THE TORPET

Second Class Mail Registration no. 5918, Shelburne, Ontario

The INDEPENDENT Commodore Users' Magazine No. 20 JUNE 1983

$2.00

VIC, PET and C-64 GAMES

Making Games, Finding Games, Playing Games. Is It Good or Bad?

0 74470 12601 06

How to get Hundreds of Free Programs for the VIC-20 and Commodore 64

see page 2

Richvale Telecommunications

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

$185.00 Canadian
$149.00 U.S.
PLUS CUSTOMS BROKERAGE, HANDLING AND MAILING CHARGE.

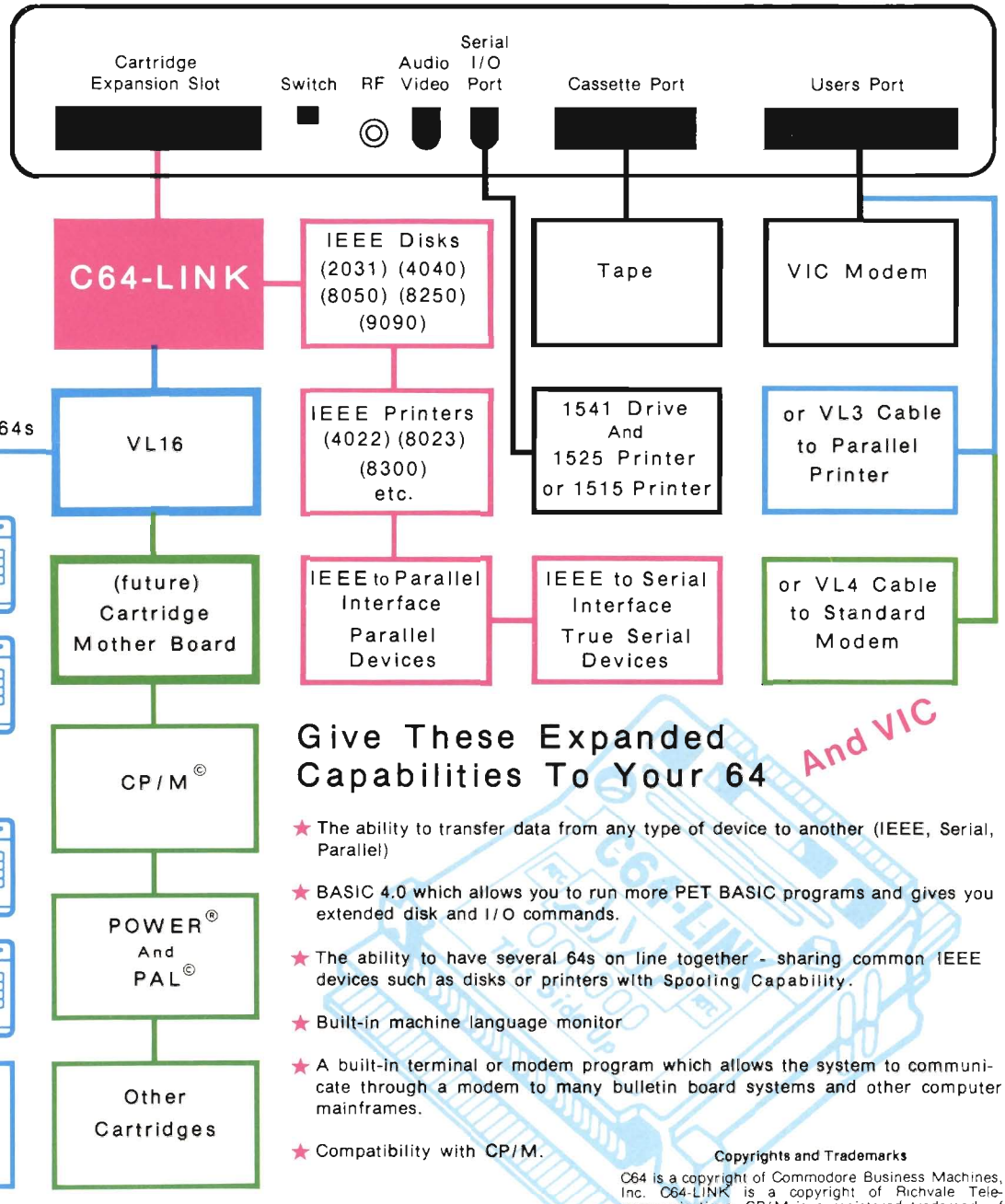
RTC

C64-LINK©
The Smart 64

Also available for VIC 20

RTC

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



Give These Expanded Capabilities To Your 64 And VIC

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

Contact your local Commodore dealer or RTC.

CONTENTS

The TORPET
published by
The Publisher
Horning's Mills, Ontario
Canada L0N 1J0
(519) 925-5376
(416) 782-9252

Bruce M. Beach, Editor
Sandra Waugh, Associate Editor

Published monthly (except April and December)
Single subscription rate - $18.00 per year

Honorary Editorial Committee:
Jim Butterfield, Associate Editor of Compute, Toronto, Ont.
Dave Williams, Contributing Editor of Info Age, Toronto, Ont.
Elizabeth Deal, well-known contributer to Commodore Magazines, Malvern, Pa.
Jane Campbell, San Diego User's Group President, San Diego, Ca.

Cover Credit: Graham Bailey

Special bulk rates to clubs

Printed In Canada
ISSN # 0821-1809
Second Class Mailing Permit Number 5918
Mailed at Shelburne, Ontario
Application to Mail at Second-Class Postage Rates is Pending at Buffalo, N.Y.
POSTMASTER: Send addresses to The TORPET, 1 Brinkman Ave., Buffalo, N.Y., 14211

EDITORIAL

Submissions

TO OUR CONTRIBUTORS: PAST, PRESENT and FUTURE:

The TORPET still pays $20 per page for articles and we welcome new submissions. We do have a **small** backlog of articles which we have not yet been able to fit in.

We are trying to get to everything. We plan, in the near future, to have letters prepared to acknowledge the received articles, but this will take some more time before it is implemented.

Send us a letter explaining the types of articles you would like to write. Frequently we need a special subject covered and, if you have indicated an interest, then we can contact you.

Submitting Material

Please type your name, address, etc. on **both** the manuscript, and in the disk file if you are submitting one. (We love disks but will retype manuscripts.)

Send any photographs and/or diagrams (or cartoons!) that go with your article. Diagrams are easiest for us to use if they are drawn in fairly dark ink, or marker, and are done on plain white paper (we will redo them if necessary).

We would also like a photograph and biography of the author.

Check your article for accuracy before you send it to us. Read any programs through an extra time, just to be sure, and **please** include a little BASIC file containing any routines in your program so we can print it directly. We don't like mistakes any more than you do.

If you send your article on a diskette, we will send it back to you with TPUG's current monthly library release. This is the easiest way for us and the most accurate for your articles. Do send us a printout of the article as well so that we can check carefully.

Send your submissions to The Publisher, in Horning's Mills(address in front of this issue).

Copyright Policy

Our editorial committee has stated that the non-acceptnce of copyrighted material rule is not cast in iron but for the present it remains TORPET's editorial policy that everything it prints is public domain so that new computer clubs can have material for their newsletters.

We also would like for you to share your thoughts about what we are doing right or wrong in TORPET. Write to the editor and tell us your advice.

TORPET needs your help to serve you the best that it can.

Sandra Waugh
Associaate Editor

Mistakes do happen...

We want to apologize to Mark Lieberman who wrote the article about the Waterloo workshop for the May TORPET which we credited to Greg Harrison, Sorry about that. Ed.

WHERE TO GET GAMES

Every now and then someone will call me and say, "How do I get those hundreds of free programs that you tell about on the cover of The TORPET?". Well, the answer is very simple. Join TPUG. Then you will have access to the world's largest public domain library.

There are actually thousands of free programs in the library. And any member of TPUG can get them in a number of different ways. If you are close enough that you can attend a TPUG meeting, there is a meeting disk given out at each meeting (recent releases are listed on the back page of this TORPET). Or you can attend one of the all day copy sessions, which again are free to members, and there you will have an opportunity to copy many disks.

If you live a long distance from TPUG then you can order disks or tapes from the club office. The programs are still free but they charge a nominal fee for the copying, materials, handling and postage. To be exact $10 for a disk ($12 for 8050/8250) and $6.00 for a tape.

There are still other ways to get the disks. From dealers who are members of TPUG (many of these just charge you for the disk material), from other TPUG members, or by joining the copy tree.

Most of the the disks and tapes have between 20 and 40 programs on them. So, you see, they really are free. To found out just what the programs are, you need to look at the TPUG club library listing. This is published approximately quarterly and is then sent to all TPUG members.

If you have bought your TORPET on a news stand, you should have a copy of of the library listing inside your TORPET.

Incidently, all TPUG members receive a free copy of the TORPET each month and since TPUG membership is just $2 more than a TORPET subscription I think it is a real deal.

Cover Story

The TPUG Office with-(from the left) Joy Bennett entering memberships into the computer, Chris Bennett- business manager (on the phone) taking some of the many inquiries received daily, standing), Doris Bradley -TPUG assistant business manager (processing the mail), Bruce Beach - TORPET editor (at word processor), and Tracy Bennett preparing new membership cards.

Requests for disks and tapes are handled in this room but the duplicating of disks and the mailing functions are handled in two additional rooms of about the same size. Some days find as many as ten people working in the office.

The TORPET editor is actually seldom there because the TORPET is produced in Horning's Mills.

Letters to the the Editor

We've been members for a few months now and wanted to tell you how good the club magazine is. Also, we appreciate the way available tapes and disks were explained and listed in the March/April magazine. By simply stating "Programs for the Vic", we could easily place the following order. We look forward to more C-64 programs--especially business applications. As soon as we see that you are offering them, we plan to buy a C-64. Until then, we will continue to learn on our VIC-20.

Roy L. Harris #4774
Highland Park, Illinois

I don't care what anyone says (referring to the A Criticism article in this months Torpet) you people really do a service to the community in trying to cater to all the different types of computers and computer nuts. If T.S. has a hangup about his obsolete SUPERPET then maybe its time he changed computers. (In the event he should want to discard it I will take it off his hands as a mercy gesture (for free of course)).

J. A. Maguire #961
Grafton, Ontario

I want to say how much I enjoyed the articles on word processors. It was a very helpful feature.

David L. Nelson #4578
Sacramento, California

I find the newsletter very interesting and valuable. Keep up the Good Work!!

