AUG"THENA=212
30 IFA$="SEP"THENA=243
31 IFA$="OCT"THENA=273
32 IFA$="NOV"THENA=304
33 IFA$="DEC"THENA=334
34 PRINT"DATE":INPUTB:IFB>31ORB<0THEN34
   :N=A+B
36 IFN>366ORN<0THEN21
37 PRINT"*****LONGITUDE*********":INPUTLO
38 IFLO>165ORLO<65THEN37
39 IFLO>65ANDLO<90THENR=5
40 IFLO>90ANDLO<105THENR=6
41 IFLO>105ANDLO<120THENR=7
42 IFLO>120ANDLO<135THENR=8
43 IFLO>135ANDLO<150THENR=9
44 IFLO>150ANDLO<165THENR=10
53 PRINT"*****LATITUDE*********":INPUTLA
56 PRINT"***RIGHT ASCENSION***":INPUTRA
```

```
 58 PRINT"***DECLINATION******":INPUTD
 60 UT=T+R
 70 S=K+.0657*N+1.0027*UT-(LO/15)
 80 IFS>24THENS=S-24
 90 REM END U.T.
120 HA=(S-RA)*15
127 REM CONV. TO RADIANS
130 HA=HA*π/180
140 D=D*π/180
150 LA=LA*π/180
160 REM END CONVERSION
175 REM ALTITUDE
180 X=SIN(D)*SIN(LA)+COS(D)*COS(LA)*COS(HA)
185 AL=ATN(X/SQR(-X*X+1))
190 REM END ALTITUDE
200 REM
205 REM AZIMUTH
210 Y=(SIN(D)-SIN(LA)*SIN(AL))/(COS(LA)*COS(AL))
215 AZ=-ATN(Y/SQR(-Y*Y+1))+π/2
220 REM END AZIMUTH
225 REM
230 REM RADIANS CONV.
231 REM TO DEGREES
235 AL=AL*(180/π)
240 AZ=AZ*(180/π)
245 REM END CONVERSION
250 PRINT"ALTITUDE= ";AL
270 IFHA=HATHEN AZ=360-AZ
280 PRINT"AZIMUTH IS";AZ
300 END
```

FOR...NEXT Sound Effects

by Jim Lockridge

This program provides a series of interesting sound effects demonstrating the use of FOR...NEXT loops. You'll hear everything from racing cars to laser blasts—even what seems to be a babbling alien creature. It begins with a series of anti-matter torp discharges and ends with a World War I biplane flying off into the sunset.

```
1 GOSUB20000
5 POKE36878,15:S=36876:S1=36875:S2=36874
10 PRINT"■■1■":REM**SOUND1**
15 FORTT=135TO241STEP1.5
20 POKES,TT:POKES1,TT
99 POKES,0:POKES1,0
100 NEXTTT
110 PRINT"■■2■":REM**SOUND2**
120 FORTT=1TO3STEP1
130 SS=241:POKES,SS:POKES,0:SS=SS-1
    :POKES,SS
140 SS1=135:POKES1,SS1:POKES1,
    0:SS1=SS1+1:POKES1,SS1
299 POKES1,0:POKES,0:NEXTTT
300 PRINT"■■3■":REM**SOUND3**
310 FORTT=1TO6STEP1
320 FORT=241TO135STEP-1:POKES1,T:NEXTT:
    NEXTTT
380 PRINT"■■4■":REM**SOUND4**
390 FORT=1TO3STEP1
400 FORTT=241TO135STEP-1:POKES,TT:POKES,
    0:NEXTTT
410 FORTT=241TO135STEP-1:POKES1,TT
    :POKES1,0:NEXTTT
420 FORTT=241TO135STEP-1:POKES2,TT
    :POKES2,0:NEXTTT
430 NEXTT
440 PRINT"■■5■":REM**SOUND5**
450 FORTT=135TO241STEP.3:POKES,TT:
    POKES1,TT:POKES2,TT:POKES,0:POKES1,0
    :POKES2,0: NEXTTT
490 PRINT"■■6■":REM**SOUND6**
500 FORY=1TO2
510 FORYY=135TO212STEP.3:POKES2,YY:
    POKES2,0:NEXTYY
550 NEXTY
590 PRINT"■■7■":REM**SOUND7**
600 FORTT=241TO135STEP-1:POKES,TT:POKES1
    ,TT:POKES2,TT:POKES,0:POKES1,0
    :POKES2,0: NEXTTT
700 PRINT"■■8■":REM**SOUND8**
710 FORV=15TO0STEP-.1
720 POKE36878,V
730 POKE36877,212:POKE36877,0:NEXTV
900 FORPP=1TO4000:NEXTPP
10000 RUN
20000 PRINT"■■":POKE36879,25
20010 PRINT"■■■■■■■■■■■■FANTASIE■ [C]"
20020 PRINT"■■■■■■■■■ BY JIM LOCKRIDGE"
20050 RETURN
```

Butterfield's Machine Language
Revisited

by Carl Robertson

In the Winter, 1982, *Power/Play* we ran an article by Jim Butterfield titled "A Little Exercise in Machine Language" that showed you how to fit a machine language program inside one line of BASIC. This program, submitted by reader Carl Robertson, is a modification of Jim's program.

```
1 REM  "MACH L INTRO 16X":REM A PROG MOD'N BY CARL ROBERTSON......
2 REM FROM AN ARTICLE IN POWERPLAY MAG (WINTER,82 PG 50) BY JIM BUTTERFIELD.....
........
3 REM THIS PROG MODIFICATION ALLOWS NEC POKES TO BE DONE FROM A DATA STATEMENT..
........
4 REM ACCURACY OF DATA LINE INPUT MAY BE CHECKED BY A 'RUN200' COMMAND..........
........
5 PRINT" THIS PROG MAY SELF-   DESTRUCT UNLESS THE   FOLLOWING ARE DONE:"
6 PRINT"1 DELETE PROG LINES 1-  9":PRINT"2 START PROG WITH       'RUN60'":PRI
NT"3 ";
7 PRINT"CHANGE LINE 220 TO    10 BEFORE RESTARTING"
8 PRINT" NOW HIT THE STOP KEY  AND START WIPING OUT    LINES 1 THRU 9!!"
9 GET E$:IFE$=""THEN9
10 REMXXXXXXXXXXXXXXXXXXXXXXXX
20 SYS4103
30 PRINT:PRINT"      THAT'S ALL"
40 PRINT" DON'T FORGET!  CHANGE LINE 220 TO 10 BEFORE RESTARTING."
50 PRINT"":LIST220
60 A=4103
70 FORN=0TO17
80 READD:POKEA+N,D:NEXT
90 T=0:FORJ=ATOA+17:T=T+PEEK(J):NEXTJ:PRINT"  TOT POKED ="T
100 PRINT" HIT SPC BAR TO CONT"
110 GETE$:IFE$=""THEN110
120 PRINT"      DEMO 'X16'            IS NOW READY":PRINT"":GOTO10
200 DATA32,228,255,201,13,208,1,96,162,16,32,210,255,202,208,250,240,238
210 FORN=0TO17:READA:T=T+A:NEXT:PRINTT:STOP:REM SHOULD = 2847
220 REMXXXXXXXXXXXXXXXXXXXXXXXX
1100 OPEN4,4:CMD4
1110 LIST
1120 PRINT#4,CLOSE4
```

Commodore Information Network File Translator

by Jeff Hand

With your Commodore computer and VICMODEM you can access the many telecommunications data bases available in the United States, including the Commodore Information Network. This issue the Commodore Information Network Systems Operator (fondly known as SYSOP) continues from where he left off in the spring, with an ingenious little program that automatically translates program listings that you saved as text into programs that WILL run. And you don't even have to lift a finger (well, hardly).

