commodore power/play

home computing

Spring 1983 Volume II, Number 1
$2.50 U.S.
$3.50 Canada

- New Cartridge Games for the Commodore 64
- Kids Learn to Program the VIC 20 with *Gortek and the Microchips*
- Making Music on the Commodore 64

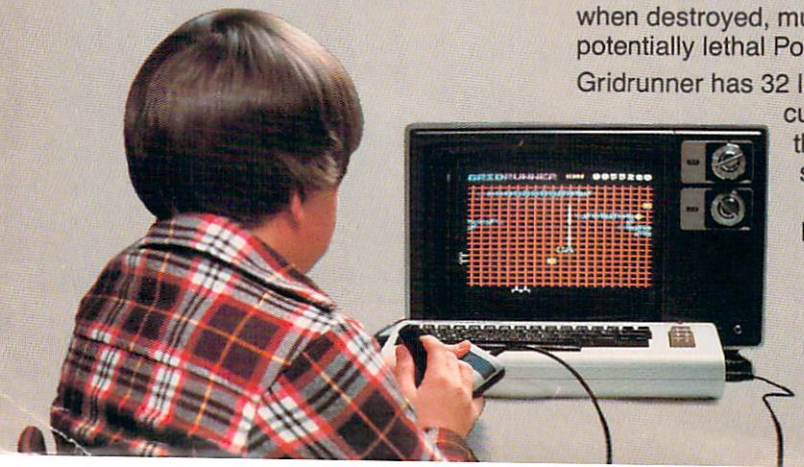
0 43467 00121

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.

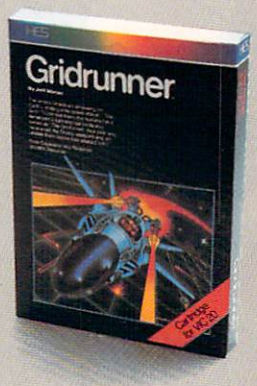
As the pilot of the Gridrunner, a combat ship, you must annihilate the various enemies traveling along the "Grid." High scores are possible only through the mastery of the patterns of the X/Y Zappers and the Gridsearch Droids which, when destroyed, mutate into potentially lethal Pods.

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

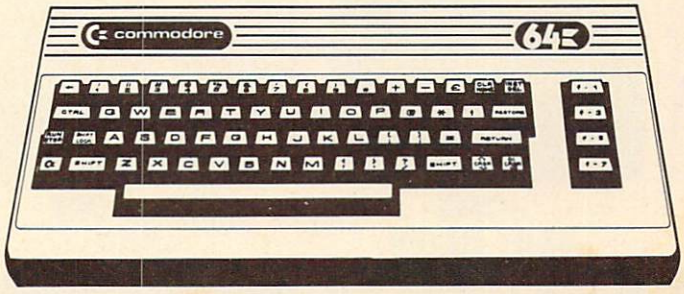
VIC 20 and **Commodore 64** are trademarks of Commodore Electronics Ltd.

BUSIWRITER

BUSIWRITER A Honey of a Word Processor

Why word processors?

Word processors allow the user to quickly and easily create letters, memos, notes, reports, term papers, manuals, poetry and any other written information using the memory of the computer as a pencil and paper. The computer display or terminal acts as a window through which the user views the information as it is entered. The outstanding advantage of using BUSIWRITER is that it acts not only as a pencil and paper but as a perfect eraser and automatic typewriter.



For Commodore 64™
Commodore 1515, 1525, Epson, C. Itoh, Qume, Diablo, NEC Spinwriter, Starwriter, Prowriter, Okidata, Microline, Gemini-10
And many more printers

BUSIWRITER The Queen Bee of Word Processors

BUSIWRITER allows the user to quickly and easily make any number of alterations to the text. BUSIWRITER will instantly reformat your text and show you exactly and continuously how the final output will appear. BUSIWRITER has more functions than any other known microcomputer word processor. With BUSIWRITER assisting in the entry of text, providing a **20 page memory** and performing an enormous number of editing/composing functions, the preparation of written data is far faster and outstandingly more accurate than if it were prepared by hand.

BUSIWRITER With the Sting Removed from the Prices

BUSIWRITER 64 . . . . . only **$99.00** for the CBM 64

BUSIWRITER AVAILABLE NOW FROM YOUR LOCAL DEALER
(800) 227-9998
FOR THE NAME OF YOUR NEAREST DEALER

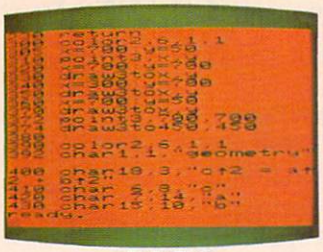
California, Canada, Alaska and Hawaii please call (415) 965-1735

Skyles Electric Works
231G South Whisman Road
Mountain View, CA 94041

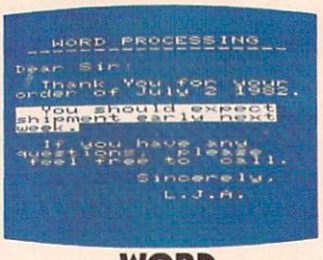
Europe please contact Supersoft, Winchester House, Harrow Wealdstone, England HA3 7SJ, Tel. 01 861 1166

IF YOU OWN A COMMODORE VIC YOU KNOW IT CAN DO ALL THIS.

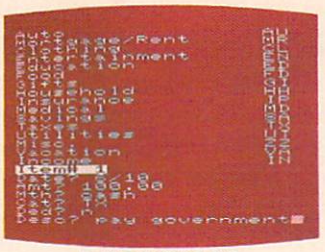
SE
AR



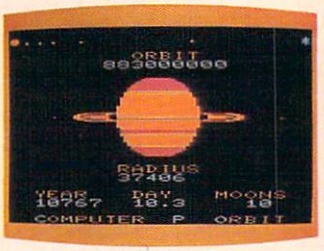
BASIC PROGRAMMING



WORD PROCESSING

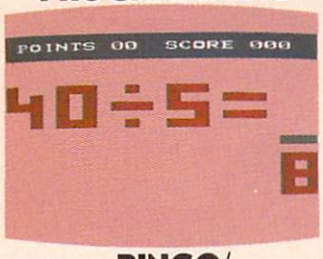


PERSONAL FINANCE

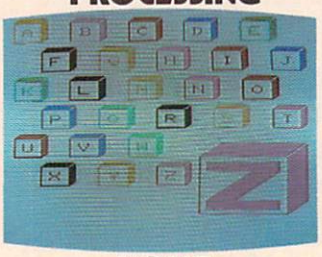


VISIBLE SOLAR SYSTEM

IG
OR

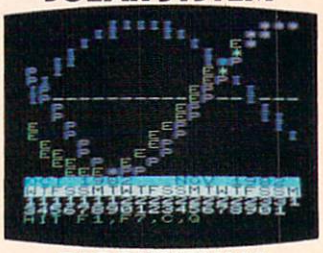


BINGO/ SPEED MATH



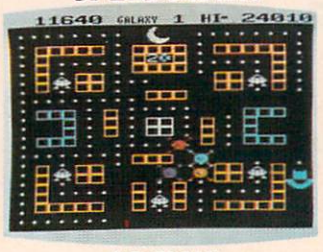
HOME BABY SITTER

ADVENTURE LAND

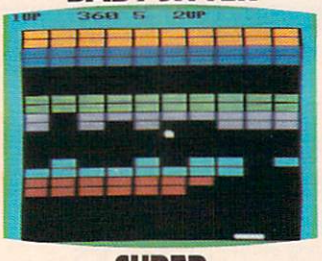


BIORHYTHMS

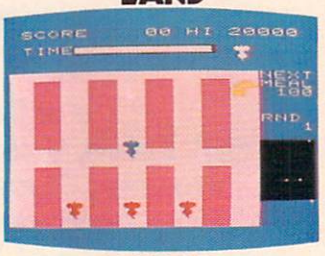
E
ORY



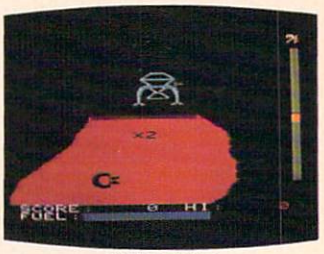
COSMIC CRUNCHER



SUPER SMASH

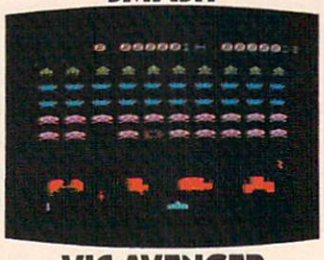


RADAR RAT RACE

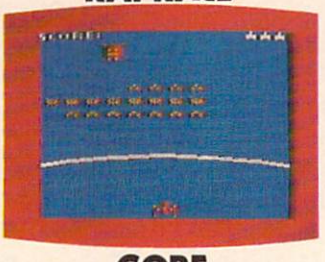


JUPITER LANDER

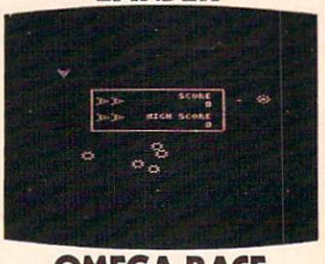
CHESS



VIC AVENGER



GORF



OMEGA RACE

What those extra few dollars get you is a simple little device called a Commodore VICMODEM.

It connects your telephone to your VIC 20™ or Commodore 64™ computer (resulting in something aptly called telecomputing), giving you access to information such as you see on the screens to your right.

Normally, you'd have to type a short program into your computer to help it make the final transition into a telecomputer.

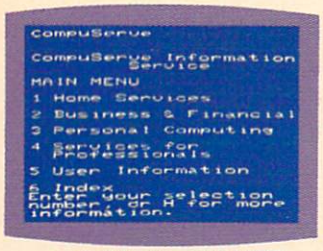
However, when you buy a VICMODEM, you'll find we've included a free software program. You just load it into your Commodore Datassette Recorder, and presto (give or take a moment or two), you have access to a vast library of information and games.

Speaking of free, Commodore also includes a free subscription and a free hour's time on CompuServe™ and Dow Jones News/ Retrieval Service,® a free trial offer on The Source,℠ and a discount program offer with Comp-U-Store and General Videotex Corp.

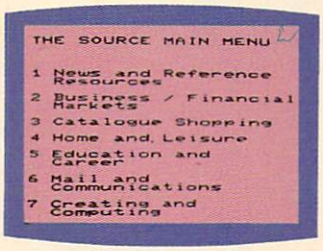
Let's see. Did we leave anything out? Oh, yes. Along with CompuServe comes a free membership in the Commodore Information Network. This is your HOTLINE to Commodore. (How often do you get to speak directly to a manufacturer?) Through it we

Certain offers subject to change. CompuServe is a trademark of CompuServe, Inc. and H.&R. Block Co. Dow Jones News/Retrieval Service is a registered trademark of Dow Jones & Co., Inc. The Source is a service mark of Source Telecomputing Corporation, a subsidiary of Reader's Digest Corporation, Inc.

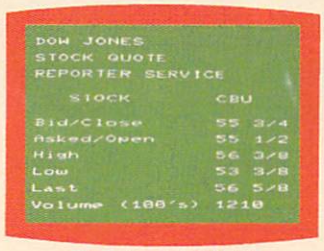
BUT DID YOU KNOW FOR ABOUT $100 YOU CAN ALSO GET IT TO DO ALL THIS?



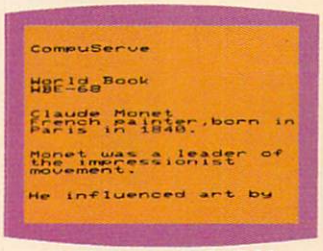
COMPU SERVE™



THE SOURCE™

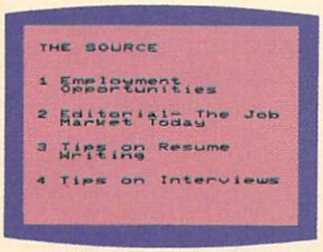


DOW JONES®

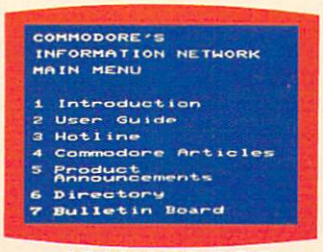


WORLD BOOK ENCYCLOPEDIA

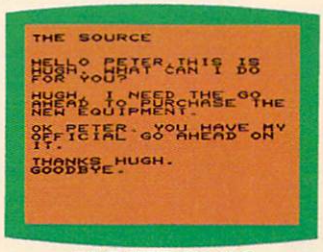
STO



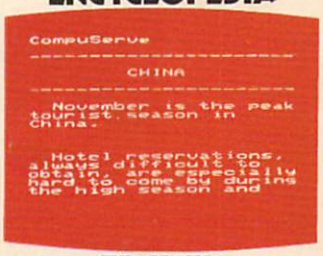
EMPLOYMENT OPPORTUNITIES



COMMODORE INFO. NETWORK

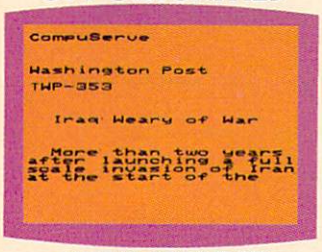


ELECTRONIC MAIL

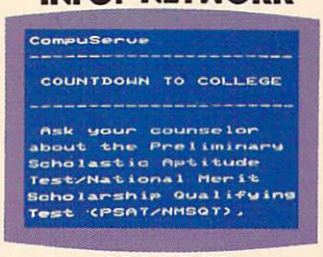


TRAVEL INFORMATION

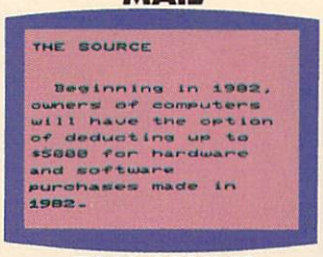
MO



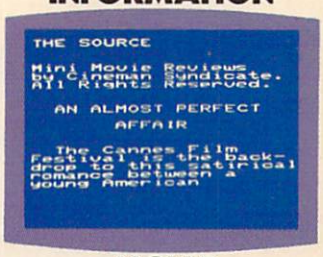
WIRE SERVICE NEWS



COLLEGE PLANNING

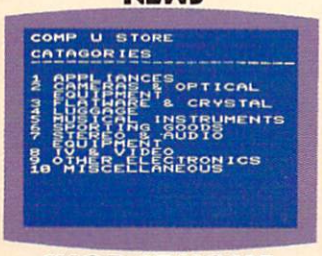


TAX ADVICE

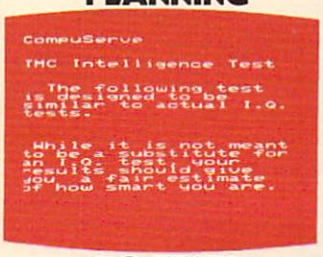


MOVIE REVIEWS

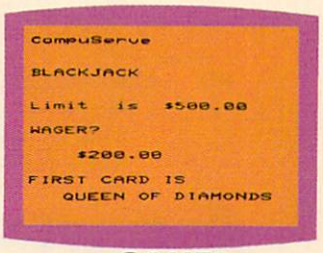
ACCE CO



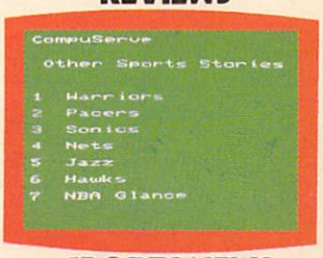
SHOP AT HOME



I.Q. TESTS



GAMES



SPORTS NEWS

BUSI

can answer any questions you might have about your computer, or programming, or anything else Commodore-related, via electronic mail.

The Commodore Information Network is also your direct line to the Commodore Bulletin Board, which Commodore owners use to keep in touch with each other, for programming tips, Public Domain Software, and technical support.

THE COMMODORE VIC 20, A REAL COMPUTER FOR THE PRICE OF A TOY.

Altogether, these little extras we've included with our VICMODEM add up to a value of $197.50. A nice return on an investment of around $100.

Some computer companies think it's reasonable to ask as much as $500 for telecomputing capabilities such as ours.

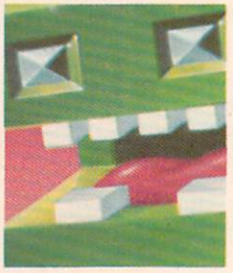
However, with the Commodore VICMODEM selling for around $100, we feel we're being a whole lot more reasonable. Don't you agree?

commodore COMPUTER

These are just a few examples from our existing library of software available for the VIC 20. U.S.A.–P.O. Box 500, Conshohocken, PA 19428; Canada–3370 Pharmacy Avenue, Agincourt, Ontario, Canada M1W 2K4.

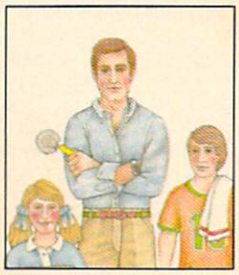
commodore power/play
home computing
SPRING 1983

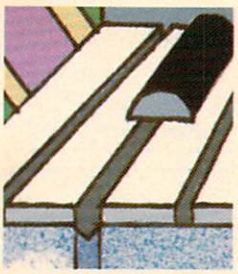
features



departments

commodore power/play
home computing
SPRING 1983





staff

Publishing Manager
Neil Harris
Editor
Diane LeBold
Technical Editor
Jim Gracely
Staff Writers
John Campbell
Joe Ferrari
Jeff Hand
Paul Higginbottom
John O'Brien
Mike Tomczyk
Tom Ziegler
Contributing Writers
Mike Abercrombie
Jim Butterfield
Betsy Byrne
David Funte
John Heilborn
Larry Holler
Myron Miller
Technical Staff
Jeff Bruette
Rick Cotton
Andy Finkel
Bill Hindorff
Cyndie Merten
Advertising Manager
Diane LeBold
Advertising Coordinator
Sharon Steinhofer
Circulation Manager
John O'Brien
Circulation Assistant
Kathy Reigel
Graphic Design
Neumann Greenberg Schlenker
Cover Art
Tom Hallman
Printing
Oak Lane Printing

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 1.

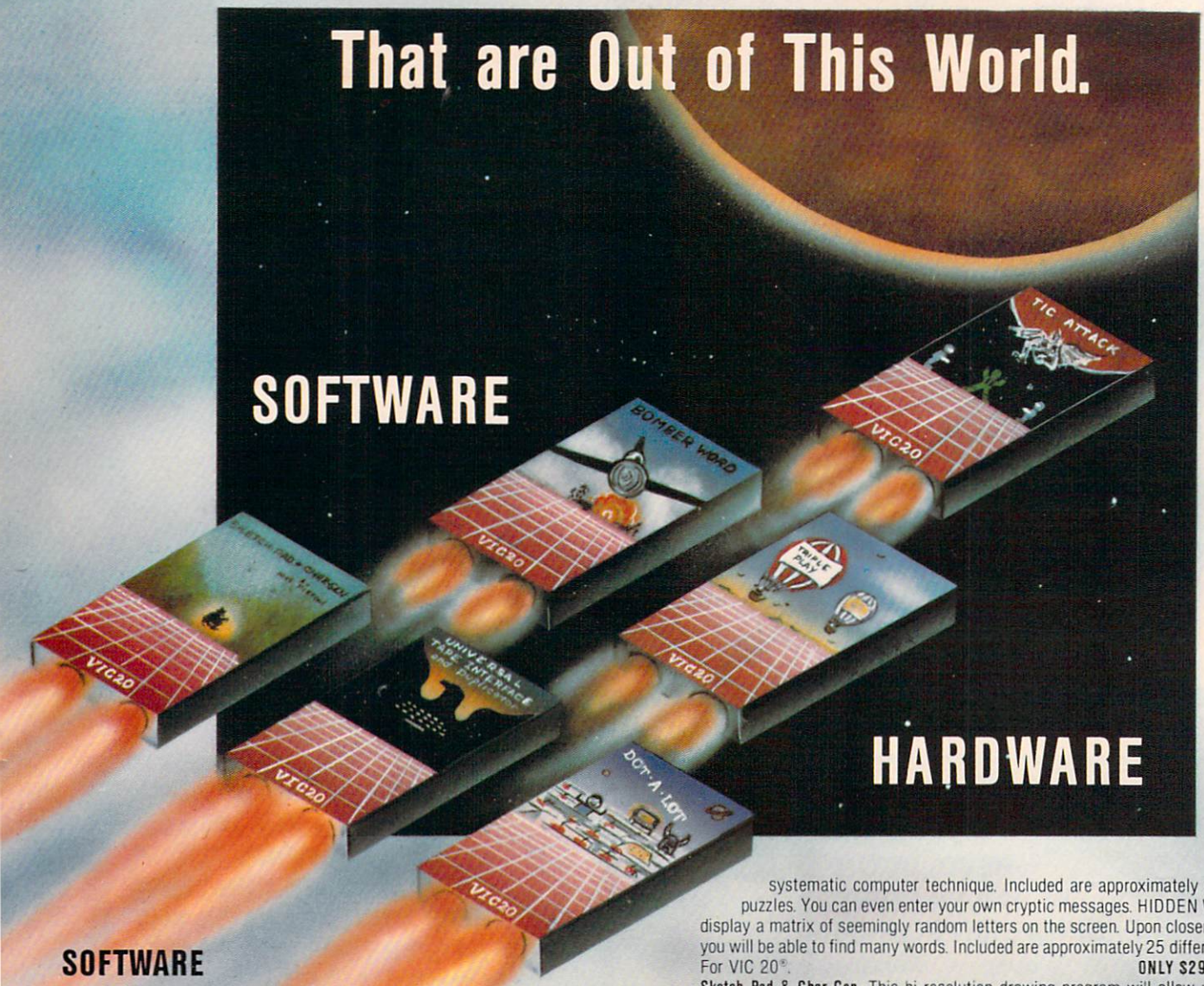
Subscription Information: U.S. subscriber rate is $10.00 per year. Canadian subscriber rate is $15.00 per year. Overseas $17.00 per year. Questions concerning subscriptions should be directed to Commodore Business Machines, Magazine Subscription Department, Box 651, Holmes, PA 19043. Phone 215-345-8112). (Pennsylvania phone 1-800-662-2444.

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

Products for Commodore 64™ and VIC 20™



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 (supports disk). **$34.95.**

ZAP!–Climbing the corporate ladder could be fun except for all that falling paperwork. This Hires arcade type game 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**

Sketch Pad & Char-Gen–This hi-resolution drawing program will allow you to draw pictures in detail. Use either the keyboard or optional joystick. A fill command will allow you to fill a block and other commands allow you to easily clear the screen. You can also save and load pictures. Char-Gen is a simple to use custom character generator that will allow you to design different characters for each printable key on the computer. This program is an excellent device to design game creatures, foreign alphabets, secret symbols, or other special characters. One set is included and you can make and store others quite easily. Both for VIC 20®. **ONLY $24.95**

HARDWARE

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

Universal Tape Interface & Duplicator–(Use on the CBM 64® and VIC 20®). With this device, you can easily load, save or even duplicate tapes easily with your recorder. Full 3 LED indication of Data transfer makes this the most reliable way to Load, Save and Duplicate. A complete I/O device with extras. NOTE: Duplication requires 2 recorders. **Only $49.95**

Universal Parallel Interfaces–Now you can use most any parallel Centronics® type printer with your VIC 20®/CBM 64®. The inexpensive model will allow you to access your printer through the user port. This cable and driver is only **$19.95**. Our other model from TYMAC is more extensive with graphic capabilities. Call or write for more information and prices.

MasterCard VISA

Dealer and Distributor Inquiries Invited

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

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

letters

Dear Commodore,

I am writing to tell you about my brother Bob. He has done so much for your company and you don't even know it!

I have a very large family—five brothers and three sisters. Your computer is all my family talks about. My brother and I used to play hockey all the time; now we play video games all the time!

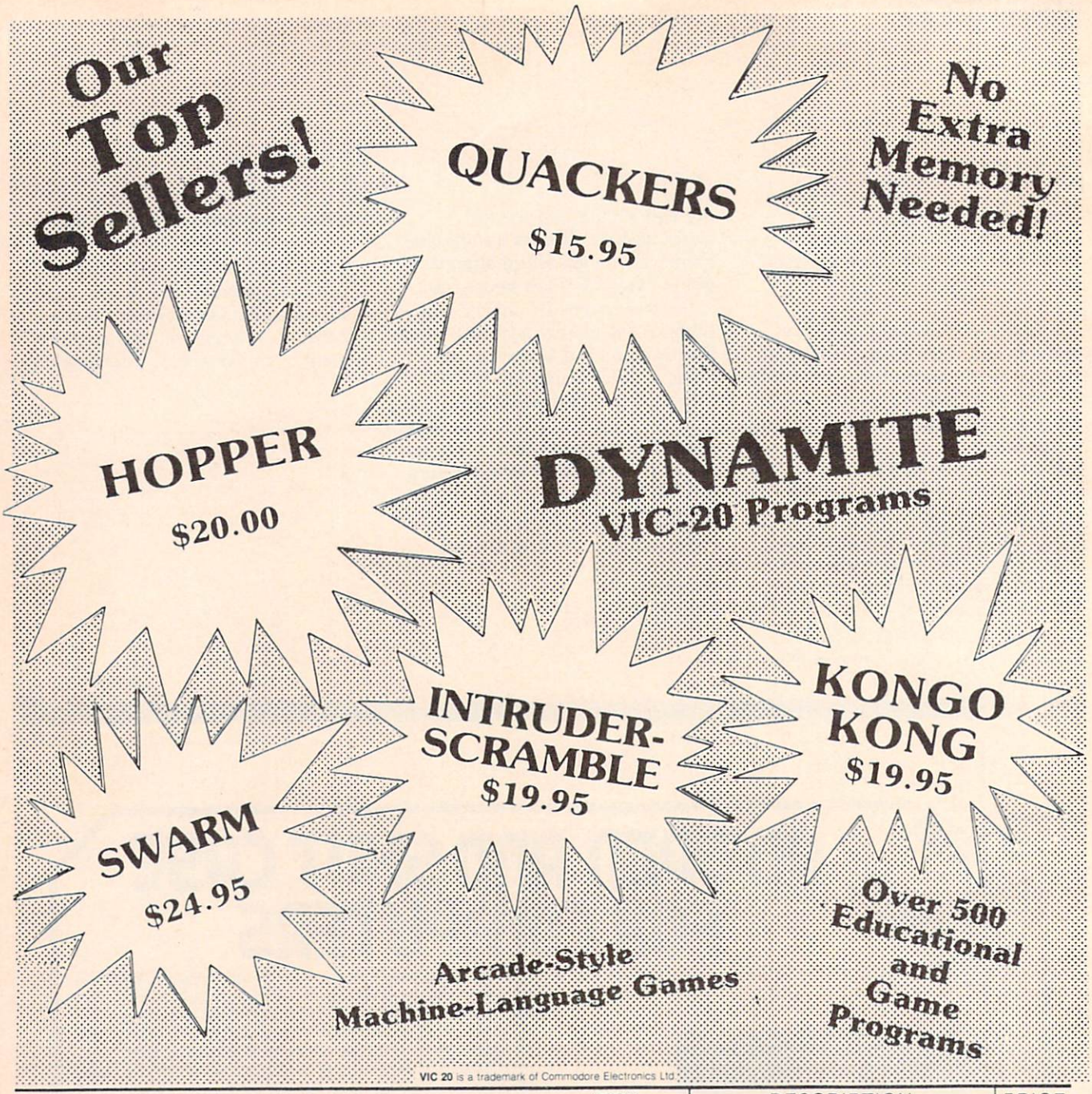
Back to my brother Bob. He bought a VIC about nine months ago. When he found all the amazing things it could do, he showed it to brother Rick. He bought one. Bob showed it to sisters Terry and Kathy. They both bought one! He showed it to my mom and she loved it, but couldn't afford one. But never fear, my dad bought her one for Christmas! He showed it to brother Frank. Come tax return time there will be one in Frank's house, too!

We're a seven-VIC family! And all of this is still not counting the several friends who have bought or are thinking about buying a VIC 20.

Everyone in my family loves their VIC, but Bob is a VIC fanatic. If he's late for his wedding later this year, I know it'll be because he's not done with his VIC! Some days he sits in front of it from morning until late at night, stopping only for dinner!

Sincerely,
Paul Jones

Thanks for letting us know about your brother, Paul. We're sending Bob complimentary subscriptions to our user magazines so he can keep up on all the latest developments here at Commodore. We wouldn't want a fan like him to miss anything.
C



VIC 20 is a trademark of Commodore Electronics Ltd.

ORDERING BLANK

To: American Peripherals
122 Bangor Street
Lindenhurst, N.Y. 11757

Ship to: Name _____

Street _____

Town, State, Zip _____

☐ *Check here for latest 64 Catalog*

☐ *Check here for latest VIC Catalog*

ITEM	DESCRIPTION	PRICE
	NY State Residents only add 7¼% tax	
	Shipping (If COD, add $1.50)	$1.50
	TOTAL AMOUNT	

U.S. Funds - Foriegn Countries add an additional $3.00

Beginnings

This issue marks both an end and a beginning. First, it's the end of our first year of publishing *Power/Play*. I won't bore you with the details of how far we've come in that year, but those of you who have been with us since the beginning know how we started, back in June of 1982, with a tentative 48 pages that has grown since then into the book you hold in your hand. Our readership has grown as well, from zero to almost 100,000—which, no matter how you look at it, is pretty phenomenal.

At the risk of sounding a little mushy, I'd like to thank everyone who helped us get here—writers, artists, technical support people, printers, Commodore management, our loyal publishing staff—and, of course, our readers.

So much for endings. Now for the "beginning" part—that's always more exciting, anyway. Since many of you are undoubtedly reading us for the first time (our readership has grown considerably since our last issue), you have no way of knowing how we've changed, just since December. With this issue we're taking on a new look and a new way of organizing material for both *Power/Play* and *Commodore: The Microcomputer Magazine* (you'll be pleasantly surprised by that one in just a few weeks). We're also beginning our first forays into the world of mass retail sales, so our million-plus users will be able to find us more easily in the near future.

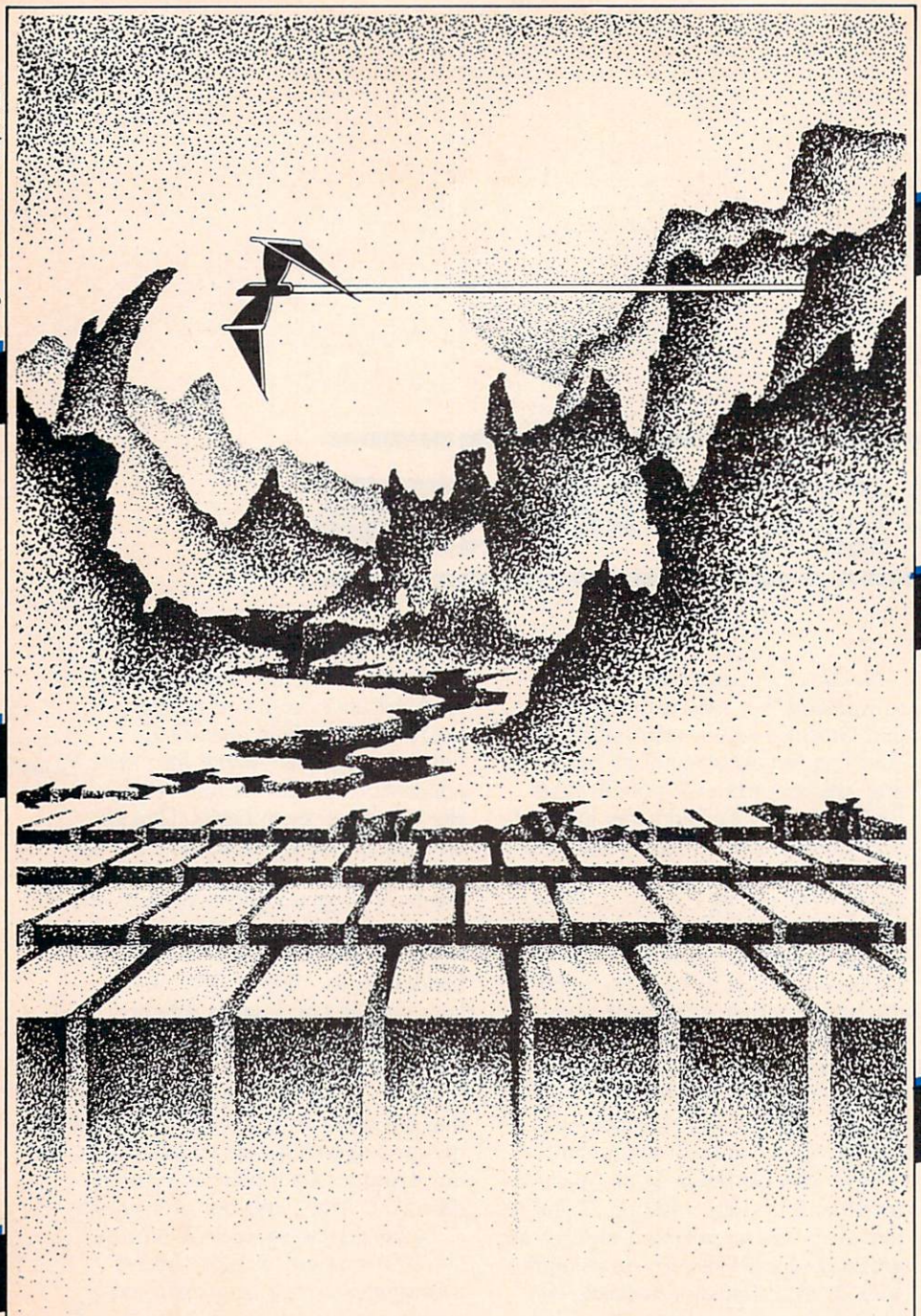
Yet another beginning—we'd like to welcome our new technical editor, Jim Gracely, who is responsible for checking the accuracy of all programs and articles before we publish them. Right now he's tucked away in an office recently abandoned by our software department (we're in the process of moving), so I hardly ever see him. When I do get back there he's usually engrossed in something on his monitor—which happens to be the hugest monitor I've ever seen, except maybe for the one that dominates Neil Harris' office—so I don't bother him much. Just remember when the programs we publish work, you can thank Jim.