William F. Osachoff #2106
Moose Jaw, Saskatchewan

I have received the March/April issue of the TORPET last week, and I found the section on the TPUG directory very useful. I also like the way you now identify your disks; it makes understanding the disk directory a lot easier.

George A. Murton #3784
Allentown, Pennsylvania

Carry on the good work! You fellows are filling a gigantic void left by Commodore after marketing the 64. (I have written them in this respect; responsibility to the customers etc., etc., and have received the lamest (?) of response.

Bruce D. Ball
Colorado Springs, Colorado

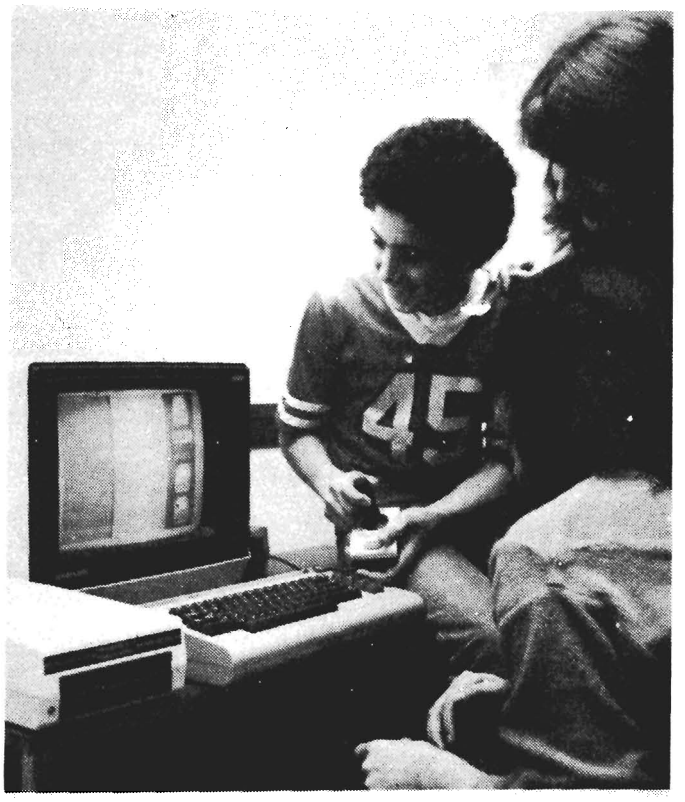
I have enjoyed being associated with TPUG and have really found the Torpet to have a great deal of information that is useful to me. I have referred several people in the area to you as I feel that your group is very helpful to novices like myself. Keep up the good work.

Wayne D. Lowery R.N. #3896
Marrero, Louisiana

FEATURE

The Feature this month is on games. The cover was supposed to be also. You just about lost ye old editor from a heart attack when he found out that a thouroughly incompetent individual at the color separator had sent the wrong picture for printing. Someday maybe you will get to see the neatest picture we have ever had for a cover. But not on this one. Not to keep you in suspense it was a couple of really cute kids playing games. Oh, well life is sometimes tough.

Thats why we need games.



Our game players on the right are Egwina Kennedy (back to camera), Chris Bennadato (on the desk) and Sue Simone (standing) who graced our cover on a previous issue. They are all grade 8 students at Our lady Help of Christians, separate school in Richmond Hill, Ont.

The computer in the picture was courtesy of RTC, without whose help we wouldn't we have gotten this issue out at all when our disk drive got struck by lightning (I'm not kidding). Its been that kind of month. Tough.

FEATURE

Microtechnology

-- Hazardous to your social health?

by Clifford Wong *St. Laurent, Que.*

The personal computer, a marvelous new machine with its costs continuously dropping, -- its variety, availability, and number of users, is ever on the increase. Although it has been around for a few years now, the microcomputer is still in its infancy. Its future is as difficult to imagine as an ancient Roman chariot racer trying to foresee the Indy 500, or an SST Jetliner. Future developments and effects of microtechnology can barely be speculated upon, but its social impact has already begun. Among its manifestations in business, industry and entertainment, nowhere is it more advertised than in home video games and personal computers.

With any technological revolution, there is a negative side, home computers not withstanding. The complaints are numerous, but what parents are concerned about is that excess preoccupation with these machines can present a real danger to a person's social upbringing, especially a young person. "Junior spends more time on that damn machine than he does with his friends. He won't even watch Love Boat any more. Look at all those quarters he's popping into the arcade. Now he wants an Intellivision game for Christmas! That's going to cost! Have you ever seen how violent those games are?"

Well readers, I'll confess that considering I am 32 years old, I'm worse off than Junior. I own an Atari game system (eat your hearts out, folks, I score 398,325 on Missile Command) and I'm on my PET everyday. If significant side effects are produced, I'll be one of the first reporters on the scene.

Indeed, I have found some negative points. Let's face it, despite the declining prices, microcomputers still cost an arm and a leg. I had to borrow to purchase mine, not usually a good, economical habit. I also spend on books, software magazines and clubs directly or indirectly dealing with my machine, and I am making plans to buy a new one. Now, this second point may alarm you, but, while my mind is so immersed in computers, some other things have become slightly less interesting to me. Terrifying, isn't it? But let's take a closer look at these problems.

Concerning time with my friends, never have I turned down a pleasant social invitation, or wanted to so that I could stay home and pump more data into my assembler. Never do I confine my conversations to computers, and rarely do I instigate such conversation. Never do I confine my social circles to micro-nuts and I never use my computer knowledge as an ego crutch. However, sometimes computers may be a better alternative for Junior than his friends, who are planning to damage public property next Friday night. Would you prefer that?

"What about the violence and destruction in most of these games?" This reminds me very much of the Pay-TV Playboy movies, where men are supposed to get the wrong impression of women. Have you ever seen the way females flock to see a nude male dancer? Talk about wrong impressions! If such movies were made about men, there would be no concern, for everyone knows women have brains that can distinguish entertainment from reality. Please give men equal respect. If you watch a movie like Dragonslayer, will you be convinced of the existence of fire-breathing monsters? May Godzilla strike me down if they do! If the Playboy movies do indeed occasionally corrupt a very young mind, the blame then belongs to the consumer, not the product. Similar accusations may have been made against hit-the-target video games.

FEATURE

When I was a youngster (perhaps I still am), I used to collect toy guns, tanks and little soldiers. The real treat was getting a toy rifle on Christmas Day that was more realistic than the previous one. Everyday we simulated wars with these toy weapons in our Cub and Brownie uniforms. None of the participants ever joined the armed forces. We all became peace-loving wimps who would prefer to live long and uneventful lives, where violence never goes beyond the screen. Throughout history, man has been violently at war with himself. After thousands of years of bloodshed, I don't think we have the right to accuse the microprocessor, or its action programs, for being responsible.

"Now what about the continuous flow of Junior's quarters into the arcade, and the high cost of these home video games and personal computers?" Come on folks, let's not overlook the obvious. Again, this is not a problem brought on by microtechnology. Any in-depth past-time will cost money. Ask an automobile or photography enthusiast how much they have spent and are planning to spend on their hobbies. I know a video tape-recording fanatic who not only owns three colour TV sets and two VCR machines, but has also spent $800 on a cabinet just to hold some of this equipment! He has only just begun sinking his money into this craze.

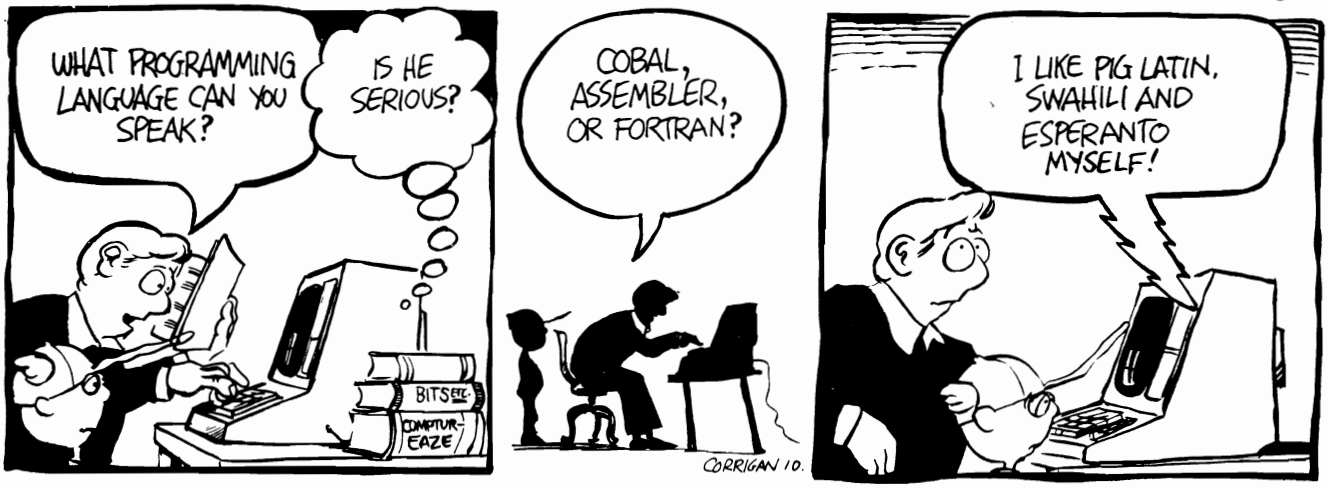
I also knew a hi-fi enthusiast who stated ten years ago, that you didn't even begin to hear a decent sound unless you had invested at least $5000 into your system. You can be sure that he listened to every dollar of sound that came out of his speakers.

If Junior's quarters don't go into video games, he will find alternatives --some, perhaps, less desirable. Although hobbies are expensive and hobbyists tend to get wrapped up in them, I have found the company of such people more pleasant. They seem to be more at peace with themselves and less offensive to others, while those without tend to be a bit lost, and often like to stick their noses in the private lives of others.

"This is all pretty easy for you to say. You are susceptible to random influences. You are an adult with an established mind." Am I? Take another look at my Missile Command score. It is an example of my devotion to a random influence. Any adult can make a poor decision, and everyone is susceptible to temptation. While teenagers still have a lot to learn, remember that they have minds too--minds connected to senses that too often hear and see things parents overlook. Accumulated knowledge is not a valid measurement of maturity or intelligence. Also remember that a strongly established mind is one that is usually slow to learn.

In conclusion, these publicized problems have little to do with microtechnology. I haven't even begun to mention the good points. The controversy concerns a human situation. Take away the machines and you still have those problems. They are only props in the play. It is people that must be dealt with.