Last issue I gave you a program called VICTERM PRINT that lets you download (save) a sequential file from a telecommunications data base to printer or datassette. That program is great for saving things like answers to hotline questions or new product information—but has one limitation. When it saves a program listing, it saves it as a sequential file—as if it were text. That's different from saving it as a program file, which is a set of commands to the computer.

So after you've downloaded a program as a sequential file, there's not much you can do with it, except take the listing and type it in by hand. That's the only way you can load it into the computer and get it to run. Otherwise you've just got a nice program listing that won't do anything.

What a pain, right? Yes, but did you really think I'd leave you with no alternatives? Of course I wouldn't. This issue I've come to your rescue with this short but very sweet little program written by Andy Finkel that will type your program in for you, allowing you to convert those sequential files into program files. And the program files it creates WILL run on your computer. It was originally written for PET and CBM computers with disk drives. Changes that make it work for the VIC 20, Commodore 64 and datassette are noted in the line-by-line explanation. Which is one way of getting you to read the explanation.


Jeff Hand, Commodore Information Network's kindly SYSOP

```
59999 REM DYNAMIC FILE TRANSLATOR FOR THE PET/CBM
60000 INPUT"DRIVE";D$:INPUT"FILENAME";N$:OPEN1,8,8,D$+":"+N$
60010 GET#1,C$:IFASC(C$)<>13THEN60010
60020 POKE174,1:PRINT"▒▒▒";:S=6:GET#1,C$:IFASC(C$)=10THEN60050
60030 PRINTC$;:IFASC(C$)<>13THENGET#1,C$:IFST=0THEN60030
60040 IFST=0THEN60070
60050 CLOSE1:PRINT"FINISHED":END
60070 FORI=623TO624:POKEI,13:NEXT:POKE158,2:PRINT"▒▒▒▒GOTO60020▒";:END
```

Program Explanation:

60000 Input drive number and file name. Open logical file one, for device 8, and open data channel for the disk drive. This can be modified to work with the datassette by using the command: OPEN 1,1,0,D$":"+N$

60010 Get each character from logical file 1 and assign it to C string. If any character in the C string is a carriage return (ASCII 13) continue to the next line of code, otherwise return to the beginning of line 60010 and get the next character.

60020 Register 174 contains the number of current open files. On the VIC 20 and Commodore 64 register 152 keeps track of open files. The reason one is POKED into this register is explained with line 60070. The print command is used to position the cursor (also explained in line 60070). Get the character from logical file one. If C string is an R then terminate the program. This GET command and "R" check is always performed after a carriage return is detected. This will detect the end of the program because the program being loaded will end with a carriage return and the word "READY". The VIC 20 and the Commodore 64 put a linefeed (ASCII 10) between the last program line and the word "READY". To run this program with either of these computers, change IF C$="R" to IF ASC(C$)=10.

60030 This line reads and prints each program line onto the screen. It first prints C$ and if C$ is not a carriage return it reads the next character from file 1 and checks the status of the disk drive. If the status is all right (ST=0), the line is executed again.

60040 This line checks the status of the disk drive again. If the last character read was a carriage return then the status was not checked in line 60030. If the status is OK, go to line 60070.

60050 Close logical file number one, print finished and terminate program.

60070 623 and 624 are the first two locations of the keyboard buffer. These locations are poked carriage returns. For the VIC 20 and the Commodore 64 631 and 632 are the equivalent keyboard location. Register 158 is the number of keys in the keyboard buffer. (For the VIC 20 and the Commodore 64 this same register is 198.) A two is poked into this register, to tell the operating system that two keys were just pressed, but they are ignored as long as the program is running. The words "GOTO 60020" are printed directly under the program line on the screen and the program is ended. Now the two carriage returns in the keyboard buffer are executed one on the program line, putting it into memory, and one on the GOTO 60020. The program starts back up at line 60020, however, once a program is ended, BASIC thinks all files are closed. By poking a one into register 174, the program is telling BASIC that one file is still open. The cursor positioning in

tele/scope

line 60020 keeps the program lines printing at the same location on the screen, and allows the carriage returns in line 60070 to occur at the correct locations.

Essentially this program will take the sequential file, print it to the screen, and at the carriage return enter the program just as if you had typed it in by hand yourself. The computer takes the print to the screen and tokenizes it into a BASIC program file and Voila! your sequential file is now a program file. Remember, however, the file transfer from CompuServe to your computer is not error-free. Therefore you may have to debug a little bit to eliminate those errors that were induced over the phone lines. But this is a lot easier than retyping the file on your own.

The program that you just converted is stored in the memory of the computer. List the program to make sure it's there. The last few lines of the program will be the dynamic file translator program, because the line numbers are large (60000). If the line numbers were lower the translated program would probably write over the file translator program. Delete those last few lines and then save your program file.

Those of you who want to pull off a sequential file that is in assembler will find the dynamic file translator program is of no benefit to you. Your best bet is to use an editor that operates on a sequential file, then process the edited file through an assembler and last through a loader. This will give you a program file that you can use on your machine.

Right now Commodore is working on several projects that will make many more programs available to our telecommunications enthusiasts. We'll be adding two more sections to the Commodore Information Network. One will contain over 600 public domain programs for the whole Commodore computer line, all of which will be free to anyone caring to download them. The other, Commodore Softex, will offer programs for sale. Both of these new sections, however, will require that you have special software in order to download the programs.

The software you'll need will have to incorporate what are called "B protocols". At the moment, the software supplied with your VICMODEM does not use these protocols, which is why you need the VICTERM PRINT and File Translator programs. They give you a way around the B protocols. But wait . . . some time in the future you'll be seeing the release of two software packages that DO use B protocols: the Executive Terminal package (for CBM, PET and Commodore 64 computers), which will be available from CompuServe, and VICTERM 40 (for the VIC 20), from Commodore. Then you'll be able to download program files directly, ready to run. C

VIXEL For Fun and Learning

by John Watkin

The three-volume VIXEL series of graphics and game programs boldly stands out in the sea of available VIC 20 software. These entertaining and educational volumes fully exploit the VIC's animation, graphics and sound capabilities. In addition to the exciting games and helpful graphics, VIXEL is a great tutorial software package due to its superb technical documentation and to the fact that the user is able to list the programs. The three VIXEL volumes include the following programs:

Volume One	Volume Two	Volume Three
Cover 1	Cover 2	Cover 3
Fire	Safari	Warp
Draw	Superfont	Fifteen
Race	Quix	Rail (needs 3K or 8K)

The Cover Programs Each of the three VIXEL volumes is introduced by a cover program. These brief programs are brilliant exercises in animation. When they are run, VIXEL's hairy little mascot dances across the screen dragging the title of the volume behind him. Each cover is different, and each is better than the previous. The cover programs use imaginative hi-res graphics combined with creative sound effects to produce three of the finest displays of VIC animation to be found anywhere.