Now let's get down to the really interesting stuff—like what you'll find in this issue. We're delighted to have an article by John Heilborn on using the VIC 20 as a mediator in doling out household chores. John is an editor with Osborne/McGraw-Hill and co-author of the *VIC 20 User Guide*. We're also featuring the continuing saga of SID (the Sound Interface Device that makes the Commodore 64 a music synthesizer) by our SID whiz Paul Higginbottom and a wonderful little program from Jim Butterfield that turns that synthesizer into a back-country "skiffle band." And we have more program listings than ever before, thanks to our enthusiastic readers, who've been sending us material that is often terrific.

Which brings us to another interesting point. We do accept material for publication from our readers. We're particularly interested in original programs—either for practical applications or just for fun—and we like "tutorials" even better. (Jim Butterfield's article in this issue is a good example of a tutorial.) If you have an article or program you think is worth sharing with other readers, drop us a note and ask for a copy of our Guidelines for Writers. The Guidelines will tell you how to submit material so our editors can deal with it most efficiently.

I'd like to wind up with a reminder that the number of Commodore user groups is growing ever larger. Check our listing to see if there is a group near you. You'll find our user groups to be invaluable sources of information and support, whether you're a computer novice or a seasoned expert. **C**

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

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
THE INTERBANK CARD

Using Those Mysterious Programmable Function Keys

by
Michael S. Tomczyk
Product Marketing Manager
(Reprinted from *Commodore Magazine*, June, 1982)

Everyone always asks how the VIC 20's programmable function keys work (The function keys are those large yellow keys on the right side of the VIC 20 keyboard).

Special function keys were added to your VIC to let you take advantage of "one-key" programming features normally found on much more sophisticated (and expensive) office computers. It doesn't take much effort to program these keys and that's what the following lesson shows you.

When you first got your VIC 20, you were probably surprised that nothing happened when you pressed the function keys. That's because the function keys are **programmable,** meaning you have to program them to do something before they work.

Before getting into the programming part, let's take a closer look at the keys themselves.

To begin with, the function keys are numbered 1 through 8. The odd-numbered keys 1, 3, 5, and 7 are obtained by simply typing those keys, and the even-numbered keys 2, 4, 6 and 8 are obtained by holding down the SHIFT key and then typing the appropriate key. This lets you use four keys to get eight separate functions.

Now, here's an important point . . . each key has a special NUMBER which you must use when programming that key. This number is called the **CHR$ NUMBER.** All of the VIC's keys have CHR$ numbers, which are listed in the ASCII AND CHR$ CODES table on page 146 of your VIC user's guide (or page 273 of the *VIC 20 Programmer's Reference Guide).*

CHR$ CODES FOR FUNCTION KEYS

FUNCTION KEY NUMBER	CHR$ CODE
f1	CHR$ (133)
f2	CHR$ (137)
f3	CHR$ (134)
f4	CHR$ (138)
f5	CHR$ (135)
f6	CHR$ (139)
f7	CHR$ (136)
f8	CHR$ (140)

Note that the CHR$ numbers are not in exact order . . . the odd numbered keys are numbered 133-136 and the even numbered keys (which you must SHIFT to use) are numbered 137-140.

Using the GET Statement—"Hit Any Key"

Before we explain how everything works, let's write a short program using a function key, to show you how it works. The first thing we do is TELL THE VIC TO LOOK AT THE KEYBOARD. This instructs the VIC to check the keyboard to see if you have pressed a key. The BASIC command for this is:

```
10 GET K$:IF K$ = " " THEN GOTO 10
```

This is called a "GET Statement" and is usually placed as a single line in your program. If you include this line all by itself in your program, the VIC will look for ANY KEY to be hit. So let's write a short program which tells the user to "HIT ANY KEY TO BEGIN." Enter this program, type the word RUN and hit the RETURN key:

```
10 PRINT "HIT ANY KEY TO BEGIN":PRINT
20 GET K$:IF K$ = " " THENGOTO20
30 PRINT " ♥ R PROGRAM BEGINS."
```

(BEGINNERS: be sure to hit the RETURN key at the end of each line to enter it.)

THIS MEANS HOLD DOWN THE **SHIFT** KEY AND TYPE THE **CLR/HOME** KEY

THIS MEANS HOLD DOWN **CTRL** AND TYPE **RVS ON** KEY

See how it works? Now let's go one step farther. After we BEGIN our program, we may want to make the program WAIT until the user is ready to continue. This technique is often used in educational programs when you want to give a student time to study something on the screen before moving on. Hold down the RUN/STOP key and hit the RESTORE key to "exit" your program. Now type these additional lines (the VIC automatically adds them to the program above, which is still in VIC's memory):

```
40 PRINT" ♥ WAIT HERE.":PRINT:PRINT
   "HIT ANY KEY TO CONTINUE.":PRINT
50 GETK$:IFK$" "THENGOTO50
60 PRINT" ♥ R PROGRAM CONTINUES."
```

Note that this program now uses the same GET statement in two different places to check the keyboard to see if a key has been pressed. You can insert line 20 almost anywhere to make the VIC wait until a key is pressed . . . but if you do this in the middle of a program, don't forget to PRINT a little message telling the user to "hit any key to continue."

Now let's take a closer look at how the GET statement lets us check the keyboard . . . and program our function keys.

More Information About the GET Statement

As already explained, the **GET K$** line tells the VIC to check the keyboard to see if a key has been pressed. There are three important things to remember about using this line in your program. First . . . notice how we used the **string variable K$** in lines 20 and 50 above? Although we used K$ as our variable (K for "Key"), you can use **any legal string variable,** such as A$, KK$, B1$, etc., in this line. For example, you could use:

```
50 GETRR$:IFRR$""THENGOTO50.
```

The second thing to remember is that the GET statement line **always GOES BACK TO ITSELF.** In other words, if you put this line at line 100, the last part of the line would read . . . **THEN GOTO 100.** This makes the VIC keep checking over and over again until a key is actually pressed. So the GET statement includes a GOTO to its own line number.

Finally, don't attach any other BASIC commands on the same line as your GET statement. For example, don't put a colon and add more BASIC commands to lines 20 or 50 above.

Keys Continued

Programming a Function Key

Now let's make a function key do something. How about amending our program above so that it only begins if you type function key 1, and only continues if you type function key 5 . . .

We'll use the same program we used above and simply modify two lines and add a few extra lines to tell the VIC to accept only designated function keys instead of "any key." To begin, type the word LIST and hit RETURN. This displays your program. Now modify your program by changing or adding the elements shown in boldface in the finished program below (If you need help editing program lines, see page 74 of the VIC 20 *Programmer's Reference Guide*, "Editing Lines.")

```
10 PRINT "HIT THE F1 KEY TO BEGIN":PRINT
20 GET K$:IF K$=""THENGOTO20
25 IFK$<>CHR$(133)THENGOTO20
30 PRINT" [♥][R] PROGRAM BEGINS"
40 PRINT"WAIT HERE.":PRINT:PRINT"HIT F5
   KEY TO CONTINUE.":PRINT
50 GETK$:IFK$=""THENGOTO50
55 IFK$=CHR$ (135)THENGOTO60
56 GOTO40
60 PRINT" [♥][R] PROGRAM CONTINUES."
```

LINE 25: Line 25 tells the VIC: "If you check the keyboard and find any other key but the f1 key being pressed, GOTO line 20 and check again. The < > sign means less or greater than (or you might interpret this as "not equal to") . . . in this case, the program treats CHR$ (133) as the number 133.

With this example, your "real" BASIC program would start at line 30. Naturally, you can alter (increase) the line numbers after line 30 to give yourself more room if you're inserting a longer BASIC program here.

CHR$ (133) is the CHR$ number for the f1 function key, and IF K$ < > CHR$ (133) means "IF f1 IS NOT BEING PRESSED." You can designate a different key by changing the CHR$ number (see page 14). For example, CHR$ (139) changes the key to f7.

LINES 55-56: Line 55 is the IF . . . THEN statement which designates the f5 key to be pressed and line 56 keeps the program "looping back" to the message in line 40 until the f5 key is pressed.

Programming Function Keys to Perform Functions

So far, we've used function keys to 1) start your BASIC program, and 2) continue a BASIC program in progress. Both of these techniques used the function keys to add some nice cosmetic touches to your BASIC programming, but they really didn't show you how to use the VIC's function keys to perform "real" functions, so that's what we're going to show you next.

Type the word NEW and hit RETURN to erase your previous program, and type in the following program. This is a VIC speaker demonstration which lets you type any note value and hear it played on any of the VIC's four internal "speakers." Function key 1 plays the note you entered on speaker 1 (the lowest tone speaker), function key 2 plays the note on speaker 2, and so on. Speaker 4 is the "white noise" or "sound effects" speaker.

```
10 PRINT" [♥] SOUND DEMO":PRINT:
   POKE36878,15:S1=36874:
   S2=36875:S3=36876:S4=36877
20 PRINT" [♥] "
30 PRINT"F1 PLAYS SPEAKER 1":PRINT
40 PRINT"F2 PLAYS SPEAKER 2":PRINT
50 PRINT"F3 PLAYS SPEAKER 3":PRINT
60 PRINT"F4 PLAYS SPEAKER 4":PRINT
70 PRINT"F5 LETS YOU ENTER A NEW NOTE
   NUMBER":PRINT
100 PRINT"INPUT A NOTE NUMBER BETWEEN
    128 AND 255":INPUTA
120 GETM$:IFM$""THENGOTO120
130 IFM$=CHR$ (133)THENX=S1:GOTO500
140 IFM$=CHR$ (137)THENX=S2:GOTO500
```

Standard VIC 20™

no additional memory needed

(CG008) Alien Panic $12.95
Race against time as your guy digs holes to trap aliens in 4 floor laddered, brick construction site. Requires joystick.

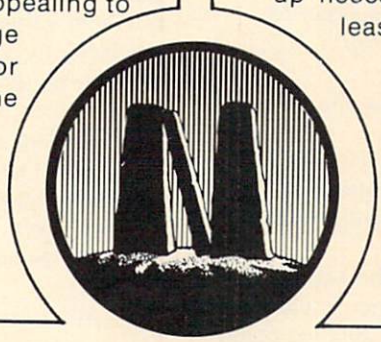
(CG096) Antimatter Splatter $24.95
This game is as good as its name. Another pure machine code game, this one is fast! The alien at the top of the screen is making a strong effort to rid the world of humankind by dropping anti-matter on them. The splatter cannon and you are our only hope as more and more antimatter falls. Joystick again is optional equipment.

(CG026) Collide $12.95
"Vic" controls one, you the other as cars go opposite directions on 4 lane track. Requires joystick.

(CG094) Exterminator $24.95
Recently scoring a rating of 10 out of a possible 10 this game was praised as "one of the best I've seen on any computer" by a prominent reviewer in a leading magazine. The idea is to shoot a centipede before it overuns you, the problem being every time you hit it, it divides into two separate shorter ones. Several other little creatures bounce around during this struggle. All of them lethal. 100% machine language makes the rapid fire action very smooth. A joystick is optional, but as always, recommended, (a trac ball is also very nice!).

(CG054) Krazy Kong $12.95
Three screens, a gorilla, barrels, and changing difficulty levels help to make this one of our most popular. Joystick optional.

(CG098) Racefun $19.95
Extensive use of multicolored character capabilities of the "Vic" make this one very appealing to the eye. Fast all machine language action, quick response to the stick or keyboard controlled throttle, combine with the challenge of driving in ever faster traffic to make it appeal to the rest of the body. Joystick controlling is an option.



(CG058) Rescue From Nufon $12.95
Must find 30 hostages in this 100 room, 5 story, alien infested, graphic adventure game. A continual big seller. Keyboard only (n. = north w = west etc.)

(CG068) The Catch . . . $12.95
Another all machine language game based on the principle that one person with one joystick guiding one catch/shield can catch everything that one alien can throw at one. The action comes slowly at first but by the fourth wave you'll be aware of . . . "The Catch" . . .

Expanded Memory Vic 20™ Games

(CG090) Defender On Tri $19.95
Pilot a defender style ship on mission to save trapped scientists from a fiery fate (they are aboard an alien vessel deep in the gravity well of sol). Excellent graphics. Short scene setting story in the instructions. "Defender On Tri" requires at least 3K added memory.

(CG092) 3D Man $19.95
The maze from probably the most popular arcade game ever, with perspective altered from overhead to eye level. The dots, the monsters, the power dots, the side exits, the game is amazing. "3D Man" requires at least 3K added memory.

(CG088) Space Quest $19.95
Our first 8K memory expander game and its a beauty. The scene (a short story is included) is far in the future, a time when man's knowledge has reduced an entire galaxy into a mapped series of quadrants. This game has stratagy (you plot your own hyperspace jumps on Galaxy map), action (against a starry background you find yourself engaged in a dogfight, laser style), exploration (you must fly your ship deep into caverns to pick up necessary fuel). "Space Quest" requires at least 8K memory expansion and a joystick.

Commodore 64™

(CG602) 3D-64, Man $19.95
This available on the expanded "Vic 20" game, has been completely rewritten for the 64 and uses sprites, sounds, and other features not available on the "Vic". This one requires a joystick.

NÜFEKOP

P.O. Box 156, Shady Cove, Oregon 97539-0156 Mastercard and Visa cards accepted **C.O.D. Orders...call (503) 878-2113**

VIC 20 and **Commodore 64** are trademarks of Commodore Electronics Ltd. Games will be on tape unless you request disk. Ask for our FREE catalog!

Keys Continued

```
150 IFM$ = CHR$ (134)THENX = S3:GOTO500
160 IFM$ = CHR$ (138)THENX = S4:GOTO500
170 IFM$ = CHR$ (135)THENGOTO10
180 GOTO120
500 FORT = 1TO200:POKEX,A:NEXTT:POKEX,
    0:GOTO120
```

Type RUN and hit RETURN.

Now, this little program is just one example of how you can assign different function keys to perform different functions. In this case, we followed some of the general rules we've already established, and introduced a couple of new programming techniques. Let's examine the program, line by line.

LINE 10: contains our "opening message." We also took advantage of the extra space on the line to turn the volume to its highest level (POKE36878,15) and assigned some easy-to-remember variables to each of our four speakers, S1, S2, S3 and S4. Obviously, it's easier to type S1 than 36874, and it also saves memory.

LINE 20 is a blank line.

LINES 30–70 provide instructions for using the program.

LINE 100 is a special INPUT instruction which asks for a NOTE VALUE to be entered into the computer. The INPUT statement assigns the value A to whatever number was typed in by the user.

LINE 120 is our GET statement to check the keyboard.

LINES 130–160 are IF . . . THEN statements which match the messages in lines 30—60. Notice that we could have written EACH LINE like this:

```
IFM$ = CHR$ (133)THENFORT = 1TO200:
POKES1,A:NEXTT:POKES1,0:GOTO120.
```

Instead, we wrote a more efficient program which put a generalized sound routine on line 500 so all we have to do in lines 130—160 is define X as the proper speaker and GOTO line 500 to execute the note, then jump back (GOTO) to line 120 to check the keyboard again.

LINE 170 tells the VIC to go back to the beginning of the program and ask for another note value if the user types f5.

LINE 180 tells the VIC to go back and check the keyboard.

LINE 500 contains a time delay loop which specifies how long the note is played when you press the function key (try changing the number 200 to something else). POKE X,A means to POKE the speaker defined as X in lines 130—160 with the note defined as A by the INPUT statement in line 100. In other words, if you entered the number 201 and pressed function key 1, the VIC responds by playing the "D" note from its middle octave. (see your user's manual for note values)

For More Advanced Programmers

There is ANOTHER way to check the keyboard to see which KEY is being held down, using the command **PRINT PEEK (197)** or **PRINT PEEK (203).** You can PEEK into either of these special memory locations to find out which key is being held down. Note that by key we mean the physical key being held down, not the symbol. In other words, f1 and f2 are interpreted as being **the same key** because we are detecting the physical key is being mechanically held down. The values for the function keys are given below (a complete chart of values for all keys is given on page 179 of the *VIC 20 Programmer's Reference Guide).*

VIC Key	Value Returned When You PEEK (197) or PEEK (203)
no key	64
f1/f2	39
f3/f4	47
f5/f6	55
f7/f8	63

The general principle involves PEEKing to see if a key is being held down. If the value returned matches the value of a specific key, you make your program perform a specific action. For example, you might write a music program which tells the VIC to play certain notes when certain keys are being held down, and to stop playing those notes when the keys are released. Here's a short program to start you off . . . when you RUN this program, it will play a note when you hit the f1/f2 key. If you get the PEEK values for ALL the keys from the *Programmer's Reference Guide*, and match them to the proper note values from the *Table of Musical Notes*, you can write your own "VIC PIANO!"

```
10 POKE 36878,15:S1 = 36874:S2 = 36875:S3
   = 36876
20 IFPEEK (197) = 39THENFORT = 1TO50:
   POKES1,200:NEXTT:POKES1,0
30 IFPEEK (197) = 47THENFORT = 1TO50:
   POKES2,200:NEXTT: POKES 2,0
40 IFPEEK(197) = 55THENFORT = 1TO50:
   POKES3,200:NEXTT:POKES3,0
50 GOTO20
```

Editor's Note: See the December, 1982 *Power/Play* for yet another version of the VIC Piano.

The key elements in line 10 are turning on the volume and defining the "speakers" as S1, S2 and S3. In lines 20—40 we PEEK (197) to see what key is being held down, insert a time delay loop (1to50) to specify how long each note is held down, then POKE a note value into the speaker we want. Then we turn off the speaker (otherwise it would keep playing). Change the number 50 to a higher or lower number to increase or decrease the duration each note is played.

Summary

This introduction to the VIC's programmable function keys is only a beginning. You can probably design programs which do much more than those described here. For example, you could make the function keys stand for different colors, and instead of POKEing note values, designate special keys to POKE different color combinations using the screen and border color chart in your user's guide.

You could specify each function key to perform a different complex calculation by combining the function keys with DEF FN statements. You might want to INPUT a series of numbers and perform several multi-step calculations which you can execute simply by pressing the function keys.

You might match the function keys to program subroutines which allow the user to access different portions of a long complex program, so he can jump from a main program into a smaller subroutine, and then come back to the main program . . . all by pressing one or two function keys.

Finally, you may have already discovered that programmability isn't limited to the VIC's function keys. In fact, ALL the VIC's alphanumeric and graphic keys are programmable, using the techniques described here in combination with the CHR$ values for each key. To program a key to perform a special function, simply use that key's CHR$ value just like the function key values described here.

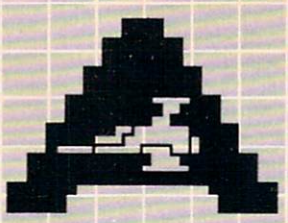
If you develop an imaginative application for the VIC's special function keys, drop a letter to The VIC Magician in care of our magazine and we'll share your discoveries with the rest of the "VIC world." **Enjoy!** **C**

Two COMMODORE cartridge products which offer built-in function key programming include the SUPER EXPANDER cartridge and the PROGRAMMER'S AID CARTRIDGE. Both cartridges let you redefine the pre-programmed function keys by typing "KEY" and hitting RETURN. See your Commodore Dealer for more information.

TODAY'S MENU

For Your VIC 20™

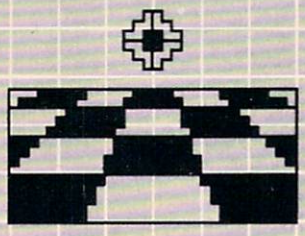
Games



ASTROBLITZ
Protect your planet by destroying enemy saucers.

TRASHMAN
Drive the garbage truck and empty the city's trash cans. But watch out for the flies.

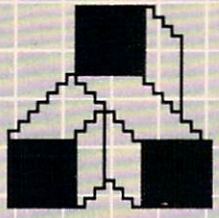
CITY BOMBER
Level a city to make it easy to land. Take off and do it again.



ACTION GAMES
SEAWOLF, BOUNCEOUT, or VIC TRAP. You'll need sharp eyes and quick hands for these.

Education

EDUCATIONAL/RECREATIONAL (I & II)
Put fun into learning math, spelling, and spatial relationships.



LOGIC GAMES
The computer is thinking. You should be, too.

Choose an Entree:

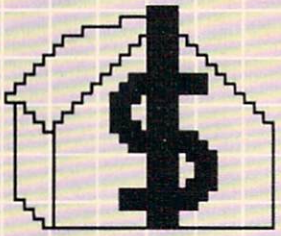
Maybe it's a colorful and challenging game like ASTROBLITZ, TRASHMAN, or CITY BOMBER; perhaps an educational game like HANGMAN or MATH HURDLER; maybe a basic diet of household concerns like HOME INVENTORY, HOUSEHOLD FINANCE, or DECISION MAKER.

Take your pick. All you have to supply is your VIC™ 20 and your own ideas about how to put our software to use. We supply the rest: competitive games, educational games, down-to-earth personal programs, and simple instructions that make everything easy to digest.

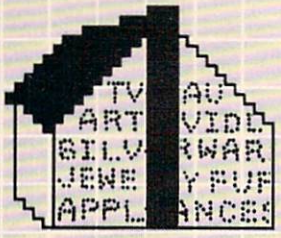
What you see here is part of the menu for today. We'll be adding more, so check our menu from time to time. It's growing, and it's all take home.

Contact your local outlet.

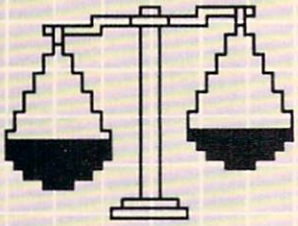
Personal



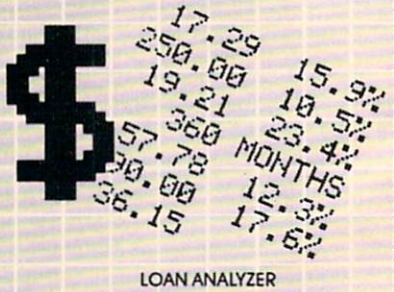
HOUSEHOLD FINANCE
Schedule the family budget, account for expenditures, and face the tax man with a smile.



HOME INVENTORY
Make a record of what you own and revise it as you go. Find out what you've got in seconds.



DECISION MAKER
What should you do about virtually anything? Here's help.



LOAN ANALYZER
Home, car, boat . . . whatever: How much can you afford? Find out.

CREATIVE SOFTWARE
A Division of ASCI, Inc.

VIC 20 is a trademark of Commodore Electronics Ltd.

201 San Antonio Circle, Mountain View, CA 94040 U.S.A. (415) 948-9595

The 64 Skiffle Band

Jim Butterfield, Toronto

OK, so the Commodore 64 has a high class music synthesizer that's great for playing anything from Bach to Bartok. But let's take a break from all the heavy stuff and get down to some foot stompin' fun with Jim Butterfield, right out of the back hills of "Commodore County."

Somewhere up in the hills of Commodore County, there's a jug band and skiffle group that gathers from time to time. I'd like to introduce you to the players, and show you how to program your Commodore 64 to hear them play.

This is a basic band playing basic music . . . so you can write your program in BASIC. Later, you may want to change instruments and add to the group's repertoire. With the 64 sound chip, it's not hard to do.

We're going to be the bandleader, so we must set up the instruments and choose the type of play. Here we go:

100 Print "Music (by Jim Butterfield)"

This identifies the maestro (that's me). Now, let's locate the musicians:

```
110 L1=54272:L2=54279:L3=54286
120 H1=L1+1:H2=L2+1:H3=L3+1
130 V1=L1+4:V2=L2+4:V3=L3+4
```

Each musician stands in a "spot" marked by a number. As band director, we'll call up the music for each one by signalling each player's spot, using his "address." The L's and H's will be used to name the pitch of the notes; the V locations are used for the other stuff.

140 POKE 54296,15

This tells the listeners to turn up their hearing aids, so they can hear the group playing. The highest volume we can set is 15, and that's the one we pick.

150 POKE V1+1,9:POKE V1+2,0

Player 1 is young Billy; he plays the hanging jugs. When he hits a jug with his corncob drumstick it will sound right away, and come out with a good clunking sound (that's the 9, for fast sound attack and fulsome decay). But he can't strum the jug or hold the sound, so the note will fade away soon after he plays (that's the 0, for no sound sustain or release).

160 POKE V2+1,36:POKE V2+2,36

Player 2 is Bearcat Zeke, who plays the harmonisqueeze. When Zeke puffs into his instrument, it takes a moment for the sound to build up (we pick 36, for a medium speed attack and moderate decay). Once he winds up the note, he can hold it for quite a while (we set 36 for medium sustain and release).

170 POKE V3+1,18:POKE V3+2,250

Old man Mose is in slot number 3; he plays a traditional gut bucket. When he strums the cord, he gets a fast, full sound (we'll pick 18 for fast attack and medium decay). Once the cord is in motion, it holds the sound for quite a while before it fades away (250 for a full slow sound decline).

180 T=TI

Let's check our digital watches so that we can get the band playing in time.

200 POKE V1,16:POKE V2,32:POKE V3,16

Here's where we give each instrument its individual sound. Billy and old man Mose have instruments that produce a clear, soft sound. For them, we pick a "triangle" waveform, code 16. Zeke's contraption makes a sharper, snarly sound; we give it an edge with the "sawtooth," code 32.

We're doing something else important here. By using *even* numbers, we're telling the boys: hands off the instruments. Billy, take your corncob away from the jugs; Zeke, don't blow into that thing; and old man Mose, hands off the string. Later, we'll tell them to "hit it" with an odd number to the same location.

It's time to play some music. Let's read the notes.

210 READ S:IF S=0 GOTO 290

First, let's get the timing. If the note timing is zero, we must be finished, because we can't play that fast . . .

220 READ X1,Y1,X2,Y2,X3,Y3

Let's get the notes. Each note is broken into two pieces; that's the way we feed it to the band. If the note is zero, we have nothing to play this time around.

230 IF X1 THEN POKE H1,X1:POKE L1,Y1:POKE V1,17
240 IF X2 THEN POKE H2,X2:POKE L2,Y2:POKE V2,33
250 IF X3 THEN POKE H3,X3:POKE L3,Y3:POKE V3,17

If a player has something to play (his note is not zero) we'll put his note into the proper slot, and then tell him to "hit it," by making his waveform number odd—remember line 200 where we told everyone "hands off?"

In fact, we're going to go back to 200 later and tell the fellows to let go. If we didn't, they wouldn't be able to play the next note. But first, let's do some timing . . .

260 T=T+S

Let's set the clock ahead so that we can tell when this note has been played long enough. This way, we can wait until the note has had its full time:

270 IF T>TI GOTO 270

When the note has played, we'll go back and do it all again, with:

280 GOTO 200

And when the music is finished, we tell the band to take a break . . . until the next tune. Turn off the instruments.

290 FORJ=L1 TO 54296:POKE J,0:NEXT J

That's all very well, but we'll need some music to play. Here it comes:

```
310 DATA 10,43,52,0,0,0,0
320 DATA 20,34,75,21,154,8,147
330 DATA 20,34,75,25,177,0,0
340 DATA 10,34,75,17,37,6,108
350 DATA 10,38,126,0,0,0,0
360 DATA 10,43,52,25,177,0,0
370 DATA 10,45,198,0,0,0,0
400 DATA 20,51,97,21,154,8,147
410 DATA 20,51,97,25,177,0,0
420 DATA 20,51;97,17,37,9,159
430 DATA 20,43,52,25,177,10,205
440 DATA 20,57,172,22,227,11,114
450 DATA 20,57,172,34,75,0,0
460 DATA 20,57,172,28,214,8,147
470 DATA 10,0,0,34,75,0,0
480 DATA 10,51,97,0,0,0,0
500 DATA 20,57,172,22,227,11,114
510 DATA 10,0,0,34,75,0,0
520 DATA 10,51,97,0,0,0,0
530 DATA 10,57,172,28,214,10,205
540 DATA 10,64,188,0,0,0,0
550 DATA 10,68,149,34,75,9,159
560 DATA 10,76,252,0,0,0,0
600 DATA 20,86,105,21,154,8,147
610 DATA 20,0,0,25,177,0,0
```

```
620 DATA 20,0,0,17,37,6,108
630 DATA 10,68,149,25,177,0,0
640 DATA 10,51,97,0,0,0,0
650 DATA 20,68,149,21,154,8,147
660 DATA 20,0,0,25,177,0,0
670 DATA 20,0,0,17,37,8,23
680 DATA 10,51,97,25,177,7,53
690 DATA 10,43,52,0,0,0,0
700 DATA 20,51,97,22,227,6,108
710 DATA 20,0,0,25,177,0,0
720 DATA 20,0,0,19,63,9,159
730 DATA 10,38,126,25,177,0,0
740 DATA 10,43,100,0,0,0,0
750 DATA 20,34,75,21,154,8,147
760 DATA 20,0,0,25,177,6,108
770 DATA 20,0,0,21,154,4,73
780 DATA 20,0,0,0,0,0,0
790 DATA 0
```

That's the whole thing. You can add to the music, change the instruments, change the timing, or do whatever else you like.

If you'd like to listen to the instruments one or two at a time, you can silence any player by changing the appropriate line 230-250 to a REM line. You could delete the line, but REM is better—it saves typing when you want to bring the instrument back.

An easy way to change the speed of a tune is to change line 260: by multiplying or dividing variable S by an appropriate factor, the band can turn frantic or ease back into lazy playing.

If you change the waveform, remember that there are two places to do it: line 200 and the appropriate line of 230-250.

There are a couple of waveforms that we haven't used: we've stayed with triangle (16) and sawtooth (32) instruments. If you want to bring in fiddlin' Fran, she will need a pulse waveform (64), and you'll need to supply a *pulse width* by setting values into (for instrument 1) V1-1 and/or V1-2. Snaredrum Sammy will want a noise waveform (128). Don't forget to add 1 when you want the instrument played.

The group mostly plays southern music. But on a pleasant warm night when visitors stop by to listen, they might just try a northern tune . . .

```
300 DATA 20,34,75,21,154,8,147
310 DATA 20,34,75,25,177,0,0
320 DATA 20,38,126,28,214,6,108
330 DATA 20,43,52,25,177,0,0
340 DATA 20,34,75,21,154,8,147
350 DATA 20,43,180,25,177,0,0
360 DATA 20,38,126,22,227,8,23
370 DATA 20,0,0,25,177,0,0
400 DATA 20,34,75,21,154,8,147
410 DATA 20,34,75,25,177,0,0
420 DATA 20,38,126,28,214,6,108
430 DATA 20,43,52,25,177,0,0
440 DATA 20,34,75,21,154,8,147
450 DATA 20,0,0,25,177,0,0
460 DATA 20,32,94,22,227,8,23
470 DATA 20,0,0,19,63,6,108
500 DATA 20,34,75,21,154,8,147
510 DATA 20,34,75,25,177,0,0
520 DATA 20,38,126,21,154,7,163
530 DATA 20,43,52,17,37,0,0
540 DATA 20,45,198,28,214,7,53
550 DATA 20,43,52,34,75,0,0
560 DATA 20,38,116,28,214,0,206
570 DATA 20,34,75,22,227,0,0
600 DATA 20,32,94,25,177,6,108
610 DATA 20,25,177,22,227,0,0
620 DATA 20,28,214,21,154,7,53
630 DATA 20,32,94,19,63,8,23
640 DATA 20,34,75,21,154,8,147
650 DATA 20,0,0,25,177,6,108
660 DATA 20,34,75,21,154,4,73
670 DATA 20,0,0,0,0,0,0
700 DATA 0
```

Next time you're passing through the hills of 64, drop in, set a spell . . . and play along. **C**

Commodore 64™ Computer

over $300 Free Professional Software
when you buy a Commodore 64™ computer

COMMODORE 64™
PROFESSIONAL SOFTWARE

Name	List	Sale
word processing pack	$89.00	$69.00
complete Data Base pack (includes Mailmerge)	$89.00	$69.00
Electronic spreadsheet pack (like visicalc)	$89.00	$69.00
Accounting pack (personal & business)	$59.00	$49.00
Programmers Helper	$79.00	$59.00
Sprite designer	**$16.95**	**$14.95**
Basic Tutor	$24.95	$19.95
Typing Tutor	$24.95	$19.95

80 column BOARD

80 characters per line on the screen at one time. Includes word processing pack (list $89.00), complete data base pack (list $89.00), Electronic spreadsheet pack (list $89.00). **ALL FOR ONLY $275.**

We have over 300 Programs for
the Commodore 64™ Computer

- 10 day free trial • We have the lowest prices
- One day delivery express mail • Free Catalogs

WE LOVE OUR CUSTOMERS

PROTECTO
ENTERPRIZES

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

Commodore 64 is a trademark of Commodore Electronics Ltd.