Bytes ***by Patrick Corrigan***



FEATURE

GAMES OF LIFE

by Harry Baecker

Calgary, Alberta

Many people, particularly in the computer field, are familiar with John Conway's Game of Life, first widely publicized by Martin Gardner in his columns in Scientific American" +1,2,3]. Now is the moment to introduce a consumer warning: the Game of Life bears little resemblance to arcade games on computers! It is deterministic, allows no "player" input after the initial configuration of entities on the playing field is established, and is an automaton, a game where chance has no play. But many find it interesting to follow the evolution of initial patterns, often of those that form other elegant patterns.

Some years ago I became interested in trying to "improve" the game. First I toyed with the introduction of bisexual entities in place of the asexual ones of the original. That's always fun. Then came heresy, to make the game non-deterministic. I wrote a few versions that way, none very satisfactory.

In February 1981, I lectured on the topic at California State University, Chico. One of the students present was inspired to adopt it as his graduate thesis topic and produced CASL, a Cellular Automata Simulation Language +4]. Most valuable were the discussions with many students who fired me with new ideas.

So evolved the present versions of LifeSim. The object is to provide an elementary framework for players, users of personal computers, to have their ecological fantasies acted out. Up to 9 different types of entities, chosen by the player, interact among themselves and with the environment on a 20 by 60 field. Whether the entity types represent different animal, plant, insect, etc., species, or the two sexes of the same species in some cases, is left to the imagination and design of the player.

As in the original game a generation is an iteration over all the cells of the field during the course of which the surround of each cell is evaluated to determine birth or survival in that cell. Given the planar square grid of the field each cell has a surround of 9 cells. The occupants of cells adjacent to the entity in the current cell have an effect on its survival. This occurs through evaluation of the interaction factors of the entity and the surrounding occupants. If the cell is empty then the surround is evaluated to determine if that surround will generate birth in the cell of an entity of a type present in the surround. In each case the environment condition of the cell is a contributory factor. Cells at the edge of the field are considered to have empty cells as neighbors there.

Since the interaction factors are provided by the player, we have a problem. We could use numeric constants only as interaction factors. But it seemed much more realistic to allow expressions and even functions that would be re-evaluated constantly during the simulation, then we could allow for variations in effect according to the current cell co-ordinates, according to time, the passage of generations, according to the proximity of other entities, and according to any other factors the player might choose to stipulate, including the output of a random number generator. Once I got carried away this far there were only two ways to go, either to build an elaborate compiler or interpreter for a simulation language, or to use APL. I chose the latter route. Crazy?

Thus the present implementation is in microAPL running on a Commodore SP9000 SuperPET. This has shaped the implementation to some extent, especially in the limits on the size of workspace possible, 32KB, which has lead to compromises in the implementation. The factors available to the player and their interaction are not as flexible as one might wish. Even so, a generation takes about an hour to evaluate in the more elaborate version. Also, the

environment conditions have been limited to one, wheras it would be more realistic to allow several, to represent independent factors such as climate, topography, etc. Of course, the player s functions can compensate for this limitation, the price is execution time. Perhaps more than 9 interacting types would be useful, but again space and time limit what is possible.

It will be evident that to play this game the player needs a certain level of expertise in the use of APL. This is unfortunate. Or, because APL is the inspiration of a Canadian, should we regard it as suitable popularization of a Canadian program product?

As an aside, I would mention that the most serious contribution to the sloth of the simulation is that subscript expressions are evaluated in floating point by the interpreter. But this dementia is not confined to this implementation of this language interpreter, it is common for interpreters on microcomputers. Implementors would be well advised not to remove the user from the native mode integer arithmetic facilities of the host chip in future.

To play the game the user loads the workspace of the relevant version (see below). The player now has to establish the functions that drive the interactions, if these are needed, and the player must also initialize any global variables needed by these functions or the interaction factor expressions. It is devoutly to be hoped that the user functions have been debugged previously, unless the player understands the global simulation functions thoroughly there is little possibility of debugging the user functions in the context of the workspaces provided.

The player then invokes the input function provided. This will request the number of types to be simulated, T, where 1<=T<=9, and the interaction factors. Each interaction factor can be any valid APL expression that is appropriate, as usual limited to one line of screen input.

The first expression is the environment condition for the cells of the field which is independent of the occupancy of any individual cell. The next T expressions are the birth factors for the T types, used to evaluate the possibility of the birth of any type in an empty cell.

Next come T sets of T+1 expressions, used to evaluate the fitness factor of an entity in an occupied cell. The first expression of a set gives the interaction value of that entity with the environment. The following T expressions give the contribution of any entities in its surround to its fitness factor. Each entity is born with a fitness factor of 1, each generation the contribution of the environment and of the surround are summed and added to that fitness factor, as long as it remains greater than 0 the entity continues to live. As we use the term "generation" it would be more accurate to say that an entity of the same type continues to occupy that cell in the next generation.

The player must next implant an initial population in the field. This is best done by a (debugged) user established function that invokes the "hooks" provided in the workspace for this purpose.

At last the serious business can begin. The player invokes the interaction function provided to evaluate the interactions on the field for the number of generations wanted. When the required number of generations have elapsed the user can display the field on the screen, can call for a detailed printed report, or can file the field and attendant data to diskette for later analysis. Of course, the player can also invoke the APL >SAVE command to continue the game later.

So far two workspaces, which we will call Dynamic and Static, have been developed for differing versions of the game.

During the Dynamic version each player provided condition expression is evaluated anew every time it is applied. The value returned can thus depend on factors such as the current cell coordinates, the environment condition and on any other variables the player chooses to maintain. The penalty is time.

The Static version evaluates each condition expression once before interaction begins. In the case of the global environment condition this means that it is evaluated for every cell of the field and that the result for each cell is stored. During interaction every occupied cell adds its environment interaction factor to the stored value for that cell, so modifying the environment by occupancy. The cell coor-

dinates are the only global variables, other than any supplied by the player, during evaluation of the global environment condition. No global variables other than those provided by the player are available for evaluation of the other expressions. The expression values are stored in a table and applied during interaction without modification. This version allows a generation to be evaluated in about 20 minutes on a SuperPET.

Entities are born with a fitness factor of 1 and die if it drops to 0. Choice of values for the condition factor expressions is therefore critical to any simulation. Only the player can decide what parameters are sensible for the scenario being simulated. A short course in population genetics and demography might be useful.

Quite evidently these are not parlor or arcade games. Setting up a simulation requires a great deal of thought and preparation and the playing time is best measured in days. They are intended as very elementary ecological models that simulate interaction in a habitat in terms of place. Their main interest lies in two aspects, that the interaction factors are specified by the player and can be about as elaborate as one could wish, and that the model is a discrete simulation that allows distribution and mingling of populations to be observed directly in the model, whereas most demographic models currently available employ continuous simulation and do not provide such pedagogic aid. In a lighter vein, we often consider "what if" questions if we think about ecology at all; here are some tools that, with a little experience, allow the curious to explore these questions.

That would have been about all that could be claimed until recently. Now there is a book published, "Laws of The Game" +51, by two eminent biophysicists, one of whom is a Nobel prizewinner, that shows how rules much simpler than ours, nearer to those of Conway's original, and a few random interactions, are sufficient to explain much of molecular biology. The book has a very engaging and provocative subtitle: How the Principles of Nature Govern Chance. Dare one hope that at the present level of elaboration is a beginning to meaningful simulation of ecological events?

The curious may also be interested in that the Game of life is mentioned in at least two science-fiction novels +6, 71.

Should there be sufficient interest in LifeSim, it would be possible to arrange for the distribution of the workspaces and documentation on diskette. Please contact the author, c/o Department of Computer Science, University of Calgary, Calgary, AB, T2N 1N4, Canada, or try to persuade TPUG to venture forth and make an APL diskette available!

BIBLIOGRAPHY

+1) CONWAY, J.H., "Regular Algebra and Finite Machines", Chapman and Hall, London, 1971.

+2) Gardner, M., 'The Fantastic Combinations of John Conway's New Solitaire Game "Life"', Scientific American, Vol. 223, No. 4, Oct. 1970, pp. 120-123.

+3) Gardner, M., 'On Cellular Automata, Self--reproduction, the Garden of Eden and the Game of "Life", Scientific American, Vol. 224, No. 2, Feb. 1971, pp. 112-117.

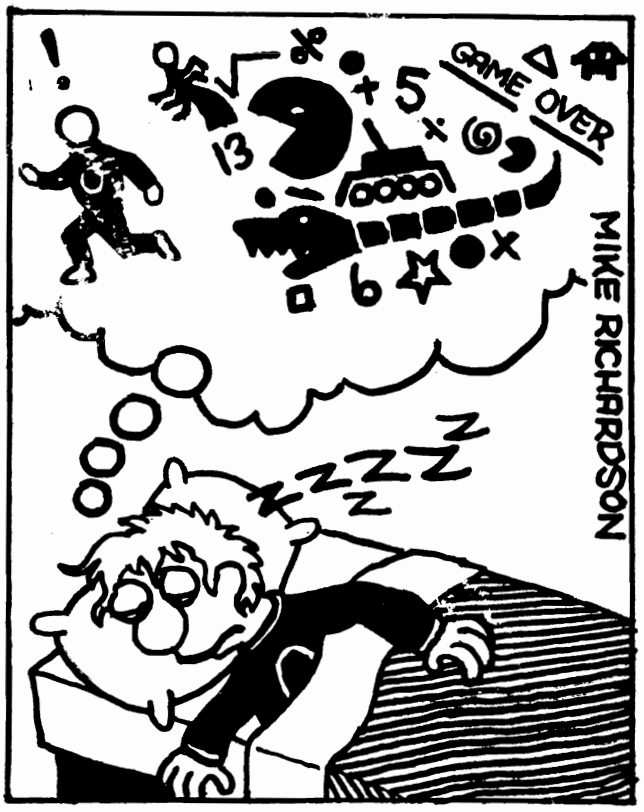
+4) Proett, T.M., "Simulation as a Game of Competition", California State University, Chico, 1981.

+5) Eigen, m. and Winkler, R., "Laws of The Game, Alfred A. Knopf, New York, 1981.

+6) Foster, M.A., "The Gameplayers of Zan", Daw Books, New York, 1977.

+7) Anthony, Piers, "OX", Avon Books, New York, 1976.

Kapute *by Mike Richardson*



Creating Sprites on the C-64

by David Bradley
Toronto, Ont.