The Graphics The graphics programs included in the VIXEL series are *Superfont* and *Draw*. *Superfont* is an invaluable aid to programmers who create their own hi-res graphics. Using a joystick or the keyboard, you can design your own hi-res characters on the large work screen either one at a time or in blocks of four. The program then will convert your creation into data statements to be used in your own programs! This is one of the easiest, most enjoyable ways of designing programmable characters available to the VIC user. No programmer should be without *Superfont*.

Draw is not as much a programmer's aid as it is an addicting toy! Do you remember the Etch-a-Sketch you played with as a child? Well, *Draw* can be described best as a multi-color, electronic Etch-a-Sketch. Either a joystick or the keyboard is used to move the line around the screen. The color of the line can be changed at any time by simply hitting any of the color keys. *Draw* enables you to back up, erase, or delete lines and you can even save your creation on tape if you wish.

The Arcade-Style Games *Fire*, *Safari*, *Warp*, *Rail* and *Race* are all hi-res, arcade-style games that challenge to differing degrees the VIC user's game-playing dexterity. All the games use some machine language routines to speed up the action.

Fire's an exciting game in which the player must maneuver a helicopter over burning apartment buildings and drop water on the spreading flames. The helicopter's water supply must be replenished frequently from a nearby water tank. While the pilot reloads, the flames continue to spread throughout the buildings. The player races the computer head-on in trying to douse the flames. *Fire*'s only drawback as an arcade-style game is that the action is not quite fast enough. Despite this minor problem, *Fire* will be a popular addition to your game library.

Safari requires the player to stalk the African jungle and shoot photos of wild animals and natives. This program involves some great hi-res graphics; the natives that run to and fro are truly artistic creations. *Safari* plays at about the same speed as does *Fire*, but again, it could be faster. The animals in *Safari* are great examples of the VIC's graphics capabilities. *Safari* is certain to be a hit.

Rail is an intriguing game in which the player must guide several trains through a train yard filled with rail switches and other trains. The keyboard is used to flip track switches to prevent collisions between trains. The goal of this tricky game is to get your train from one side of the screen to the train depot on the other side. Once you successfully get your train through the yard, another train is added and you must guide two through simultaneously. If you master the game, you will eventually be able to steer eight trains successfully to the depot. Sound easy? Not so! *Rail* is an exciting game that requires a lot of practice to master.

Race is a simple little game in which you race your car around a circular track avoiding the computer-guided oncoming cars. The simplicity of *Race* makes it a good game for the young kids around the house. For adults, *Race* will serve as a fun diversion.

Of VIXEL'S five arcade-style games, **Warp** is by far the best. It is challengingly difficult and it rarely becomes boring. In *Warp*, the player, using a joystick, must maneuver a spaceship down into a winding tunnel. The farther he proceeds, the narrower the tunnel becomes. If the spaceship careens into the tunnel wall, it is destroyed and the game ends. *Warp* is a refreshingly simple game, yet at the same time it is intriguing and challenging. *Warp* requires a lot of skill and patience to master. It is a simple matter to become addicted to *Warp*!

The Puzzle Games The VIXEL series includes two puzzle games that test your logic and recall skills. *Quix* is a game in which the computer presents a random sequence of color and sounds that flash in four boxes on the screen. The player must then repeat the pattern by hitting the corresponding function keys. As the game progresses, the patterns get longer. Soon, the player is frantically hitting function keys, trying to remember the last pattern. *Quix* is an enjoyable game that will appeal to both young and old.

In the game *Fifteen*, the screen displays a large square filled with fifteen randomly numbered squares. The object is to "slide" one square at a time until the numbers are in sequence. The player can slide a square by pushing the joystick in a given direction. *Fifteen* is a clever and creative application for the VIC 20. The game is designed very well and is a fun challenge.

The Documentation Thorough, professional documentation accompanies all VIXEL software. The documentation includes both a layman's guide to the programs as well as technical outlines of the programs themselves. These technical descriptions are very valuable to the programmer who wants to decipher and learn from the VIXEL programs. The software is not protected so the user can list and study any of the programs. This combination of listability and documentation is by far the biggest selling point of the VIXEL software. Few software companies can match VIXEL'S superb technical documentation.

VIXEL software belongs in every VIC 20 user's program library. The wide variety of games and graphics will appeal to everyone from young kids to serious programmers. Unlike much of the software currently available for the VIC 20, the VIXEL programs are easily listed. This advantage, in accompaniment with the superb documentation, will prove to be a great asset to the programming crowd. VIXEL Volumes One, Two and Three are available from The Code Works, Box 550, Goleta, CA 93116. Their number is (805) 683-1585. The price of each volume is $12.95. C

"We're Glad You Asked"

Winter, 1982

A method for disabling the RUN/STOP key was presented in the Winter 1982 *Power/Play* question and answer section. The procedure is to POKE 788,194 to disable the key and POKE 788,191 to re-enable it. A side effect of using this is that whenever the RUN/STOP key is disabled the real time clock (TI and TI$) is also disabled. This is unacceptable for some applications which need the real time clock (such as home sentry programs which turn lights, alarms or coffee pots on and off).

A second method can be used to disable the RUN/STOP key without this side effect (although it does have its own). You can disable the key by using POKE 808,109 and re-enable it by using POKE 808,112. The real time clock runs as normal now, but.... The side effect of this method is that the program won't LIST properly until the POKE 808,112 is performed.

The VIC Magician

Winter, 1982

The very last line of that article, which is a program line that is supposed to produce a beep, is wrong. It should read:

```
35 POKES3,200:FORT=1TO200:NEXT:POKES3,0
```

Tele/Scope

Spring, 1983

In the program listing on page 88, line 385 should read:

```
385 IF ZT=0 THEN OPEN 4,4,7:PRINT #4,
RT$ RT$
```

"Deflection"

Winter, 1982

For those of you who conquered the little tiny listing and typed in the *Deflection* program from the Winter issue, more pitfalls. A routine was included at the end of the program that didn't nearly work. Here are the lines to change:

```
5700 CT = 0: FOR L=SS+ WD TO SE- WD: IF PEEK(L) <> 32 THEN 5900
5800 POKE L, 102: CT= CT+ 1
6000 FOR L=1 TO CT- A
6100 X=INT( RND( 1)* WD* HT)+ SS: IF PEEK(X) <> 102 THEN 6100
```

If you set the speed control to a number higher than the recommended 10, the ball flickers on for only a short time. You can correct this by changing to this new version of line 3300:

```
3300 P0=NP: POKE Q0,NP: POKE P0,81: FOR Q5=1 TO Q4*5: NEXT: Q0=P0:
     GOTO 1900
```

Also, since we've had a number of beginners frustrated by the marginal legibility of that program listing, we thought we'd give everybody another crack at it. So here it is, in its entirety, with corrections.

For those who are new, this program will run on any Commodore computer. Look at lines 1050-1052. Pick your machine and eliminate the words REM and the computer name from that line. That will set the factors for your particular computer.