Congratulations to Commodore Challenge Contest Winner

James Dunn of Dallas, Texas
for "Wheel of Gold"

We received a slew of superb software from our talented readers over these past few months, but, unfortunately, could choose only one winner. Many thanks to all of you who submitted programs. They were all terrific! And keep up the good work. Who knows . . . 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

WHEEL OF GOLD

James Dunn of Dallas, Texas

For Unexpanded VIC 20 and Joystick

The point of this game is to get the little man up the ramps to the bags of gold at the top of the screen. But don't let him get squashed by the boulders that keep rolling down at him! Use the fire button to make him jump over the boulders, and grab that gold! (See page 7 for the key to graphic symbols.)

```
5 REMWHEEL OF GOLD BY J DUNN
6 REMP.O.BOX 64761 DALLAS TEXAS 75206
10 POKE52,28:POKE56,28:POKE51,0:POKE55,0:CLR
20 FORD=7168TO7679:POKED,PEEK(D+25600):NEXT
30 READF:IFF=-1THEN150
40 FORC=0TO7:READE:POKE7168+F*8+C,E:NEXT
50 GOTO30
60 DATA0,0,0,0,0,16,40,40,16
70 DATA27,184,84,20,16,40,70,66,192
80 DATA28,0,0,0,0,16,40,40,16
90 DATA29,58,84,80,16,40,196,132,6
100 DATA30,0,0,0,16,40,170,84,56
110 DATA31,16,16,56,68,68,198,0,0
120 DATA35,255,255,255,129,129,255,255,255
130 DATA38,56,84,214,186,186,214,84,56
140 DATA37,0,108,56,84,84,84,84,56,-1
150 PRINT"■":POKE36869,255
160 REM INITIAIZE
170 DIMMS(2,2):POKE37139,0:DD=37154:PA=37137:PB=37152
180 FORI=0TO1:FORM=0TO1:READMS(M,I):NEXTM,I
190 DATA-23,-22,-21,-1,0,1,21,22,23
200 S1=7680:S2=38400:Q=0:A=483:N=70:J=156:Z=316:W=397:CT=0
210 CA=0:CB=2
220 POKE36878,15
230 REM DRAW BOARD
240 FORX=4TO10:Y=4:GOSUB670:NEXT
250 FORX=11TO13:Y=5:GOSUB670:NEXT
260 FORX=14TO17:Y=6:GOSUB670:NEXT
270 FORX=18TO21:Y=7:GOSUB670:NEXT
280 FORX=0TO4:Y=8:GOSUB670:NEXT
290 FORX=5TO6:Y=9:GOSUB670:NEXT
300 FORX=7TO8:Y=10:GOSUB670:NEXT
310 FORX=9TO13:Y=11:GOSUB670:NEXT
320 FORX=14TO16:Y=12:GOSUB670:NEXT
330 FORX=17TO21:Y=13:GOSUB670:NEXT
340 FORX=0TO4:Y=14:GOSUB670:NEXT
350 FORX=5TO8:Y=15:GOSUB670:NEXT
360 FORX=9TO12:Y=16:GOSUB670:NEXT
370 FORX=13TO16:Y=17:GOSUB670:NEXT
380 FORX=17TO21:Y=18:GOSUB670:NEXT
390 FORX=0TO3:Y=19:GOSUB670:NEXT
400 FORX=4TO6:Y=20:GOSUB670:NEXT
410 FORX=7TO14:Y=21:GOSUB670:NEXT
420 FORX=15TO21:Y=22:GOSUB670:NEXT
```

```
430 POKES1+27,37:POKES2+27,7:POKES1+29,37:POKES2+29,7:POKES1+31,37:POKES2+31,7
440 TI$="000000"
450 REM
460 POKES1+A,32:POKES1+A-22,32
470 GOSUB720 IFPEEK(S1+A)=38THEN680
480 POKES1+A,CB:POKES2+A,6:POKES1+A-22,CA:POKES2+A-22,6
490 IFH>=483THENH=70:POKES1+483,32
500 IFJ>=483THENJ=70:POKES1+483,32
510 IFZ>=483THENZ=70:POKES1+483,32
520 IFW>=483THENW=70:POKES1+483,32
530 POKES1+H,32:IFPEEK(S1+H+22)<>35THENH=H+21
540 IFPEEK(S1+H+1)<>32THEN680
550 POKES1+H+1,38:POKES2+H+1,0
560 POKES1+J,32:IFPEEK(S1+J+22)<>35THENJ=J+21
570 IFPEEK(S1+J+1)<>32THEN680
580 POKES1+J+1,38:POKES2+J+1,0
590 POKES1+Z,32:IFPEEK(S1+Z+22)<>35THENZ=Z+21
600 IFPEEK(S1+Z+1)<>32THEN680
610 POKES1+Z+1,38:POKES2+Z+1,0
620 POKES1+W,32:IFPEEK(S1+W+22)<>35THENW=W+21
630 IFPEEK(S1+W+1)<>32THEN680
640 POKES1+W+1,38:POKES2+W+1,0
650 H=H+1:J=J+1:Z=Z+1:W=W+1
660 POKE36876,200:POKE36876,0:GOTO450
670 POKES1+22*Y+X,35:POKES2+22*Y+X,2:RETURN
680 PRINT"■"
690 PRINT"■■■■■■■■■■■■■■■■■■■■■■■   YOU'RE DEAD"
700 FORT=255TO0STEP-5:POKE36876,T:NEXTT
710 PRINT"■":GOTO200
720 REM
730 POKEDD,127:B3=-((PEEK(PB)AND128)=0):POKEDD,255
740 P=PEEK(PA):B2=((PAND16)=0)
750 FR=-((PAND32)=0)
760 Q=Q*2:IFQ<>0THENGOTO870
770 IFFRANDPEEK(S1+A-45)=37THENPOKES1+A-45,32:A=483:CT=CT+1:RETURN
780 IFFRTHEN840
790 IFB3ANDPEEK(S1+A+45)=35THENA=A+22:CA=28:CB=29:IFA>483THENA=483:GOTO830
800 IFB3THENA=A+1:CA=28:CB=29:IFA>483THENA=483:GOTO830
810 IFB2ANDPEEK(S1+A-1)=35THENA=A:CA=0:CB=27:GOTO830
820 IFB2THENA=A-1:CA=0:CB=27:IFA<70THENA=70:GOTO830
830 PRINT"■■■■■■■■■■■■■■";TI$:RETURN
840 Q=1:A=A-23:CA=30:CB=31:FORT=150TO250STEP10
850 POKE36875,T:NEXTT
860 POKE36875,0:RETURN
870 REM
880 IFPEEK(S1+A+22)=38THEN910
890 IFPEEK(S1+A+22)<>32THENA=A-1:Q=0:CA=0:CB=27:RETURN
900 IFPEEK(S1+A+21)=35THEN680
910 A=A+21:Q=0:CA=0:CB=27:IFCT=3THENPRINT"■■■■■■■■■■■■■■";TI$:END
920 RETURN
```

high scores

Up until now we've been accepting scores on the honor system. But we think it's 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 Schrieber, New York, NY
DRAW POKER	17,410 R. Callia, Torrance, CA
JUPITER LANDER	207,400 Christopher Champlain St. Petersburg, FL
GORF	60,410 Andy Ralston, Fairfax, VA
MIDNIGHT DRIVE	14.11 km Nathan Mehl, Newark, DE
MOLE ATTACK	331 Heda Takaya, Saskatoon, Saskatchewan
OMEGA RACE	260,050—5 ships Ben Piper, Chico, CA
PINBALL SPECTACULAR	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!

C

The New Mexico Commodore Users Group and How It Grew

by Betsy Byrne

My husband, Danny, and I moved to New Mexico from Detroit, Michigan, in the early part of 1977. We were driven out of Michigan by one of the worst winters ever. We were already carrying with us electronic bits and pieces that Danny planned would someday be part of a computer. Our passenger list also included an eight year-old daughter, a set of thirteen month-old twin boys, and a dog. After a year in Albuquerque, the number of children increased to four and the bits and pieces had increased to boxes and boxes, but neither children nor parts were very well organized.

During the next three years we moved three times, lugging boxes, PC boards, stacks of catalogs, circuit diagrams, and at least a ton of magazines, with (to me) mysterious names such as *BYTE* and *Kilobaud Microcomputing*.

Unfortunately somewhere along the way the twins, Timmy and James, discovered Dan's collection and made definite inroads into inventory before they were noticed. I vacuumed resistors out of the shag carpet for months. Where some of the other little things ended up is better left unmentioned, but the twins can truthfully boast to their friends that they ingested advanced computer technology with their soy formula. The computer building schedule had been seriously set back.

After spending three years at a series of low-paying, back-breaking jobs (including painting fire engines!), Danny went to work for General Electric, making aircraft engines. Shortly thereafter, he acquired a Z80 microprocessor chip, and it looked as if our computer would finally become a reality. Working nights at G.E., he busily spent mornings and weekends soldering, and muttering in some language I didn't understand. With the help of neighbors, I did definitely eliminate Spanish, and then Navaho, which had been a strong candidate since I thought that ROM and RAM might have something to do with sheep.

Then the find of finds occurred. While shopping the flea market for twin bicycles with training wheels, we discovered the only major necessity that had not yet been acquired, a computer KEYBOARD! The seller of this treasure had no idea where it had originated, but apologetically insisted that he had to get at least $15.00 for it. Danny reached for his wallet immediately, but I kicked him in the shins and bargained the price down to $10.00. We were shopping for bicycles, after all.

The computer might have become a reality that year (1980) if only the twins had not inherited their father's interest in electronics. While I was at a political meeting one evening, they escaped the grandmotherly eye of their babysitter ("They were being so quiet and good," she said). The boys had spent a busy two hours or so DISassembling their father's many moons of assembly. Worst of all they had destroyed the precious chip. Danny dejectedly salvaged what he could, chastized the children, fired the babysitter, and put his pet project on the shelf. I had to gently dissuade him from relegating the boys to a neighboring shelf! It looked as though the computer had gone the way of our model railroad layout, another project the twins had shown great interest in.

Then in February of 1982 the most amazing fact came to our attention. A regular person could evidently BUY a computer. It was called the VIC 20, priced dollars less than it would cost Danny to finish his. Dan was stunned, but not too stunned to make a fast phone call to the G.E. Credit Union. After singing the praises of the educational and profes-

sional benefits of computer ownership he got approval for a loan to buy a VIC 20 computer. In spite of my mentioning things such as "What about the hole in the livingroom carpet? You promised as soon as we had the money . . . ," Danny was on the doorstep of the Commodore dealer bright and early the next morning. By noon, the Byrne family membership had been increased by one and although *I* certainly wasn't aware of it, all our lives had been irrevocably changed.

Now we had a computer. What next? Even Danny was a little uncertain. Of course we could play games on it, but that had definitely not been our goal for six years. So Danny plunged in with a will, and spent all his free time with the little blue book that had come in the box with the VIC. Sometimes he couldn't find the book, and amazingly enough, it was usually traced to our oldest daughter Beth, now 12. This was the girl who had often been heard declaring that "Computers are so-oo-oo boring" and other phrases of the same ilk. I decided it was a conspiracy when I found Timmy and James (who had somehow survived to age 5), copying in print commands to make their names float across the VIC screen. Then one day three year-old Molly interrupted me while I was bragging on the phone to my mother about my five year-old 'programmers' (the twins). Molly said "Come and see my NAME Mommy!" And by golly there it was, in lights and looking ten feet tall to me. So I shoved aside my hesitancy (and yes, fear!) and sat down with the rest of the family to type in programs, learn BASIC, do the family budget, and even play a few games!

I have saved what I think may be the best part of our story for last. In only four months, Danny taught himself programming! He did this by going through Commodore's instruction books and, when he got stuck, asking questions at our local Commodore dealer.

Several months later we were both offered jobs at the Commodore dealer, Danny to answer clients' questions about programming on the VIC 20 and the Commodore 64, myself in sales. I had somehow become very good at explaining the merits of computers in the home and the difference they can make in the development of children.

Now we get to the good part. We also found we had to organize a Commodore users group, that now has over 100 members. Why? So we could have lots of friends to share ideas with. Friends that spoke the same strange language we now did.

So the New Mexico Commodore Users began, as I would assume many other Commodore groups have, with an outpouring of enthusiasm and ideas from users who met by chance at the Commodore dealership where Danny and I worked. It seemed natural to approach the owners to discuss using the store for meetings after hours and also to request their sponsorship to get us started. We also contacted other local Commodore dealers about helping sponsor the group, but none of them felt they were able to get involved to that extent, although they thought it was a good idea.

Armed with whole-hearted approval and a promise of cooperation from the dealer, Danny and I called Commodore headquarters in Pennsylvania and were put in touch with Jeff Hand, who is on the staff of *Commodore Magazine, Power/Play,* and is Systems Operator for the Commodore Information Network. Jeff was very supportive and gave us lots of encouragement, explaining that Commodore likes to interact with as many users as possible and

that the user groups are one of the best formats for doing this. We also discussed the obvious benefits of sharing information and ideas, and the ability of the group to maintain a public domain software library and serve as a foundation for specific interest clubs such as sound and graphics or telecommunications.

Another important phone call we made was to Jim Butterfield in Toronto, who needs no introduction to readers of *COMPUTE!* and *Commodore Magazine*. Jim is a member of TorPUG (Toronto PET Users Group) one of the oldest and most respected Commodore user groups. He had several suggestions as to how we might structure our group and he certainly instilled a large amount of the confidence we needed to organize it! Now, feeling we were on firm ground we put ads in the classified section of all the local newspapers under "Computers," put up a sign at the dealer and called the other stores that sell Commodore products so they might inform their customers.

When the day of the meeting arrived, we had about fifteen people signed up. I borrowed folding chairs so that we had seating for 22 people in the Commodore dealer's store. A half hour or so before the meeting I went out to pick up some coffee and refreshments. Picture my amazement when I returned to the store to find more than fifty people waiting for me!! In spite of somewhat(!) crowded conditions, the meeting was an unqualified success.

We have now moved to a larger meeting place provided by a local bank. Our meetings usually follow this format:

1. We have a short business meeting and vote on such things as plans and major expenditures.
2. A speaker is introduced or a demonstration is given of interest to the entire group.
3. We break up into sub-groups and gather around our respective computers, usually for a computer-specific demonstration followed by a question and answer session.

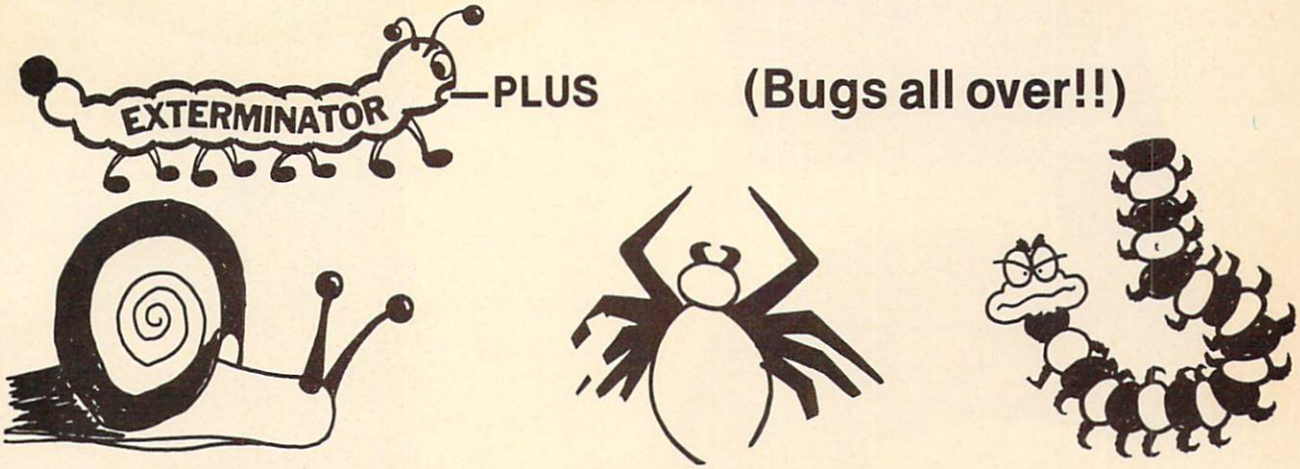
Then general discussion and socializing takes over, with people practically waiting in line to talk to some of our more knowledgeable members!

Our group organization is as follows: Several members are chosen from each sub-group to serve as an executive board. The executive board elects the officers and makes executive decisions for the group. A monthly newsletter that is free to members is published by the group with additional copies sold in local stores to help pay for the printing costs.

In between the main Commodore group meetings are meetings of the PET, VIC 20, Commodore 64, and other clubs including machine language, graphics, and women's BASIC study. We are organizing two children's groups called the "Commodore Kids"—a preteen, and a 13-to-17 year-old group. One of our main goals at the moment is to obtain a permanent place from which to administer our software and print libraries, and to provide a hotline phone service staffed by volunteers to answer questions and solve problems.

We now have over one hundred members and we are still rapidly growing due to the increasing popularity of Commodore products. Our advice to anyone who has been thinking about joining or organizing a Commodore user group is: DO IT! You won't believe how much fun it is and how rewarding it can be until you try it. **C**

TWO VIC20™ PERSONAL COMPUTER TOP SELLING ARCADE GAMES!!



—PLUS (Bugs all over!!)

This game is literally full of bugs. We guarantee hours of exciting entertainment trying to remove them. Bugs you will come up against are spiders, snails, fleas, and centipedes in this rapid fire, 100% machine language, great graphics, exceptional quality game. Our highest recommendation runs in standard 5K VIC!

10 DAY FREE TRIAL—BECAUSE WE LOVE OUR CUSTOMERS!

EXTERMINATOR—PLUS
LIST PRICE $24.95 SALE PRICE $19.95

NUCLEAR DISASTER!

NUCLEAR DISASTER!

A more dastardly alien could scarce be found than one who would wipe out an entire civilization by dropping anti-matter anti-cannisters, right? If your opinion of this alien trouble maker is the same as ours, probably your first thought was, get some matter! We say calm down! All is not lost. A mobile rapid splatter cannon capable of both breaking through his standard alien moving force fields and laying waste to the every increasing number of anti-cannisters is even now hovering above us. If only our cannoneer hadn't called in sick . . . say, what are you doing today? This is a must for VIC-20 owners.

Anti-Matter Splatter is 100% machine language and runs in standard 5K VIC.

10 DAY FREE TRIAL—BECAUSE WE LOVE OUR CUSTOMERS!

ANTI-MATTER SPLATTER
LIST PRICE $24.95 SALE PRICE $19.95

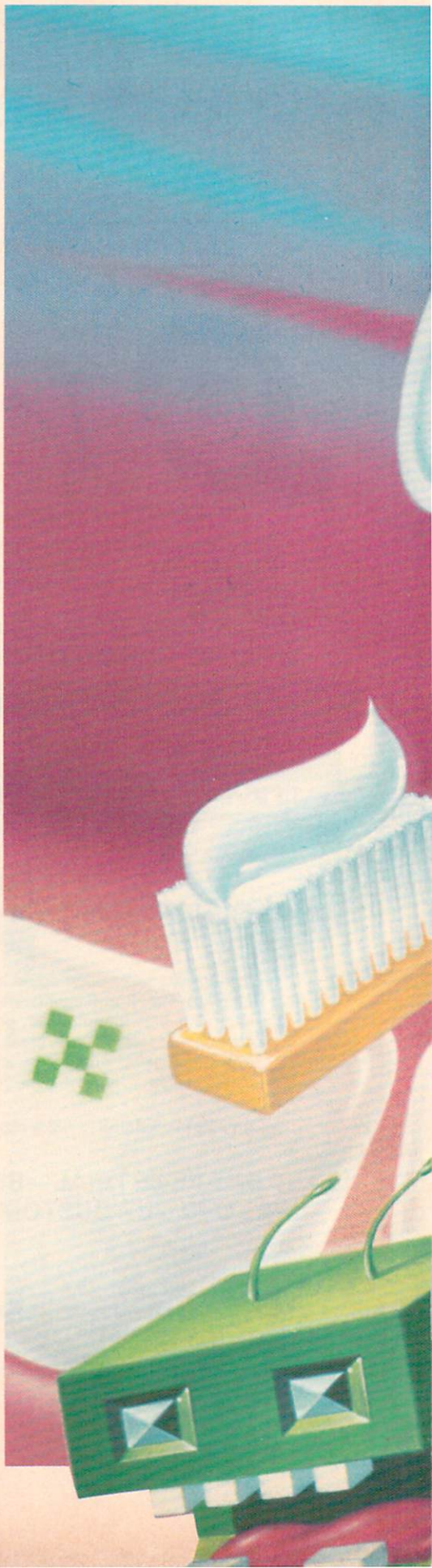
we are commodore VIC experts!!

VIC 20 is a trademark of Commodore Electronics Ltd.

PROTECTO ENTERPRIZES

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

Illustration by TOM HALLMAN

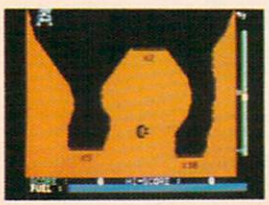


5 NEW GAME CARTRIDGES FOR THE COMMODORE 64

by Neil Harris

Commodore's game development group is now shipping the first set of arcade-style games for the Commodore 64. Of the five, three are conversions of popular games for the VIC 20, one is a new game licensed from the Bally/Midway arcade company, and the fifth is an all-new one.

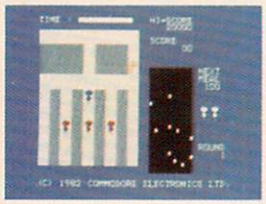
1



Jupiter Lander gives you control of a space ship's retro rockets. You try to pilot your ship to a safe landing on one of three sites. You must maneuver your ship through some tricky terrain to get the largest possible score. Obstacles include the ever-

growing pull of gravity, a limited fuel supply, and the walls around the landing sites. Your fuel supply is replenished by landing safely, and diminished by each crash landing.

2



Radar Ratrace is a maze game with a twist: you can only see a small part of the maze on the screen at one time. You control a mouse in the maze whose goal is to eat all ten cheeses that are hidden in the maze. To aid you in finding your way is a radar screen that shows the relative locations of the cheeses, yourself, and the enemy mice. In the first round there are three enemy mice chasing you, and more in the following rounds. There are also stationary cats poised to pounce on you. Other features are clouds of "magic stars" that confuse pursuers, progressively higher scores for eating many cheeses without being "killed," and special bonus rounds with no pursuers but very high speeds.

3

Speed Math/Bingo Math contains two educational games for grammar school children. You choose the problem type at the start of the game, either addition, subtraction, multiplication, division, or a mixture of all types. **Speed Math** gives thirty problems. You get ten points for a correct answer within a couple of seconds, with the possible score decreasing the longer you wait. If the answer given is wrong, the computer shows the correct response and goes on to the next problem. **Bingo Math** is a two-player competition game, in which you must pick the correct answer from a bingo board. The first player to get five correct answers in a row on the board is the winner.

4

A new addition to Commodore's line-up is **Kickman.** This game from

Getting Serious On The 64

In addition to these great new games for the Commodore 64, Commodore will soon be releasing some exciting applications and educational software that will warm the cockles of your RAM. At the very top of our long list of software-in-development are the following, all of which will be at your dealer soon, if they aren't there already. For in-depth coverage of these products, see upcoming issues of *Commodore Magazine.*

Screen Editor: This software developer's tool is used as an aid in setting up screens on the Commodore 64. It adds commands in BASIC that allow you to change screen colors, draw horizontal lines, input and edit data from the screen, maintain status lines, and disable the RUN/STOP key. If you're writing software for the 64 that requires color changes and/or special screens for data collection or cursor movement, the Screen Editor will save you much time and effort. Consistent with Commodore's approach to computing, the Screen Editor is designed for use by novices and experienced programmers alike.

Assembler 64: Assembly language programmers will be delighted to learn about this one. The Assembler 64 contains everything you need to create, assemble, load and execute assembled code for the Commodore 64. Not only that, but you get a text editor and an excellent monitor program included at no extra cost! Assembler 64 contains the following components: full screen editor with more commands than provided in BASIC, DOS Wedge program that provides additional commands to make disk access easier, two loader programs that allow the user to load an object program anywhere in memory, a powerful monitor program for debugging, and a

cross reference program. Use assembly language routines as complete programs in themselves or incorporate them into BASIC programs for functions that require speed and efficiency.

EasyLesson 64/EasyQuiz 64: Using EasyLesson 64, a teacher or parent can create and edit questions about any subject. Once the master pool of questions is established, it is saved on disk or cassette. This pool can then be used to generate any number of different tests, quizzes or work sheets on the screen or printer. EasyQuiz 64 administers tests prepared in EasyLesson 64 and corrects them on the spot (students may think this is a dubious benefit at first). Students learn immediately what their scores are and—here's the part students may like—are given a chance to improve the scores. This package can ease some of the tedious, repetitious kinds of work teachers normally have to do in devising, administering and grading tests.

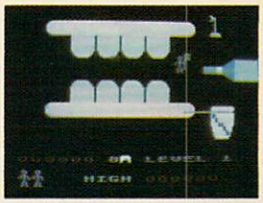
PILOT: For those of you who are new to computers, PILOT is the language that helps educators create programs for Computer Assisted Instruction (CAI). It provides a whole raft of special commands so educators can easily set up quiz-style formats for end users who have little knowledge of computers (i.e., their students). This particular version of PILOT is probably the most sophisticated one available, since it allows access to all the graphics and music capabilities of the Commodore 64. It features, among other things, easy graphics commands for entering and animating sprites, full upper/lower case text, the capability for split-screen displays and the ability for the author to plug in machine language subroutines as "hooks" for integrating things like laser disks and video cassette recorders into programs.

the Bally/Midway arcade collection features cartoon graphics and some nice sound effects. You control a clown on a unicycle, moving from side to side to catch, pop, or kick balloons that fall from the sky. When all the balloons are gone you advance to the next round, where there are more balloons that fall faster. In the higher rounds some of the balloons are shaped like Pacman and the ghosts—and I must admit that it gives me a perverse satisfaction to kick Pacman around the screen!

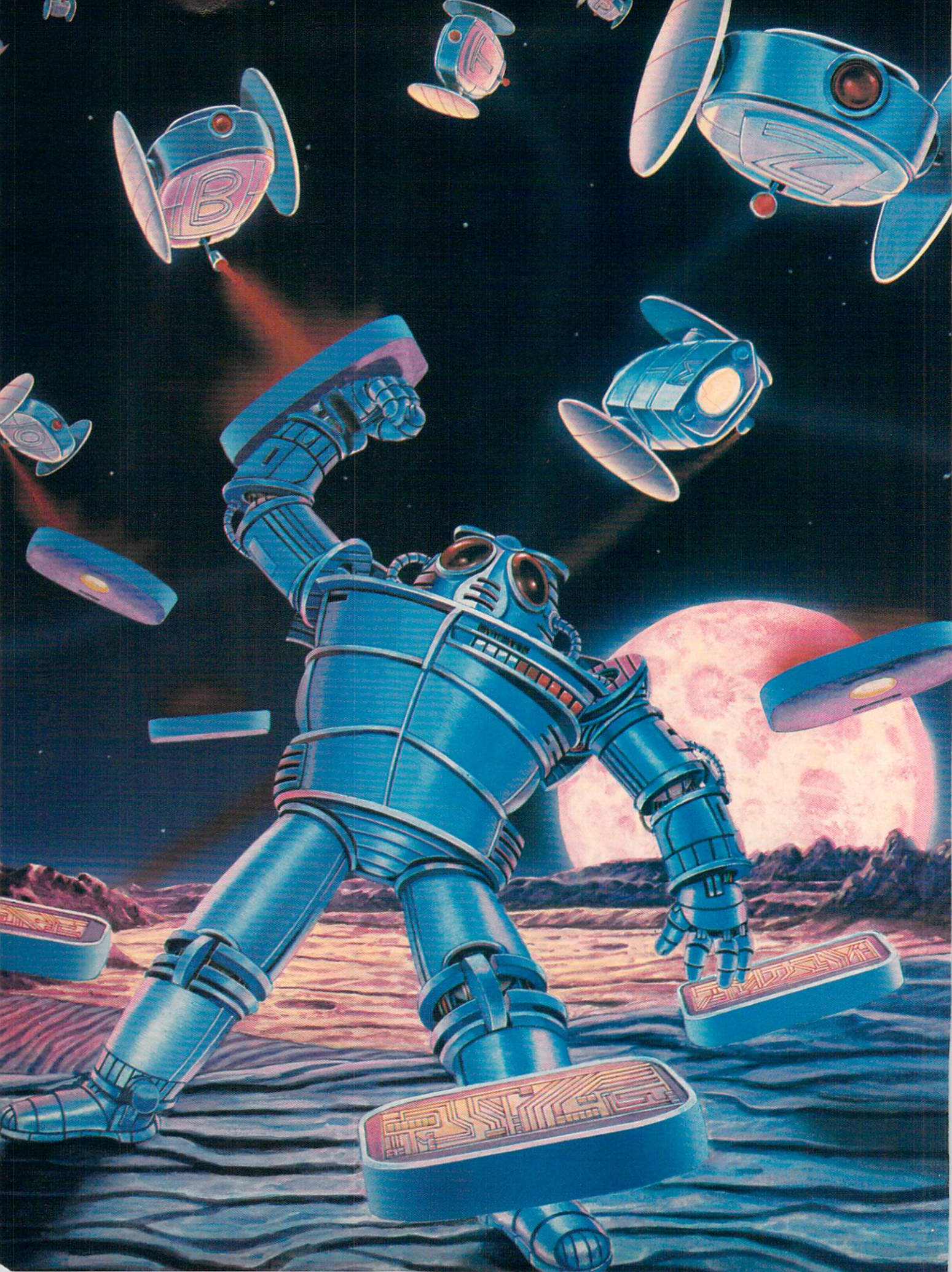
Now for my favorite:

5

Tooth Invaders. This reminds me of the commercial for the cereal that is good for you and tastes so good even Mikey likes it. This game, developed in cooperation with the American Dental Association, is as fun as it is educational. You control the hero, called Plaqueman, in his battle to fight D.K. Germ. D.K. wanders around the teeth spreading plaque—enough plaque on a tooth and it will decay and fall out! Your weapons in fighting D.K. are a toothbrush (don't forget to load up with fluoride toothpaste), dental floss for between teeth, and the fluoride rinse. The screen shows the whole mouth or, at the touch of the joystick's button, a closeup of a tooth. You must decide whether to brush the teeth or floss between teeth. When the teeth are clean, they sparkle, and you get twenty seconds to kill D.K. (to the tune of the "William Tell Overture"). D.K. will kill Plaqueman if touched at any other time. The round ends when all teeth are sparkling and the fluoride rinse appears, giving bonus points for each tooth remaining. The game ends when all three Plaquemen are destroyed, or when only four teeth remain. All in all, a unique game playing experience.



All these games carry a suggested list price of $29.95. They are available in limited quantities until April, when mass production starts. Have fun! **C**



Commodore's new tutorial for children teaches programming on the planet Syntax.

I PROGRAM WITH GORTEK

by Betsy Byrne

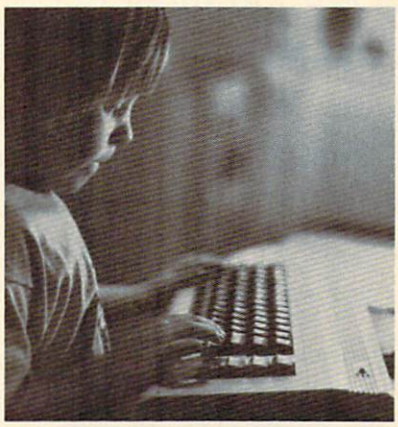
Beyond the oblivion of all numbers, far out in the galaxy, and even farther. . . This is where my four children and I have been spending a lot of our time lately. Four year-old Molly, thirteen year-old Beth, Timmy and James (six year-old twins), and I have been sojourning on the planet Syntax. There, we have been helping a robot named Gortek and strange little beings called Microchips protect an all-knowing computer known as Creativity from an invasion of misguided nasties, the Zitrons. The Zitrons are bent on the destruction of all the planets but their own.

Does this all sound suspiciously like another clone of a space-type video game? It's not!

Instead, *Gortek and the Microchips* is the first in a series of tutorials for children to be released by Commodore. Two versions should be available soon—one for the VIC 20 and the other for the Commodore 64, both priced at about twenty dollars.

With *Gortek*, the people at Commodore have at last answered the pleas of those of us whose children were not old enough or were not motivated to learn BASIC programming from the (excellent) books provided with the VIC and the 64. *Gortek* was written by three British school teachers to make learning BASIC programming fun for children. Judging by the reactions of my four children and their friends, they have done a pretty good job of it.

Gortek was designed for children aged 10 to 13 but is easily used by much younger children with the help of a parent or an older sister or brother. It begins with a stirring description of the plight of the inhabitants of the planet Syntax and an appeal for the child's help in repelling the Zitron invasion. The manual then explains how to LOAD and RUN the first program on the first tape.



Four year-old Molly Byrne learns the cursor keys while playing "Splat."

Described by Timmy and Jamie as "a really neat game, Mom!" Zitrak is formatted to help your child quickly and easily learn the location of the letters on the keyboard. Any parent familiar with a child's frustration at not being able to immediately spot the letters needed to type an important word or load a program will see the wisdom in beginning with a game that protects the planet Syntax from a bombardment of alien letters! Zitrak is followed by a spelling game and a "recreational" multiplication game. By the time my children and the Microchips began their "vital instructions in computer programming" they were competent and comfortable using the keyboard, including all four modes of the cursor keys.

An outline of the BASIC instruction part of the microchip manual would not greatly differ (believe it or not!!) from an outline of the course my husband, Danny, and I are taking in BASIC at the University of New Mexico. But what a difference in the way it is presented. *Gortek*, whether teaching simple print statements or introducing your child to strings and memory locations, (information on Syntax is kept in boxes), immediately gives the trainee things to do with this knowledge that are both useful and *fun*.

The second tape contains games and examples that reinforce what the children are learning from the manual. They are encouraged to list some of the programs and experiment with changing them. The manual uses teaching methods such as scrambled line numbers, and encourages the kids to use input statements to make their programs as friendly as possible.

I suppose that this all sounds a bit too good to be true, a BASIC learning program that is educational and fun, easy enough for a four year-old like Molly to learn some aspects of, but challenging enough to hold the interest of a teenager like Beth.