First, get some graph paper and set up a grid 24 columns wide by 21 rows deep and number it as shown in Figure 1. This is the space that you have in which to create your sprites.

Look at Figure 2. OK, so it isn't the best Commodore flag, but it serves its purpose. The next thing I should tell you is how to change that grid into data. Look at the top of Figure 2. Note the way that the columns are numbered. If a square is filled in, it is considered 'on' or a logical 1. So change the filled in squares to 1's and the blank square to 0's such as Figure 3. Now, look at byte 1, row 1. All of the bits are 0. So the first byte will be 0. The bits in byte 2, 3 are also made up of 0's. So the data for the first line will be:

DATA 0, 0, 0

Now we look at row 2. The first byte is all 0's so it, like all the bytes in row 1 is equal to 0. But look at the next byte, byte 2, row 2. It is '0 1 1 1 1 1 1 0 0'. So look at it this way:

```
1 0 0 0 0 0 0 0
2 6 3 1 0 0 0 0
8 4 2 6 8 4 2 1

0 1 1 1 1 1 1 0 0
```

0+64+32+16+8+4+0+0=124

So what you are doing is adding up the bits by replacing all of the 1's with the value of their columns and leaving the 0's as 0's.

Now it is time to put this data into the computer and see what the sprite looks like. So type in the following program, add the data statements for my sprite and then run the program.

To save you some time, here is the data for Figure 2.

```
Byte   1    2    3
Row 01-000, 000, 000
Row 02-000, 124, 000
Row 03-001, 254, 000
Row 04-003, 254, 000
Row 05-007, 254, 000
Row 06-015, 130, 000
Row 07-030, 001, 255
Row 08-060, 001, 254
Row 09-056, 001, 252
Row 10-056, 001, 248
Row 11-056, 000, 000
Row 12-056, 001, 248
Row 13-056, 001, 252
Row 14-060, 001, 254
Row 15-030, 001, 255
Row 16-015, 130, 000
Row 17-007, 254, 000
Row 18-003, 254, 000
Row 19-001, 254, 000
Row 20-000, 124, 000
Row 21-000, 000, 000
```

```
10 B=53248
20 FOR I =0 TO 62
30 READ A
40 POKE 64 * 200 + I, A
50 NEXTI
60 POKE 2040, 200
70 POKE B + 21,1
80 POKE B, 160
90 POKE B + 1, 127
100 DATA...
```

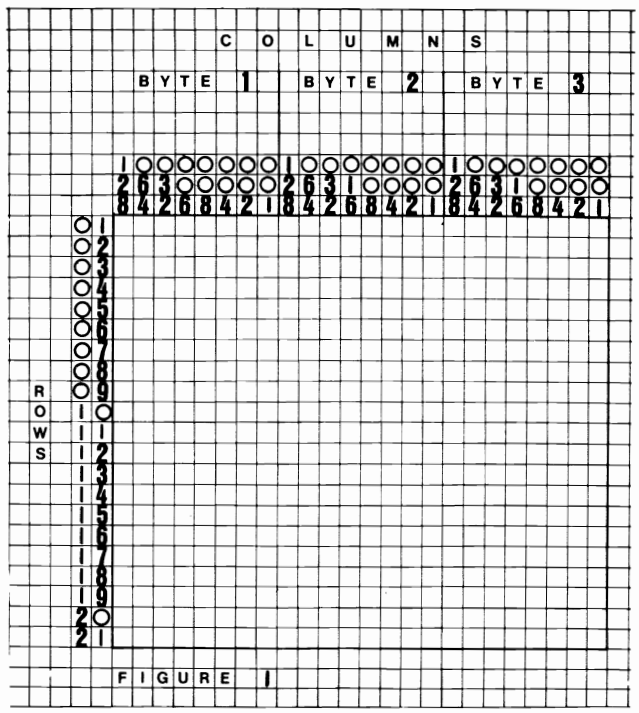


FIGURE 1

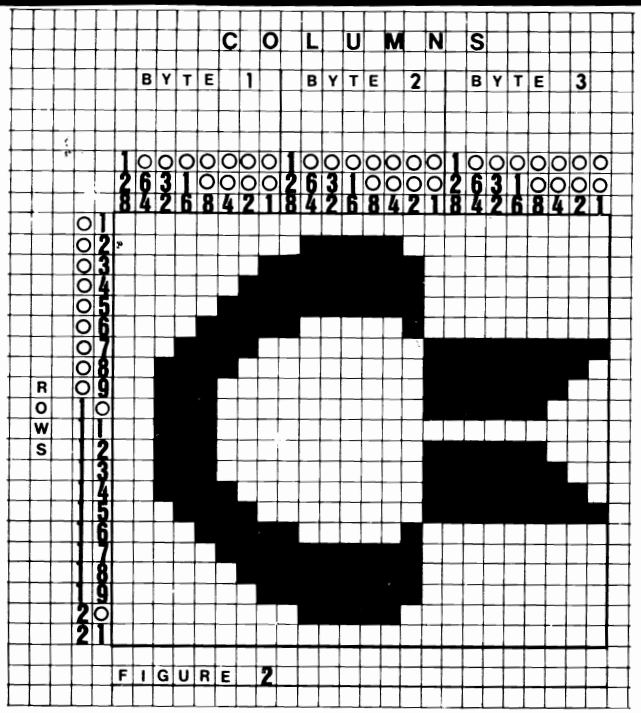


FIGURE 2

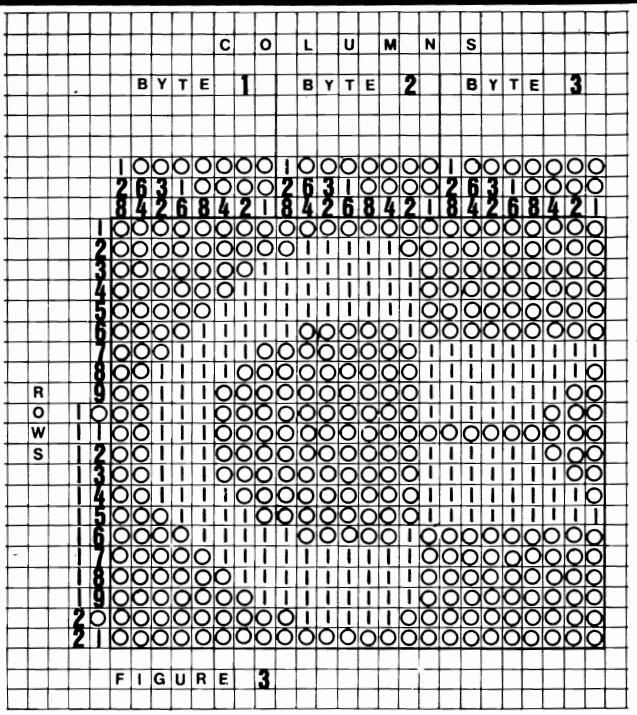


FIGURE 3

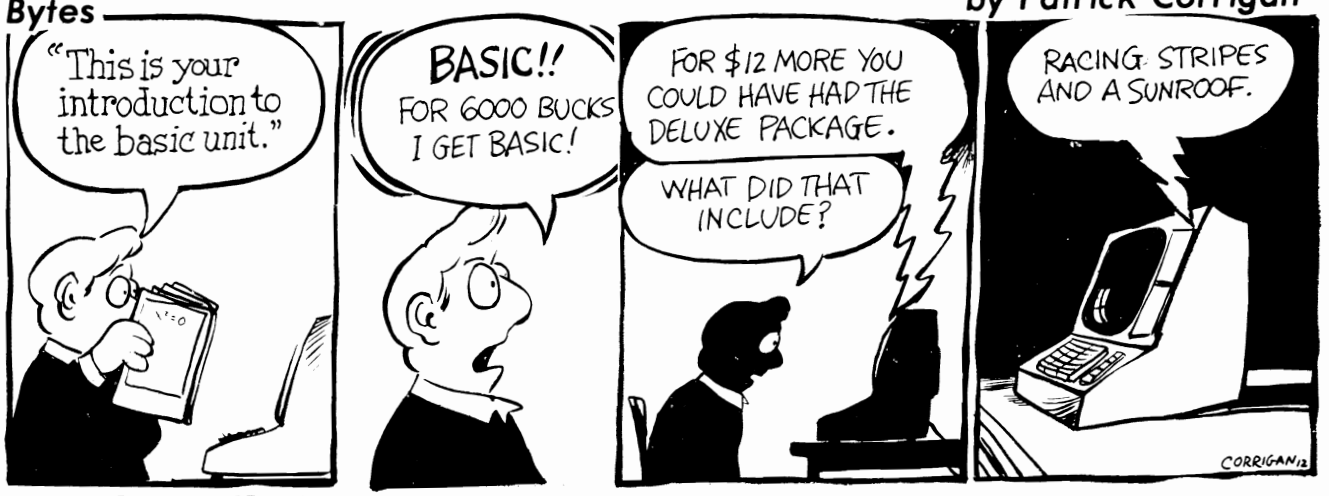
If you did the program correctly and got the data right, you should now see a Commodore flag in approximately the middle of your monitor. You have just created your first sprite!

Line(s) and Description

10 Sets the value of B to the start of display chip
20 Start of read loop
30 Reads A
40 Pokes A. The 200 * 64 is the place in memory that you will have to point the sprite so it knows what data it is supposed to use.
50 End of read loop
60 Pokes (points) 2040 (sprite 1) to get its data starting at memory location 200 * 64.
70 Turns on sprite 1
80 Sets vertical position for sprite 1
90 Sets horizontal position for sprite 1
100 The data you entered.

Hopefully this will assist you in understanding sprites. If you have any questions send them to the TPUG business office addressed to me....David Bradley. Or call the NORTEC BBS at (416)-487-2593 and leave me, the SYSOP, a message.

Bytes — by Patrick Corrigan



PETSPEED

Now for COMMODORE 64!

FAST ENOUGH FOR THE HUMAN RACE

Our alien won't hang around for slow software. He wants crisp responses and really fast processing.

For the human race too, slow PET BASIC is not good enough. When we run a program, whatever it is, we want fast efficient action.

PETSPEED, the compiler recommended by Commodore, is now available for the 64 and CBM 2. It can make any BASIC program run many times faster. It even speeds up disk handling. We guarantee that PETSPEED is easier to use and generates faster code than any other BASIC compiler for Commodore Systems.

Using PETSPEED is simple. Just type in the name of the program wait a few minutes and then watch your software run up to 40 times faster.