```
1000 REM-->DEFLECTION #9 BY NEIL HARRIS
1050 REM CBM SS=32768:WD=80:HT=25:SE=SS+WD*HT-1:POKE59468,12
1051 REM PET SS=32768:WD=40:HT=25:SE=SS+WD*HT-1:POKE59468,12
1052 REM VIC SS=7680:WD=22:HT=23:SE=SS+WD*HT-1:POKE36879,8
1053 SS=1024:WD=40:HT=25:SE=SS+WD*HT-1:POKE53281,0
1100 Z=-10:INPUT"[CLR][DOWN][DOWN]INSTRUCTIONS";A$:IFLEFT$(A$,1)="N"THEN1400
1200 IFLEFT$(A$,1)="Y"THEN4300
1300 Z=Z+10:GOTO4300
1400 INPUT"[DOWN]TARGETS";A$:A=VAL(A$):IFA<1ORA>.8*WD*HTTHENZ=Z+10:GOTO5000
1450 INPUT"[DOWN]SPEED (0-9)";Q4:IFQ4<0THENZ=Z+10:GOTO1450
1500 PRINT"[CLR]";:FORL=0TOWD-1:POKESS+L,96:POKESE-L,96:NEXTL
1510 FORL=WDTO(HT-2)*WDSTEPWD:POKESS+L,96:POKESS+WD+L-1,95:NEXTL
1530 IFA>440THEN5700
1550 FORL=1TOA
1600 X=INT(RND(1)*WD*HT)+SS:IFPEEK(X)<>32THEN1600
1700 POKEX,102:NEXTL
1750 FORL=SSTOSE:IFPEEK(L)<>32THENNEXTL
1800 P=0:T=A:P0=L:Q0=L:D=SGN(RND(2)-.5):TI$="000000":IFRND(3)>.5THEND=WD*D
1900 NP=P0+D
2200 J=PEEK(NP):GETA$:IFJ=32THEN8100
2225 IFJ=96THEN5200
2250 IFJ=81THEN5250
```

glitch fix

```
2300 IFJ=102THEN3400
2350 GETA$:IFA$="Q"THEN5100
2400 IFJ=77THEN2800
2500 IFABS(D)=1THEN2700
2600 D=-D/WD:GOTO5250
2700 D=-WD*D:GOTO5250
2800 IFABS(D)=1THEN3000
2900 D=D/WD:GOTO5250
3000 D=D*WD:GOTO5250
3100 IFA$="Q"THEN5100
3150 IFA$="/"THENP=P+1:POKENP,78:GOTO1900
3200 IFA$="£"THENP=P+1:POKENP,77:GOTO1900
3300 P0=NP:POKEQ0,32:POKEP0,81:FORQ5=1TOQ4*5:NEXT:Q0=P0:GOTO1900
3400 T=T-1:IFT>0THEN3100
3500 PRINT"⊐⊠⊠YOU GOT";A;"TARGETS USING";P;"PADDLES"
3600 PRINT"⊠YOUR TIME WAS ";MID$(TI$,3,2);":";RIGHT$(TI$,2);"⊠"
3700 R=16*LOG(A*15000/(P*P*VAL(TI$)/2))-Z-2*Q4:PRINT"YOUR RATING";R
3701 IFR<=0THENR=.1
3800 RESTORE:FORL=1TO10-R/10:READA$:NEXTL
3900 PRINT"...";A$
3950 IFZ>90THENNEW
4000 INPUT"⊠ANOTHER GAME";A$:IFLEFT$(A$,1)="N"THENEND
4100 IFLEFT$(A$,1)<>"Y"THENZ=Z+10:GOTO3950
4200 GOTO1400
4300 PRINT"⊠THE OBJECT OF THE GAME
4400 PRINT"IS TO DEFLECT THE BALL
4450 PRINT"BALL THROUGH THE
4500 PRINT"TARGETS. ONCE THE LAST
4550 PRINT"TARGET HAS BEEN HIT
4575 PRINT"THE GAME ENDS.
4600 PRINT"⊠YOU DEFLECT THE BALL
4650 PRINT"USING THE £ AND /
4700 PRINT"KEYS.  ONCE A
4750 PRINT"DEFLECTOR IS CREATED
4800 PRINT"IT IS IN PLACE
4825 PRINT"PERMANENTLY.
4850 PRINT"⊠SPEED FACTOR OF ZERO
4875 PRINT"IS MAXIMUM."
4900 PRINT"⊠HIT Q AT ANY TIME
4950 PRINT"TO QUIT"
5000 PRINT"⊠YOU MAY CHOOSE FROM
5050 PRINT"1 TO"INT(.8*WD*HT)"TARGETS":GOTO1400
5100 Z=Z+20:PRINT"⊐YOU QUIT":GOTO4000
5200 D=-D:P0=NP:NP=P0+D:GOTO2200
5250 P0=NP:GOTO1900
5500 DATAAMAZING,PROFESSIONAL,"VERY GOOD",FAIR,"KEEP PRACTICING","TRY HARDER"
5600 DATAWIMP,SPAZZ,"TRY A DIFFERENT GAME"
5700 CT=0:FORL=SS+WDTOSE-WD:IFPEEK(L)<>32THEN5900
5800 POKEL,102:CT=CT+1
5900 NEXTL
6000 FORL=1TOCT-A
6100 X=INT(RND(1)*WD*HT)+SS:IFPEEK(X)<>102THEN6100
6200 POKEX,32:NEXTL
6300 GOTO1750
```

Win A FREE Commodore 64™ Computer

Can You Beat PARATROOPER?

10 Prize Winners

A New Commodore 64 will be awarded to the person with the highest score. The winning entry must contain a photograph of the highest score of the game.