I must admit, parents, that I did find one (almost fatal) flaw. *Gortek and the Microchips* is packaged in a sturdy, hinged, booklike container that is designed to blend into any bookcase. Inside is the storybook-manual (not quite washable but at least wipe-offable!) and two cassette tapes. Also included is one Gortek and the Microchips Badge, which the "trainee" earns the right to wear. That's right parents, (Commodore marketing people, please note!) one and only one. I am happy to report that the four Byrne children are all alive and recovering nicely from the conflict that was triggered by this unfortunate lack of foresight in packaging. As a result, wherever I go nowadays the one indispensible item of adornment that I always pin to some part of my apparel is that badge proudly proclaiming: **C**

"I PROGRAM WITH GORTEK."

Betsy Byrne is the president of The New Mexico Commodore Users Group. She is married to Danny Byrne, who among other things, is the author of Commodore's Alpha Draw program for the VIC 20. The Byrnes are the parents of four children, aged 4-13 and have a major interest in education software for children. Betsy is in the process of organizing both a Commodore Kids Group and a New Mexico Commodore educators exchange. She and Danny work part time as Commodore consultants, raise Pekin and Gray Call "Show Ducks" and edit the N.M. Commodore Users Group newsletter, **ONLINE.**

Illustration by MIKE ADAMS

For owners of the Commodore 64 who want to get the most out of their computer's music synthesizer, this series by our resident SID expert is a must. In Part 2, Paul shows you how to make your synthesizer sound like a piano, and explains exactly what your SID is doing to create that effect.

by Paul Higginbottom

MAKING FRIENDS

Hello again. Last issue we got acquainted with some of the parts of the SID (Sound Interface Device) chip on the Commodore 64 and discussed some of the terms associated with music synthesization. We're now equipped to learn some more advanced things about the synthesizer to, for example, produce more than one note simultaneously and then create software that can play actual pieces of music.

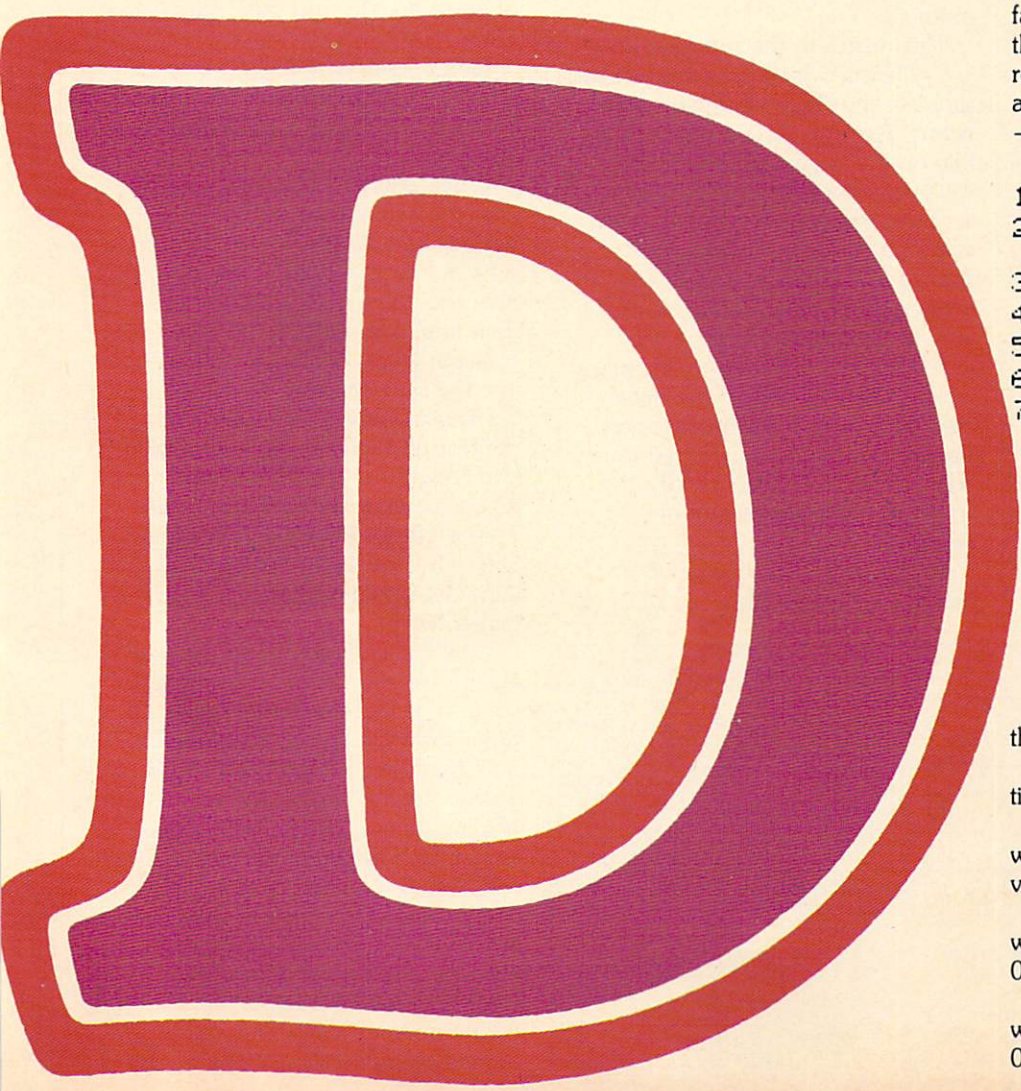
So there are, in fact, two areas I plan to deal with in this and subsequent articles:

1) Defining the capabilities of the SID

2) Explaining some software techniques to make SID perform.

This time I'd like to put some of the last article's theory into practice by giving some parameters for the SID that will make it sound similar to musical instruments. I think this would be useful so you will be able to see that a music synthesizer is not limited to beeps and pops and other sounds that seem to come from television shows like "The Twilight Zone!"

WITH



In the last article I outlined the various parameters of a voice, except for the filter in the SID. Don't worry about understanding filters fully, yet, since we aren't ready to use the filtering capabilities of the SID. Just remember that the filter will (as is implied) filter the sound output from any of the voices in a number of ways. "Filtering" a sound means quieting the voice to varying degrees above, below or around a given "cutoff" frequency. I wanted to make you aware of this feature in the SID so you won't be taken by surprise in the future!

To begin, let's try to emulate one of the simplest sounds: a piano. When a piano key is struck, the sound begins immediately and then fades away in about two seconds if the key is held down. If the key is released before the sound has faded away, it will fade much more rapidly —in say, half a second.

Try this program:

```
10 SID=54272
20 FOR I=0 TO 24:POKE
   SID+I,0:NEXT
30 POKE SID+24,15
40 POKE SID+5,10
50 POKE SID+6,9
60 KEY=197
70 POKE SID+1,16
80 GET A$:IF A$=""
   GOTO 80
90 POKE SID+4,33
100 IF PEEK(KEY)<>64
    GOTO100
110 POKE SID+4,32
120 GOTO 80
```

Explanation of program:

Line 10 defines the variable SID as the start location of the SID chip.

Line 20 POKEs all the SID locations with a zero to initialize the chip.

Line 30 sets SID register 24 to 15, which sets the chip to maximum volume.

Line 40 sets SID register 5 to 10, which makes the attack of voice one 0 and the decay value 10.

Line 50 sets SID register 6 to 9, which makes the sustain of voice one 0 and the release value 9.

Line 60 defines the variable KEY as the zero page memory location, which holds the keyboard matrix number of the key being depressed, or 64 if no key is depressed.

Line 70 sets SID register 1 to 16. This sets the high order byte of the frequency of voice one. Therefore, the frequency of voice one = 16*256. (See Part 1 of this series for an explanation of low and high bytes.)

Line 80 waits for a key by GETting a keypress from the keyboard. If the keypress is null, the program will GOTO the same line and keep waiting.

Line 90 sets SID register 4 to 33. This gates voice one on (starts the attack) with a triangular waveform. (See Part 1 for explanation. [33 = 32 + 1]).

Line 100 checks to see if a key is still depressed, as with a piano. If it is (i.e., variable KEY is still something other than 64), the program will GOTO the same line and check again.

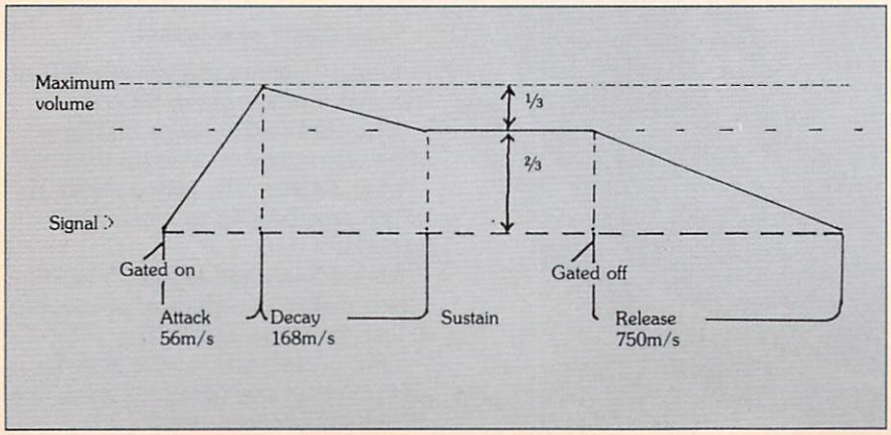
Line 110 sets SID register 4 to 32, which gates voice one off (releases it) now that no key is depressed. Voice one still has a triangular waveform.

Line 120 simply goes back to line 80 to allow the program to continue indefinitely. (To stop the program the STOP key must be pressed.)

Something that needs to be understood here that when a voice is gated off (i.e., released), the envelope releases from whatever volume level it had reached. A diagram would probably be the best way to show this:

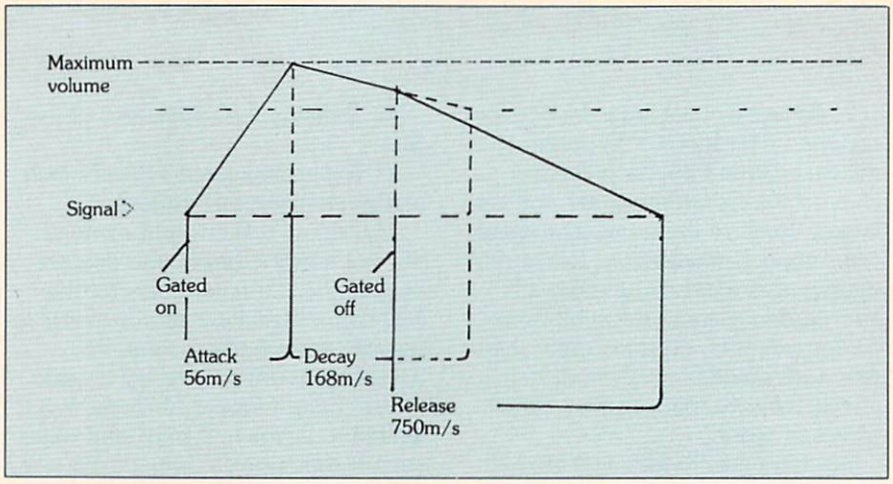
Figure 1.

ATTACK = 5 (56 milliseconds)
DECAY = 5 (168 milliseconds)
SUSTAIN = 10 (two-thirds of maximum volume)
RELEASE = 8 (750 milliseconds)



Now using the same parameters, only releasing at a different point:

Figure 2.



In Figure 2 it can be seen that the voice was released before the envelope had decayed to its sustained level. Notice that when the voice was released it simply started the release from the point it had reached at that moment.

This relates to the program you just entered, which uses this fact to simulate the "feel" of a piano keyboard. As soon as you release the key on the keyboard the envelope will begin its release cycle, which is set at 9, one less than the decay value (10). This gives the same response as a piano key by fading away quicker once the key is released. Of course the force with which the key is hit in the first place, which on a piano gives the initial volume, cannot be simulated here, because a key on any computer keyboard is either DOWN or UP; the speed of transition cannot be detected.

Similarly, if a voice is gated on before the release cycle has finished, the attack begins at the volume level the envelope has reached at that moment. To see this more clearly, enter the following to change our program:

```
40 POKE SID+5,11*16+13
50 POKE SID+6,9*16+11
```

Line 40 sets SID register 5 to 11*16 + 13, which makes the attack of voice one 11 and the decay value 13.

Line 50 sets SID register 6 to 9*16 + 11, which makes the sustain of voice one 9 and the release value 11.

When you RUN the program this time the actual path of the envelope will be more audibly clear. When you press a key and hold it down, you'll hear the volume rise (attack) and then fade some (decay) to a constant (sustain) level. When you release the key the tone will fade away to nothing (release). However, if you depress and release the key quickly, you'll hear that the tone never reaches a very high volume at all. That is because the release occurs before the envelope has reached either its maximum volume or its sustained level.

Conversely, if you press the key again very soon after releasing it, you'll note that the sound builds up again from the level it had faded to, and if you keep depressing and releasing the key—provided you're holding the key down for slightly longer than the time you're not—you will "pump up" the volume.

A program that would allow us to test envelope and waveform combinations would certainly be useful. It would enable us to experiment with the parameters available to create a desired sound. It would also be use-

ful if we could "play" the Commodore 64's synthesizer from the keyboard.

Before we can get that far, though, I'd like to explain how to derive the frequencies for musical notes on the Commodore 64.

In a musical scale, the ratio of pitch between one octave and the next is 2:1. If we had the frequencies of the twelve semi-tones of the top octave, we could generate the values of all the lower notes by continually dividing all twelve by two to derive the pitch of the semi-tones on the next octave lower. It is not necessary to go into the math here, but if the ratio between octaves is 2:1, the ratio between semitones is 2(1/12):1.

The frequency of middle A on a piano is 440Hz. To convert harmonic frequencies to the fundamental frequencies we need to put in the SID registers, we need to multiply the harmonic frequencies by a constant. This constant is derived from the frequency of the internal clock in the SID chip and the system clock.

$$\text{Harmonic frequency} = \frac{\text{fundamental frequency x system clock speed}}{\text{SID clock speed}}$$

Therefore:

$$\text{Fundamental frequency} = \frac{\text{harmonic frequency x SID clock speed}}{\text{system clock speed}}$$

Which turns out to:
Fundamental frequency = harmonic frequency x 16.404 (approximately)

Therefore middle A, whose harmonic frequency is 440Hz, would have a fundamental frequency of 440 x 16.404 or approximately 7217. "A" in the next octave up would be 880Hz, or 880 x 16.404 = 14435 in the SID chip. The maximum fundamental frequency value in the SID chip is 65535 (255 in both the low and high byte). Therefore, by doubling again: 1760 x 16.404 = 28871. And again: 3520 x 16.404 = 57742.

The number 57742 is fairly near the top end of the SID frequency value range and doubling once more would push it beyond 65535, so we will base our frequency range around 3520 Hz. To create a two-dimensional array (subscripts being "octave" and "semi-tone") of frequencies, we could use the following program:

```
100 fr=3520:rem note 'a' in top octave
110 co=2(1/12):rem constant multiplyer
    for next semitone
120 for i=1 to 9: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 dim f(7,11):rem frequency array (octave,
    semitone)
300 for i=0 to 11:rem cycle through 12
    semitones
310 s=fr*fc:rem calculate sid value of
    semitone in top octave
400 for j=7 to 0 step-1:f(j,i)=s:s=s/2
410 next:rem calc 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
500 print "frequency table"
510 print "---------------"
520 print "oct sem frequency"
600 for i=0 to 7
610 for j=0 to 11
620 print i;tab(4);j,int(f(i,j))
630 next j,i
```

The REMarks in the program explain how it works.

Add the following lines to hear the frequency array:

```
470 s=54272:rem start address of sid chip
475 for i=0 to 24:poke s+i,0:next:rem
    initialise sid chip
480 poke s+24,15:rem set volume
485 poke s+5,11:rem attack=0:decay=0:sustain=0
    :release=11
624 poke s+4,32:rem gate off the voice first
625 h=int(f(i,j)/256):rem calc high byte of
    frequency
626 l=f(i,j)-h*256:rem calc low byte
627 poke s,l:poke s+1,h:rem put in frequency
628 poke s+4,33:rem now gate it on
629 for k=1 to 100:next:rem wait a bit
```

When you RUN the program this time, as the frequencies are listed each pitch will be sounded.

I would imagine that this is quite enough to absorb this time, and we'll get onto the parameter testing program next time. Make sure you understand what has been covered so far, otherwise subsequent articles will slowly become difficult to follow. Have fun. **C**

Navigate your Federation starship into the vast reaches of space, but be sure to stay alert. An Empire commander could enter the game and launch a sneak attack, as you try to master the galaxy in this interactive telecommunications war game, available on the CompuServe network.

by Diane LeBold

The Federation starship Lexington, under the command of Captain Thrash, has just captured a neutral planet and docked. The captain has begun standard procedures for building the planet into a Federation outpost. Meanwhile, the Lexington's maintenance crew is working desperately to repair the damage incurred in a recent battle with an enemy starbase.

Suddenly a harsh, bone-chilling voice interrupts communications: "I will crush ignorant human cretins!"

Captain Thrash, seasoned by years of starship duty, recognizes this typically Romulan intimidation tactic. He calmly gives a command to his ship's computer: "List enemy ships." When the computer reports there are no enemy ships within range, the captain relaxes. As he had suspected, it was just another empty threat from a Romulan too far away to be a problem.

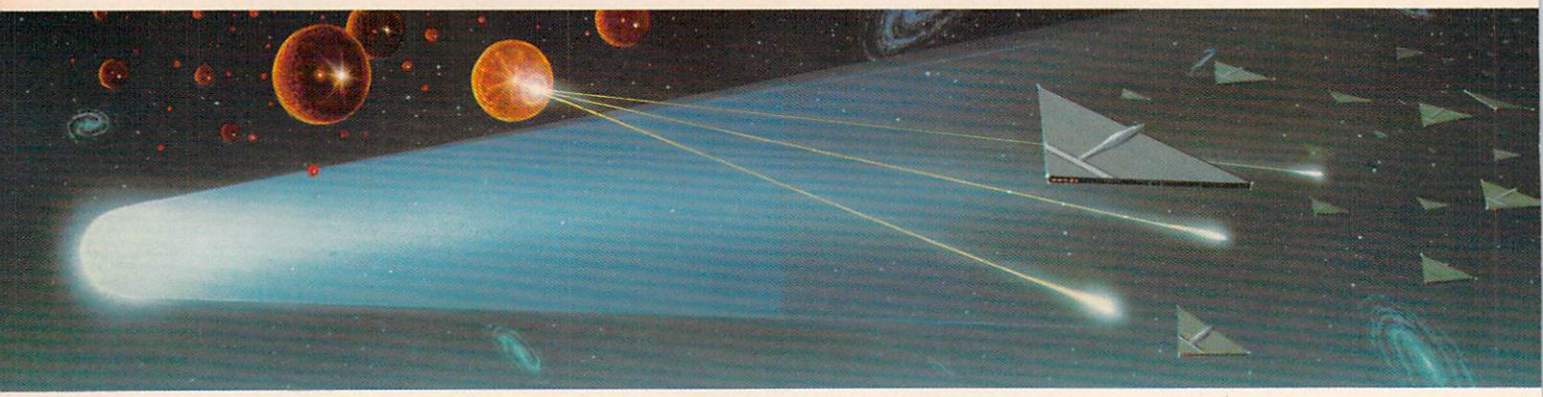
"I wonder if those guys have anything better to do than spew garbage into other people's computers," he thinks to himself. "What a waste of time." But he also knows that just a few stardates ago one of those

Illustration by KARL KOFOED

Romulan pirates had done some real damage to a distant Federation starbase. The Lexington had not been able to go to the starbase's aid—moving too far too fast could result in serious engine damage for a starship—but Captain Thrash is reading the battle reports carefully.

In fact, the Lexington is just recovering from engine damage, caused by a too-rapid retreat from a nearby Empire starbase. Captain Thrash has been examining those battle reports as well, trying to find the flaws in his attack plan. Why did his second torpedo miss its mark? How did the enemy deflect his phasers? has the Empire developed a new phaser shield?

Because of the extent of the damage to his ship, the captain expects to be docked at this new outpost—his temporary haven—for at least six or seven stardates. This is making him nervous. He really can't afford to lose that much time in his mission. On the other hand, he muses, he can use the time to study the battle reports from that last debilitating encounter. He needs to



Black holes also add excitement—not to mention confusion—by sucking you into oblivion.

know what went wrong, because he plans to renew his attack on the Empire starbase as soon as all systems are back to normal.

This may sound like an episode from your favorite TV space series, but the fact is that all this—and much more—actually happened, one morning about 11:00 on programmer Bill Hindorff's computer. As a result of his 37 minutes of play that morning, Bill (aka Captain Thrash) ended up in first place in the ongoing DECWAR competition and was awarded the Emerald Star Cluster for outstanding service to the Federation.

Now we get to the really good part. With your Commodore computer and modem, you can play DECWAR, too! Get right in there with captains from all over the U.S. and battle roving Romulans, blow up enemy starbases, fend off attack from lethal enemy starships. Choose your side—Empire or Federation—get command of a starship and plunge into the fray. But remember, in DECWAR you've got to use your head, stay alert and make quick, accurate decisions if you want to gain points—and avoid getting blown up. It's not just a matter of simple eye-hand coordination like the games you may be used to playing.

A quick note. Although it's possible to play on any size screen, for easiest readability DECWAR needs at least 40 columns. That means it reads well on any Commodore computer except—at least at the moment—the VIC 20. However, VIC 20 owners can take heart, because once VICTERM 40 terminal software becomes available they, too, can easily enter the world of warp shields, phasers and remote planetary outposts.

Since Commodore's Captain Thrash is a proven DECWAR master, we asked him if he could advise determined DECWAR cadets on how to play. A few days later the Captain showed up with a computer printout in his hand—the hard copy of his award-winning 37 minutes—to offer these words of wisdom.

To begin at the beginning, DECWAR is accessed through CompuServe, a national telecommunications network based in Columbus, Ohio. (A free subscription to CompuServe is included with every VICMODEM.) Once you sign onto CompuServe, if you want to go directly to DECWAR without plodding through all the menus, simply type GO GAM-208.

Once you're in the game, you have a choice of either reading the instructions or going directly into play. For the sake of all you novices, let's take a look at the instructions. They tell you, among other things, that the game is suggested for people age 12 and up and that there is no minimum number of players. However, there is a maximum of ten players because there are only five ships available on each side at any one time. (That means if you sign on during prime playing time—usually evenings—you may sometimes have to wait for an opening before you can get into the game.) You also can choose to play at the beginner, intermediate or expert level.

The instructions also tell you whether there are Romulans and/or black holes in the game. If there are only one or two players signed on, Romulans are thrown in to add challenge. They attack anybody—regardless of race, creed or political affiliation. As soon as a third ship enters, however, the Romulans quit. Black holes also add excitement—not to mention confusion—by sucking you into oblivion.

To get a list of commands—which is essential to your survival—type HELP. A feature you may not pick up right away, according to Captain Thrash, is the LIST option. Since you can view only a small area of the galaxy at one time on your screen, the LIST feature is essential for spotting enemy activity—or finding allies—outside your range of vision. Use this option to list closest (cl) friendly (fr) or enemy (e) ships (sh) or bases (ba), or to find a nearby neutral (ne) planet (pl), for instance. That can be important information when you're in trouble—or looking for it.

Next you choose your side (Empire or Federation) and your ship (depending on which ships aren't in use at the time). Once you're in the game, Captain Thrash suggests you use the HELP INPUT command to get information on some finer points of play. For instance, in DECWAR you move around, capture planets and fire at enemies by giving the appropriate command and the x-y coordinates of your objective (actually the y-x coordinates, since you put y first). Depending on your own preference, you can use either relative or absolute coordinates. If you

If there are only one or two players signed on, Romulans are thrown in to add challenge. They attack anybody—regardless of race, creed or political affiliation.

want to find out more about using coordinates and applying commands most efficiently, be sure to use the HELP INPUT option.

Captain Thrash, at this point, suggests a short range scan. What you'll see on your screen is a 15 x 15 grid of dots, with your ship in the middle. You'll also see asterisks (stars), @ symbols (planets—no letter beside it means a neutral planet, F means a Federation and E an Empire-controlled planet), perhaps some < > symbols (Federation starbases) and some > < symbols (Empire starbases). Starships, both Federation and Empire, are represented by the first letters of their names.

Now you have several options. Check for potential targets and begin an attack, capture any neutral planets in range and begin to build them into outposts—and eventually starbases—check the status of your ship's systems and/or check to see where enemies are located off-screen, so nobody sneaks up on you when you're not looking.

If you decide to capture a neutral planet or attack an enemy, both of which gain you points, Captain Thrash warns to stay on the alert for the positions of stars. If you hit one by accident, not only will it blow up and take everything nearby with it, but you'll lose 100 points. A neutral planet will fight back, by the way, when you move in to capture it, but will generally not do fatal damage to your ship. However, an enemy starbase will blow you to smithereens if you get too close, so the trick is to get close enough to be accurate, but stay far enough away to avoid getting hit. (A starbase's range is five dots in every direction.) You can most effectively attack a starbase from six dots away, with no objects—like stars or planets—in your path.

Two more things to remember when you're launching an attack. Don't fire too many phasers in a row or you stand a chance of overheating. And try not to move more than six dots at a time, or you could damage your engines. In its desperate escape from a particularly persistent enemy starbase, the Lexington, under the command of Captain Thrash, traveled eight dots in one move and almost did her engines in.

Once you've captured a planet, destroyed—or failed to destroy—an enemy, or otherwise used up some of your resources, you should consider docking at a friendly starbase or outpost to re-energize. Captain Thrash almost always does this as soon as he can, so he's never caught with his warp shields down, so to speak.

If there aren't any friendly bases around, the best thing to do, according to our venerable Captain, is create one by capturing a neutral planet and giving as many "build" commands as you can. Five "builds" will create a new starbase, but if your side is already operating the maximum number of allowable starbases, you'll be stopped at four "builds" and will have to settle for creating an outpost for the time being.

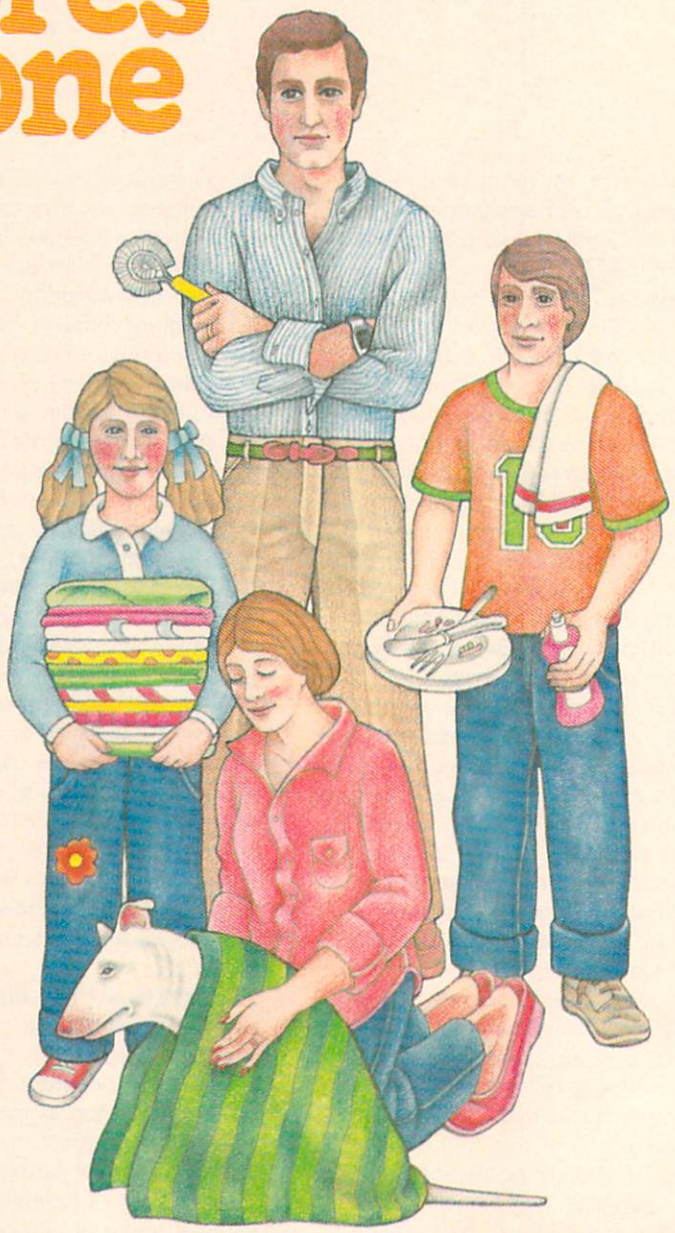
The advantage of docking is that your ship gets fixed. The disadvantage is that you lose time—and points are figured "per stardate." If you've got 1500 points and you get hung up for a lot of stardates at a docking station, your points-per-stardate obviously go down. If your ship is badly damaged, it could take five or six "dock" commands (each "dock" command uses up one stardate) to get her up to par. The trick, then, is to keep from getting hurt too badly.

Meanwhile, while you're sitting in New York busily taking care of your part of the galaxy, what happens if an enemy out in Albuquerque signs on in a part of the galaxy you can't see?? Or if a player signs off unbeknownst to you, reducing the number of players to two, and a couple of Romulans get thrown into the game? That's where you've got to remember to check on the status of your enemies frequently. Just like your favorite TV starship commander, right? Never take anything for granted.

Regardless of how much advice you get, however, we all know that the only way to learn to play a game well is to get in there and play it. That's just about what Captain Thrash said as he folded up his printout, gazed blandly around for a moment as if to re-orient himself, and strolled out of my office. If you stay alert, you may run into him on the system someday. When that happens, get on the radio and say howdy.

For more information on DEC-WAR, contact CompuServe customer service at 800-848-8990 (In Ohio 614-457-8600). **C**

Getting the Chores Done



A program for the VIC 20 that will help you divide up household chores equitably.

by John Heilborn

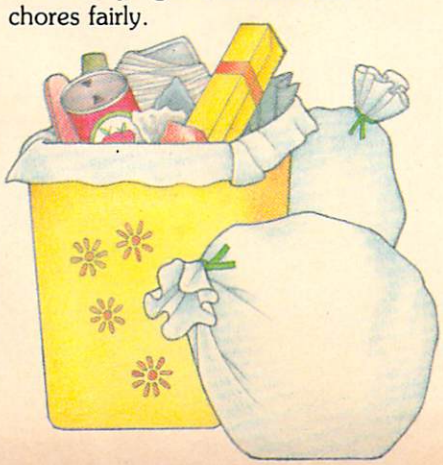
Illustration by JEAN GARDNER

I have a simply amazing ability to misplace things. I can't count the number of times I have found myself looking for a matching sock or the book I was supposed to return to the library. Unfortunately, the only way I know of to keep these things in order is to keep the house clean. But I hate housework . . . and so does my wife (who shares my talent for losing things). So we wind up going back and forth trying to figure out how to avoid being the one to (for example) wash the dishes after dinner or clean the bathroom on Saturday.

Another problem is my own procrastination. For instance, I really hate taking out the garbage, but I've been known to agree to take it out (on rare occasions). The problem is, I'll put it off until the morning the garbage truck comes by. The ensuing race to get to the side of the house before the garbage man does can be quite exciting.

None of this is much fun. More often than not, each of us winds up trying to assign the nasty chores to the other, knowing full well that the chores need to be done but not wanting to do them ourselves.

Without an arbiter, we had no simple way of determining who should do what. Finally in desperation, I turned to my VIC 20 for help. After consulting for several hours, we (the computer and I) came up with a workable program to hand out chores fairly.



If you have a similar situation in your house, you may want to settle your arguments with your trusty VIC and the same program I used. Here's how I did it:

First you'll need to put the chores into memory:

```
10 REM ** CHORE ROUTINE **
20 PRINT "HOW MANY CHORES
   ARE YOU ENTERING"
30 INPUT C
40 DIM A$(C), B(C)
50 FOR R=1 TO C
60 PRINT "CHORE"; R
70 INPUT A$(R)
80 B(R)=0
90 NEXT
```

If you have a datassette, you can store the chores in a data file by adding these lines to the program above:

```
15 OPEN 1,1,1,"CHORES"
35 PRINT#1, C
75 PRINT#1, A$(R)
100 CLOSE 1
```

To save the chores on a disk instead of a datassette, replace line 15 with:

```
15 OPEN 1,8,4,"@:CHORES,W"
```

(The @: allows you to change the chore list from time to time).