Petspeed is not simply a compiler, it contains a powerful OPTIMISER. While PETSPEED is compiling, it breaks your program down into tiny fragments and reassembles it removing the unnecessary and simplifying the complex. Dazzling graphics. Lightning sorts. With PETSPEED anything is possible.

Also available INTEGER BASIC COMPILER – 150 to 200 times the speed of Basic. Integer Basic is for those applications where the speed of machine code is required without the inconvenience of assembly level programming. Ideal for scientific and educational users. Compatible with Petspeed.

PETSPEED (Commodore 64)	$199
PETSPEED (8000 or 4000 series)	$249
INTEGER BASIC (8000 or 4000 series)	$249
SPECIAL OFFER: Petspeed PLUS Integer Basic	$400

All prices in Canadian dollars
Mail Orders Accepted.

COMPUTERS WORKSHPS LTD

465 KING STREET EAST, UNIT 9
TORONTO, ONT. M5A 1L6
(416)366-6192

DEALER INQUIRES WELCOME

COMPU $ENSE

CS1	**Quick Brown Fox** The Word Processor of this decade!	$55.00

COMMODORE SOFTWARE

VIC-1211A **VIC-20 Super Expander** $55.00
Everything Commodore could pack into one cartridge - 3K RAM memory expansion, high resolution graphics plotting, color, paint and sound commands. Graphic, text, multicolor and music modes. 1024x1024 dot screen plotting. All commands may be typed as new BASIC commands or accessed by hitting one of the VIC's special function keys. Includes tutorial instruction book. Excellent for all programming levels.

VIC-1212 **Programmer's Aid Cartridge** $45.99
More than 20 new BASIC commands help new and experienced programmers renumber, trace and edit BASIC programs. Trace any program line-by-line as it executes, pause to edit. Special KEY command lets programmers redefine function keys as BASIC commands, subroutines or new commands.

VIC-1213 **VICMON Machine Language Monitor** $48.99
Helps machine code programmers write fast, efficient 6502 assembly language programs. Includes one line assembler, disassembler.

GAMES FOR YOUR VIC-20®

CSN **Snackman** - Better than Packman $16.95

CC500 **Intruder-Scrambler** - In your bomber, invade the defending scramble system, dodging rockets, to blow up enemy posts, etc. 17.95

CS1 **Flags of Nations** - A game that challenges players to identify flags of various widely-known nations of the world 10.95

CS2 **Flags of Nations** - Second Edition - A field of 34 flags of lesser known nations of the world 10.95

CS3 **Cities and States** - A game that draws a map of a state or states and asks players to name key cities in those states 10.95

CS4 **Cities of the World** - Deals with important cities of nations throughout the world 10.95

CS5 **Mountains and Rivers** - Draws large geographical area maps. You identify major mountain ranges, rivers & bodies of water 10.95

CF1 **Galactic Blitz** 22.95

CF3 **Sidewinder** 25.95

CS3 **Centipod** 17.95

CV108 **Bomber** 8.95

Hangman 8.95

NEW GAMES FOR YOUR C-64

Tank Arcade (Also for VIC-20) - Pre-determine how many hits it will take to wipe out your opponent. Then, on with the battle! Battlefield changes. $13.95

Roadracer - Choose the type of track & a time or lap race. Use steady control at speeds of 50 to 200 miles per hour. Hit the wall & lose valuable time. 13.95

Shootout at the OK Galaxy (Also for VIC-20) - 30 alien warships have entered your war zone. Shields up? Energy level OK? Defend yourself! 19.95

Galaxy - Have you ever wanted to conquer the universe? Send your galactic fleets out to explore solar system by solar system. From 1 to 20 players. 19.95

Bomber Attack - Ground to air warfare. You're in command of a supersonic bomber over enemy terrain. Drop all 25 bombs on key locations. 14.95

Midway Campaign - Your computer controls a huge force of Japanese ships trying to conquer Midway Island. Your only advantage is surprise. 19.95

Dnieper River Line - A fictionalized engagement between Russian & German forces in 1943. Soviet forces, controlled by the computer, seek to overrun your line and capture sufficient objectives to attain victory. Four levels of difficulty. 25.00

Tanktics - Armored combat on the Eastern front of WWII. You start outnumbered 2 to 1 but you choose your tank types before the battle. 24.50

Guns of Fort Defiance - You are the commander of a 19th artillery piece in a besieged fort. Choose type of ammo. Set the cannon's elevation, deflection. 20.00

Computer Baseball Strategy - You, the manager of the home team, test you skill against a wily and unpredictable opponent, your computer. 15.95

Lords of Karma - Like an intriguing puzzle! Decipher secrets while explo ring a mythical, magical city & countryside. Avoid the lurking monsters! 20.00

North Atlantic Convoy Raider - It's the Bismarck convoy raid of 1941! The computer controls the British ships. Will you change history? 19.95

Planet Miners - Compete against others and the computer to stake valuable mining claims throughout the solar system in the year 2050. 19.95

Conflict 2500 - In 2500 AD, earth is threatened by attacking aliens with an infinite # of attack strategies with which to tease the defending player. 19.95

Nukewar - Nuclear confrontation between two hypothetical countries. Defend your country with espionage, bombers, missiles, submarines, etc. 19.95

Computer Acquire - New Second Edition! The object is to become the wealthiest person in this "business" game - hotel acquisitions & mergers. 20.00

Andromeda Conquest - Vast scale space strategy game of galactic colonizing and conquest. Strange life forms & alien technologies - exciting! 19.95

Telengard - Microcomputer Dungeon Adventure game. Time fantasy and role-playing. 50 levels of ever-more complex mazes to explore & survive! 25.00

MORE — MORE — MORE

Shipping & Handling Charges:
First two (2) items - $2.00 per item.
Three (3) or more items - $1.00 per item.
For orders over $100 total, surface shipping will be paid by CompuSense. Blue Label or special handling will be paid by the customer.
Additional $2.00 C.O.D. fee on all C.O.D. orders.
MasterCard and Visa accepted. Give card number and expiration date on order form.
Allow three (3) weeks for personal checks.

TO ORDER:
P.O. Box 18765
Wichita, KS 67218
(316) 263-1095

WRITE FOR FREE CATALOG

Prices subject to change.

VIC-20® is a registered trademark of Commodore

C-64	**Compute**	$399.00
VIC-20®	**Personal Computer**	147.00
VIC-1515	**Printer**	334.95
VIC-1530	**Datasette**	67.50
VIC-1541	**Disk Drive**	347.00
VIC-1010	**Expansion Module**	139.95
VIC-1311	**Joystick**	9.95
VIC-1312	**Game Paddles**	19.95
	Telephone Modem	99.95
VIC-1210	**VIC 3K Memory Expander Cartridge** Plugs directly into the VIC's expansion port. Expands to 8K RAM total.	34.95
VIC-1110	**VIC 8K Memory Expander Cartridge** 8K RAM expansion cartridge plugs directly into the VIC.	52.50
CM101	**VIC 16K Memory Expander Cartridge**	99.95
CM102	**24K Memory Expander Cartridge**	119.95
VIC-1011A	**RS232C Terminal Interface** Provides interface between the VIC-20 and RS232 telecommunications modems. Connects to VIC's user port.	39.95
	PETSPEED - Basic Compiler for Commodore Compile any Pet Basic program. The only optimizing compiler. Programs compiled with Petspeed run up to 40 times faster. Petspeed code is unlistable and compiled programs cannot be tampered with. No security device required for compiled programs. Available NOW for the Commodore 64.	140.00
	Star Gemini 10 Printer	350.00
	Star Gemini 15 Printer	450.00
	SND Monitor	347.00
	16K Memory Expander	50.00

CARDCO

CARDBOARD 6 $87.50
An expansion interface for the VIC-20. Allows expansion to 40K or accepts up to six games. May be daisy chained for more versatility.

CARDBOARD 3 35.95
Economy expansion interface for the VIC-20.

CARD "?" CARD/PRINT 76.00
Universal Centronics Parallel Printer Interface for the VIC-20 or CBM-64. Use an Epson MX-80 or OKIDATA or TANDY or just about any other.

CARDETTE 30.95
Use any standard cassette player/recorder with your VIC-20 or CBM-64.

LIGHT PEN 29.95
A light pen with six good programs to use with your VIC-20 or CBM-64.

All CARDCO Products have a lifetime warranty.

HOME & BUSINESS PROGRAMS For VIC-20 & C-64

CW-107A **Home Calculation Program Pack** $48.95

CPV-31 **Data Files** - your storage is unlimited 14.95

CPV-96 **Household Finance Package** - to keep records of all your household expenses 30.95

CPV-208 **Bar-Chart** - display your numerical data 8.95

CH **Turtle Graphics** - learn programming 34.95

CH **VIC Forth** - is a powerful language for BASIC programming 49.95

CH **HES MON** - is a 6502 machine language monitor with a mini-assembler 34.95

CH **HES Writer** - time-saving word processing tool 34.95

CH **Encoder** - keep your personal records away from prying eyes 34.95

CT-21 **Statistics Sadistics** - statistical analysis 14.95

CT-121 **Total Time Manager 2.0** - creates personal or business schedules 15.95

CT-124 **Totl Label** - a mailing list and label program 13.95

CT-125 **Totl Text BASIC** 15.95

CT-126 **Research Assistant** - keep track of reference data 17.50

CT-140 **Totl Text Enhanced** 29.95

CM-152 **Grafix Designer** - design graphic characters 12.95

CQ-5 **Minimon** - allows you to program, load, save, or execute machine language programs 13.95

CT-3 **Order Tracker** 15.95

CT-4 **Business Inventory** - to maintain record of inventory 15.95

CS **Home Inventory** - lists your home belongings 17.95

CS **Check Minder** - (V-20 & 64) keep your checkbook the right way 14.95

CS **General Ledger** - a complete general ledger 19.95

CHC-504 **HES Writer** - word processor 39.95

CHC-503 **Turtle Graphics II** - utilizes the full graphics of your 64 49.95

CHC-502 **HESMON** - machine language monitor w/mini-assembler 34.95

CHP-102 **6502 Professional Development System** 29.95

CFC **Data Files** - a management program 27.95

CPV-327 **HESCOM** - transfers data and programs bidirectionally between VICs at three times the speed of a disk drive 40.95

CPV-328 **HESCOUNT** - monitors program execution 19.95

CHV **HESPLOT** - Hi-res graphics subroutines 12.95

CTV-367 **Conversions** - figures, volume, length, weight, area, and velocity to all possible configurations 7.95

CC **The Mail** - your complete mail program Cassette 24.95 / Disk 29.95

CPV-220 **Client Tickler** 16.95

CPV-221 **Club Lister** 13.95

CPV-224 **Depreciator** 9.95

CPV-236 **Investment Analyst** - keep track of investments and investment opportunities 12.95

CPV-251 **Present Value** 10.95

CPV-269 **Super Broker** 12.95

CPV-270 **Syndicator** - calculates whether to buy or sell 13.95

CPV-274 **Ticker Tape** - maintains investments profile 14.95

CPV-276 **Un-Word Processor** - screen editor 16.95

CPV-286 **Phone Directory** - never lose a phone number again 9.95

CS-111 **Checkbook** - home "utility" program 14.95

CPV-294 **Calendar My Appointments** - print a calendar for every month in any year 14.95

CPV-296 **The Budgeter** - place your personal finances in order 12.95

DEVELOP-20

FIVE POWERFUL SOFTWARE DEVELOPMENT TOOLS

Plus The Exciting New Book

INSIDE THE VIC

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 registered TM of Commodore Business Machines Inc.