PARATROOPER *By Nic Dudzik*

REALISTIC SMOOTH ACTION AND TRUE HI-RES GRAPHICS

JOYSTICK OR KEYBOARD

1st Prize
CBM 64

2nd Prize
Disk Drive

3rd Prize
$200 worth of Software

4th - 10th Prize
$100 worth of Software

Runs on the Standard VIC 20™

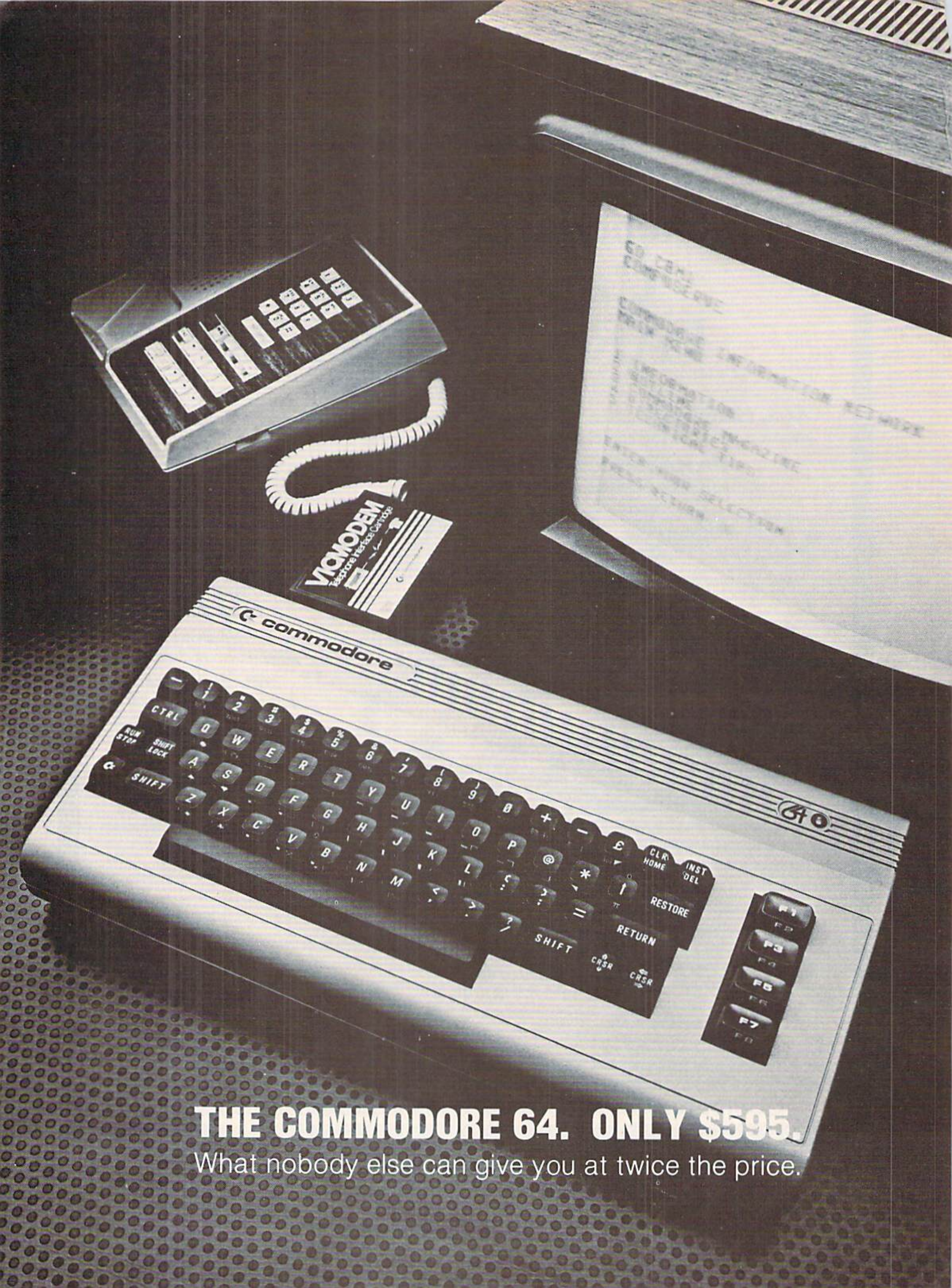
PARATROOPER a High Resolution game that doesn't let you make any mistakes. You are in command. Helicopters fill the sky, (and we mean fill the sky!), dropping paratroopers. Your mission is to keep 3 paratroopers from hitting the ground on either side of your gun. But that's just the beginning. You score 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 you VIC. PARATROOPER IS OUR #1 SELLING ARCADE GAME, you've got to see this game to believe it. **$19.95**

Order your copy today or see your dealer.
Will you get the top score?

RULES

1. All entries must be mailed, as postmarks are required to determine the earliest winning entry. In the event of a tie the postmark will determine the winner.
2. Deadline for entries is July 31, 1983.
3. Proof of purchase must be provided. Return your entry with package front and proof of purchase slip and photo.
4. **ENTRIES MUST BE MAILED TO:** PARATROOPER CONTEST, P.O. Box 388, Lake Havasu City, AZ 86403
5. Game contest void where prohibited.
6. Contest begins April 2, 1983. All entries must be postmarked by July 31, 1983 Contest ends July 31, 1983.
7. Winners will be notified by mail. Public notice of winners will be printed in this and other computer magazines.
8. Only one entry per person please — all duplicates will be discarded.
9. There will be 10 winners for a 1st, 2nd & 3rd prize. Prizes for 4th - 10th will be for software.
10. Contest is limited to U.S.A., Properties and Canada only.

ComputerMat • P.O. Box 1664X • Lake Havasu City, Arizona 86403



THE COMMODORE 64. ONLY $595.

What nobody else can give you at twice the price.

new products

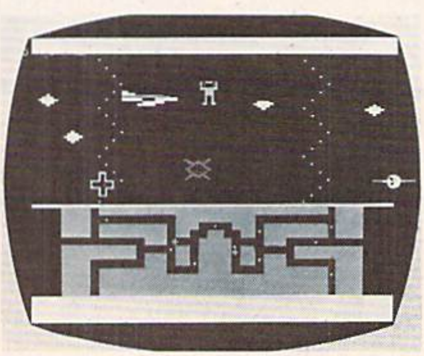
The following information is taken from product announcements sent to us by independent manufacturers, and is provided simply to keep our readers abreast of developments in the field. Commodore does not endorse any of the products mentioned, has not tested them and cannot vouch for their availability.

Company:
Sirius
10364 Rockingham Drive
Sacramento, CA 95827
916-366-1195

Product:
Four new games on cartridge for the VIC 20 and Commodore 64—**Turmoil:** Fight off evil aliens attacking from both sides. **Fantastic Voyage:** You and your submarine are reduced to microscopic proportions and injected into the blood stream of a critically ill patient. You must navigate past deadly defense cells, bacteria, antibodies and enzymes to destroy the life-threatening blood clot near the patient's brain. **Deadly Duck:** The crabs in the pond take to the air, armed with bricks and bombs. Deadly Duck's job is to shoot them down with his gun-barrel bill. **Fast Eddie:** Eddie is on a treasure hunt, climbing up and down ladders everywhere in search of prizes, while pesky Sneakers keep him jumping.
Price: See your local Sirius dealer.

Company:
Twentieth Century-Fox
Games of the Century
4701 Patrick Henry Drive, Bldg. #9
Santa Clara, CA 95050
408-988-6666

Product:
Flash Gordon:—Space rescue game on cartridge for the VIC 20. Maneuver through underground tunnels and battle spider warriors to save American spacemen trapped in Spider City. Package features a full-color poster from the film.
Price: Contact company.



Flash Gordon

Company:
Avalon Hill Game Company
4517 Hartford Road
Baltimore, MD 21214
301-254-5300

Product:
Eight multi-cassette games for the Commodore 64—**B-1 Nuclear Bomber, Midway Campaign, North Atlantic Convoy Raider, Nukewar, Planet Miners, Draw Poker, Computer Stocks & Bonds, Andromeda Conquest, Computer Football Strategy** and **Telengard.**
Price: From $16.00 to $24.00

Company:
EPYX/Automated
Simulations
1043 Kiel Court
Sunnyvale, CA 94086
408-745-0700

Product:
Jumpman—Science fiction game on disk for the Commodore 64. The player is Jupiter Jumpman, last-resort secret weapon against the enemy Alienators. Jupiter Jumpman must defuse bombs on all 30 levels of headquarters by scaling ladders, girders and ropes. Obstacles include Alienator bullets, robots, dragons, birdmen, flying saucers, crumbling girders and vanishing escape routes.
Price: $39.95

Company:
Tronix Publishing, Inc.
8295 S. La Cienega
Inglewood, CA 90301
213-671-8440

Product:
Three games for the VIC 20—**Deadly Skies:** A "shoot 'em up" in which the player has a squadron of

Moondust

Rat Hotel

Pipes

five helicopters with which to destroy an enemy military base. Features 32 levels of play. **Gold Fever!:** The player is a gold miner who must gather all the gold in a mine shaft while avoiding runaway box cars, boulders and claim jumpers. The miner must do all this before oxygen runs out in the mine. Nine skill levels. **Scorpion:** Pits a scorpion struggling for survival against such enemies as dragons, frogs, venus fly traps, worms and pods. 32 play levels.