Okay, now that you've told the VIC what your chores are, you'll need to tell it who the folks are that will be doing them. The routine to do that is similar to the one above:

```
200 REM * PEOPLE ROUTINE *
210 PRINT "HOW MANY
    PEOPLE ARE YOU
    ENTERING"
220 INPUT C
230 DIM A$(C),B(C)
240 FOR R=1 TO C
250 PRINT "PERSON"; R
260 INPUT A$(R)
270 B(R)=0
280 NEXT
```

Once again, to store this on the datassette you'll need to add a line to OPEN the file, one to store the number of people, one to store the data and one to CLOSE the file:

```
215 OPEN 1,1,1,"PEOPLE"
235 PRINT#1, C
275 PRINT#1, A$(R)
290 CLOSE 1
```

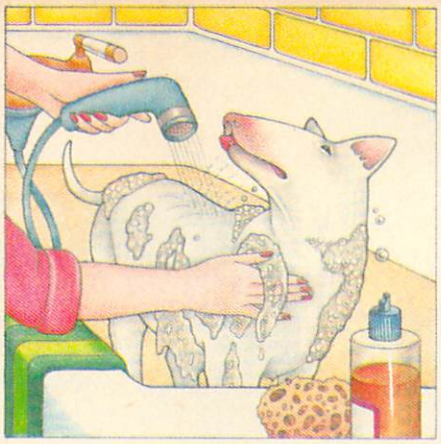
If you are using a disk drive change line 215 to:

```
215 OPEN 1,8,4,"@:PEOPLE,W"
```

Having determined what your chores are, and who's going to do them, the only task remaining is to match up the chores to the people. This is how it's done. I'm assuming that the number of people you have is an even multiple of the number of chores you have. If this isn't the case, pad your chore list with some extra chores, delete some, or add some chore items called "free time"—we can all use some of that.

```
600 REM * MASTER PROGRAM *
610 T=INT(C/X)
630 REM -- THE PEOPLE LOOP
640 FOR R=1 TO X
650 REM -- THE CHORE LOOP
660 PRINT P$(R): PRINT
670 FOR RR=1 TO  T
680 Z=INT(RND(0)*C)+1
690 IF B(Z)=0 THEN PRINT
    A$(Z): B(Z)=1: GOTO 710
700 GOTO 680
710 NEXT
720 PRINT"---------------"
730 NEXT
```

If you are running this program with a disk drive, precede this routine with:



```
300 REM ** DISK REVISION **
310 REM -- GET CHORES  --
320 OPEN 1,8,4,"CHORES,R"
330 INPUT#1, C
340 DIM A$(C), B(C)
350 FOR R=1 TO C
360 INPUT#1, A$(R)
370 NEXT
380 CLOSE 1
390 REM -- GET PEOPLE  --
400 OPEN 1,8,4,"PEOPLE,R"
410 INPUT#1, X
420 FOR R=1 TO X
430 INPUT#1, P$(R)
440 NEXT
450 CLOSE 1
```

If you want to use this program with a datassette, you'll have to make two changes:

The first one is to change the OPEN statements:

```
320 OPEN 1,1,0,"CHORES"
```

and

```
400 OPEN 1,1,0,"PEOPLE"
```

The second change is to store the programs in the correct order. This isn't necessary with the disk drive version because the disk drive can look for these routines and the data randomly.

Store the master program on tape first. The next program should be the tape version of the chores routine. After saving the chores routine, run it. By running it, you will set your data in the right place. The last program you need to save on tape will be the people routine. After saving it, run it also.

To run the master routine and set a chore list, rewind the tape and enter LOAD and hit return. The VIC will read the first routine and load all of your data automatically.

Using a Printer

You can print out the names and chores as well as displaying them on the screen. All you'll need to do is add a statement at the beginning of the master routine that OPENs the printer such as:

```
315 OPEN 2,4
```

and add a line that prints each line on the printer following each PRINT statement:

```
665 PRINT#2, P$(R):
PRINT#2
725 PRINT#2,"------------"
```

finally, you'll need to change line 690 to:

```
690 IF B(Z)=0 THEN PRINT
    A$(Z): PRINT#2,A$(Z):
    B(Z)=1: GOTO 710
```

and CLOSE the file.

```
740 CLOSE 2
```

It may be of value to note that this program will run on PET computers and the Commodore 64 as well as a VIC 20. For the PETs and older CBM machines you will need to replace the 0 in the RND function of the master program (line 680) with −TI. **C**

John Heilborn is an editor at Osborne/McGraw-Hill. He has recently co-authored the VIC 20 User Guide *with Ran Talbott. The price is $14.95 and it's available from: Osborne/McGraw-Hill, 2600 Tenth Street, Berkeley, CA 94710*

VIC SUPER PIANO

Last issue we ran a short program called "VIC Piano," which was a variation on a program in the VIC 20 user manual. To show you how programs evolve and grow, we thought you'd enjoy Super Piano, which lets you not only play tunes on your computer, but save them and replay them, complete with a changing color display.

by Myron Miller

For a couple of years now, I have been *silently* hacking away with my trusty PET. So when I brought home my new VIC, the first thing I wanted to try was music. I found a neat little program on page 78 of the *VIC 20 Personal Computer Guide* called "The VIC Piano." Being a hacker, I couldn't resist making a few changes.

Voila! VIC Super Piano! With Super Piano your VIC can:

display a different screen color for each musical note.

give you a choice of eight octave selections, three individual and five mixtures.

memorize a tune and replay it.

save a memorized tune on tape.

load a memorized tune from tape for replay.

print the keystrokes on the screen for tune composing.

Super Piano runs on an unexpanded VIC—all you need is a little bit of time to type it in.

Super Piano has three operating modes and a number of program commands. All of the modes and commands are selected by a single keystroke. First, let's take a look at the modes; then we will cover program commands.

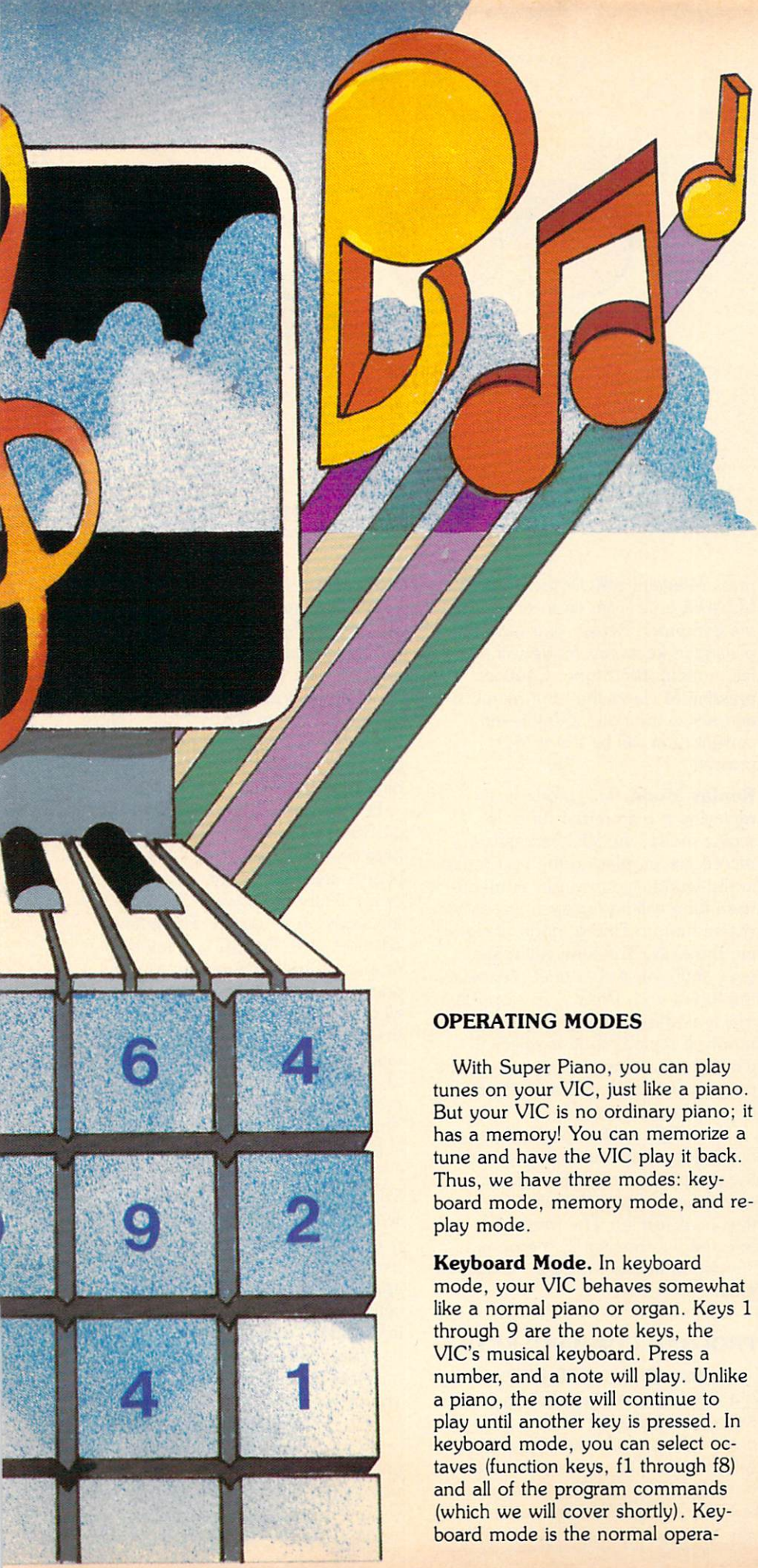


Illustration by JACK FREAS

OPERATING MODES

With Super Piano, you can play tunes on your VIC, just like a piano. But your VIC is no ordinary piano; it has a memory! You can memorize a tune and have the VIC play it back. Thus, we have three modes: keyboard mode, memory mode, and replay mode.

Keyboard Mode. In keyboard mode, your VIC behaves somewhat like a normal piano or organ. Keys 1 through 9 are the note keys, the VIC's musical keyboard. Press a number, and a note will play. Unlike a piano, the note will continue to play until another key is pressed. In keyboard mode, you can select octaves (function keys, f1 through f8) and all of the program commands (which we will cover shortly). Keyboard mode is the normal operational mode of the program and is selected by default. That is, unless you choose memory mode or replay mode, the program will automatically select keyboard mode. You will use keyboard mode to play the VIC like a piano and to select the program commands.

Memory Mode. Do you like a certain tune? Have you composed a piece of music that should be preserved? Super Piano can memorize your tune. Once memorized, a tune can be replayed by the VIC. You can sit back and hear your favorite tune without slaving over the keyboard. Also, you can save a memorized tune on tape and load it back into the VIC for future replay.

To memorize a tune, you must select memory mode. Do this by pressing the M key, the program command key for "memorize." Now

just play the tune as you would in keyboard mode. When you have completed the tune, press X to exit from memory mode to keyboard mode. That's all there is to it. Now you can hear your tune by selecting replay mode, or you can save the tune on tape.

When a tune is memorized, the computer stores the notes and octave selections in a "tune memory" (two 100 element arrays; one for key identification, the other for the time duration between keystrokes). The tune memory is limited to a total of 100 notes and octave selections. If you attempt to memorize more than 100 keystrokes, the program will automatically return to keyboard mode with 100 notes of your tune memorized. A counter is displayed at the top of the screen to let you know how many notes remain in the tune memory.

You may notice that the key response is a bit slow in memory mode as compared to keyboard mode. This is due to the additional operations required to memorize a tune. The response is fast enough for most tunes.

Memory mode is for memorizing tunes only. As such, the only keys that are operational are the note keys (number keys), the octave selectors (function keys, f1 through f8), the RETURN key (for memorizing carriage returns for the screen print-out during replay) and X. All the other program command keys will have no effect in memory mode.

Once memorized, a tune will remain in the tune memory until you press M again, reRUN the program, LOAD a tune from tape, or shut off the computer. Replay of a tune, or playing in keyboard mode will not affect a memorized tune. Caution: pressing M clears the tune memory and resets the note counter—any current tune will be lost if M is pressed.

Replay Mode. This mode is for replaying a memorized tune. In replay mode, the VIC becomes a record player, playing the best music in the world—your music! A memorized tune will be replayed just as you entered it in memory mode, including mistakes. The tune will repeat over and over with a brief silence at the tune's end. Press X to stop the replay and exit back to keyboard mode. If replay mode is selected when the tune memory is empty, the program will automatically return back to keyboard mode.

This mode is for replaying memorized tunes only. As such, none of the program command keys are operational except X (exit to keyboard mode). The mode is selected by pressing R, the program command key to replay a memorized tune.

PROGRAM COMMAND KEYS

1 through 9: NOTE KEYS. The number keys are the VIC's musical keyboard. They are like the keys of a piano. Press a number key, a note will play, and the screen will display one of nine colors. The 9 key does not play a note. Rather, it shuts off any playing note, but does not end the program (unlike VIC Piano in the owner's manual). The 9 key is used like any other note key, but it is used for silence (such as ending a tune or pauses). The note keys are operational in keyboard and memory modes but not in replay mode. Table I lists the musical note and screen color assignments for each of the nine note keys.

f1 through f8: OCTAVE SELECTORS. The function keys (large tan keys on the right side of the keyboard) are used to select an octave or a mixture of octaves. They are like the stops on an organ. The octave selectors give the VIC a greater range of musical notes. Some tunes require going up or down an octave to cover all of the notes in the melody. For these tunes, use the individual octaves: low, medium, and high (f1, f3, and f5). A quick example is the tune from "Close Encounters Of The Third Kind." To play it, press:

f5 5 6 4 f3 4 8 or
f3 5 6 4 f1 4 8.

You may want to face your TV screen to the night sky for this example. You can change octaves in the middle of a tune just by pressing the appropriate function key immediately before pressing the required note key. It takes a little dexterity, but so does everything else in music. The mixtures give an interesting sound. The three voices of f7 sound very close to an organ, and f8 has white noise for those who enjoy weird music. Like the note keys, the octave selectors are operational in keyboard and memory modes but not in replay mode. Table II has the octave assign-

ments for each function key.

M: Memorize a Tune. The M key selects memory mode. Note: remember, the instant that M is pressed, any current tune in the tune memory will be lost.

R: Replay a Tune. Use R to select replay mode. Note: both M and R are operational in keyboard mode only. This may seem a bit confusing. Once M or R have been pressed, the computer is no longer in keyboard mode. Pressing M or R, now, will have no additional effect.

X: Exit to Keyboard Mode. The X key is used to exit or return to keyboard mode from either memory or replay modes. It is operational in all of the modes but serves no real purpose in keyboard mode.

NOTE! The remaining command keys are operational only in keyboard mode (except the RETURN key). Pressing these keys will do nothing when you are in memory or replay modes. However, some of the commands will remain in effect during memory and replay modes. The command key must be pressed before the mode is selected. An example is the P command key. If it is pressed before entering replay modes the tune's notes will be printed on the screen as the tune is replayed.

S: Save Tune on Tape. If a tune has been memorized, you can save it on tape to replay. When S is pressed, the VIC will ask you for a file name. The file name is stored on the tape as well as the tune and is used to identify this particular file. Be careful with file names. You must enter the exact same file name when the tune is to be loaded. Keep an accurate record of the file names that you save the tunes with. Also, a saved tune on tape is for loading back into the VIC, not for an audio tape recorder.

CAUTION: Before saving a tune, make sure that you have removed your Super Piano program tape from the tape drive. Otherwise, you may end up recording the tune over the program. You may want to remove the "write protect tab" from the back of the cassette housing to prevent accidents.

L: Load a Tune from Tape. Use L to load tunes that have been saved on tape back into the VIC. Don't forget, you must enter the same file name that you used when you saved the tune. If the name is not exactly the same, the VIC will not find and load the tune. After the tune is loaded, you can hear it by pressing R for replay. Note: loading a tune from tape will clear the tune memory of current tunes.

P: Print Note Characters. P is used to display the note characters (keystrokes) of a tune on the screen. This is sheet music for us nonmusicians. With the editing commands that follow, you can keep an accurate printout of the keys needed to play a tune.

Q: Quit Printing. Q stops the printing of note characters, but it does not clear (erase) the screen. You can restore the printing by pressing P again. The printing will resume at the same point as where it left off. Use Q to preserve the accuracy of your note print-out by not printing when you are not sure of which notes are next.

RETURN: Carriage Return. RETURN performs its usual function; it starts the printout on the next line. Use it to divide your note printout into lines that follow the natural pauses in the tune. Note: RETURN is operable in memory mode as well as keyboard mode. If you press RETURN while memorizing a tune, the tune can be printed out in logical lines during replay. See the hint near the end of the article.

DEL: Delete Note Character. Press DEL whenever you hit a sour note. It will delete the most recent character from the screen printout. This will help keep your VIC sheet music accurate.

CLR: Clear Printout. CLR (shifted CLR/HOME key) stops the printout and clears (erases) the screen of all characters.

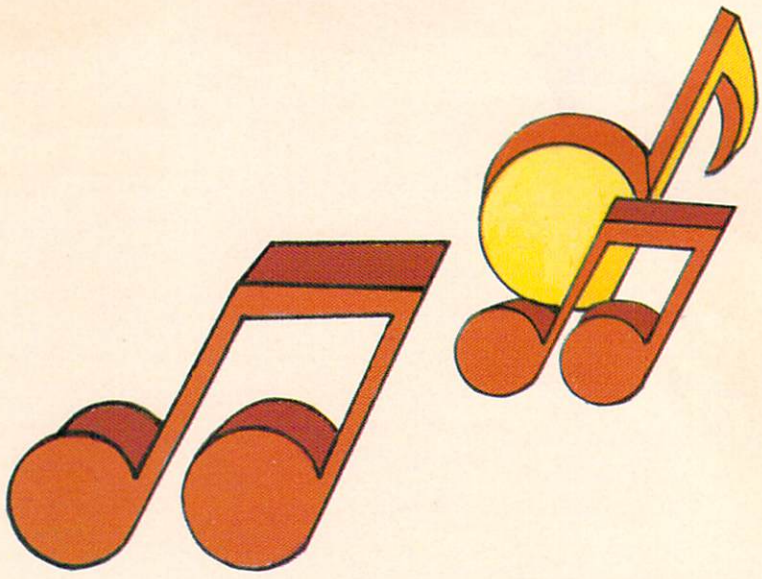
D: Disable Color. If you find that the flashing colors on the screen are driving you batty, press D. This will disable further color changes. When D is pressed, the screen will continue to display the present color.

C: Color Enable. Pressing C restores the screen's ability to change colors after D has been pressed.

K: Key Guide. Pressing K will display the key guide that appears at the beginning of the program.

To END the program, hold down the STOP key and press the RESTORE key, just like any other program.

Now that we have covered all the program command keys, I would like to comment further on the PRINT



and related commands. These commands are primarily for nonmusicians like myself. For us, trying to plunk out a tune for the first time is an exasperating procedure of trial and error. We randomly press the note keys until we find the right combination. The PRINT command will save fooling around with a pencil and paper until you get the tune right.

Think of a tune, now plunk around the number keys until you find the first note. Press P, enter the first note, and then press Q. OK, start plunking around for the second note. When you find it, again press P, enter the note, and then press Q. Do this for each note of the tune. If you make a mistake, use the DEL key to delete the sour note from the screen. At pauses in the tune, start printing on a new line by pressing the RETURN key.

When you are done, all of the note keys needed to play the tune will be on the screen. Now get your pencil and paper, and write down the keystrokes for future reference. The musical note keys are printed in normal video. The octave selectors appear in reverse video characters. Don't forget that the function keys are shifted for even numbers, unshifted for odd. None of the other keys are printed on the screen. I have found the PRINT and related commands to be a big help when trying to compose a tune.

HINT: The printout will remain on the screen in memory mode. You may find it easier to follow a tune when it is printed on the screen than on paper. If so, print the tune on the screen (press P and play the tune) before memorizing it. Also, you can get a print-out of a tune that is in memory, perhaps loaded in from tape. First, clear the screen with the CLR key. Now press P, and then press R. As the tune is replayed, the notes will be printed on the screen. If you used the RETURN key when the tune was memorized, the printout will be in lines that follow the natural pauses of the tune.

VIC Super Piano has many commands and features. The best way to learn the program is to experiment with it. Before long, you will be playing your favorite tunes like a pro. Super Piano may not sound like a Steinway, but then again, try playing GORF or balancing a checkbook on a grand piano. A computer is a playground for your mind and your imagination. I hope that you will find Super Piano to be a playground for your musical creativity. Who knows, Super Piano and your VIC may bring out the Bach in you! Here is a familiar tune to get you started.

```
f3  5  5  6  5  8  7
 5  5  6  5 f5  2  1
f3  5  5 f5  5  3 f3  8  7  6
f5  4  4  3  1  2  1  9
```

REFERENCES

Personal Computing On The Vic 20. Fifth Edition. Commodore Electronics Ltd. 1982. pp. 78-79.

Finkel, A., and others. *VIC 20 Programmer's Reference Guide,* Commodore Business Machines, Inc. 1982. **C**

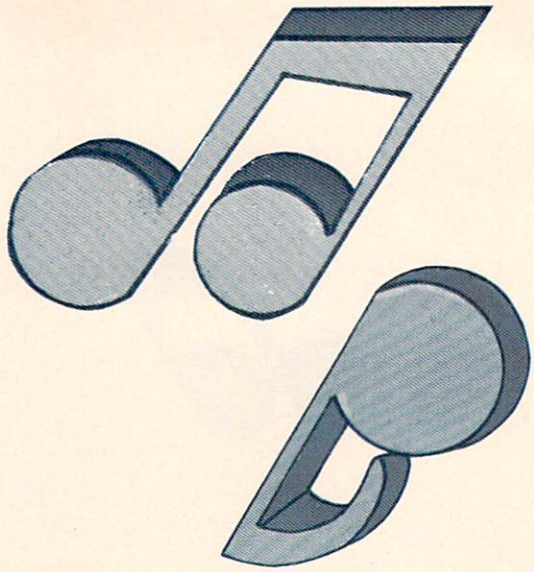
KEY	MUSICAL NOTE	SCREEN COLOR
1	DO	BLUE
2	RE	CYAN
3	MI	GREEN
4	FA	YELLOW
5	SOL	ORANGE
6	LA	RED
7	TI	PINK
8	DO	PURPLE
9	SILENCE	WHITE

TABLE I. Musical note and screen color assignments for the number keys.

FUNCTION KEY	OCTAVE SELECTION
f1	LOW
f3	MEDIUM
f5	HIGH
f7	LOW, MEDIUM, & HIGH
SHIFTED	
f2	LOW, & MEDIUM
f4	LOW, & HIGH
f6	MEDIUM, & HIGH
f8	LOW, MEDIUM, HIGH, & WHITE

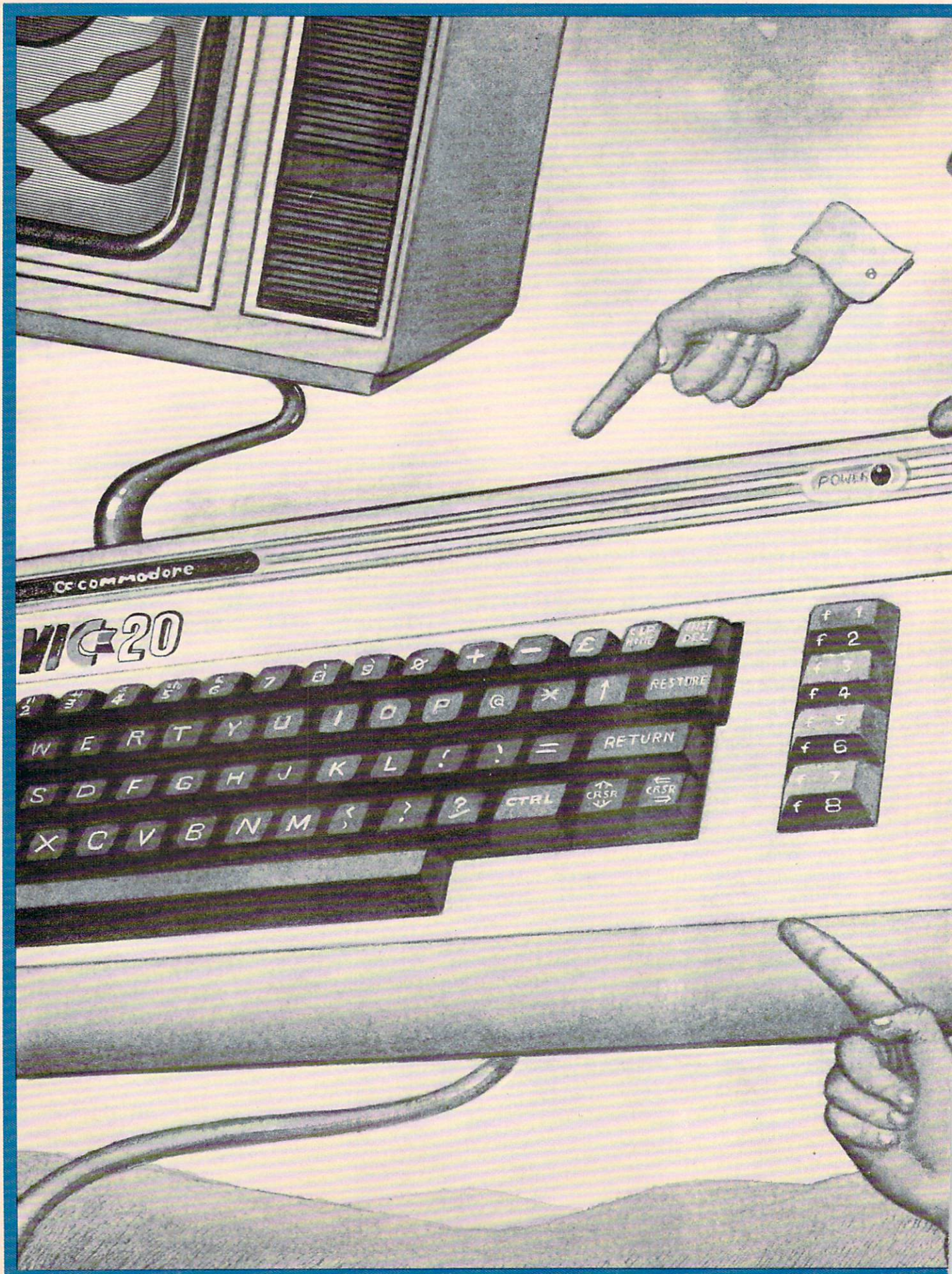
TABLE II. Octave selections for the function keys.

```
10 REM SUPER PIANO  MODIFIED PIANO PROGRAM PG 78 VIC USER MANUAL
20 REM MODIFIED BY  MYRON MILLER
100 REM INITIALIZATION
110 DIMA%(100),T%(100)
120 S1=36874:S2=36875:S3=36876:S4=36877:V=36878:C=36879
130 FORX=1TO8:READN%(X),B%(X),OC%(X):NEXTX
140 DATA223,104,49,227,56,?1,230,88,53,231,120,55
150 DATA234,136,50,236,40,52,238,168,54,239,72,56
160 BL$="":FORX=1TO21:BL$=BL$+" ":NEXTX
200 REM INSTRUCTIONS
205 GOSUB1000:PRINTCHR$(147)CHR$(144)
210 PRINT" KEY GUIDE":PRINT:PRINT" 1 TO 8: NOTES"
215 PRINT" 9: SILENCE"," F1 TO F8: OCTAVES"," M: MEMORIZE TUNE"
220 PRINT" R: REPLAY TUNE"," X:EXIT M OR R ABOVE"," S: SAVE TUNE ON TAPE"
225 PRINT" L: LOAD TUNE OFF TAPE P: PRINT NOTES"," Q: QUIT PRINTING"
230 PRINT" RETURN: NEW LINE"," DEL: DELETE NOTE CHR"
235 PRINT" CLR: CLEAR PRINT"," D: DISABLE COLOR"," C: COLOR ENABLE"
240 PRINT" K: KEY GUIDE"," STOP & RESTORE: END"
245 PRINT"PRESS ANY KEY TO CONT";
250 GETX$:IFX$=""THEN250
255 POKEV,15::PRINTCHR$(147)
300 REM READ KEYS
305 PRINTCHR$(19)BL$CHR$(19)"KEYBOARD READY"
310 IFM=1THEN720
315 IFR=1THENRETURN
320 GETA$:IFA$=""THEN320
325 T=TI:IFP=1 OR Q=1THEN335
330 PRINTCHR$(19)BL$CHR$(19)
335 IFA$="X"THENM=0::Q=0:GOSUB1000:GOTO305
340 Z=VAL(A$):IFZ>0THENGOSUB510:GOTO310
```



```
345 FORX=133TO140:IFA$=CHR$(X)THENFK=X-132:GOTO430
350 NEXTX:IFA$=CHR$(13)AND(P=1 OR M=1)THENPRINT:GOTO310
355 IFM=1ORP=1THEN310
360 IFA$="D"THENC=65535:GOTO320
365 IFA$="C"THENC=36879:GOTO320
370 IFA$="P"THENP=1:GOTO420
375 IFA$="Q"THENQ=1:P=0:GOTO320
380 IFA$=CHR$(20)THENPRINTCHR$(20)CHR$(20);:GOSUB1000:GOTO320
385 IFA$=CHR$(147)THENP=0:Q=0:PRINTCHR$(147):GOTO305
390 IFA$="M"THENGOSUB1000:PRINTCHR$(19)BL$,BL$CHR$(19)"MEMORIZE TUNE":GOTO710
395 IFA$="R"THENGOSUB1000:PRINTCHR$(19)BL$,BL$CHR$(19)"REPLAY TUNE":GOTO810
400 IFA$="S"THENPRINTCHR$(147)"SAVE TUNE ON TAPE":GOTO900
405 IFA$="L"THENPRINTCHR$(147)"LOAD TUNE FROM TAPE":GOTO950
410 IFA$="K"THEN205
415 GOTO320
420 IFQ=1THENQ=0:GOTO320
425 PRINT:PRINT:GOTO320
430 IFP=0THEN310
435 PRINT" "CHR$(18)CHR$(OC%(FK))CHR$(146):GOTO310
500 REM PLAY NOTE
510 IFP=1THENPRINT" "A$;
520 IFZ<9THENGOSUB1010:FORX=1TO25:NEXTX:POKEC,BX(Z):GOTO540
530 FORX=15TO1STEP-1:FORY=1TO25:NEXTY:POKEV,X:NEXTX:GOSUB1000:POKEV,15:RETURN
540 ONFKGOTO550,560,570,580,590,600,610,620
550 POKES1,NX(Z):RETURN
560 POKES2,NX(Z):RETURN
570 POKES3,NX(Z):RETURN
580 POKES1,NX(Z):POKES2,NX(Z):POKES3,NX(Z):RETURN
590 POKES1,NX(Z):POKES2,NX(Z):RETURN
600 POKES1,NX(Z):POKES3,NX(Z):RETURN
```

```
610 POKES2,N%(Z):POKES3,N%(Z):RETURN
620 POKES1,N%(Z):POKES2,N%(Z):POKES3,N%(Z):POKES4,N%(Z):RETURN
700 REM MEMORIZE
710 FORX=1TO100:A%(X)=0:T%(X)=0:NEXTX:M=1:MC=0:Q=1:P=0:TM=TI:GOTO320
720 MC=MC+1:T%(MC-1)=T-TM
730 IFMC>100THENPRINTCHR$(19)CHR$(17)"TUNE MEMORY FULL":GOTO750
740 A%(MC)=ASC(A$):TM=T:PRINTCHR$(19)BL$CHR$(19)"NOTES LEFT: "100-MC:GOTO320
750 GOSUB1000:M=0:Q=0:MC=100:FORX=1TO1000:NEXTX:GOTO305
800 REM REPLAY
810 R=1:RC=0:IFMC=0THENPRINTCHR$(19)CHR$(17)"NO TUNE IN MEMORY":R=0
820 PRINT:PRINT:FORX=1TO1000:NEXTX:IFR=0THENQ=0:P=0:GOTO305
830 GETX$:IFX$="X"THENGOSUB1000:R=0:Q=0:P=0:GOTO305
840 RC=RC+1:IFRC>MCTHEN880
850 A$=CHR$(A%(RC)):TR=TI+T%(RC):GOSUB325
860 IFTI<TRTHEN860
870 GOTO830
880 Z=9:GOSUB520:FORX=1TO1000:NEXTX:RC=0:GOTO830
900 REM SAVE
910 GOSUB1000:NA$="":INPUT"ENTER TITLE";NA$:CLOSE1:OPEN1,1,2,NA$
920 PRINT#1,MC:FORX=1TOMC:PRINT#1,A%(X):PRINT#1,T%(X):NEXTX
930 CLOSE1:PRINT"SAVE COMPLETED":GOTO990
950 REM LOAD
960 GOSUB1000:NA$="":INPUT"ENTER TITLE";NA$:CLOSE1:OPEN1,1,0,NA$
970 INPUT#1,MC:FORX=1TOMC:INPUT#1,A%(X):INPUT#1,T%(X):NEXTX
980 CLOSE1:PRINT"LOAD COMPLETED":GOTO990
990 FORX=1TO2000:NEXTX:PRINTCHR$(147):GOTO305
1000 POKEC,24
1010 POKES1,0:POKES2,0:POKES3,0:POKES4,0:RETURN
```


POWER
Commodore
VIC 20
RESTORE
RETURN
CTRL
f 1
f 2
f 3
f 4
f 5
f 6
f 7
f 8

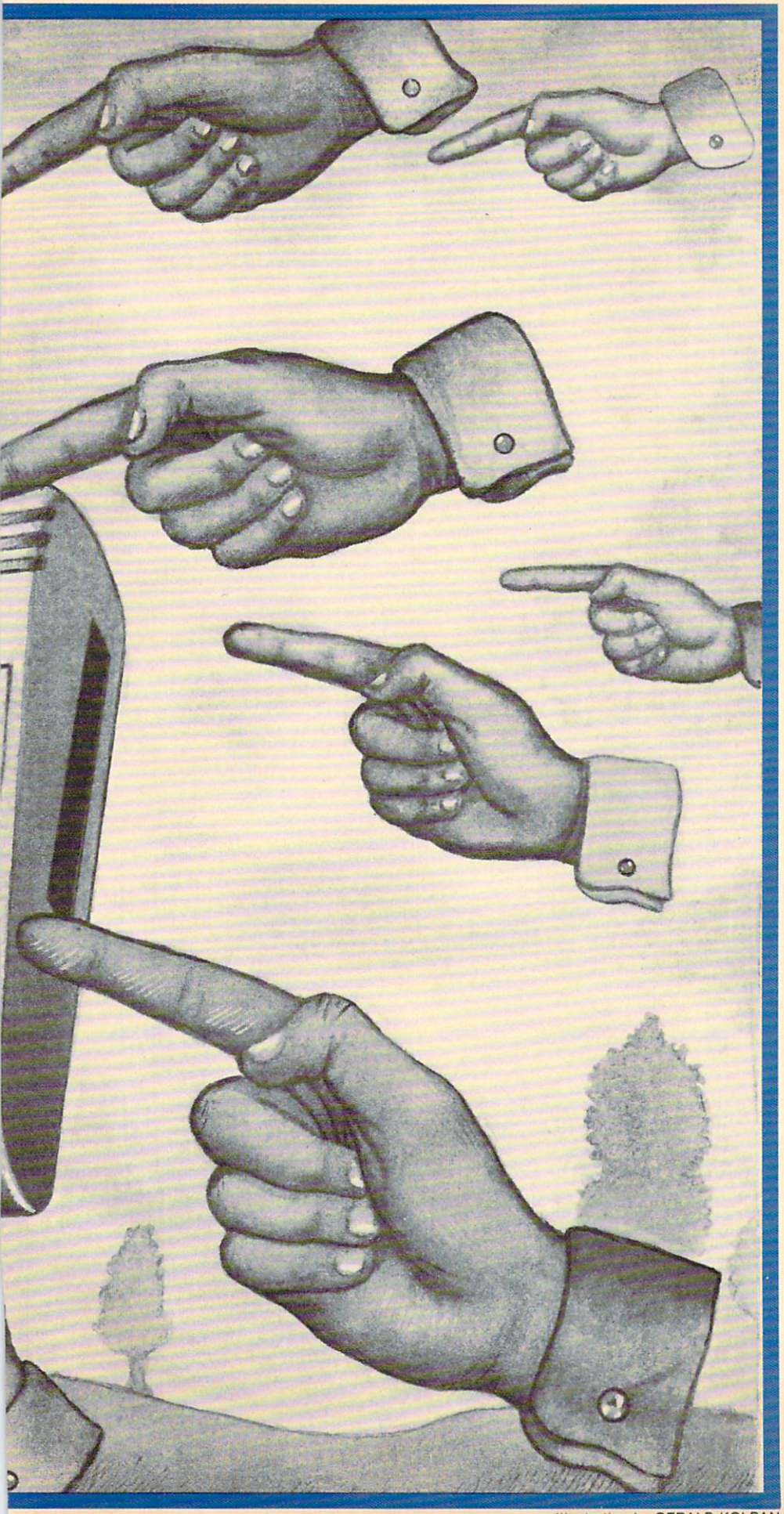


Illustration by GERALD KOLPAN

MACHINE LANGUAGE IS A POKEr's GAME

Machine language subroutines are terrific devices for enhancing your programs, but can seem mysterious and difficult to beginners. In this article David Funte provides a couple of tricks for getting machine language subroutines into your programs without a lot of struggle, and helps give you a feel for this sophisticated programming technique.

by David H. Funte

When you read all the articles on machine language you can sometimes get the impression that it's not so difficult to learn. But those impressions quickly fade when you start to read the textbooks. Learning machine code takes patience; you can't learn it all in one day.