Some Mixed-mode Graphics Subroutines In BASIC

by William R. Frenchu Princeton, NJ.

The C-64 Programmer's Reference Guide mentions (but doesn't give examples of) using the raster scan interrupt to mix bit-mapped and text modes on the screen. For machine language programmers this presents few problems, but for users with only BASIC experience it would often be advantageous to be able to present high resolution plots with text mode labels quickly and easily. This method would also side-step another difficulty of using the raster interrupt method to mix graphics, that is, the inability to mix graphic types on the same raster. For applications where a vertical axis must be plotted, this restriction eliminates much of the screen from containing text. The BASIC routines presented here enable beginning programmers to mix upper case (and graphic characters),lower case and bit-mapped graphics on the same line and screen with minimal effort.

ROUTINE 1 (Lines 10000-10290)

Routine 1 prints a string (including cursor, case and reverse control characters) starting at a given position while in the high resolution mode. It's capable of using any character set pointed to by variable B1. This allows the use of user defined character sets in addition to the ROM set defined by the C-64. User sets should be stored in the same order as the ROM set (64 character blocks of upper case, graphics, reversed upper case, reversed graphics, lower case, shifted lower case, reversed lower case, shifted reversed lower case) in order for the case and reverse control keys to retain their expected function. Using the ROM set requires the routine to temporarily disable the interrupts and 'switch in' the ROM while getting the character definitions. For this reason the stop key is disabled whenever a string is actually being printed. When a user character set is accessed, this restriction does not apply and the POKEs to locations 1 and 56334 may be eliminated.

Routine 1 makes use of several flags and variables set by the user. The lower case flag (L) is set to '1' when a string is to be printed using the lower case set and '0' otherwise. The reverse flag (R) performs a similar function for reverse on/off. Case and reverse can also be changed at any time from within a string by including the following special characters:

Change to upper case	(reversed shift 'n')
Change to lower case	CTRL-N
Turn reverse on	CTRL-9
Turn reverse off	CTRL-0

Thus, any string may contain characters from both upper and lower case character sets. The programmable function keys could be easily added to the decoding section (lines 10030-10130) to produce frequently used effects such as tabs, super- and sub-scripting.

The position of the printed string is determined by the variables X and Y. These refer to the usual cursor positions, X from 0 to 39 and Y from 0 to 24. B$ is the string to be printed. Strings may be up to 255 characters in length and the subroutine will automatically continue a string that is too long on the next line. Strings printed at the same X,Y positions will be 'overstruck' if flag 'O' is equal to '1' and replaced if 'O' is equal to '0'.

ROUTINE 2 (Lines 20000-20080)

This routine supports user input to the program. As it calls Routine 1, the X,Y,R

and L variables retain the same functions as above, but now B$ is the prompt string for the 'input statement'. Input is returned as a string, I$, and must be converted to a numerical value with the VAL function if necessary. The special characters from the decoding section of Routine 1 are returned in I$, but are not echoed on the screen during input. The final flag is BL which when set to '1' causes the prompt and input strings to be blanked after a return is received. If BL is '0' the prompt and input strings will remain on the screen. This routine also allows use of the DELETE key to correct errors in input. Do not use cursor controls to correct errors as these keys are returned in the string I$. The INSERT key is not supported.

ROUTINE 3 (Lines 30000-30400)

Routine 3 draws a line from pixel coordinates X1,Y1 to X2,Y2 where X ranges from 0 to 319 and Y from 0 to 199. The Reference Guide recommends always using points two pixels wide to decrease 'Chroma Noise'. This could be a simple modification (or two lines could be drawn side by side) but since it decreases resolution it hasn't been implemented here.

ROUTINE 4 (Lines 16 and 1000)

This is a 'one line' routine that turns on a given pixel X,Y as above. It is called by Routine 3. Line 16 sets up a table containing the powers of two from 7 to 0 for use by line 1000. This was done because exponentiation on the C-64 is very slow. (Integer exponentiation is even slower than floating point!) It was found that calculating the powers of two for each X,Y pair more than doubled the time necessary for plotting.

ROUTINE 5 (Line 1001)

The final routine turns off a given X,Y pixel. If called instead of Routine 4 (by Routine 3) lines may be 'unplotted', too.