Price: $39.95 each

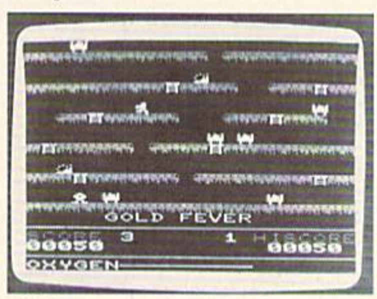
Company:

Creative Software
230 E. Caribbean Drive
Sunnyvale, CA 94086
408-745-1655

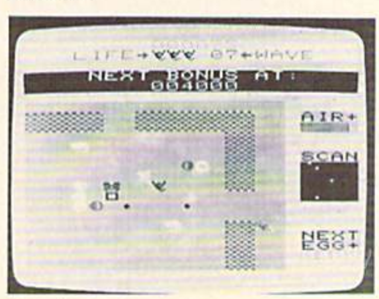
Product:

Nine titles for the Commodore 64—Four cartridge games and five home application programs on tape or disk. Games include conversions of two VIC 20 games, **Astroblitz** and **Trashman,** and two new games, **Moondust** and **Save New York!** In **Moondust** the player draws colorful trails of moondust crystals across glowing concentric circles. Music, composed according to joystick action, accompanies every move. The object is to drag the moondust to the center of the circles without bumping the space walker. The point of **Save New York!** is to destroy the above-ground and subterranean mutants before they destroy the city. Home application programs, all conversions from the VIC 20, include **Household Finance, Home Inventory, Loan Analyzer, Car Costs** and **Decision Maker.**
Price: Contact company


Deadly Skies



Gold Fever



Scorpion

Product:

Two new cartridge games for the VIC 20—**Rat Hotel:** Using a joystick, the player has three minutes to maneuver Ermine the Rat from the attic down six floors to the basement, where he can eat Le Grand Cheeseball. Ermine must avoid Waldo the Maintenance Man and traps. **Pipes:** A home education game intended to teach the concepts of spatial relationships and economics. Arlo the Plumber must connect all the houses in town to the main water supply by buying various kinds of pipe from the factory. He selects the pipe and carries each piece to the work site, where he attaches it. The point is to hook up all the houses, with no leaks, without running out of pipe or using too much money.
Price: $39.95 each

Company:

RAK Electronics
P.O. Box 1585
Orange Park, FL 32073

Product:

Commodore 64 Maillist—A multipurpose mailing list management system on cassette or disk. Allows you to construct, sort, maintain and print out a mailing list of over 300 addresses. Subfiles created from the main file allow growth beyond the 300-address limit.
Price: $14.95 cassette, $17.95 disk plus $2.00 shipping per order.

new products

Company:
COMPUTER:applications, Inc.
13300 S.W. 108 St. Circle
Miami, FL 33186

Product:
Two games for the VIC 20—**ZAP!:** With keyboard or joystick, players work their men up the many levels of corporate structure while avoiding the obstacles that can zap their chances for success. **KEYQUEST:** Explore the many levels of an ancient dungeon while gathering valuable treasures and gaining experience points. Destroy monsters and search for the key to each dungeon level.
Price: $29.95 each

Company:
MicRo Information Systems
P.O. Box 73
Wayne, NJ 07470
201-696-3296

Product:
Zorlok—Adventure game for the VIC 20 with 8K expansion. Enter the castle of Zorlok the wizard, wipe out the plague of monsters and regain his treasures. Multiple skill levels.
Price: $39.95

Company:
Parr Programming
2664 Tyler Street
Gary, Indiana 46407
219-885-0611

Product:
Two sports strategy games on cassette for the VIC 20—**Baseball Adversary:** Make tough managerial decisions against your computer opponent. Additional players on the bench allow for pinch hitting and relief pitching. **Football Adversary:** Places you in the role of head coach opposing the computer. A selection of 31 plays and displays of halftime/game statistics. Requires 8K memory expansion.
Price: Baseball $10.95; Football $14.95.

Company:
Computer Software Associates, Inc.
50 Teed Drive
Randolph, MA 02368
617-961-5700

Product:
VIC-PAK #1—Seven basic programs for the VIC 20 on cassette. **Mortgage** allows the user to find the unknown variable for principle, monthly payment, term and annual interest of a mortgage loan. **Elements** provides a short quiz on naming the chemical symbol associated with each of the chemical elements. **Statistics** is a brief introduction to the VIC 20 as a sophisticated calculator. **Calendar** products any calendar month from 1 A.D. to 9999 A.D. **Marblestat** illustrates the computer's ability to imitate real life events via programming. **Expectancy** calculates a rough estimate of the user's life expectancy. **U-Draw** allows you to draw simple blocked graphics in eight colors.
Price: $19.95

Company:
Scholastic, Inc.
730 Broadway
New York, NY 10003
212-505-3000

Product:
Family Computing—A national consumer magazine aimed specifically at families with home computers, to be published monthly beginning in September, 1983.
Price: Single copies $1.95; year's subscription $17.97 **C**

commodore product list

Order Number	Product Name and Description
Commodore Consumer Range—VIC Series	
	VIC 20™ Full-featured, expandable color computer system. 5K RAM, includes RF Modulator & TV switch box.
Commodore Personal Computer Range—64 Series	
	Commodore 64™ 64K RAM personal computer features 16 colors, music synthesizer, high resolution graphics (320 × 200 pixels), 8 movable sprites, and CP/M* option. Provides access to 2 joysticks, 4 paddles or a light pen. Display: 40 column × 25 lines
Commodore Personal Computer Range—4000 Series	
	PET® 4032N System includes graphic keyboard and numeric keypad; Display: 40 column × 25 lines; 32K RAM
Commodore Business Computer Range—8000 Series	
	CBM™ 8032B Typewriter style keyboard, numeric keypad, 80 column × 25 line display. 32K RAM, BASIC 4.0.
	CBM 64K Memory Expansion Board 64K Add On-Memory. Expands CBM 8032 to 96K RAM.
	Upgrade Board Upgrades CBM 8032 to SuperPET
Commodore Advanced Computer Range—9000 Series	
	SuperPET™ Computer SP9000 Enhanced 8032 with additional 6809 microprocessor; total 134K memory with 96K RAM. Languages include Waterloo microBASIC, Waterloo microAPL, Waterloo microPASCAL, Waterloo microFORTRAN, Waterloo microCOBOL, interpreters and Waterloo 6809 Assembler.

Order Number	Product Name and Description
Commodore Systems Peripherals—Disk Drives	
VIC 1530 (C2N)	**Datassette** Cassette storage for PET/CBM/VIC/C64
VIC 1541	**Single Disk** Single drive intelligent 5¼" floppy disk system for the VIC 20 and Commodore 64. Media compatible with CBM 2031 and 4040 diskettes. (170K)
CBM 2031	**Single Disk** Single drive intelligent 5¼" floppy disk system. 170K (DOS 2.6).
CBM 4040	**Dual Drive** Dual drive intelligent 5¼" mini-floppy disk system. 340K (DOS 2.1).
CBM 8050	**Dual Drive** Dual drive intelligent 5¼" mini-floppy disk system. 1 Megabyte (DOS 2.5).