If you decide to tackle it anyway, I recommend a good machine language monitor like VICMON. If you wish to write a program for the VIC 20 using more than 3.5K RAM, you will also need an expansion cartridge and an expansion board to connect it all together. That's how to do it, under ideal conditions. But if you're at the stage where your wallet can't handle it, you have to make do with what you have.

For instance, I have an 8K expander and a VICMON cartridge, but I don't own an expansion board to connect the two. For this reason it seemed impossible to add a machine language subroutine to a program requiring more than the standard 3.5K RAM. I would have to type in POKE (address), (value): POKE (address), (value). . . etc. This is a long and monotonous process.

To make life a little easier I wrote the program VIC POKEr, using ideas from past issues of *Power/Play* and *Commodore* magazines.

For the Key to reverse graphic symbols, see page 7.

```
50 POKE650,128
51 PRINT"■■■■■■ENTER COMMAND"
52 PRINT"■■■ERASE PROGRAM LINES"
53 PRINT"■■■ISPLAY CON ENTS OF MEMORY"
54 PRINT"■■■IC POKER"
55 GETZ$:IFZ$=""THEN55:REM DECIDE WHICH SUBROUTINES TO USE
56 IFZ$="E"THENGOSUB91:GOTO80
57 IFZ$="D"THENGOSUB90:GOTO70
58 IFZ$="V"THENGOSUB90:GOSUB91:GOTO60
59 GOTO55
60 PRINT"■■■■■":FORT=XTOX+19:REM BEGINNING OF VIC POKER ROUTINE
61 Z=Z+1
62 IFT=X+19THENPRINTZ"■POKE"T","PEEK(T)"■":GOTO68
64 PRINTZ"■POKE"T","PEEK(T)
68 NEXT:END
70 PRINT"■":FORT=XTO65000:REM BEGINNING OF DISPLAY ROUTINE
72 PRINT"LOC #"T"="PEEK(T)
74 NEXT
80 PRINT"■■■■■":FORT=Z+1TOZ+20:REM BEGINNING OF ERASE ROUTINE
81 IFT=Z+20THENPRINT"■":GOTO84
82 PRINTT
84 NEXT:END
89 REM QUESTIONS ASKED BY PROGRAM
90 PRINT"■■■■■WHAT IS THE FIRST":PRINT"MEMORY LOCATION?":INPUTX:RETURN
91 PRINT"■■■■■ENTER FIRST LINE #":INPUTZ:Z=Z-1:RETURN
92 REM VIC POKER (C)1983 DAVID H. FUNTE
```

Let's browse through the program. Line 50 POKEs 128 into address 650. This sets all the keys to repeat when you hold them down. Having all the keys set to repeat makes it easy to create REM statements—and REM statements are convenient locations for machine language subroutines.

For example, enter POKE 650,128 and hit RETURN. Now type the following line, but don't hit RETURN this time:

10 REM

Hold the "A" key down and in no time at all you have four rows of A's. Now hit RETURN.

Since 65 is the token the VIC uses to store the A, when you look through memory for the location of this REM statement, you only have to look for a long string of 65's. (A list of VIC 20 tokens is on page 121 of the *VIC 20 Programmer's Reference Guide.)*

Lines 51-54 display the command options of the program. Lines 55-59 tell the VIC which subroutines to use. Line 55 is a GET statement that loops back to itself if no key is hit.

Lines 56-58 make all the decisions of the program. For example, if "E" is hit, the program GOSUBs to line 91, which asks for a line number, INPUTs Z and RETURNs to line 56. Then the program GOes TO line 80.

Line 59 goes back to line 55 if the key that was hit was not a legal entry. Lines 60-68 contain the subroutine that draws twenty program lines on the screen and then ENDs. At this point VIC is in the screen edit mode. For example:

1000 POKE 4102,65
1001 POKE 4103,65
1002 POKE 4104,65 . . . etc.

The lines are not actually entered as program lines. They can be, however, simply by hitting RETURN on each line.

Lines 70-74 contain the subroutine that displays the contents of memory. Lines 80-89 contain the subroutine that allows you to erase program

lines with ease. Its use will be described later.

Running the Program

As I mentioned previously, a good location for a machine language subroutine (which I will call an MLS from now on) is the REM statement. Since REM statements are ignored by BASIC, the MLS will be ignored until you call it with BASIC's SYS command. If you put the REM statement at the beginning of your program it will be located near the beginning of RAM, and will be easy for the computer to find. However, if you bury it inside the program the statement will take longer to find.

Using the display function of VIC POKEr will make this task a simple one. Enter the following line:

99 REMZZZZZZZZZZZZZ etc. (four rows of Z's).

Let's say you want to place an MLS inside this statement. First you would have to find where it is in memory. To do that, first RUN the program. The first display tells you to "enter command." Enter "D" for display. Now the VIC asks for the first memory location. Since the REM statement has to be located in RAM, enter the address of the first byte of RAM (4096) and hit RETURN.

Wow! The VIC starts displaying the contents of each address from 4096 on. You can slow down the display a bit by using the control key. Remember, you're looking for the REM statement containing all those Z's. The token for Z is 90, so you should look for a large number of bytes that have 90 as their value. When you locate the REM statement, hit the STOP key.

Your next task is to find the address of the first Z. If you think you've passed it, reRUN the program and display the contents of memory again. When you find the first Z, write its address on a piece of paper.

On my VIC the first Z was at address 4630, so from now on I will use 4630 as the "first address." If it was somewhere else on your VIC, don't worry. As you work through the examples, simply substitute the address that is correct for your VIC.

Using the VIC POKEr Routine

Here is where you write your MLS. As an example, we will use the following program, which will simply print the alphabet.

```
      LDX #$41
start TXA
      JSR $FFD2
      INX
      CPX #$5B
      BNE start
      RTS
```

If you haven't had much experience with assembly language, one look can be enough to discourage you. Just remember how strange BASIC looked when you were first learning it and keep reading.

Assembly language commands have their own set of tokens. A list of these tokens is on page 164 of the *VIC 20 Programmer's Reference Guide*. They are labeled 00-FF in hexidecimal. To make the list easier to read, you may wish to re-label the numbers in decimal (0-255).

The first program lines can be converted to the tokens 162 and 65. The token for LDX (load X register with) is 162. The value we want loaded into the X register is #$41 in hexidecimal. Without getting into a lengthy explanation of the hexidecimal system, let me just state that #$41 in hexidecimal is 65 in decimal.

Using the same logic it is possible to find the tokens for the rest of the program. A list of tokens for the complete assembly language program is below.

Assembly	Token
LDX	162
#$41	65
TXA	138
JSR	32
$D2	210
$FF	255
INX	232
CPX	224
5B	91
BNE	208
start	247
RTS	96

We will now POKE these tokens into the REM statement using the VIC POKEr routine. RUN VIC POKEr and for the command hit "V." The computer will ask for the first memory location (location of the first Z in the REM statement). Enter 4630 (or whatever your first address is) and hit RETURN. Now the VIC asks for a line number. Enter 1000.

Just watch now as the VIC does most of your work for you. The display shows twenty lines that can easily be entered as program lines by hitting RETURN. Before you hit RETURN, however, let's make some changes.

Move the cursor over to the 90 on line 1000 and change it to 162 (the first token of the assembly language program). Hit RETURN. You have just entered the line into the program. Do the same now for the second token (65). Place it in line 1001. Complete the same process for each of the tokens, remembering to hit RETURN after each change.

When you have finished, LIST the program. You should see that lines 1000-1011 have been added to the program. Clear the screen and LIST 99. The REM statement hasn't changed. Now enter RUN 1000 and when the VIC says READY, LIST 99 again. You should see a big change in line 99 this time. The line should now read as follows:

99 REMNEWARUN DIM(symbol for Pi)CLRSAVE(INPUT#—ZZZZZ etc.)

You should be able to recognize a few BASIC commands in there like NEW, RUN, DIM, etc.

If you look at the table of BASIC tokens (page 121 in the *Reference Guide)*, you will see that the token for the command NEW is 162. Now look at the tokens we used for the MLS. The token for the first com-

mand (LDX) is also 162. So how does the VIC tell the difference?

When you LIST a program, the computer reads all the tokens in RAM as BASIC tokens. The same thing happens when you RUN the program. But BASIC ignores REM statements. That is why we put the MLS inside a REM statement.

Then how do we RUN our MLS? Very simple. The VIC, in its infinite wisdom, has a BASIC command that allows us to leave BASIC and enter the world of machine language. This command is SYS. SYS is similar to BASIC's GOSUB, in that the VIC jumps to a specified address and then returns to the original location. The command that tells VIC when to return is RTS (return from subroutine) and is the same command (token 96) we used at the end of our MLS.

Let's try to RUN the MLS. Enter SYS4630 and hit RETURN. (Remember that if your "first address" is different from mine, you must SYS to that location). The VIC should promptly display the alphabet. If it did not, list lines 1000-1011 and check for mistakes. When you find the mistake, correct it using the VIC's screen editor and reRUN line 1000. When the VIC says READY, try RUNning the MLS again.

Erase Function

Now that we have the MLS in place, lines 1000-1011 are no longer needed. In fact, they are taking up valuable memory that might be needed later on. This is where the erase function of VIC POKEr comes in handy.

RUN VIC POKEr, but this time hit "E" for the command. The computer now asks for the first line number. Enter 1000 (the first line we want eliminated). The display shows the numbers 1000-1019 and places the cursor on line 1000. Just hit RETURN through line 1011 and those lines are gone forever.

User Tips

In the beginning we created a REM statement with four rows of Z's. This made it easy to find the REM statement in RAM. But since we needed only twelve bytes for the MLS, we could have eliminated all but the first twelve Z's. This would have saved about 60 bytes of memory (one byte for each Z). If you want to save memory, remember to remove excess bytes BEFORE placing the MLS into the REM statement, because attempting to alter the REM statement afterwards can destroy your program.

Machine language can act very weird at times. If you enter commands in an illogical order, the entire system can "crash." The only way to regain control of your VIC if this happens is to turn it off and then back on again. For this reason it is a good idea to SAVE your program often as insurance against such a disaster.

Also, the address of the start of your MLS can move. If you add a program line before the MLS, the VIC will shove the rest of your program up in memory to make room for the addition. If you remove a line (or even a single byte), the VIC will squeeze the program together to fill the hole. These shifts in memory will not affect the MLS itself. However, the shift will move the MLS to a different location. To find the new location use the display function of VIC POKEr and look through memory for the first two or three tokens of your MLS. Once you find it, you will have to SYS to the new location to call your MLS.

Short Cuts

If you own a VICMON cartridge it can tell you which tokens to use for your MLS. Write your MLS on VICMON and then display the contents of memory, using the "M" function of VICMON. For example, type the following program into VICMON:

```
1400 LDX #$41
1402 TXA
1403 JSR $FFD2
1406 BRK
```

Now enter the following line:

```
M 1400 1406
```

The VIC now displays the contents of memory between 1400 and 1406 in hexidecimal:

```
1402 A2 41 8A 20 D2
1405 FF 00
```

It's now up to you to translate this into decimal.

When you're beginning to write a program that will have an MLS, LOAD VIC POKEr and write your program around it. In doing this, you can easily call upon the special talents of VIC POKEr by RUNning line 50. On the other hand, line 0 can be used to bypass VIC POKEr. Just make line 0 GOTO your program.

Lines 1-49 can be used for your machine language subroutines. By placing the subroutines at the beginning of your program, you are insuring their starting address. In other words, when you alter your BASIC program, the MLS will not shift to a different location.

VIC POKEr uses about 500 bytes of memory, so when you are finished using it, erase it. But remember if there is an MLS after the spot where VIC POKEr used to be, the MLS will move when VIC POKEr is erased.

Conclusion

The explanations I've given here are over-simplified and, by themselves, cannot give a thorough picture of how machine code works. If you have a desire to learn more about programming in machine code there are some good books that can be of help. Two of these are *Programming the 6502* by Rodney Zaks and *Programming and Interfacing the 6502* by Marvin L. DeJong.

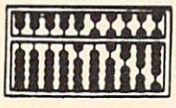
When you read about something that's new, try to use it on your VIC. Knowledge is acquired by reading, but the only way to learn is by doing. **C**

You Can COUNT On Abacus Software

VIC™ COMMODORE 64™ PET®

*SPRITE-AID sprite editor with joystick option for CBM-64™	$14.95
*SYNTHY-64 music & Sound Synthesizer.Fantastic! for CBM-64™	$29.95
*SCREEN-GRAPHICS-64 add graphics commands to BASIC incl. sprites	$24.95
*SKIER-64 exciting gameware	$14.95
*Tiny Basic Compiler for Vic,™ CBM-64™ or Pet.®	$19.95
*BUDGETEER Visual planner for Vic,™ CBM-64™ or Pet.®	$19.95
*QUICK CHART presentation chartmaker for CBM-64™ or VIC 20™	$14.95
*TINY FORTH language for CBM-64™ or VIC 20™ (April 25th)	$19.95
VIC GREAT BALLOON RACE another exciting game	$14.95
VIC I-CHING oriental fortune teller (8K expander)	$24.95
VIC SUPER EXPANDER SCREEN DUMP prints your graphics	$14.95
VIC JOYSTICK PAINTER	$14.95
VIC OR PET VIGIL games language with 9 games	$29.95
VIC OR PET PIPER the music machine!	$19.95
VIC HIRES / MULTICOLOR GRAPHICS UTILITIES (no extra memory)	$19.95
VIC GRAPHVICS super full-screen graphics (req. 3K or 8K mem.exp.)	$24.95
VIC TINY PILOT educational language	$17.95
CRIBBAGE (VIC 20™ req. 16K) for CBM-64™ or VIC 20™	$14.95
VIC MACHINE LANGUAGE GUIDE ($6.95 foreign)	$ 5.95
PET MACHINE LANGAGE GUIDE ($7.95 foreign)	$ 6.95
PET TINY PASCAL PLUS + (req. 32 K)	$39.95
BASIC REFERENCE CARD ($2.00 foreign)	$ 1.50

Write for FREE Catalog or for fast service, call our Order Line.

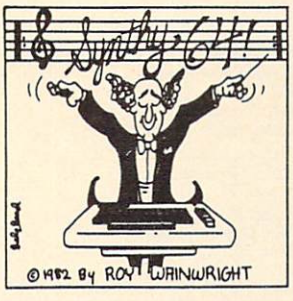


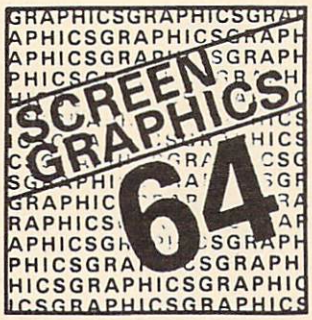
Abacus Software MasterCard VISA

P.O. Box 7211, Grand Rapids, MI 49510 616 / 241-5510

All software packages come complete with instructions or manuals. Postage and handling $1.50 U.S. and Canada. $3.00 elsewhere. For disk enclose $3.00 per software package. Payment acceptable in U.S. dollars by check, international money order, VISA, MC, ACCESS, Barclaycard

PET is a registered trademark of Commodore Business Machines, Inc.
VIC 20 and Commodore 64 are trademarks of Commodore Electronics Ltd.





TINY PILOT

access: commodore user groups

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
1206 N. Fraser Drive
Mesa, AZ 85203
Contact: Paul V. 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

Lawrence Hall of Science
UC Berkeley
Computer Project, Room 254
Berkeley, CA 94720
(415) 527-9286

PALS (PETS Around Livermore Society)
886 South K
Livermore, CA 94550
Contact: John Rambo

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) 346-6555
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

BAMBUG
1450 53rd Street
Emeryville, CA
(415) 523-7396

North Orange County Computer Club
3030 Topaz, Apt. A
Fullerton, CA 92361
Dave Smith

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
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

Vincent Yanniello's VIC 20 Software Exchange Club
2130 Colby Avenue
West Los Angeles, CA 90025
(213) 479-3000

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

64 Users Group
6029 Conta Costa Rd.
Oakland, CA 94618
E. R. Kennedy

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

COLORADO

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

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

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

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

DuPage-64
708 Prairie Ave.
Glen Ellyn, IL 60137
(312) 858-0637
Gus Pagnotta

Oak Lawn Commodore Users Group
The Computer Store
11004 S. Cicero Ave.
Oak Lawn, IL 60453
Bob Hughes

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

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

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

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.
Winnsboro, 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

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

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

VIC Commodore Club
2765 Bristol Rd.
Dowling, MI 49050
Bill Kelley

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

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

Clearwater Club
Clearwater School
Star Route
Piedmont, MO 63957
Contact: Carolyn Polk

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

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

access: commodore user groups

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

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

OKLAHOMA

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

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

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

Bucks County Commodore
User Club
72 E. Rambler Dr.
Holland, PA 18966
(215) 322-0394
Jim Dubrouillet

ABEPET Users Group
P.O. Box 6051
Lehigh Valley, PA 18001
Bill Thomas

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

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 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

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

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

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

Tidewater Commodore Users Group
417 Bethune Dr.
Virginia Beach, VA 23452
H. Thomas Baise, III

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.

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

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

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 Daniel Cayer
R.R. 6
Simcoe, Ontario
Canada N3Y 4K5

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 R2V0H9
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

KOREA

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

MEXICO

Asociacion De Usuarios 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

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:

Florida

VIC-Heartland User Group
1220 Bartow Rd. #23
Lakeland, FL 33801
813-666-2132
Tom Keough

VIC User Group
4796 31 St. North
St. Petersburg, FL 33714
Chuck Fechko

Indiana

Commodore User Group
2019 E. Madison
South Bend, IN 46617
219-289-8634
John Vance

Oklahoma

Central Oklahoma Commodore User Group
566 Buchanan
Norman, OK 73064
405-364-2154
Joe Tillison

North Carolina

Computer 2000/64 Users Group
Mohammed Company
6164A Falls of Neuse Rd.
Raleigh, NC 27609
919-872-2022

Classified

WHAT'S SMALLER THAN A BUSINESS CARD?

FASTER THAN CASSETTES?

AND FAR LESS EXPENSIVE THAN DISKS?

Why the ESF-20/64 Stringy Floppy from Exatron, of course. Our exciting little storage alternative gives you near floppy disk speed and reliability at a budget-minded price. Our high quality digital mass storage system is the perfect product to fill the gap between cassettes and floppy disks. And that's especially true in the case of the Commodore VIC-20® and Commodore 64® microcomputers.

The Exatron Stringy Floppy system is based on a small endless loop tape cartridge we call "The Wafer". This wafer measures only 2 3/4" x 1" x 3/16" — or about the dimensions of a standard business card. Wafer was born to run fast — at around 7200 baud, or 14 times faster than a standard cassette and has a memory capacity of 64K bytes. The ESF-20/64 system costs less than $200 and wafers are less than $3.00 each. No wonder computer people call us the "poor man's disk system".

If you'd like to get the world on a string, and bring your VIC-20® or Commodore 64® into the 20th Century, just fill out the coupon below and mail it to EXATRON, 181 Commercial Street, Sunnyvale, CA 94086. If you need to place an order call (408) 737-7111, outside California (800) 538-8559 and ask for Chrissy or Natalie.

The ESF-20/64 Stringy Floppy System from Exatron for use with the Commodore VIC-20® and Commodore 64® microcomputer.

exatron

For exciting alternatives in mass storage.

VIC-20 AND COMMODORE 64 are trademarks of commodore electronic ltd.

Name _____

Address _____

City/State/Zip _____

Mail today for information on the ESF-20/64 from EXATRON, 181 Commercial St., Sunnyvale, CA 94086

Two New Cassette Sixpacks For The VIC 20

by John Campbell

Commodore is pleased to announce two new high quality cassette sixpacks for the VIC 20. The first sixpack is an educational mathpack designed to utilize the dynamic powers of the computer to entice a child to actually want to learn. The second sixpack is a conglomeration of high quality programs constructed to both amuse and entertain the entire family. The most obvious strength of both of these packages is the diversity of the tapes within each group.

The programs in the Math Sixpack (VT 108) are not just your "run of the mill" pre-school counting programs. Instead, these programs can really challenge students from first grade and up. The six cassettes are:

1) NUMBOWL—mathematically bowl your way as close to 30 as possible, without going over, by manipulating three randomly generated numbers.

2) LCM MACHINE—use the LCM (Least Common Multiple) Slot Machine to rack up enough dollars to get to go to the "BIG EVENT." This game features three levels of play.

3) SECTOR FIVE—you save the colony by estimating the number of attacking Kuminons. This game rates you on your performance.

4) BACKFIRE—correctly identify all of the divisor(s) of a given number and blast them.

5) RULER DUELER—you command a space ship which fires targets on the ground. In order to successfully neutralize the target, you must press the key associated with the proper ruler mark.

6) SCARE CITY MOTEL—you are the owner of a motel and your job is to maximize revenues by charging the appropriate room rate. A simple illustration of supply and demand, and of course the "best price" changes each time you play.

All of the tapes include strong graphics, sound effects, and rating schemes so that the educational value of the programs can be maximized.

The Sampler Sixpack (VT 109) includes Art, Music, Adventure and Arcade games all in one package! The six cassettes are:

1) VIC SYNTHESIZER—Turn your VIC into a sound synthesizer and musical keyboard. This user-friendly program can help you define your own sounds to imitate real instruments or create music that no ordinary instrument could make.

2) CRAWLER—Juice up your joystick for intense arcade-quality action. This is a high quality machine language program that squeezes enjoyment out of your VIC.

3) ALPHA DRAW—Use a joystick to draw on the screen in any of the available character sets and then save your creation to cassette.

4) SUPER SEEKERS—Travel among the universes gathering points, but watch out for the black holes!

5) BIG BAD WOLF—A classic adventure game that is definitely not just for kids. You are alone in an abandoned farm house and the Big Bad Wolf is on his way. Can you figure out how to stop him?

6) TREASURES OF THE BAT CAVE—An amazing 3D action game. Search through an underground labyrinth to find treasure. Beware of the vampire bats!

As you can see, Commodore has done its best to produce quality cassette software. We feel that this represents only the beginning of an aggressive and diligent effort to bring our customers the finest software available. Enjoy! **C**

jiffies

The Art of Playing Pinball On The VIC 20

by Joe Ferrari
Current *Pinball Spectacular* Champ

As with any video game, in order to be successful at playing it, the participant must acquire and develop some basic skills. Hand-eye coordination for instance, can be attained by many hours of playing (brute force). While these skills are being developed, the player learns many of the game's characteristics and at the same time, experiments with various strategies for attaining higher scores. In many cases, the techniques that are usually employed yield good results quickly (short term). In the case of Pinball Spectacular, *I decided to take a different approach (long term). In the following paragraphs, I will share with you the method that allowed me to attain a score of over 1,500,000!*

CHARACTERISTICS

Before I can proceed to describe my method of playing *Pinball Spectacular,* a description of the game's major characteristics is in order. One of the key features of the game is the letters located on the left and right sides. When the letters on one side are all turned green, a bonus ball is awarded. This feature allows indefinite continuation of play.

The faces horizontally aligned near the bottom are also important. Turning all these sad faces into happy ones will add on to the bonus multiplier (maximum of 3). The final feature that greatly contributes to high scores is the ghosts located at the top center. 15,000 points can be accumulated here with just one shot! All other features should be considered secondary with little consequence on the score.

The bricks on the upper portion of the screen (both sides) have several purposes. Clear one colored layer (any side) and 1,000 points are added to the bonus. When the entire side is cleared, if the life saver (located at the bottom) on that same side has been lost, it is restored. One thing I should mention at this point; upon hitting any of the bricks in the uppermost color layer (pink), the ball will gain speed. This I might add, is the game's only characteristic that is negative from the player's point of view.

To restore the ball to a speed that is more controllable, a gate located just before the haunted house is provided. Direct the ball through the gate and the ball will slow down.

STRATEGY

A close look at the game's characteristics reveals that changing all the letters on one side to green is of greatest priority and is my first aim. Attaining a bonus ball ensures survival and opportunity to continue play. To accomplish this task, effective use of the top and bottom paddle is required.

The strategy I use here is to play with the bottom paddle only until the letter is turned green and only then do I use the top paddle to flip to the next letter. The only time this objective is abandoned is when the ball's speed is too fast. At this time, the primary goal is to direct the ball through the gate and slow it down. When this has been accomplished, I resume my original goal.

Once I have attained the bonus ball, my next objective is to bring the bonus multiplier to 3. The ball is kept in the upper portion of the screen with the top paddle until the fall of the ball is aiming for a face that is sad. Again, if the ball is moving too fast, I abandon the current objective and place all efforts on slowing the ball down.

With the above objectives met, I can now concentrate all my efforts on directing the ball through the gate and into the haunted house. This particular strategy serves many purposes. For one, hitting the ghosts yields bonus points and at the same time, the ball has a good chance of eating away at the bricks from the inside. The bricks are very important at this stage as each layer that is cleared yields 3,000 points.

CONCLUSION

Although the above constitutes the main strategy that enabled me to attain very high scores, *Pinball Spectacular* has many subtle characteristics that ultimately influence the final outcome of the score. Unfortunately, the scope of this article was not to go into these subtleties, but instead to outline the general strategy that worked for me. GOOD LUCK! **C**

jiffies

Using the Programmable Character Set Editor

by Michael Tomczyk

Reprinted from *Commodore Magazine*

The VIC Programmable Character Set & Gamegraphics Editor is a useful aid for both BASIC and machine language programmers, and is friendly enough for first-time computerists to use with little or no experience.

We wanted to tell you about this fascinating product, and thought the best way to do that would be to give you an example of what you can do with it. THE TUMBLER is a screen animation program that you can type directly into your VIC 20 and RUN.

You can use the editor to write the SAME type of program yourself, LIST, SAVE and RUN it in BASIC. Try typing TUMBLER exactly as shown below, and type RUN. You can also SAVE this on tape for future use by using your Commodore datassette.

```
10 REM:TUMBLER
11 K=7168:FORJ=32*8TO32*8+7:POKEK+J,0:
   NEXTJ
12 READCH:IFCH=1THENGOTO20
14 POKEK,CH:K=K+1:GOTO12
20 PRINTCHR$(147):X=64:POKE36869,255
25 PRINT:PRINTSPC(6)CHR$(18)CHR$(30)
   "TUMBLER!"CHR$(146)CHR$(31)
30 POKE36878,15:S=36875:B=128:E=138:
   PRINTCHR$(31)
40 FORM=BTOESTEP5:POKES,M:NEXTM:POKES,0
50 PRINTTAB(80-X)CHR$(X)CHR$(32)CHR$(145)
60 FORM=BTOESTEP5:POKES,M:NEXTM:POKES,0
70 IFX=79THEN:POKE36869,240:PRINTCHR$(147)
   :END
80 FORT=1TO150:NEXTT
90 X=X+1:B=B+5:E=E+5:GOTO40
500 REM: TUMBLER CHARACTERS
505 DATA24,24,127,24,56,40,40,40
510 DATA96,96,60,50,240,72,36,18
515 DATA96,98,60,16,114,78,66,192
520 DATA50,52,152,113,50,44,32,24
525 DATA0,4,104,113,62,48,73,134
530 DATA65,34,20,217,242,60,192,0
535 DATA65,34,148,89,50,252,192,0
540 DATA48,8,228,28,12,63,40,0
545 DATA64,32,24,248,28,42,24
550 DATA0,64,32,24,248,28,42,34
555 DATA1,2,4,254,30,40,72,132
560 DATA38,22,8,24,52,82,144,144
565 DATA12,76,56,14,8,24,36,34
570 DATA48,178,84,56,16,16,40,68
575 DATA177,178,84,56,16,16,40,68
580 DATA48,48,16,56,84,16,16,48,-1
```

The key is the DATA starting on line 505. What the character editor lets us do is this:

1. First we created a series of special characters using the character set editor.
2. We then typed "SAVE" from the editor's command menu and typed "LIST." The list command gave us a list of 8-number DATA statements for each of the special characters we created. Each special character is represented by ONE LINE of DATA statements.
3. The BASIC program we then wrote includes a section which lets us identify our programmable characters as DATA statements, tells the VIC to READ that DATA and create characters in memory and on the screen (POKE 36869,255) and PRINT those special characters in line 50 (the X is the special character). Immediately after we PRINT the character (CHR$(X)) we PRINT a space to erase that character (CHR$(32)) and tell the VIC to move one space to the left and PRINT the next character one space over. In this way we create the illusion of animation on the screen.
4. When you RUN the program, the programmable characters appear and move. If you want to experiment, you can change some of the numbers in the DATA statements and watch how the characters change.

The character editor retails for only $14.95 and is available from most VIC dealers. It comes with a 16-page instruction manual which talks you through all the features, step-by-step.

The character editor lets you perform all sorts of fascinating tricks with your VIC 20 . . . but most importantly it lets you create your own characters! This means you can create foreign language character sets, game graphics, special notation symbols . . . anything your imagination conjures up. **C**

(Thanks to H. George Hain of Saskatchewan, Canada, for his revisions to this program. Ed.)

Cursor Positioning On The VIC 20

by Tom Ziegler

You can use this little machine language subroutine for many things—one of which is to move characters around the screen in a game program.

There are a few ways in BASIC to position the cursor at a specific location on the VIC 20. You can PRINT combinations of the cursor keys (PRINT CURSOR DOWN CURSOR DOWN CURSOR RIGHT CURSOR RIGHT . . .) or you can use a KERNAL routine that moves the cursor around faster and easier than you could in straight BASIC. The KERNAL (see page 182 of the *VIC 20 Programmer's Reference Guide)* is a set of standard machine language routines built into your VIC. You can access the KERNAL routines from BASIC with the SYS command. In the following example, READ and DATA statements are used to POKE a small machine language subroutine into the memory area normally used for the cassette buffer. (The cassette buffer is a handy place for machine language subroutines.) Once the subroutine is in place, you can change the cursor position with two POKEs (one for the row, one for the column) and a SYS. Once the cursor is repositioned, you can use the normal PRINT (or INPUT) commands to display your information.

The following little sample program will move a red ball diagonally from upper left to lower right using the subroutine described above.

```
10 PRINT"■"
20 FORI=828TO836
30 READA:POKEI,A
40 NEXT
50 DATA24,160,0,162,0,32,240,255,96
60 X=1:Y=1
65 FORI=1TO20
70 POKE830,Y::POKE832,X:SYS828
80 PRINT" "
90 Y=Y+1:X=X+1
95 POKE830,Y:POKE832,X:SYS828
100 PRINT"■"
110 NEXT:PRINT"■"
```

A brief explanation of the program follows:

Line 10 clears the screen.

Lines 20-50 put the machine language code into locations 828 through 836 (the cassette buffer).

Line 60 initializes the values of X and Y (i.e., row 1 and column 1).

Line 70 puts the column position (Y) into location 830, the row position (X) into location 832 and uses SYS828 to tell the BASIC program to branch to the machine language subroutine starting at location 828.

Line 80 prints a space to erase the previous position of the character.

Line 95 repositions the cursor.

Line 100 prints the red ball.