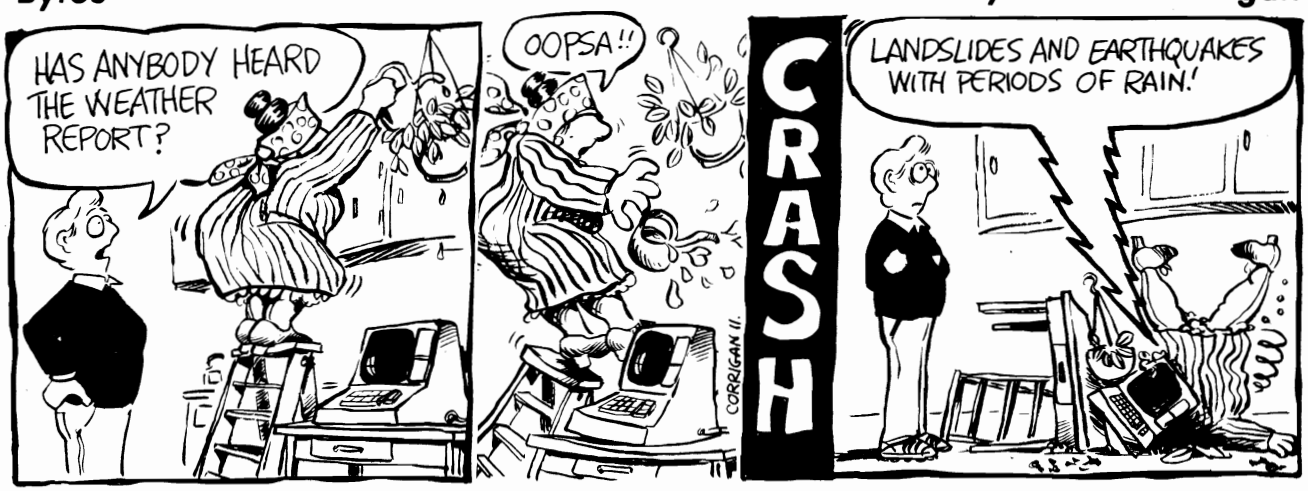
```
0 rem ***change screen color***
1 rem ***print wait message ***
2 poke 53280,11:poke 53281,0
4 print"[clearing high res screen . . "
6 print"[please wait 35 seconds"
7 print"[ commodore-64 hi-resolution demo "
8 rem *** clear hi-res screen ***
10 for i=8192 to 16192:poke i,0:next
13 rem *** set up powers of 2 table ***
14 rem ***    for routines 4 & 5    ***
16 for i=0 to 7:p(i)=2^(7-i):
   p1(i)=255-p(i):next
17 rem ***   start hi-res mode and   ***
18 rem *** set hi-res screen at 8192 ***
20 print"[":poke 53265,peek(53265)or32:
   poke 53272,peek(53272)or8
25 rem ***    set hi-res colors    ***
26 rem ***upper nybble for "1" bits***
27 rem ***lower nybble for "0" bits***
30 for i=1024 to 2023:poke i,192:next
100 rem ***print strings using***
101 rem ***      routine 1      ***
102 l=0:r=0:x=0:y=13:bl=53248:o=1:
    b$="-2[ ":gosub 10000
103 x=20:y=13:b$="0":gosub 10000
104 x=37:y=13:b$="[":
    gosub 10000
105 x=19:y=3:b$="F+1":gosub 10000
106 x=19:y=21:b$="D-1":gosub 10000
107 r=1:x=1:y=23:b$="[ commodore-64
    Hi-Resolution Demo [":gosub 10000
108 rem *** print axis using ***
109 rem ***    routine 3      ***
110 x1=0:y1=100:x2=319:y2=100:
    gosub 30000
113 rem *** print axis using ***
114 rem ***    routine 4      ***
115 for y=25 to 174:x=158:gosub 1000:
    x=157:gosub 1000:next
116 rem *** get user input with ***
117 rem ***      routine 2      ***
120 bl=1:x=0:y=0:r=0:b$="input
    period ? ":gosub 20000:j=val(i$)
125 rem *** plot sine curve ***
126 rem *** using routine 4 ***
130 for x=0 to 319:z=sin((x-158)/25*j):
    y=int(100-70*z*z*z):gosub 1000:next
131 rem *** label plot with input ***
132 rem ***    using routine 1    ***
133 l=1:r=0:x=4:y=1:b$="y=[SIN(3[
    (":gosub 10000
134 b$=i$:gosub 10000:b$="*x)":
    gosub 10000
137 rem *** pause loop: when "a"  ***
138 rem *** is received go back   ***
139 rem ***   to standard mode    ***
140 rem ***      and stop         ***
145 get a$:if a$="" then 145
```

```
150 poke 53265,peek(53265)and223:
    print"";:poke 53272,peek(53272)
    and21:end
982 :
984 :
990 rem ***      routines 4 & 5     ***
992 rem ***     for plotting and    ***
994 rem ***    unplotting points    ***
996 rem *** see ref. guide pg 125 ***
997 :
1000 b=int(y/8)*320+int(x/8)*8+(yand7)
     +8192:poke b,peek(b)orp(xand7):
     return
1001 b=int(y/8)*320+int(x/8)*8+(yand7)+
     8192:poke b,peek(b)andp1(xand7):
     return
9980 :
9982 :
9990 rem *** routine 1: for printing ***
9992 rem ***   strings in hi- res    ***
9993 :
9994 rem *** disable interrupts & ***
9996 rem ***  switch in char rom  ***
9998 rem *** calculate char base  ***
10000 poke 56334,peek(56334)and254:poke
     1,peek(1)and251:b2=b1+r*1024+1*2048
10010 rem ***  get a character  ***
10012 rem *** from input string ***
10020 for i=1 to len(b$):
      c=asc(mid$(b$,i,1))
10026 rem *** special characters ***
10028 rem ***  decoding section  ***
10030 if c=145 then y=y-1:next:
      return:rem ** cursor up **
10040 if c=17 then y=y+1:next:return:
      rem ** cursor down **
10050 if c=29 then x=x+1:next:return:
      rem ** cursor right **
10060 if c=157 then x=x-1:next:return:
      rem ** cursor left **
10070 if c=18 then r=1:b2=b1+1024+1
      *2048:next:return:rem **
      reverse on **
10080 if c=146 then r=0:b2=b1+1*2048:
      next:return:rem ** reverse off **
10090 if c=19 then x=0:y=0:next:return:
      rem ** cursor home **
10100 if c=14 then 1=1:b2=b1+r
      *1024+2048:next:return:rem
      ** start lower case **
10120 if c=142 then 1=0:b2=b1+r*1024:
      next:return:rem ** stop lower
      case **
10130 if c=255 then c=126:rem ** "π" is
      special case **
10132 rem ***  translate chr$ codes  ***
10134 rem *** to screen codes: chars ***
10136 rem *** patterns in rom stored ***
10138 rem ***    by screen code      ***
10140 on c/32+1 goto 10150,10200,10170,
      10160,10150,10170,10190,10170
10150 c=32:goto 10200
10160 c=c-32:goto 10200
10170 c=c-64:goto 10200
10180 c=c-96:goto 10200
10190 c=c-128
10192 rem *** calculate starting pos ***
10194 rem ***   for string and char  ***
10196 rem ***       definition      ***
10200 z=y*320+x*8+8192:c=c*8+b2
10220 rem *** poke definition into ***
10222 rem ***   hi-res location    ***
10240 for j=0 to 7:poke z+j,
      (o*peek(z+j))orpeek(c+j):
      next:x=x+1:next
10260 rem *** re-enable interrupts ***
10262 rem ***  and switch out rom  ***
10290 poke 1,peek(1)or4:poke 56334,
      peek(56334)or1:return
19880 :
19882 :
19900 rem *** routine 2-user
      input    ***
19901 :
19902 rem *** initialize input
      string ***
19904 rem *** save start position
      and ***
19906 rem ***    length of prompt
      ***
20000 i$="":hx=x:hy=y:hb=len(b$):
      gosub 10000
20008 rem *** get a char ***
20010 get b$:if b$="" then 20010
20014 rem *** check for special
      chars ***
20016 rem ***    only first two are
      ***
20018 rem ***different from routine 1 ***
20020 if b$=chr$(13) then 20070:
      rem *** return ***
20030 if b$=chr$(20) then 20045:
      rem *** delete ***
20031 if b$=chr$(145) then 20041
20032 if b$=chr$(17) then 20041
20033 if b$=chr$(29) then 20041
```

```
20034 if b$=chr$(157) then 20041
20035 if b$=chr$(18) then 20041
20036 if b$=chr$(146) then 20041
20037 if b$=chr$(19) then 20041
20038 if b$=chr$(14) then 20041
20039 if b$=chr$(142) then 20041
20040 rem *** echo character ***
20041 gosub 10000
20042 i$=i$+b$:goto 20010
20043 rem *** delete key:
      dont delete ***
20044 rem ***    if nothing there
      ***
20045 if len(i$)=0 then 20010
20046 rem *** move back and blank one
      ***
20047 rem ***    char; update input
      ***
20050 x=x-1:z=y*320+x*8+8192:
      for i=0 to 7:poke z+i,0:next:
      i$=left$(i$,len(i$)-1)
20058 rem *** get next char ***
20060 goto 20010
20066 rem *** blank input if desired ***
20068 rem ***      else return       ***
20070 if b1=0 then return
20074 rem *** starting address for  ***
20076 rem *** blanking and blanking ***
20078 rem ***          loop          ***
20080 z=320*hy+8*hx+8192:for i=0 to
```

```
29880 :
29882 :
29900 rem ***  routine 3     ***
29902 rem *** draw a line    ***
29903 :
29904 rem ***  calculate slope and  ***
29906 rem ***   decide whether to   ***
29908 rem ***    increment x or y   ***
30000 xd=x1-x2:yd=y1-y2
30010 ifxd=0then30200
30020 ifyd=0then30300
30030 m=yd/xd:s=y1-m*x1
30040 ifabs(m)<=.5then30400
30050 m=xd/yd:s=x1-m*y1
30060 rem *** calculate x  ***
30062 rem *** step along y ***
30100 fory=y1toy2stepsgn(y2-y1):
      x=m*y+s:gosub1000:next:return
30160 rem *** vertical line ***
30162 rem ***  step along y ***
30200 x=x1:fory=y1toy2stepsgn(y2-y1):
      gosub1000:next:return
30260 rem *** horizontal line ***
30262 rem ***   step along x  ***
30300 y=y1:forx=x1tox2stepsgn(x2-x1):
      gosub1000:next:return
30360 rem *** calculate y  ***
30362 rem *** step along x ***
30400 forx=x1tox2stepsgn(x2-x1):
      y=m*x+s:gosub1000:next:return
```

Bytes — ***by Patrick Corrigan***



GENERAL

Planning For Obsolescence

by Ron Kushnier

In a recent Compute article, Jim Butterfield matter-of-factly stated that, in his opinion, the VIC-20 will "fade away" in a few years. I have no argument with that statement. I too believe it. Yet, seeing it in print made my mind gasp. Things are moving so fast, products are going on and off the market at such a rapid rate, that it is not surprising the following event occurred at a recent computer club meeting.

I had brought in my original 8K PET which, I must say, is still in mint condition. A young member of the club came running up and exclaimed, "Boy, another new model! Commodore is really something! Look at that, a built-in cassette unit! What will they think of next?"

It broke my heart to tell him that what he was seeing was the great-grandfather of the present-day CBM computer.

But that's the way things are in this "Future Shock" world of micros.

How can we live with such goings on?

How can we decide when to buy and when to wait?

And, more important, how can we plan for obsolescence?

In the world of computers, obsolescence is a subjective term. My single board KIM (vintage 1976) would be considered by many as obsolete. Yet it still performs the same functions as it did back then. It has all the bells and whistles, all the options that were ever made, and has never had a failure. But try to find a buyer for it -- impossible!

My experience with "KIM" brings out three areas to consider when dealing with obsolescence.

The first question which must be asked is "For whom is the product obsolete?"

We can categorize buyers into three types.

There is the **"Applications Buyer"** .

This is a person who buys a computer with a particular application in mind, and who satisfactorily solves his problem with that computer. He certainly can not complain that his computer is obsolete.

There is the **"Ubiquitous Computer Buyer"**.

This person expects his computer to do everything from high density color graphics to 80-column word processing, all at super speed and precision. This type of person is apt to be disappointed and dissatisfied with any computer he buys. He will constantly be on the lookout for something newer or better.

Finally, there is the **"Computer Experimenter"**.

The "Computer Experimenter" is more fascinated by the idea of what a computer can do than actual applications. He is the guy involved in advancing the technology and probably accounts for most of the published computer articles.

The experimenter finds himself in an unfortunate situation. Unless he is independently wealthy, he can never keep up with the rapid changeover in equipment. To him, machines become obsolete before they've even had a chance to be fully explored.

Perhaps I should mention a fourth category of buyer, the **"New Educational Buyer"**. This person is just breaking into the computer field and is not sure what his needs will be. He usually settles for a low-end micro such as the VIC-20 or the Sinclair ZX-81. The small initial cost can be written off as an educational expense.

At first, the newcomer is usually ecstatic with his purchase. However, once the novelty and educational value have diminished, the "New Educational" buyer is reduced to one of the three previously-defined categories, and is faced with the same decisions.

Each of our three buyers has his own view and definition of "obsolete".

An example of how each would view the purchase of a VIC-20 might be enlightening.

The "Applications Buyer" probably saw the VIC as one of three possible Games machines, the others being the ATARI and INTELLIVISION. The VIC provided more flexibility at only a slightly higher price. So the purchase was made. He is satisfied

with his machine because it does everything that he expected it to do.

The other two buyers are not happy. They are satisfied with what the VIC is, but they are not satisfied with what it is not. They complain about slow tape speeds, lack of a "proper" amount of memory, and only a 22-column screen.

This leads us to the second area of concern and to another question.

Where are we headed?

In the "KIM" example, it was not until I had amassed a large amount of memory, an ASCII keyboard, and a huge assortment of other hardware and software, that I asked myself, "Where am I headed?". The answer was that I was heading toward a system that spoke "BASIC". Unfortunately, by the time I achieved that end, my "computer" covered an entire table, and had to be turned on through a complex procedure by three separate power supplies. My KIM was obsolete by now, at least to the buyers market, and all the money I had spent on "add-ons" was lost.

What I am proposing then is that you ask yourself that all-important question, now. If you are not happy with your system as it is, wouldn't it be better to trade up now while your present computer still has value? It seems foolish to me to start with the "add-ons" only to produce a bigger "obsolete" system a few years from now.

The third area of concern affects all of us buyers. This is the area of product discontinuance. Even our "Applicatations Buyer", snug and secure with his programs and machine, is shaken by this one.

Every year, new car models come on the scene. Yet, we can still get parts for a '57 Chevy or, for that matter, even a "Model T". But it seems that, once a computer model has been discontinued, it stands alone and unsupported. Resale values crash and it finally lands up in the back row of some computer Flea Market.

This should not be. Computer manufacturers have a responsibility to support their product for more than just the one year of its sales life. As I have tried to point out, obsolescence is a relative term. Old computers are not "dead". If they can do your job and meet your needs, then they are just as good as the new machines.

Now, this brings up another area dealing with "Software Obsolescence". Does the quality of a computer consist solely of its hardware? Or, is the merit of a computer system only dependent on the number of programs available to it? Obviously, it must be a combination of both. But, when the pendulum swings to one extreme or the other, it may mean the death of a particular micro.

The original PET had its hardware and firmware bugs. However, the users found ways around virtually all of them. Commodore's decision to throw out compatibility with their new operating system sounded the death knell for the 8K PET.