CBM 8250	**Dual Drive** Dual drive intelligent 5¼" floppy disk system. 2.1 Megabyte, dual sided (DOS 2.7).
D9060	**Hard Drive** 5¼" "Winchester" drive (DOS 3.0) 5.0 MB
D9090	5¼" "Winchester" drive (DOS 3.0) 7.5 MB

All Commodore disk units feature compatible DOS command language and file handling capabilities to allow upward expansion of systems as required.

*CP/M is a Registered TradeMark of Digital Research, Inc.

Order Number	Product Name and Description
Printers	
VIC 1525	**Graphic Printer** VIC dot matrix printer for the VIC 20 and Commodore 64. 30 CPS, 80 column; prints full VIC character set; tractor feed.
CBM 4022	**Printer** Dot matrix printer with tractor feed, 80 column, prints full PET graphics, variable line spacing, bi-directional, 80 CPS.
CBM 8023P	**Printer** Dot matrix printer with tractor feed, 136 column. Fully programmable graphics, bi-directional, pseudo letter quality, 150 CPS.

commodore product list

Order Number	Product Name and Description
CBM 8300P	**Printer** True letter quality, daisy wheel printer with 10, 12, 15 pitch or proportional spacing and optional tractor feed. Bi-directional; 40 CPS; supports APL & foreign language print wheels.

Cables

	IEEE-IEEE Designed to connect more than one peripheral to any PET/CBM computer.
	PET-IEEE Used for connecting one peripheral to any PET/CBM computer.

Accessories

CT 500	**Commodore Computer Desk** 26″ × 1⅛″ T-molding with rounded corners; heavy steel legs with adjustable leveling glides; typewriter height with a ventilated shelf for the disk drive.
VIC 1311	Joystick for VIC 20 and Commodore 64
VIC 1312	Paddles for VIC 20 and Commodore 64
VIC 1600	**VICMODEM** Telephone Interface Cartridge. Lets VIC 20 & Commodore 64 owners communicate by telephone, access telecomputing and timesharing service. Includes VICTERM I program on tape. Plus FREE CompuServe Subscription.
VIC 1525	**Printer Paper** 1000 Sheets

Peripheral Compatibility...
CBM Peripherals can be used with the PET, CBM, and SuperPET range of computers, and also with the VIC 20 and Commodore 64 computers if a special IEEE cartridge is purchased.

Commodore Marketed Applications Software for PET/CBM

400010	TCL Pascal version 1.6
400020	Assembler Development
400030	Integer Basic Compiler
400040	CMAR Record Handler
400050	UCSD Pascal (without board)
400060	PETspeed BASIC Compiler
500010	OZZ—8050
500011	OZZ—4040

Order Number	Product Name and Description
500039	Dow Jones Portfolio Management System (80 col. RS232)
500041	BPI Accounts Receivable
500043	BPI General Ledger
500046	BPI Job Cost
500047	BPI Inventory
500048	BPI Payroll
500049	BPI Accounts Payable
500050	Legal Time Accounting
500051	Medical Accounting System
500052	Atlas 1200 Service & Maintenance
500053	Titan Job Cost System
500060	Freight Rating and Invoice
900040	I.R.M.A. II (Information Retrieval & Management Aid)
500100	Commodore General Ledger
500101	Commodore Accounts Receivable
500102	Commodore Accounts Payable

Commodore Marketed Applications Software for the Commodore 64 Business/Education

600001	EasyCalc 64—Electronic Worksheet
600002	EasyPlot 64—Prints charts & graphs
600003	EasySchedule 64—Schedules time & resources
600004	EasyFinance 64—Analyze financial opportunities
600005	EasyFile 64—Database (define, edit, enter, retrieve)
600006	EasyScript 64—Multi-function wordprocessor
600008	Word/Name Machine—Simple wordprocessor/mail list
600009	EasyMail 64—Full feature name & address program
600010	PET Emulator—Allows use of PET software
600011	Easy Lesson/Quiz 64—Create/Administer test quizzes

Entertainment

600050	C-64 Avenger
600051	C-64 Jupiter Lander
600052	C-64 Road Race
600053	C-64 Omega Race

Special Documentation

	Commodore Software Encyclopedia A catalog of over 1000 software programs, books, and interfaces from Commodore and third parties. (406 pages)

Order Number	Product Name and Description
	VIC 20 Programmer's Reference Guide VIC 20 reference manual with information on VIC BASIC. 6502 Machine Code Programming, Input/Output ports, VIC microprocessing chips and tips for all levels of programmers. (289 pages)
	Education Marketing Resources Book 185 Public Domain Programs for Education including 4 disks of software. Details of Users' clubs, educational vendors, and educational applications. (380 pages)
	Users Magazines Information on our products is continually updated in the following publications: **COMMODORE The Microcomputer Magazine**...a user magazine devoted to all aspects of Commodore computers including business and education. (6 issues) **POWER/PLAY Fun, Games & Beyond with Commodore Home Computers...** a user magazine devoted primarily to personal fun, entertainment and learning with Commodore Computers. (4 issues)

VIC Memory Expansion Cartridges

Order Number	Product Name and Description
VIC1110	**VIC 8K Memory Expander Cartridge** 8K RAM expansion cartridge plugs directly into the VIC.
VIC1111	**VIC 16K Memory Expander Cartridge** For use with the VIC 20 and/or VIC1010 EXPANSION MODULE.

VIC Interface Cartridges

Order Number	Product Name and Description
VIC1011A	**RS232 Terminal Interface** Provides interface between the VIC 20 & C-64 and RS232 telecommunications modems. Connects to the VIC's & 64's user port.

VIC Recreational Games on Cartridge

Video Arcade Series

Order Number	Product Name and Description
VIC1901	**VIC Avenger** It's an invasion of space intruders and you're the VIC "Avenger". Space action for arcade enthusiasts.
VIC1904	**Superslot** Colorful slot machine game works just like the real thing! Great music and sound effects.
VIC1907	**Jupiter Lander** Pilot your "Jupiter Lander" through the treacherous crevices of a mysterious planet. Variable rocket thrust, anti-gravity, horizontal retros.
VIC1908	**Draw Poker** Casino-style poker recreates the real thing! Superb animation and sound effects add to the fun, mystery, and luck.
VIC1910	**Radar Rat Race** The magical mouse maze makes for a fast-paced, challenging game of wit, strategy and reflexes. Excellent graphics.
VIC1913	**Raid on Fort Knox** You're scurrying through a complex of tunnels below Fort Knox. Just ahead you spot the gold... now grab it and try to escape before the guards find you.
VIC1919	**Sargon II Chess** SEVEN challenging play levels. Called the "best" microcomputer chess program by experts. SARGON II makes the VIC a challenging chess opponent.
VIC1920	**Pinball Spectacular** Plays just like a true pinball machine... only computerized... flashing lights... quick "flipper" action!
VIC1921	**Super Smash** The World Championship is at stake! As a finalist do you have the cat-like reflexes to return the speeding ball and take the crown? On your toes... this is racquetball at its toughest!
VIC1922	**Cosmic Cruncher** Maneuver your "Cosmic Cruncher" through the Milky Way and "crunch" all the pulsars in the galaxy... eleven challenging levels of play... over 300 color/maze combinations. Exciting arcade action!