By modifying the X and Y values, you can position the cursor anywhere on the screen. For example if you change line 90 to read "90 X = X + 1", the ball will just move down the left hand side of the screen, because you don't change the column value.

For those interested, the assembly language translation of the subroutine POKEd into locations 828 through 836 is:

```
828 24   CLC
829 160 LDY
830 0
831 162 LDX
832 0
833 32   JSR $FFF0 (the FFF0 is taken from lines
         834 and 835)
834 240 (F0)
835 255 (FF)
836 96   RTS
```

Line 833 jumps to the PLOT KERNAL routine (memory location FFF0). This routine will put the cursor at the location specified by the X and Y registers (lines 830 and 832).

With a little imagination, the possibilities for this subroutine are virtually endless. It can be used in games to move characters around. It can be used for formatted data entry screens, or it could be used for error messages. Almost any program could benefit from quick and easy cursor positioning. **C**

jiffies

Controlling Graphics Movement

by Mike Abercrombie
Wichita, KS

The use of the "GET" statement is not a very "friendly" method of controlling graphics movement on the screen. For example:

```
10 A$="": GETA$: IFA$=""
   THEN 10
20 IF A$<>"A" THEN 10
30 PRINT A$;: GOTO 10
```

If you enter this routine and "RUN" it you will observe that no matter how long you hold down the "A" key, the letter "A" will be printed on the screen only one (1) time. Observe also that if you hit the "A" key several times in very rapid succession, the "A" may continue to be displayed after you have stopped hitting the key. This is due to the VIC's keyboard buffer. This can have some undesirable effects if you are controlling the movement of a game graphic because it may continue to move after you've decided to stop. Also, you can get muscle spasms and callouses from repeatedly hitting the movement control keys to dodge the enemy or to position yourself for attack.

An alternate method of controlling the graphics on the screen is through the use of the PEEK value of memory location "197" (HEX C5). This address contains the value of the key currently being held down. This value is in a range from 0 to 64 (the values of all the keys are documented on page 179 of the *Programmer's Reference Guide VM110).* For the purpose of demonstration, the only value we need is the value of the "A" key which is "17." Note however that a value of "64" means that no key is currently being held down. Key in the following routine:

```
10 D%=PEEK (197): IF D%<>
   17 THEN 10
20 PRINT "A";: GOTO 10
```

When run, this routine will display the letter "A" repeatedly as long as you hold the "A" key down. When you release the "A" key it will immediately cease to display and it will resume display when it is depressed again. No more keyboard buffering problems, and no more muscle cramps! A few words of caution are in order. My keyboard may have a slight flaw in some of the relays because every once in a while the value in address "197" will not correspond to the key being held down. This does not cause any serious problems as long as correct logic is used. For example, let's say that a game uses the logic for graphic movement where "A" moves the graphic to the left. "D" moves the graphic to the right, and any other key will cause the graphic to "fire" a rocket or something. You might code the routine as follows:

```
10 D%=PEEK(197)
20 IF D%=64 THEN 10: REM
   NO KEY BEING HELD DOWN
30 IF D%=17 THEN 100: REM
   MOVE GRAPHIC LEFT
40 IF D%=18 THEN 200: REM
   MOVE GRAPHIC RIGHT
50 GOTO 300: REM FIRE
   ROCKET
```

The above logic will work 99% of the time. However every once in a while, pressing the "A" or "D" key will result in something other than a value of 17 or 18 in address "197." This will cause an unwanted branch to the "FIRE" logic. To avoid this problem, specify another key for the "FIRE" logic such as the "SPACE BAR" which has a keyboard value of "32." This will allow you to avoid the default to the "FIRE" logic. Example:

```
10 D%=PEEK(197)
20 IF D%=64 THEN 10: REM
   NO KEY BEING HELD DOWN
30 IF D%=17 THEN 100: REM
   "A" KEY SO MOVE LEFT
40 IF D%=18 THEN 200: REM
   "D" KEY SO MOVE RIGHT
50 IF D%=32 THEN 300: REM
   SPACE BAR SO FIRE
60 GOTO 10: REM INVALID
   KEY PRESSED OR KEY-
   BOARD ERROR
```

If you keep these words of caution in mind, screen movement of graphics should be much more player "friendly." C

Lazy Line Numbering

Somewhere between there and here the letter we assume was attached to this article got lost, and the author didn't put his or her name on the article itself. (See where laziness gets you?) But we thought it was worth printing, regardless. We hope to hear from the author, soon.

Got your Commodore 64 and don't have a package that gives you automatic line numbering yet? (Or don't want to buy the package with all those other zipper features just to get automatic numbering?) If you were born lazy, like I was (not to mention cheap), you'd like to have somebody at least do the line numbers for you.

You're in luck! Let VIC do 'em for you.

I call this little program "PAD" (for "scratchpad") and load it every time I start a new program. First the code, then the explanation:

```
1 GOTO 63993
2 GOTO63988
3 REM**
63993 INPUT"■■■■■BEGINNING
      LINE NUMBER AND
      INTERVAL";S,I
63994 INPUT"■■■NUMBER OF
      LINES (UP T  22)";N
63995 PRINT"■ ;
63996 FORK=1T N:PRINTSTR$
      (S+(K-1)*I)"REM**"
      :NEXTK
63997 END
63998 POKE53280,1:POKE53281
      ,1:PRINTCHR$(152)
63999 END
```

With this program loaded, "RUN" jumps to 63993 and prompts for a line number and the numbering interval. I usually start with "100,100" and "22" for the number of lines. (More than 22 scrolls the first line off the top and loses it.) Line 63996 then generates a screen display of 22 numbered lines:

```
100 REM**
200 REM**
    •
```

ending with "2200 REM**".

Just press "HOME" and as fast as you can hit 22 returns, you've written 22 "REM**" lines into your program. It's then easy to hit "HOME", cursor right, and write over the line with your program code.

To try out the program at any stage, type in "GOTO3" and your program executes from its first true line. (Line 3 is, in effect a NO OP, as is any other line you may have generated but not used—but be sure to have a "STOP" or "END" before 63993!)

There is no danger of typing in a wrong number and inadvertently wiping out valid code; the numbered "REM**" lines do not become program lines until you type a return for each line.

You can, of course, start with any line number and any interval, and generate as many lines (up to 22) as you like. So you can easily insert, say, seven lines between 800 and 900 by using 810 to start, an interval of 10, and calling for 7 lines.

When you get the program running and debugged, of course, simply delete the original lines (1-3 and 63993-63999).

Line 63998, by the way, converts the screen display to something I prefer. It "removes" the border (but no change in the usable screen space, of course), gives a light background, and changes the characters to a less contrasty grey. I could have typed "GOTO63998" but that's 4 strokes more than "GOTO2". (Told you I was born lazy.) **C**

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

Crazy Balloons

Reprinted from the Wichita Commodore Users Group Newsletter

The object of the game is to move the balloon from 'START' to 'GOAL'. The balloon sways back and forth over its basket, and if the balloon hits a wall or obstacle you lose. The following four keys are used to move the basket:

@ Up /Down :Left ;Right

```
280 POKE36878,15:BL=3
290 POKE36879,8:T$="000259"
300 VR=7680:V1=38400
310 Y$="■■■■■■■■■■■■■■■■■■■■■■■■■"
1000 PRINT"■■TIME 00:00■■■F.M CLUB":TIME$="000000":Y=15:X1=15
1010 FORA=22TO43:POKEVR+A,160:POKEV1+A,7:POKEVR+A+462,160:POKEV1+A+462,7:NEXT
1020 FORA=22TO462STEP22:POKEVR+A,160:POKEV1+A,7:POKEVR+A+21,160:POKEV1+A+21,7:NE
XT
1030 D=1:U=1:K=0:K1=0:K2=0:K3=0:K4=0:K5=0:K6=0:K7=0:K8=0:K9=0:K0=0:YI=Y:XI=X1
1100 ONMSGOTO1400,1700
1110 PRINT"■■■*****   ******■ GOAL ■"
1120 PRINT"■***      *        ■ ■"
1130 PRINT"■**               ■ ■"
1140 PRINT"■*                ■ ■"
1150 PRINT"■*                ■ ■"
1160 PRINT"■                 ■ ■"
1170 PRINT"■                 ■ ■"
1175 PRINT"■       *   ***■     ■"
1180 PRINT"■      ***************"
1190 PRINT"■      ***************"
1200 PRINT"■*     ***************"
1210 PRINT"■      ************"
1220 PRINT"■       ****■ START  "
1230 PRINT"■       *        ■ "
1240 PRINT"■**              ■ "
1250 PRINT"■*               ■ "
1260 PRINT"■*               ■ "
1270 PRINT"■**              ■ "
1280 PRINT"■***             ■ "
1290 PRINT"■*******     **■     "
1300 GOTO2000
1400 PRINT"■■■*****     **■ G O A L "
1410 PRINT"■**         **■ ■     ■ "
1420 PRINT"■**          ■ ■      ■ "
1430 PRINT"■*           ■ ■      ■ "
1440 PRINT"■*           ■ ■      ■ "
```

Crazy Balloons Continued

```
1450 PRINT"*        "
1460 PRINT"                    *"
1470 PRINT"                    *"
1480 PRINT"       *            *"
1490 PRINT"      **            *"
1500 PRINT"      ***          **"
1510 PRINT"       **         ***"
1520 PRINT"        ***** START  "
1530 PRINT"         *           "
1540 PRINT"*                    "
1550 PRINT"*                    "
1560 PRINT"*                    "
1570 PRINT"**                   "
1580 PRINT"***                  "
1590 PRINT"*****        *       "
1600 GOTO2000
1700 PRINT"****        ** G O A L "
1710 PRINT"**                   "
1720 PRINT"**                   "
1730 PRINT"*                    "
1740 PRINT"*                    "
1750 PRINT"*                    "
1760 PRINT"              *   **"
1780 PRINT"       *      *"
1790 PRINT"       *      *"
1795 PRINT"       *      *"
1800 PRINT"        *"
1810 PRINT"        *"
1820 PRINT"            "
1830 PRINT"            "
1840 PRINT"            "
1850 PRINT"            "
1860 PRINT"            "
1870 PRINT"         *        *"
1880 PRINT"S T A R T**      **"
1890 X1=4:GOTO2000
```

Crazy Balloons Continued

```
2000 PRINTLEFT$(Y$,1+YI)SPC(XI)"   [illegible]   [illegible]   "
2010 POKEK8,32:X=X1-(3-U):K=Y*22+X+VR:K1=K+1:K2=K+2
2015 K3=K+22:K4=K+24
2020 K5=K+44:K6=K+45:K7=K+46:K8=K+67
2030 KK=PEEK(K)+PEEK(K1)+PEEK(K2)+PEEK(K3)+PEEK(K4)+PEEK(K5)+PEEK(K6)+PEEK(K7)
2040 IFKK<>256THEN3000
2050 PRINTLEFT$(Y$,1+Y)SPC(X)"[illegible] [illegible]   [illegible] [illegible]":YI=Y:XI=X
2060 POKEK8,86:POKEK8+30720,1:POKEK9,32:POKEK0,32
2070 IFU-3>0THENK9=K+88:GOTO2100
2080 IFU-3=0THENK9=K+89:GOTO2100
2090 IFU-3<0THENK9=K+90
2100 K0=K+114-U:IFPEEK(K9)+PEEK(K0)<>64THEN3000
2110 POKEK9,86:POKEK0,102:POKE30720+K9,1:POKE30720+K0,7
2120 U=U+D:A=PEEK(197):IFU=5ORU=1THEND=-D
2130 IFA=30THENPOKE36874,254:Y=Y+1:POKE36874,0
2140 IFA=22THENPOKE36874,254:X1=X1+1:POKE36874,0
2150 IFA=45THENPOKE36874,254:X1=X1-1:POKE36874,0
2160 IFA=53THENPOKE36874,254:Y=Y-1:POKE36874,0
2170 PRINT"[illegible]"SPC(5)MID$(TIME$,3,2)"[illegible]"RIGHT$(TIME$,2)
2180 IFTIME$>T$THEN3000
2190 IFY<5ANDX1>13GOTO2500
2200 GOTO2000
2500 ME=ME+1:IFME=3THENME=0
2510 BL=BL+1:FORA=1TO15:POKE36874,253:FORB=1TO80:NEXT:POKE36874,0:FORB=1TO20:NEX
T:NEXT
2520 GOTO1000
3000 PRINTLEFT$(Y$,Y)SPC(X1)"[illegible]   [illegible] [illegible]  [illegible]  [illegible] [illegible]"
3100 FORA=254TO0STEP-1:POKE36874,A:NEXT
3110 FORP=1TO2000:NEXT:BL=BL-1:IFBL=-1GOTO3500
3120 GOTO1000
3500 POKE198,0:PRINT"[illegible]     REPLAY";:INPUTA$
3510 IFA$="Y"THENRUN
3520 PRINT"BYE BYE !"
```

Fill-In-The-Blanks

```
0 rem vic fill-in-the-blanks, by Pat ubbs, converted by *rick cotton*
1 printchr$(14)"■";
2 Print"■This is a VIC 9ame,   you fill in the
4 Print"missin9  Parts of     sPeech. Good luck!"
6 Print
15 Prin"■Type your name":inPutz$
20 Print"■ TyPe an adJective":inPuta$
30 Print "■ TyPe the name of a 9irl":in ut b$
40 Print "■ TyPe an adJective":inPut c$
50 Print"■ TyPe the name of a man":inPut d$
52 Print"■TyPe an adverb":inPut e$
54 Print"■ TyPe a noun":inPut f$
60 Print"■ TyPe a noun": inPut 9$
70 Print"■ TyPe an adverb":inPut h$
80 Print"■ TyPe an exclamation":inPut i$
90 Print"■ TyPe an ad ective":inPut J$
1 0 Print"■ TyPe an adverb":inPut k$
110 Print"■ TyPe an adJective":inPut l$
120 Print"■ TyPe an ad ective":inPut m$
13 Print"■ TyPe a noun": inPut n$
132 Print"■TyPe a noun":inPuty$
134 Print"■TyPe a verb":inPutx$
135 Print"■■■■■■■"
140 Print"       ■A S ORY       ","■■■■■■■■■■■■■■■by":Print"■■■■■■■■";z$:for i=1to1
500:next
150 P int"■";
```

Fill-In-The-Blanks Continued

```
160 print"■ There was a sudden ";a$;" knock";
170 print" on the door. ";b$;" turned to ";
180 print"her ";c$;" cousin. Quick, ";d$;
190 print " she cried ";e$;". Hide in";
200 print" the ";f$;" closet.";
210 print" It was her boy friend. "
212 print"■■■■ ■■hit any key■■"
214 geta$:ifa$=""then214
216 print"■";
220 print"I forgot my ";g$;", he said ";h$;". ";i$;
230 print" he added. Where did this ";j$;" ";y$;
240 print" come from? What ";y$;"?";
250 print" she asked ";k$;".";
260 print" Don't give me any of that ";l$;
270 print" stuff, he shouted. Next time";
280 print" I catch you ";x$;"ing my ";y$;",";
290 print" I'm going to whale the ";m$;" "; $;
300 print" out of y u!"
310 pri t"■■■■■■■■■■THE END■■"
320 print"■   HIT ■KSpace bar>■          to  read aga n"
330 geta$:ifa$=""then330
340 ifa$=" 'then135
350 run
```

Craps

An updated version of the game published in Commodore's *Introduction to BASIC, Part 1.* by Larry Holler

```
10 REM CRAPS
20 PRINT"■"
27 TAB56  B$="     |    |  |    |"
30 PRINT"THE GAME OF CRAPS"
40 PRINT "IS PLAYED WITH TWO"
50 PRINT"DICE.FIRST YOU BET"
60 PRINT"AND THEN YOU THROW.IF"
70 PRINT"YOU GET A SCORE OF■■ 7 ■■"
80 PRINT"OR■■ 11 ■■,YOU WIN.IF YOU"
90 PRINT"THROW■■ 2■,■ 3 ■■ OR■■ 12■■,YOU"
100 PRINT"LOSE.IF YOU THROW ANY"
110 PRINT"OTHER NUMBER YOU DON'T";
120 PRINT"WIN OR LOSE STRAIGHT"
130 PRINT"AWAY:YOU KEEP ON"
140 PRINT"THROWING UNTIL YOU"
150 PRINT"EITHER"
160 PRINT"THROW THE SAME AS YOU"
170 PRINT"DID FIRST TIME(AND"
180 PRINT"WIN)"
190 PRINT"OR"
200 PRINT"THROW A■■ 7■■(AND LOSE)"
220 PRINT: PRINT"■■ HIT ANY KEY TO GO ON ■■"
240 GET A$
250 IF A$=""THEN 240
261 A$="■     ┌──┐   ┌──┐"
270 B$="     |  |  |  |"
280 C$="     └──┘   └──┘■"
290 PRINT"■"
300 INPUT "■■STARTING■ CAPITAL■■";C
302 PRINT
310 PRINT"■■HIT A KEY FOR NEXT BET■■"
330 GET R$
340 IF R$=""THEN 330
350 PRINT"■■■ YOUR CAPITAL NOW IS  ■■":PRINTC
370 PRINT"■HOW MUCH DO YOU ■■BET■ ":INPUT"";W
390 IF W<=C THEN 420
400 PRINT"■YOU CAN'T AFFORD IT■"
410 GOTO 310
420 PRINT"■■■■FIRST THROW (BET=";W;")■"
425 FORZ=1TO1900:NEXT
430 PRINT"■■■■";A$
440 FOR J=1 TO 5
450 PRINT B$
460 NEXT J
```

Craps Continued

```
470 PRINT C$
480 Q=INT(10+50*RND(0))
490 FOR Z=1 TO Q
500 A=INT (1+6*RND(0))
510 B=INT(1+6*RND(0))
520 POKE 36878,15
530 POKE 36876,254-A*B
540 PRINT"
541 PRINT:PRINT
542 PRINT"     |  "A;"   |  ";B
550 FOR M=1 TO 50
555 NEXT M
560 NEXT Z
570 POKE 36876,0
580 POKE 36878,0
590 T=A+B
600 IF T=7 THEN 1000
610 IF T= 11 THEN 1000
620 IF T=2 THEN 1100
630 IF T=3 THEN 1100
640 IF T= 12 THEN 1100
650 PRINT:PRINT:PRINT
660 PRINT
670 PRINT
680 PRINT"YOU HAVE TO MAKE"
690 PRINT T;"BEFORE  7"
700 PRINT:PRINT:PRINT:PRINT: PRINT" HIT ANY KEY TO GO ON "
710 GET R$
720 IF R$=""THEN 710
730 PRINT"NEXT  THROW(BET=$";W;")"
740 PRINT"MAKING";T
750 FORZ=1TO1900:NEXT
755 PRINT"
760 PRINTA$
770 FOR J=1TO5
780 PRINTB$
790 NEXT J
800 PRINTC$
810 Q=INT(10+10*RND(0))
820 FOR Z=1 TO Q
830 A=INT(1+6*RND(0))
840 B=INT(1+6*RND(0))
850 POKE 36878,15
860 POKE 36876,254-A*B
```

Craps Continued

```
870 PRINT"┌┬┬┬┬┬┬┬┬┐
871 PRINT:PRINT:PRINT
872 PRINT"│    │■"A;"│   │■";B
880 FOR M=1TO50
890 NEXT M
900 NEXT Z
910 POKE 36876,0
920 POKE 36878,0
930 IF A+B=T THEN 1000
940 IF A+B=7THEN1100
950 GOTO700
1000 PRINT:PRINT:PRINT:PRINT"■        YOU WIN■"
1010 C=C+W
1020 POKE 36878,15
1030 FOR J=1 TO 20
1040 POKE 36876,240
1050 FOR M=1 TO 25
1060 NEXT M
1070 POKE 36876,0
1080 FOR M=1TO25
1085 NEXT M
1090 NEXT J
1095 GOTO 310
1100 REMPLAYER LOSES
1110 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT    "■       YOU LOSE          ■"
1120 POKE 36878,15
1130 FOR J=220 TO 127 STEP-1
1140 POKE36874,J
1150 POKE 36875,J
1160 FOR M=1 TO 5
1170 NEXT M
1180 NEXT J
1190 POKE 36878,0
1200 C=C-W
1210 IFC>0THEN 310
1220 PRINT"■YOU ARE NOW BROKE■"
1221 FORZ=1TO1900:NEXT
1230 RUN
```

Phone Home

```
100 PRINT"□"
110 PRINTSPC(71)"PHONE HOME"
120 PRINTSPC(88)"BY DENNIS MCCORMACK
130 PRINTSPC(88)" OUCH ANY KEY.."
140 GETG$:IFG$=""THEN140
150 PRINTSPC(44)"CAN YOU GUESS"
160 PRINT"ET'S PHONE NUMBER?"
170 PRINTSPC(24)"???-???-????"
180 A=INT(RND(1)*900)+100:A$=STR$(A):A$=MID$(A$,2,3)
190 B=INT(RND(1)*900)+100:B$=STR$(-B)
200 C=INT(RND(1)*9000)+1000:C$=STR$(-C)
210 E$=A$+B$+C$
220 INPUTG$
230 GOSUB410
240 IFG$=E$THENGOSUB330:PRINT"PLAY AGAIN?...":GOTO140
250 FORN=1TO12
260 IFMID$(G$,N,1)=MID$(E$,N,1)THENPRINTSPC(N+1)MID$(G$,N,1):GOTO280
270 PRINTSPC(N+1)"?"
280 PRINT"□□":NEXT
290 PRINT"■SORRY, WRONG NUMBER!"
300 FORX=1TO999:NEXT
310 PRINT"□"
320 GOTO220
330 FORJ=1TO155:PRINTG$" THAT'S IT"
340 POKE36879,J:NEXT
345 POKE36879,27
350 PRINT
360 PRINT"HELLO!"
370 PRINT"THIS IS ET'S HOUSE!"
380 PRINT
390 PRINT"HE IS NOT HERE NOW. HE WENT OUT TO CHANGE HIS PHONE NUMBER AGAIN!"
400 RETURN
410 POKE36878,15
420 FOR =1TO2:FORK=1TO50
430 POKE36876,220
440 FORL=1TO5:NEXT
450 POKE36875,0:NEXT
460 FORK=1TO2000:NEXT:NEXT
470 POKE36878,0
480 RETURN
```

Thanks to the VIC-NIC News, Volume 2, Number 1, for this program.

Closing The Communications Gap

by Jeff Hand

Use these programs to save text from any telecommunication service and then print it out on your printer. Before you get too excited, though, this software will NOT let you directly download programs. It will let you save a program listing as a string (as if it were text), and print it out on your printer. But to get it to actually run on your computer, you still have to type it in. (The capability to download directly, however, will be available soon.) Nevertheless, this software will be of great use to all telecommunications enthusiasts who ever wished they had a hard copy of information they've read on a network—like the technical tips or answers to hotline questions on the Commodore Information Network, for instance.

Here is a program that lets you save information from a telecommunications service such as CompuServe, Dow Jones News/Retrieval or THE SOURCE to your printer or datassette. It was originally designed for use with a VIC 20, VICMODEM and 8K expander, but includes some simple modifications for use with a Commodore 64. It not only lets you save information but also allows you to transmit and receive in upper and lower case. If you read the description of each line you'll see the program can be easily modified if you have a different equipment configuration.

The program serves a dual purpose in this article. First, it is a utility that will expand your telecommunications capabilities. Second, it is a teaching tool to help you learn about programming and telecommunications. I'll explain each line of code in detail so you can understand the program thoroughly and eventually use that knowledge to write your own programs or modify this one to your specific needs.

Many beginning programmers expect to sit down at the computer and program a complete, self-sufficient piece of software within a few hours. It doesn't work that way. The process is more akin to sculpting—a process of shaping, adding more clay and reshaping, until after many changes the sculpture resembles your idea.

This particular version of our program, for instance, was written by Joe Brown. But the original arrays and translation routine were written by Jim Butterfield, then modified by me for the VICMODEM manual. The blinking cursor routine was written by Neil Harris. Then Neil and I again modified the program for up-down logic and added a few lines of code to let you know when the buffer was full. As you can see, this kind of public domain software involves a lot of cut-and-paste to get it sculpted to your needs.

Program Operation

The main program on page 88 is for the VIC 20. Modifications for the Commodore 64 appear on page 92.

On the first screen of the program you are asked, "do you want to store the received data?" The next screen will notify you that the program is setting up the arrays. The list of function settings will appear on your screen next, as follows:

Function Key Settings

F1-CTRL C	F2-Transmit User ID
F3-CTRL S	F4-Transmit Password
F5-CTRL Q	F6-Transmit Logoff
F7-Escape	F8-Return to Function Menu

On the next screen you are notified that the terminal is open. When the buffer is full you will be notified by a "buffer full" prompt. A CTRL S is automatically sent to the other computer to stop transmission and the option menu appears on your screen as follows:

Your Options:

1. Terminal Emulator
2. Print From Memory
3. Record From Memory
4. Clear Buffer
5. End

Print from memory will take the data in the buffer and send it to the printer. Selection three will store the information on the datassette. After you've saved your information it is necessary to clear the buffer with selection four if you are going to save more information. When you're ready to go back to the telecommunications service select number one. When your screen prints "terminal open" press F5; this will start the information flow exactly where you left off without missing one bit of data.

As a precaution, it is a good idea to slow down the transmission rate of the telecommunications service to assure that this BASIC program is not overwhelmed by a lot of data. This can be accomplished by changing the service's terminal defaults to have a 20 character delay following the transmission of a <RETURN>. On CompuServe this can be done in command mode with the following:

TER CRD [20]
(Terminal carriage return delay of 20)

```
10 poke52,32:poke56,32:open5,2,3,chr$(6):zt=0
20 rt$=chr$(13):id$="idnumber":pa$="password":of$="off":mt=16380:mb=8193
30 poke36879,110:printchr$(142)"■■■■   ■                    ■"rt$"   ■ vic term pri
 ■
40 print"   ■                  ■"rt$"■■■■■          by"rt$"■■   joseph l. brown
60 print"■■■do you want to store"rt$"received data?"rt$"■■■    ■yes or ■no"
70 getms$:ifms$=""then70
80 ifms$="n"then110
90 ifms$<>"y"then70
100 da=mb:open4,4,7:print#4,rt$rt$:zt=1
110 print"■■■setting up"
120 dimf%(255),t%(255):forj=32to64:t%(j)=j:next
130 t%(13)=13:t%(20)=28:rv=18:ct=0
140 forj=64to90:t%(j)=j+32:next
150 forj=91to95:t%(j)=j:next
160 forj=193to218:t%(j)=j-128:next
170 t%(133)=3:t%(134)=19:t%(135)=17:t%(136)=27
180 forj=0to255:k=t%(j)
190 ifk<>0thenf%(k)=j:f%(k+128)=j
195 next
200 print"■■functions:"rt$"■f1 = ctrl c"rt$"■f2 = your user id"rt$"■f3 = ctrl :
210 print"■f4 = your password"rt$"■f5 = ctrl 9"rt$"■f6 = log off"rt$"■f7 = (esc
e)"
220 print"■■press any key■■":geta$:ifa$=""then220
230 print"■    terminal open"rt$"■    vic to ascii"chr$(14)
240 get#5,a$:ifa$=""orst<>0then290
250 a=asc(a$+chr$(0)):ifms$="y"thenpokeda,a:da=da+1
255 ifda=mtthen365
260 print" "chr$(157)chr$(f%(a));
270 iff%(a)=34thenpoke212,0
280 goto240
290 printchr$(rv)" "chr$(157)chr$(146);:geta$:a=asc(a$+chr$(0))
295 ifa=137thenprint#5,id$:goto240
300 ifa=138thenprint#5,pa$:goto240
310 ifa=139thenprint#5,of$:goto240
320 ifa=140then440
330 ifa<>0thenprint#5,chr$(t%(a));
340 ct=ct+1
350 ifct=6thenct=0:rv=164-rv
360 goto240
365 m$="n"
366 print#5,chr$(19);
367 print"***Buffer Full***"
369 goto 470
380 print"■■■■■   printing memory":forpt=mbtoda
385 if zt=1then390:open4,4,7:print#4:rt$:rt$
390 print#4,chr$(f%(peek(pt)));:next:print#4,rt$rt$:goto440
400 print"■■■what is the file name":inputa$:open1,1,1,a$
```

```
410 forpt=mbtoda
420 Print#1,chr$(f%(Peek(pt)));:next:Print#1,chr$(92):print#1,chr$(92)
430 Print"closing tape file":close1
440 Printchr$(142)chr$(147);
470 Print"options are:"
480 Print"1. terminal emulator
490 Print"2. Print from memory"rt$"3. record from memory"rt$"4. clear buffe
491 Print"5. end"
500 Print"your choice: "
510 geta$:ifa$<"1"ora$)"4"then510
520 onval(a$)goto200,380,400,540
530 close2:Print"  terminal closed":end
540 m$="y":da=mb:Print"buffer clear":close4:open4,4,7:Print#4,rt$rt$:goto470
```

Program Explanation

10 Memory location 52 is a pointer to the top of the string storage area. The value 32 represents the top of memory in the 8K expander. 32 x 256 = 8192. The memory location 56 is a pointer for the top of memory with the 8K expander. The 32 is POKEd into this location for the same reason as above. If you change your memory configuration you'll also have to change the values POKEd into both register 52 and 56.
The open command is opening logical file 5 (can be any value from 1 to 225) to device 2 which is the RS232 port; 3 is a syntax placeholder for the command number. (Refer to the *VIC 20 Programmer's Reference Guide*, page 37 for more on the syntax of the open statement.) CHR$ (6) is the string that is loaded into the control register of the RS232 port. This sets up one stop bit, eight bit word, and a 300 baud rate. Since there is no other string here, zero is loaded into the command register of the RS232 port, which sets up no parity, full duplex and 0-3 line handshake. If you needed a different setting for the system you are tying into, say half duplex, this is what you would do: open 5, 2, 3, CHR$ (6) CHR$ (16). This will change the duplex to half. For more information refer to the *Programmer's Reference Guide*, pages 251 to 260.
The zt = 0 setting is a marker that is set when the printer is opened. This is done so that the printer isn't opened twice, which would give an error.

20 RT$ = CHR$ (13) sets the string RT$ equal to a carriage return. ID$ represents your user ID on the telecommunications system. PA$ is your password for the system and OF$ represents the logoff command of the system. ID$, PA$, and OF$ have to be typed in by you.
MT represents the top of memory with an 8K expander and MB is the bottom of memory. This is where all your data will be stored before being sent to the printer or datassette.

30 Register 36879 is for the screen and border colors. POKing 110 into this register will give a blue border and blue screen. CHR$ (142) switches to upper case mode.
For a listing of how the reverse characters are generated within the quotes refer to the listing on page 7 of this magazine.

70 Get whatever key is pressed on the keyboard and assign that value to MS$ string. If MS$ equals null then continue to poll the keyboard. If there is any character depressed then the computer will go to the next line of code.

80 If MS$ is "N" (no) then don't open up the printer, but go directly to the set up routine.

90 If MS$ is any character other than a "Y" then continue to poll the keyboard. If MS$ equals a "Y" then the code program falls through to the next line of code.

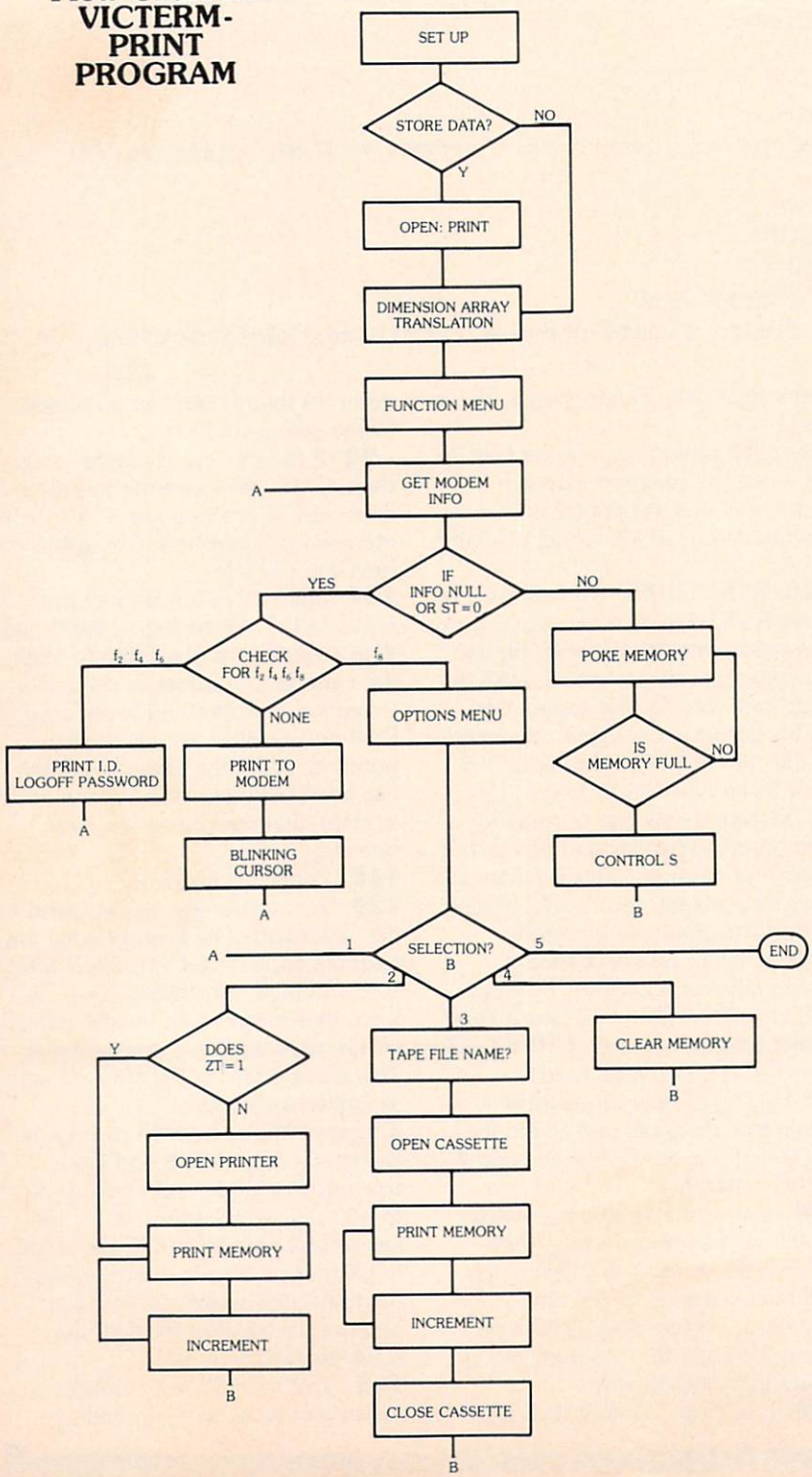
100 Set the bottom of memory equal to DA. Open logical file 4 and device number 4 (the printer). With the command number 7, print all information in upper and lower case. Print two carriage returns to the printer to show the user the printer has been opened. ZT = 1 is a marker to show that the printer has been opened.

110 Prints "Setting Up".

120 Two arrays are dimensioned in this statement. The F array is for the information received and the T array is information transmitted. The "%" is for dimensioning an integer array which takes up less memory space. This is important in the VIC 20 with its limited memory.
This starts the translation process between standard ASCII and Commodore ASCII. J is set to equal 32 to 64. T% (J) = J says that for the values of Commodore ASCII of 32 to 64 there is no need to translate the transmitting array to ASCII because the values represent the same characters in both.

130 T% (13) = 13 is a carriage return and is the same in both

Flow Chart for VICTERM-PRINT PROGRAM



systems. T%(20) = 28 means that when the INST/DEL key (20) is pressed a file separator (28) is sent. RV = 18 is the reverse key. It has no translation into ASCII. CT = 0 is the counter for a blinking cursor, which will be handled later in the program.

140 For Commodore ASCII between 64 to 90 there is a discrepancy of 32 with standard ASCII. Therefore all transmitted characters are adjusted by the addition of 32.

150 Commodore ASCII and standard ASCII are the same for all character values between 91 and 95.

160 Commodore ASCII uses numbers up to 255 to define characters whereas ASCII defines characters up to 127. In Commodore ASCII the numbers from 193 to 218 are for graphics characters that standard ASCII cannot interpret. Therefore they are translated into alphabet characters that ASCII can intrepret. For a table of ASCII and Commodore ASCII values refer to pages 272 to 274 in the *VIC 20 Programmer's Reference Guide*.

170 133 represents F1 (function key). When that key is depressed a three is sent out, which is a CTRL C in ASCII.

F3 (134) = CTRL S
F5 (135) = CTRL Q
F7 (136) = ESCAPE

180 K is assigned to the transmission array for all values of J.

190 This defines F% and T% to be identical.

210 Explanation of function keys for this program.

230 Terminal open notice to the user.

240 Get the information coming in from logical file 5, the RS232 (modem) channel, and assign it to A$.

If A$ is null or creates a status (ST) of any value but zero then go to the test for functions keys.

250 A is assigned the ASCII equivalent of A$ and CHR$(0), which is a null.

Test MS$ to see if the user wanted to save data. If yes put "A" into the present location for the bottom of memory. Increment DA to the next higher memory location.

255 This is a test to see if the program has run out of memory. If the bottom of memory (DA) now equals the top of memory then tell the user the buffer is full.

260 Print a blank to the screen then cursor back (CHR$157) to that location and put the Commodore ASCII translation for the character A in the receiving array (F%).

270 If A equals a quotation mark (34) then POKE location 212 with zero for a direct cursor.

280 Continue to get information from the modem.

290 Get A$. Translate the string into an ASCII variable.

295 If the character is an F2 (137), F4 (138), or F6 (139) then send the ID, password, or logoff through the modem to the mainframe you are connected to. Then go back to poll the information coming in from the modem.

320 If F8 (140) is pressed then go to the options menu.

330 IF A is any value but zero then send it to the modem.

340 Increment the counter (CT) for the blinking cursor control.

350 If the counter is equal to six then reset CT to 0. Reverse the cursor. This will give the cursor a blinking effect.

360 Get more data from the modem.

365 Set MS$ to "N". This will stop any more data from coming into the full buffer.

366 Send a CTRL S (19) to the telecommunications service, so no more information is sent while you print or store your data. To start the information flow from the mainframe press CTRL Q.

367 Tell the user the memory buffer is full. Next go to the options menu.

380 Notice to user the information is to be sent to the printer. The range of memory used (PT) is set from the bottom of memory (MB) to the top of memory (DA).

385 If the printer has already been opened (ZT = 1) then proceed to printing the information to the printer. If the printer hasn't been opened then open the printer channel and send two carriage returns to the printer to let the user know the channel is opened.

390 Take (PEEK) the data in a given memory register (PT) in the received array (F%) and convert it to Commodore ASCII. Then send it to the printer. When all the information is sent to the printer go to the options menu.

400 Asks the user what file name he or she wishes to assign to the tape file. A$ is assigned to the new file name. Next a channel is opened to the datassette.

410 Set the range for PT from the bottom of memory to the top of memory.

420 This code will print the data to cassette tape. When the file is completed two English pound signs (CHR$92) are placed at the end of the file. This is a helpful tape-to-printer program that will allow you to take data from the datassette and send it to the printer.

430 Close tape file.

440-500 Options menu.

510 Test to see which selection was made from the options menu. Get the character from the keyboard. If the character is not 1, 2, 3, or 4 then go back and get another character.

520 If A$ is 1 go to the terminal emulator portion of the program and so on for each of the selections.

530 Close the modem channel and print "terminal closed."

540 Clear the buffer. Set MS$ equal to "Y," so that more data can

be stored on tape or printer. Reset DA to the bottom of memory. Close and reopen the printer channel to avoid any problems. Go back to the options menu.

To make this program operate on the Commodore 64 a few POKES are necessary:

```
384 POKE53265,11
441 POKE53280,246:
    POKE53281,252
```

Tape-to-Print

Since the data in this program is saved in ASCII, you need to include a program to print from the tape. The prompts are self explanatory. And see if you can figure out the code yourself. Portions of the code are identical to the code in the main program. **C**

```
10 rt$=chr$(13):poke36879,110:printchr$(14)"
20 print"  ASCII TAPE TO PRINT "rt$"
"
30 print"  Joseph L. Brown"rt$" All Rights Reserved"
40 forx=1to4e3:next
50 poke36879,94:print"What is file name":inputb$
60 te=0:open1,1,0,b$
70 open4,4,7:print#4:print#4
120 print"Setting Up"
130 dimf%(255),t%(255)
140 forj=32to64:t%(j)=j:next
150 t%(13)=13:t%(20)=8
160 forj=65to90:k=j+32:t%(j)=k:next
170 forj=91to95:t%(j)=j:next
180 forj=193to218:k=j-128:t%(j)=k:next
190 forj=0to255:k=t%(j)
195 ifk<>0thenf%(k)=j:f%(k+128)=j
200 next:print"Printing file"
210 get#1,a$
215 ifa$=""then210
220 ifa$=chr$(92)then280
230 print#4,chr$(f%(asc(a$)));
240 goto210
280 te=te+1:ifte<>2thengoto210
290 print"File Print Completed":close1:print#4:print#4:close1:end
```

DEVELOP-20

FIVE POWERFUL SOFTWARE DEVELOPMENT TOOLS

Plus The Exciting New Book

INSIDE THE VIC 20™

By: Don French

THE BOOK

- Written for both beginners and professionals.
- Clear, complete explanation of the internal workings of the VIC.
- Machine language explained so you can understand it.
- Hexadecimal and binary made clear.
- How to do fast-action graphics, program for joysticks, game paddles and sound effects.
- Complete list of the internal VIC operating programs and how to use them in your own programs.
- Auto-start cartridges and how to make your own.
- Step-by-step guide to the use of the development tools.
- Sample programs fully explained.

THE TOOLS

DECODER - Turns machine language programs (like game cartridges, utility cartridges or even the VIC's own operating programs) into an English-like language (Assembly) you can understand. Produces listings to screen, printer and cassette. Programs produced can be improved, customized or studied to see how they were written.

EDITOR - Used to create or modify assembly language programs, accepts the output from the decoder as input. Enables you to make, save and update Assembly language programs.

ASSEMBLER - Converts Assembly language back into machine language. Lets you use labels and complex address expressions in your programs. Saves the machine language output on tape. Described by Jim Butterfield of COMPUTE magazine as "a remarkable feat". Given a four-star review by Gregory Yob of Creative Computing. Called "elegant" by Jim Strasma of Midnite Software Gazette/The Paper.

LOADER - Loads the programs created and saved with the other tools. Also lets you save machine language programs onto tape so they may be loadable with usual "LOAD" command.

MONITOR - Lets you single-step through your program one instruction at a time, displaying all the registers and status bits. Memory display and modify made easy. Bypass any instruction with ease.

ALL FOR $49.95 PLUS $2.00 POSTAGE AND HANDLING

Standard version runs on any system with Datasette (5K and up)

Add $5.00 for disk version, $5.00 for extended features (minimum 8K)

Send check, M.O., VISA/MC ($2.00 S.C.) or specify C.O.D. (add $3.00) to:

French Silk smooth ware

P.O. Box 207, Cannon Falls, MN 55009

507-263-4821

VIC 20 is a trademark of Commodore Electronics Ltd.

COMPUTE!'s First Book of VIC

Published by COMPUTE! Books, A division of Small System Services, Inc., Greensboro, North Carolina.

reviewed by John O'Brien

What is this book? Hmm, it's not a user's guide. No, it couldn't be, it doesn't go through and teach a little about every aspect of getting started with your computer. It's not thorough enough to be a programmer's reference guide either. Let me see, the subtitle of the book is "Games, programs, and other helpful information for owners and users of the Commodore VIC 20 personal computer." Oh, I get it, that's exactly what this book is.

As the title says this is *COMPUTE!* magazine's first VIC book and since the title implies it won't be the last, let's hope they are all this good. The book is comprised almost exclusively of articles from *COMPUTE!* magazine printed between 1981 and 1982. Long time *COMPUTE!* readers should keep this in mind before they buy the book—you may already have most of the information. The only other VIC owners I wouldn't recommend this book to are people who have absolutely no interest in programming and are unwilling to type in long programs.

COMPUTE!'s First Book of VIC features some familiar names that we associate with Commodore and the VIC. The "VIC Magician" himself, Mike Tomczyk, opens up the book with a history of the VIC. There is also a short history of computers in general so you can find out that your home computer, 35 years ago, would have been the size of your home. Another familiar name that pops up is Jim Butterfield. Any book with five articles by Jim Butterfield can't be all bad.

Most of the articles teach you some trick or subroutine and then demonstrate it in a program. Although it is not a teach yourself BASIC book, it does ease into the more difficult chapters. Chapters one and two are fairly light and often use games as examples. You'll learn more about using paddles and joysticks in your programs, with games like "Breakout" and "Pong" to help teach.

Chapter three, "Programming Techniques", is over 50 pages long—making it the longest chapter in the book. Here the concepts start to get more complicated. This chapter is full of handy little tips and programs that will help make you a better programmer. For example, "How to get the most out of 5,000 bytes" will help streamline your programs. Chapter four is similar but gives helpful hints on using color and graphics in programs.

The last two chapters, five and six, get into the real "heavy" stuff. Chapter five consists mostly of memory maps and chip diagrams. It also gives you some POKE's that do interesting things like disable the runstop and restore key, or totally disable the keyboard. People interested in machine language will like chapter six, although it has only two articles. One is TinyMon by Jim Butterfield, a small machine language monitor for VIC which unfortunately, must be typed into a PET first. The second article tells how you can type that straight into the VIC.

The book will help build your software library with some good game programs such as Meteor Maze, Zap and Starfight3. Slightly more serious programs are also included. One shows how to double the size of the characters on the screen, and another puts items in alphabetical or numerical order. There is also a nice little amortization program that can help you with your taxes.

The chapter groupings may sound similar to some in the *VIC Friendly Computer Guide* or the *Programmer's Reference Guide*. Don't let this fool you. In most cases the book covers different aspects of the same subject or takes off where the other leaves off. All in all this is a book that should be in most VIC owners' libraries. **C**

Best Books

VIC 20™ User Guide by John Heilborn with Ran Talbott. Osborne/McGraw-Hill. To be released soon. Look for a review in the next issue of Power/Play.

An Introduction to Microcomputers: Volume 0—The Beginner's Book and **Volume 1—Basic Concepts** by Adam Osborne. Osborne/McGraw-Hill.

Practical BASIC Programs edited by Lon Poole. Osborne/McGraw-Hill. Forty fully documented programs for small business and household applications. Require no programming experience. **C**

True Names

by Vernor Vinge
Reviewed by Neil Harris

Although this isn't a standard computer-user-magazine-type book review, we thought you might like to know about a piece of fiction that has strong appeal to computer users, especially those who have played Adventure games and/or have experience with telecommunications. True Names *explores the outer limits of personal computing, postulating a telecommunications network accessed directly through your subconscious when you attach special electrodes to your head. Through the network, users reach the "other plane", where a kind of ultimate Adventure game is taking place.*

Every once in a while, a science fiction story appears with an idea that strikes close to the heart of a particular subject. It just *feels* right, like Arthur C. Clarke's weather satellites. Such a story is Vernor Vinge's short novel (novella?), *True Names*, a nominee for 1981's Hugo Award (science fiction's Oscar).

True Names describes the way the future will be, at least the future of telecomputing. Before getting into the actual plot of the book, I'll fill you in a little on what telecomputing is now, and where the state of the art seems to be going. But let me assure you—all my computer buddies here at Commodore Computers are sure that the possibilities portrayed in *True Names* are the real thing!

When I first learned to program (ten years ago, in the misty past of computers), the only way to talk to the computer was with a telephone. We used a clunky monstrosity called a teletype along with a telephone connector called a modem, and dialed the special telephone number of a computer across town. After typing in our high school's account number and secret password, we could use the computer and the programs stored on it. The teletype itself had no "brains", it only served to allow a human to talk to a computer and vice-versa. All the brains were inside the computer, far away.

Now people can buy personal computers for much less than the price of just a teletype. The VIC 20 is selling for under $200, and does more than the big computer could do ten years ago, at least for one user. There are modems for personal computers, too, so they can be used to talk to other computers. Information utilities have been set up to provide news, stock and financial quotations, restaurant guides, airline reservations, and similar services. This whole area of the field is growing phenomenally.

Information in the olden days was sent at a rate of about ten characters per second, or about 110 baud (bits per second). Modern modems go at 30 characters per second (300 baud), with some capable of using 1200 baud and even 4800 baud.

In order to allow people throughout the country to share use of the same computer systems, special data networks have been established, like Tymnet, Telenet, and ArpaNet. They allow the computer user to make a local telephone call from virtually anywhere in North America, tie into the network, and specify which computer in the network they want to speak with.

What has all this got to do with Vinge's story? Imagine a future world where *everything* is computerized. Everyone uses some kind of terminal from their homes to perform their job

functions. Today, a typical home computer system has between 4 and 64 kilobytes of storage; in Vinge's future world, they have "a few thousand megabytes." Many networks exist in Vinge's story, transferring data over 50,000 baud lines.

In the future Vinge creates, computers no longer utilize TV screens. Instead, the new data links provide information via all five senses. Therefore, the user actually experiences the information, instead of just reading it.

Books have been superceded by computer programs, also known as "novels" even though they are electronic. These electronic novels provide sensory cues at the rate of 50,000 baud. Even though it's hard to imagine enough cues even at that high a rate, Vinge makes a good case for providing enough subtle cues to trigger more complex responses. Doing a little arithmetic, let's speculate that a human could deal with a maximum of fifty sensory cues per second. This leaves 1000 bits per cue per second, or a variety of 2 raised to the 1000th power different cues. This is an incredibly large number (my computer can't even calculate it—must be hundreds of digits long), more by far than the number of words in the English language.

Not just novels, but the entire world of computers takes on a whole new meaning in *True Names*. In the early part of the book, the hero tunes into several networks, and patches his consciousness into a communications satellite, gazing down at the earth from above. In order to get into a specially guarded section of the main computers, the hero must get through a maze that most resembles a modern "adventure" game, where you must win out in a series of fantasy-oriented situations to achieve a goal.

Using computers, a new world—indeed a new plane of existence—becomes possible for people of Vinge's future world. Your appearance in the new plane depends not on accidents of genetics, but on whatever you program in. A character using the "nom-de-puter" of Robin Hood for instance, shows up on the "other plane" resembling Errol Flynn. A group of other-plane residents look like computers, including a creepy character known as "The Mailman," who appears on this plane as an old teletype. This shifting of identities and appearances reminds me of nothing more strongly than a science fiction convention, where people's own identities can merge with those of characters they portray.

The plot is a diverting mystery involving a group of other-plane characters, who use their expertise to commit minor acts of vandalism against the large systems. The government has found that the aforementioned Mailman has been engaging in major sabotage against the programs that run the government, to unknown purpose. The government enlists the unwilling aid of another other-plane persona, who eventually discovers the Mailman's true identity and purpose, after a few red-herrings and a major battle that threatens the computing power of the entire world.

In all, this is a story to be reckoned with, especially in years to come. It finished a strong second in the Hugo competition, with outstanding support from computerists. The story is published in the paperback Binary Star #5 collection from Dell paperbacks, and is an absolute must for any modern computer freak. **C**

glitch fix

Rhino!
Fall, 1982 Power / Play

Some of the reverse graphic characters used in the program listing were unusual, which evidently made it hard for beginners to get the program working. For those of you who've been struggling with it, here they are:

Reverse S: The home key
Reverse E: Control 2
Reverse R: Control 9
Reverse Q: Cursor right
Reverse ■: Function key 1
Reverse ■: Function key 3

To help our readers keep their sanity, we will publish the complete list of all the "hidden" characters for both the VIC 20 and Commodore 64 in every issue from now on. See page 7.

VIC Baseball
Winter, 1982 Power / Play

Somehow line 200 got buried at the end of line 169, which caused the "GOTO 200" statement in line 415 to get quite confused. As an additional aid to entering the program, our resident experts suggest you abbreviate the POKE statements in line 420. Use P ©SHIFT□ O to abbreviate. For the scoop on those insidious reverse graphic characters, please see our listing on page 7.

Omega Race:
The Finer Points
Winter, 1982 Power / Play

In putting together these instructions, one of us evidently didn't know right from left. Whether it was the editor or the author was a hotly disputed question for a while. Since the main point is for our readers to get this straightened out (regardless of who was the cock-eyed one), the author submits these corrections for your edification.

Page 38: First column, near bottom. "Therefore, when pointing left (<)" should read "Therefore, when pointing right (>)"

Page 38: Second column, section 3d. "(make sure you turn the paddle clockwise . . .)" should read "(make sure you turn the paddle counterclockwise . . .)"

Page 39: Second paragraph. ". . . the DROIDS will fire at you while you are right of point A or left of point C" should read ". . . the DROIDS will fire at you while you are left of point A or right of point C." **C**

OUR MOST POPULAR "ARCADE GAME" TAPE PROGRAM FOR VIC 20™ COMPUTERS

SUPER PARATROOPER!!

FANTASTIC SOUND AND ACTION WITH SUPER HI-RES GRAPHICS
(A PROTECTO ENTERPRIZE EXCLUSIVE)

By Nic Dudzik

JOYSTICK OR KEYBOARD

SUPER PARATROOPER is a High Resolution game that doesn't let you make any mistakes. You are in charge of a big gun that sweeps back and forth by your command. Helicopters fill the sky, (and we mean fill the sky!), dropping paratroopers. Your mission is to keep 3 paratroopers from hitting the ground on either side of your gun. But that's just the beginning. You score by hitting the helicoptors 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 helicoptor blades whirring and you can hear the howitzer pumping shells. When you hit a parachute you hear this ripping sound and the paratrooper falls struggling to the ground! NOW HEAR THIS! — If you let three paratroopers land, they bring in a tank from either side and blast you!!! This game really shows off the sound and graphic capabilities of your VIC. SUPER PARATROOPER IS OUR NO. 1 SELLING ARCADE GAME — you've got to get this game to believe it — we are so sure you'll like it we'll give you "10 DAY FREE TRIAL."

LIST $24.95 – SALE $19.95

VIC 20 is a trademark of Commodore Electronics Ltd.

RUNS ON STANDARD VIC 20™

we are commodore VIC experts!!

PROTECTO
ENTERPRIZES (FACTORY DIRECT)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

new products

The following information is taken from new product announcements sent to us by various 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. If you have any problems with any of the products listed here, please write to us.

Company:
RAK Electronics
P.O. Box 1585
Orange Park, FL 32073

Product:
SALES/EXPENSE Programs for the Commodore 64 and VIC 20—A home or small business program that maintains a full calendar year's sales and expense record for each month by three sales categories and ten expense categories. Totals are calculated for each month. Total year and average month calculations are provided. Also calculates profit. Data is saved on tape for later use and updating. A print routine is provided that works with both the VIC 1515 and VIC 1525 printers. Requires an 8K memory expander.

Price: $7.95 plus $2.00 shipping and handling

Company:
DATA 20 Corporation
23011 Moulton Parkway,
Suite B10
Laguna Hills, CA 92653

Product:
VIDEO PAK for the VIC 20—Provides a standard 24-line, 40-or 80-column screen display with upper and lower case characters. At the same time increases memory up to 70K. Also includes a terminal emulator and screen print feature. Plugs into the VIC expansion port.

VIDEO PAK

Product:
Expansion chassis for the VIC 20—Runs a series of four compatible cartridges—memory, software or games—with the standard 22-pin edge connector. Protects the VIC 20 power supply with a built-in 500ma fuse.

Product:
Memory Cartridge for the VIC 20—Boosts the brainpower of the VIC to 20K and features 200ns RAMs for added reliability.

Prices: Contact company

Company:
Cardco, Inc.
3135 Bayberry
Wichita, KS 67226
316-267-6525

Product:
Atari Cartridge Adapter for the VIC 20—Allows game cartridges designed for use on Atari video game machines to be used on a VIC 20 computer. Produced in cooperation with the Computer Works of Phoenix.

Price: $89.95

new products

Company:
Automated Simulations/EPYX
1043 Kiel Court
Sunnyvale, CA 94086
408-745-0700

Product:
"Games Thinkers Play" for the Commodore 64—Additions to the EPYX line include:
Temple of Apshai—fantasy role-playing game that won the 1981 "Computer Game of the Year" award from the Game Manufacturer's Association. On disk $39.95
Upper Reaches of Apshai—expansion module for Temple of Apshai with over 150 rooms, gardens, berry patches and caverns. $19.95
Curse of Ra—a second expansion module for Temple of Apshai set in the deserts of ancient Egypt. On disk $19.95
Crush, Crumble and Chomp!—a movie monster game that lets the player become any of six man-eating beasts wreaking havoc in a major city. On disk $29.95
Sword of Fargoal—takes the player into an ever-changing bottomless dungeon. Also available on cassette for the expanded VIC 20. $29.95

Company:
M-R Information Systems, Inc.
P.O. Box 73
Wayne, NJ 07470
201-696-3296

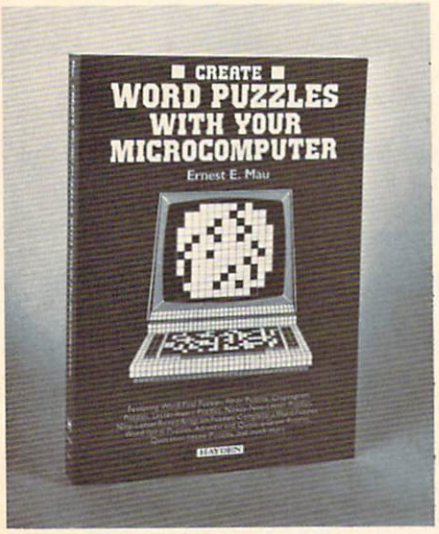
Product:
MicRo Math Blaster for the VIC 20—A combination of arcade games and basic mathematics drill in addition, subtraction, multiplication and division. Color graphics and sound. Levels of difficulty from grades 1-8. Requires 3K expansion.
Price: $15.95

new products

Company:
Hayden Book Company
50 Essex Street
Rochelle Park, NJ 07662
201-843-0550

Product:
Create Word Puzzles with Your Microcomputer—a book by Ernest E. Mau, master puzzlesmith. A collection of puzzle programs in Microsoft BASIC that allow a personal computer to create acrostics, cryptograms, word-finds, quote-falls, fall-ins and other word puzzles.

Price: $14.95 paper



Company:
Fox Video Games, Inc.
4701 Patrick Henry Drive, Bldg #9
Santa Clara, CA 95050

Product:
Revenge of the Beefsteak Tomatoes—game for the VIC 20. Features giant beefsteak tomatoes that have turned into killers by dropping tomato bombs. Players must build brick walls around the deadly tomatoes, trapping them, before they take over the world. The photo shows two of the tomatoes planted at the 1983 Super Bowl as part of the company's promotion.

Price: Contact company. C

commodore product price list

EFFECTIVE FEB. 1, 1983

Order Number	Product Name and Description	Retail Price
SYSTEM HARDWARE AND PERIPHERALS		
VIC20	**VIC 20—The Friendly Computer** Commodore's revolutionary personal computer features color, sound, graphics, programmable function keys, built-in BASIC, expandable memory, low priced peripherals and more! Includes RF modulator, switchbox, cables and self teaching instruction book.	$199.95
C-64	**COMMODORE 64** • 64K RAM • 16 colors, music synthesizer, high resolution graphics	595.00
1530	**COMMODORE DATASSETTE** Provides handy economical storage of user-written or pre-recorded programs using ordinary audio tape cassettes. Works like a standard tape recorder, includes tape counter.	75.00
1541	**1541 SINGLE DISK DRIVE** Fast, high capacity storage and retrieval of data on standard 5¼-inch floppy diskettes. Stores up to 170K on each diskette, with read/write compatibility with PET/CBM computer systems.	399.00
1525	**GRAPHIC PRINTER** Economical dot matrix printer makes paper copies of BASIC programs, letters, business data and graphic displays. Connects directly to the VIC, prints all characters including letters, numbers, and graphics. Uses 9½ inch paper and prints 30 characters per second. Reliable tractor feed mechanism, Device 4/5 and test switch. Accepts sprocketed or sheet paper.	395.00
1520	**COMMODORE COLOR PRINTER/PLOTTER** Draws charts and graphs using 4 color pens. Prints text in 4 sizes. Connects directly to the Commodore VIC 20 and 64.	199.00
1600	**VICMODEM (Telephone Interface)** Lets owners communicate by telephone, access telecomputing and timesharing services. VICMODEM cartridge plugs into the user's port, connects to any modular-jack phone handset. Includes VICTERM I terminal program on tape.	109.95
1701	**COMMODORE COLOR MONITOR** 14 inch screen with outstanding resolution. Special Commodore computer circuit. Video recorder compatible . . . easy practical size.	299.00
1311	**JOYSTICK** Single joystick for playing computer games, remote control computing. Connects to VIC Game Port.	8.95
1312	**TWO PLAYER GAME PADDLES** Double paddle controllers for two player game action. Full rotation with "fire button". Connects to Game Port.	14.95
VIC MEMORY EXPANSION CARTRIDGES		
VIC1110	**VIC 8K MEMORY EXPANDER CARTRIDGE** 8K RAM expansion cartridge plugs directly into the VIC.	49.95
VIC1111	**VIC 16K MEMORY EXPANDER CARTRIDGE** For use with the VIC 20 and/or VIC 1010 EXPANSION MODULE.	79.95
VIC INTERFACE CARTRIDGES		
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. (Not required for modem.)	49.95
VIC RECREATIONAL GAMES ON CARTRIDGE		
VIDEO ARCADE SERIES		
VIC1901	**VIC AVENGER** It's an invasion of space intruders and you're the VIC "Avenger". Space action for arcade enthusiasts.	29.95
VIC1904	**SUPERSLOT** Colorful slot machine game works just like the real thing! Great music and sound effects.	29.95
VIC1907	**JUPITER LANDER** Pilot your "Jupiter Lander" through the treacherous crevices of a mysterious planet. Variable rocket thrust, anti-gravity, horizontal retros.	29.95

Order Number	Product Name and Description	Retail Price
VIC1908	**DRAW POKER** Casino-style poker recreates the real thing! Superb animation and sound effects add to the fun, mystery, and luck.	29.95
VIC1910	**RADAR RAT RACE** The magical mouse maze makes for a fast-paced, challenging game of wit, strategy and reflexes. Excellent graphics.	29.95
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.	29.95
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.	39.95
VIC1920	**PINBALL SPECTACULAR** Plays just like a true pinball machine . . . only computerized . . . flashing lights . . . quick "flipper" action!	29.95
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?	29.95

commodore product price list

Order Number	Product Name and Description	Retail Price
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!	29.95
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 ships, saucers, aliens . . . it's terrific!	39.95
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.	39.95
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.	29.95
VIC1935	**COMMODORE ARTIST** A true lightpen drawing game . . . you are the artist . . . you create the picture . . . multi color!	29.95
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!!	29.95
VIC1938	**TOOTH INVADERS** Prevent tooth decay and battle D.K. Germ using your toothbrush and dental floss. A great arcade/educational game.	29.95
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.	29.95

Order Number	Product Name and Description	Retail Price

SCOTT ADAMS ADVENTURE GAMES

VIC1914 **ADVENTURE LAND ADVENTURE** 39.95
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** 39.95
"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** 39.95
"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** 39.95
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** 39.95
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** 29.95
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** 29.95
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 **SPEED BINGO MATH** 29.95
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.

VIC1928 **HOME BABYSITTER** 29.95
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** 29.95
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 planetery 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** 39.95
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** 69.95
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 x 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** 59.95
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 com-

commodore product price list

Order Number	Product Name and Description	Retail Price
	mand 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.	59.95

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.	24.95
VL103	**BASIC PROGRAMMING – PART II** A continuation of the excellent BASIC programming series with more programs, lessons, and instructional aids.	24.95

Order Number	Product Name and Description	Retail Price

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*** 59.95
Car Chase—Fast-paced road action
VIC 21—Casino-style blackjack
Blue Meanies from Outer Space—Space game
Biorythm/Compatibility—Compare biorythms
Spacemath—Math improvement grades 1-6
Slither/Super Slither—Dexterity game

VT107A **HOME CALCULATION PROGRAM SIX PACK*** 59.95
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

VT108 **MATH IMPROVEMENT SIX PACK** 59.95
Grades 2-6

Numbowl	Backfire
LCM Machine	Scare City Motel
Sector Five	Ruler Dueler

VT109 **SAMPLER SIX PACK** 59.95

Big Bad Wolf	Treasures of the Bat Cave
Music Synthesizer	Crawler
Alpha Draw	Searcher

VT164 **PROGRAMMABLE CHARACTER SET/GAMEGRAPHICS EDITOR** 14.95
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** 9.95
A handy VIC terminal program on tape which converts any VIC to a terminal for use with a telephone modem.

commodore product price list

Order Number	Product Name and Description	Retail Price
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.	5.95
VM110	**VIC 20 PROGRAMMER'S 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.	16.95
ACCESSORIES AND REPLACEMENT PARTS		
1515P	**GRAPHIC PRINTER PAPER** 1000 sheet pack, tractor feed, 15 lb. bond	15.00
1515R	**PRINTER RIBBON CARTRIDGE** For VIC Graphic Printer	9.95
90477801	**TV SWITCHBOX**	7.95
90510101	**RF CABLE**	2.95
902505	**POWER SUPPLY**	19.95
32145301	**MODULATOR**	19.95

This price list supersedes all previous price lists.
Prices subject to change without notice.

*Bally Midway games developed under Commodore's licensing agreement with Bally Manufacturing Company.

COMMODORE BUSINESS MACHINES, Inc.
Consumer Products Division
1200 Wilson Drive, West Chester, PA 19380

advertising index

COMPUTE!'s First Book Of VIC

Authors: **COMPUTE!** Magazine contributors
Price: $12.95
On Sale: Now

Finally, it's VIC's turn!

Users of other popular personal computers have been enjoying their **COMPUTE! Books**: *COMPUTE!'s First Book Of PET/CBM, Programming The PET/CBM,* and others.

Now, there's a book devoted exclusively to the Commodore VIC-20™ Computer: *COMPUTE!'s First Book Of VIC.*

The editors of **COMPUTE!** Magazine – the leading resource for the VIC-20 – gathered together the best VIC-20 articles published since the summer of 1981 and added some new material. The result is more than 200 pages of valuable information – information that goes beyond the instruction manuals. In the **COMPUTE!** tradition, it is carefully edited to be easily understood and useful for beginners and experts alike.

COMPUTE!'s First Book Of VIC is spiral-bound to lie flat, and includes ready-to-type program listings and articles such as "The Joystick Connection: Meteor Maze," "STARFIGHT3," "Train Your PET To Run VIC Programs," "Renumber BASIC Lines The Easy Way," "High Resolution Plotting," "Custom Characters For The VIC," "VIC Memory – The Uncharted Adventure," and "A Simple Monitor For The VIC."

At only $12.95, less than most computer manuals, *COMPUTE!'s First Book Of VIC* is among the best resources a VIC user can own.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2 shipping and handling. Outside the U.S. add $5 for air mail, $2 for surface mail. All orders prepaid, U.S. funds only.
VIC-20 is a trademark of Commodore Electronics Limited.

commodore

1200 Wilson Drive
West Chester, Pa. 19380

Address Correction Requested

ISBN 0-88731-004-4