VIC1923	**Gorf* (The smash-hit arcade game!)** Midway's incredible coin-operated game is now on cartridge for the VIC! Includes 4 completely different games, multiple levels of difficulty, some of the best cartoon graphics ever devised for video games. Invaders, gorfies, death

commodore product list

Order Number	Product Name and Description
	ships, saucers, aliens... it's terrific!
VIC1924	**Omega Race* (The smash-hit arcade game!)** The ultimate space game. You've got one Omegan fighter maneuvering against droid ships, command ships, death ships, photon mines and vapor mines. Fantastic "rubber band" boundaries, multiple levels of difficulty... all the features that make the Bally/Midway game so successful! One or two players.
VIC1931	**Clowns*** Come one, come all... see the amazing jumping clowns... direct from their show-stopping Bally/Midway arcade tour... A true arcade "classic"! Colorful acrobatics with high scoring skill.
VIC1935	**Commodore Artist** A true lightpen drawing game... you are the artist... you create the picture... multi color!
VIC1937	**Sea Wolf*** As submarine commander you are in charge of sinking and destroying all enemy ships... destroyers, freighters and P.T. boats... an explosive Bally/Midway "arcade classic". Fast action fun!!
VIC1938	**Tooth Invaders** Prevent tooth decay with Plaqueman, using your toothbrush and dental floss. A great arcade/educational game.
VIC1939	**Star Post** 3D space action... you control the laser to destroy flying aliens and objects. Allow less than 8 hits on your base and advance to the next level.
Scott Adams Adventure Games	
VIC1914	**Adventure Land Adventure** You wander through an enchanted world trying to recover the 13 lost treasures. You'll encounter wild animals, magical beings, and many other perils and puzzles. Can you rescue the blue ox from the quicksand or find your way out of the maze? For beginning Adventurers and veterans alike.
VIC1915	**Pirate Cove** "Yo-Ho-Ho and a bottle of rum..." You'll meet up with the Pirate and his daffy bird, and encounter many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover Long John Silver's lost treasures?
VIC1916	**Mission Impossible Adventure** "Good Morning, your mission is to..." and so it starts. Can you complete your mission in time? Is the world's first automated nuclear reactor doomed? This one "radiates" with excitement!
VIC1917	**The Count** You wake up in a large brass bed in a castle somewhere in Transylvania. Who are you, what are you doing here, and WHY did the postman deliver a bottle of blood? It's LOVE AT FIRST BYTE!
VIC1918	**Voodoo Castle** Count Christo has a fiendish curse put on him by his enemies. There he lies, and you are his only hope. Can you rescue him, or is he forever doomed? (Beware of the Voodoo Man!)
Children's Series	
VIC1911	**The Sky is Falling** Help Chicken Little by catching pieces of the sky as they fall! A great "first game" to teach motor skills... fun and challenging!
VIC1912	**Mole Attack** A colorful "cartoon action" game. You're trying to keep those nasty moles underground where they belong but they keep popping up! How many can you clunk before time runs out? Fast, fun, frantic!
VIC1933	**Bingo Speed Math** Two learning games in one. Learn to add, subtract, multiply and divide... while having fun. Teach your child to think and respond quickly while having fun. Math is made simple and fun with SPEED/BINGO math.

*Bally Midway games developed under Commodore's licensing agreement with Bally Manufacturing Company.

Order Number	Product Name and Description
VIC1928	**Home Babysitter** Three preschool game programs help teach your preschool children counting, the alphabet and relationship skills. Keep your children occupied for hours while giving him or her a headstart in school. The parent's manual helps your child get the most out of this excellent development aid.
VIC1930	**Visible Solar System** Astronomy buffs love this incredible game-science-learning tool. Journey to the major planets of our solar system, learn key statistics about each planet... OR... record atmospheric conditions and compare planetary statistics using "Astro Calc." A wonderful astronomy tool which gives you a tour of the solar system and teaches you about major planets along the way.

Home Improvement Cartridges

VIC1929	**Personal Finance** Your budget, expenses and personal finances are simply computerized through this business package. Personal Finance will organize, arrange and computerize your home financial expenses.

VIC Programming Aid Cartridges

VIC1211A	**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. 1024 × 1024 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.
VIC1212	**Programmers 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.
VIC1213	**Vicmon Machine Language Monitor** Helps machine code programmers write fast, efficient 6502 assembly language programs. Includes one line assembler/disassembler.

VIC Teach Yourself Programming Series

VL102	**Introduction to BASIC Programming—Part I** A gentle but thorough introduction to BASIC programming. Excellent first book for any new computerist. Tutorial lesson tapes included.
VL103	**BASIC Programming—Part II** A continuation of the excellent BASIC programming series with more programs, lessons, and instructional aids.

VIC Application Programs on Tape

The following pre-recorded programs are designed for use with the Commodore Datassette Tape Recorder. Programs on tape come in several varieties and are color coded by category as follows: Recreation (red), Education (blue), Business/Calculation (green), Home Utility (orange), and Computing Aid (black).

VT106A	**Recreation Program Six Pack*** Car Chase—Fast-paced road action VIC 21—Casino-style blackjack Blue Meanies from Outer Space—Space game Biorhythm/Compatibility—Compare biorhythms Spacemath—Math improvement grades 1-6 Slither/Super Slither—Dexterity game
VT107A	**Home Calculation Program Six Pack*** Personal Finance I—Home Budget Personal Finance II—Home Budget VIC Typewriter—Word processor for home use Expense Calendar—Income, expenses, appointments Loan & Mortgage Calculator—Decision-making aid Home Inventory—Home belongings list

commodore product list

Order Number	Product Name and Description
VT108	**Math Improvement Six Pack Grades 2-6** Numbowl, Backfire, LCM Machine, Scare City Motel, Sector Five, Ruler Dueler
VT109	**Sampler Six Pack** Big Bad Wolf, Treasures of the Bat Cave, Music Synthesizer, Crawler, Alpha Draw, Searcher
VT164	**Programmable Character Set/ Gamegraphics Editor** Lets the VIC user create up to 64 programmable characters and use them in BASIC programs. The Editor takes only one-half kilobyte of program space, works with tape, disk and printer.
VT232	**Victerm I—Terminal Emulator** A handy VIC terminal program on tape which converts any VIC to a terminal for use with a telephone modem.

VIC 20 Books and Manuals

VM100	**Personal Computing on the VIC 20** The "Friendliest" computer instruction guide available. This owner's manual comes free with every VIC 20 but is also in demand by teachers who use it in the classroom, and by "VIC families" who want more than one guide for each family member.
VM110	**VIC 20 Programmers Reference Guide** The master VIC 20 reference manual includes information on VIC BASIC, 6502 Machine Code Programming, Input/Output ports, VIC microprocessing chips, and tips for all levels of programmers. Indispensible.

Accessories and Replacement Parts

1515P	**Graphic Printer Paper** 1000 sheet pack, tractor feed, 15lb. bond
1515R	**Printer Ribbon Cartridge** For VIC Graphic Printer
90477801	**TV Switchbox**
90510101	**RF Cable**
902505	**Power Supply**
32145301	**Modulator**

This price list supersedes all previous price lists.
Prices subject to change without notice.

C

advertisers index

commodore
1200 Wilson Drive
West Chester, Pa. 19380
Address Correction Requested