When the KIM was in its golden age, software abounded. Yet, when new systems became available, the amount of software flow decreased and finally trickled to zilch. The KIM ceased to be.

The "Computer Experimenter" can become involved in a concept I call Computer Mainstreaming". This is a negative feedback mechanism. If, over a several-month period, he sees a decrease in the number of published articles dealing with his particular computer, he immediately panics. He feels that he and his machine are no longer in the mainstream. It is, therefore, time to purchase the "new" leader. This, of course, does lead to fewer articles and the cycle continues.

Therefore, the computer magazines themselves have a hand in shaping product obsolescence.

Once upon a time, there was a company called Data General which, from the beginning of the mini-computer era, produced a hardware product that never changed. Oh sure, there were mods to the system and improvements, but software compatibility was strictly maintained. After many years, their hardware was considered obsolete by many. But a strange thing was happening. People continued to buy Data General. Why? Well, throughout the years, they had amassed such an overwhelming abundance of software that it seemed stupid to use anyone else.

In the Micro world, if one looks closely, one can see two philosophies emerging. Some companies put out a new product what seems to be once a month. Others stay with the old hardware as long as possible. An example of "Rapid Hardware

Inc." is ,of course, Commodore. Some slow movers are Apple and the AIM-65. Radio Shack, I would consider, is somewhere in the middle. It is a little too early to tell about ATARI, although, if one uses their game product as a basis, then they seem to be very stable.

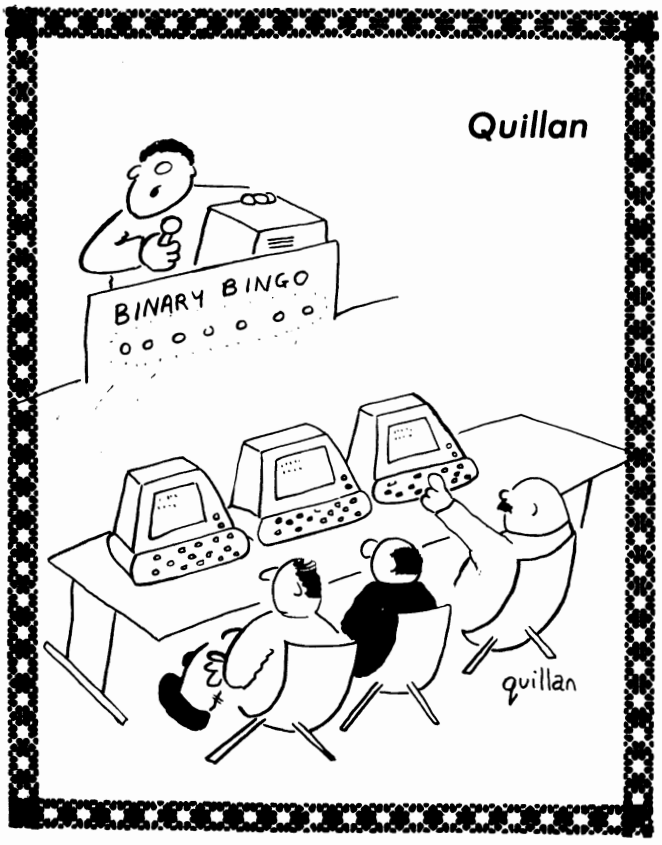
Texas Instruments is an interesting and unique example. They introduced their system early in the game. Yet, because of a lack of advertising, their high price and lack of software, their computer sat for years on the dealers' shelves. Now, T.I. appears to be making its move. The price dropped dramatically, software and firmware are becoming available, and Bill Cosby loves his Pudding Pops and the T.I. Computer. The hardware never changed, but the support made the product respectable.

What can we conclude then, when we must plan for obsolescence?

Well, to summarize:

Obsolescence is relative. It exists in the eye and the mind of the buyer.

If you are dissatisfied with the features of your present computer, trade up now.

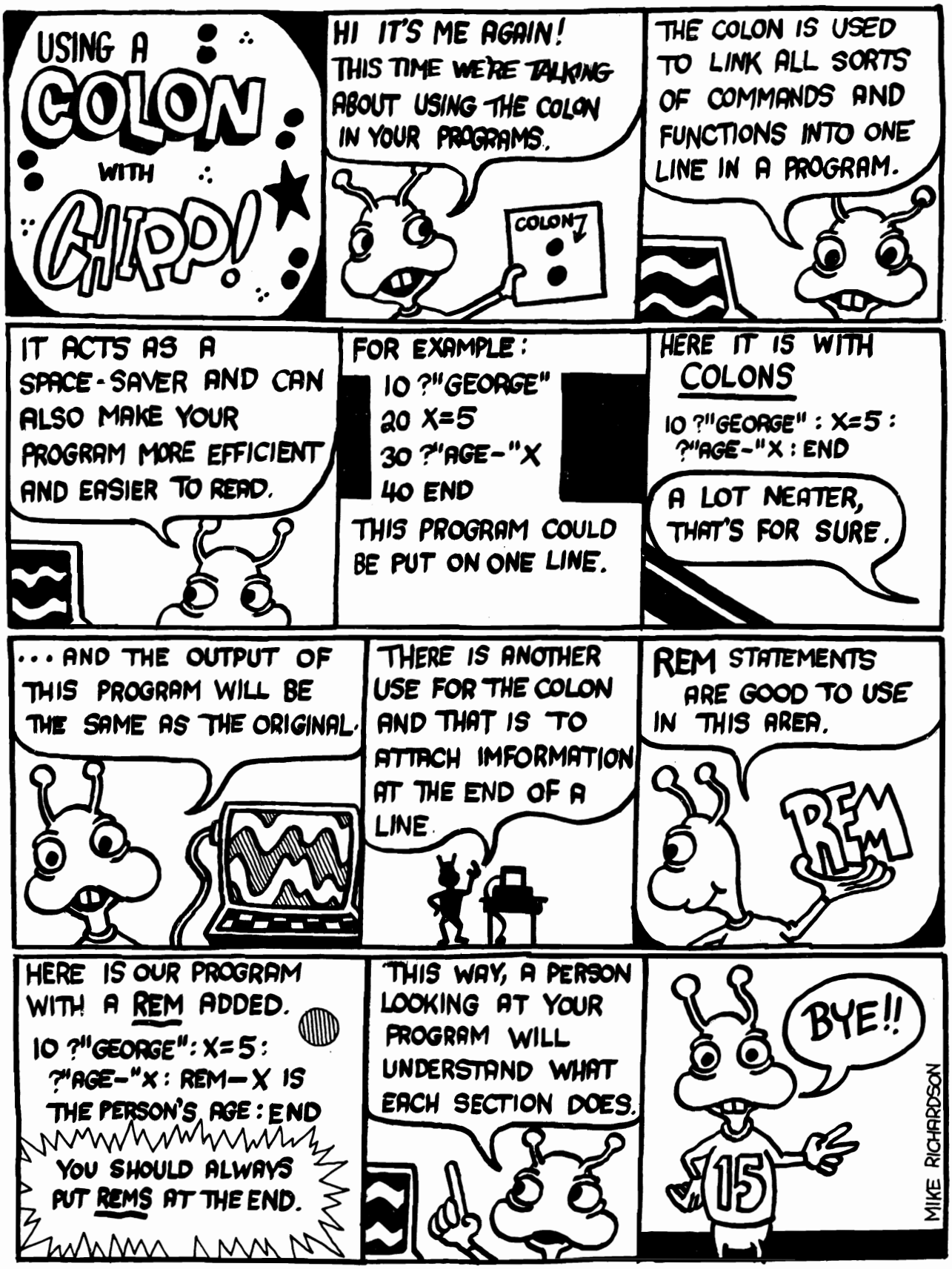


Don't spend money on add-ons which can never make your computer the machine you want it to be.

Don't buy a new computer just because it's new. Examine your needs and your computer's capability to see if they match.

To the "Computer Experimenter" -- Means must be found to fulfill your infinite curiosity without breaking your bank account. Writing articles and programs for profit is one way. Another avenue is to review new hardware and software for stores, customers or others who are willing to lend you the new systems. For that end, you get to play with the goodies and they receive valuable information.

The computer manufacturers must retain a parts supply for products they have produced for at least as long a period as other products on the market. This will ensure that both old and new computers can co-exist and provide years of valuable service to their users.


USING A COLON WITH CHIPP!
HI IT'S ME AGAIN! THIS TIME WE'RE TALKING ABOUT USING THE COLON IN YOUR PROGRAMS.
COLON?
THE COLON IS USED TO LINK ALL SORTS OF COMMANDS AND FUNCTIONS INTO ONE LINE IN A PROGRAM.
IT ACTS AS A SPACE-SAVER AND CAN ALSO MAKE YOUR PROGRAM MORE EFFICIENT AND EASIER TO READ.
FOR EXAMPLE:
10 ?"GEORGE"
20 X=5
30 ?"AGE-"X
40 END
THIS PROGRAM COULD BE PUT ON ONE LINE.
HERE IT IS WITH COLONS
10 ?"GEORGE" : X=5 : ?"AGE-"X : END
A LOT NEATER, THAT'S FOR SURE.
... AND THE OUTPUT OF THIS PROGRAM WILL BE THE SAME AS THE ORIGINAL.
THERE IS ANOTHER USE FOR THE COLON AND THAT IS TO ATTACH IMFORMATION AT THE END OF A LINE.
REM STATEMENTS ARE GOOD TO USE IN THIS AREA.
REM
HERE IS OUR PROGRAM WITH A REM ADDED.
10 ?"GEORGE": X=5: ?"AGE-"X: REM-X IS THE PERSON'S AGE: END
YOU SHOULD ALWAYS PUT REMS AT THE END.
THIS WAY, A PERSON LOOKING AT YOUR PROGRAM WILL UNDERSTAND WHAT EACH SECTION DOES.
BYE!!
15
MIKE RICHARDSON

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.)

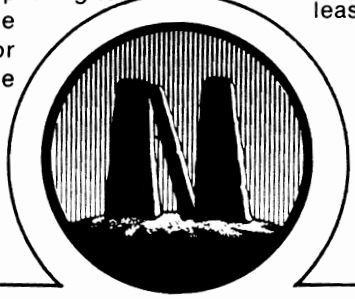
(CG068) The Catch . . . $12.95
Another all machine language game based on the principle that one person with one joystick guiding one catch/shield can catch everything that one alien can throw at one. The action comes slowly at first but by the fourth wave you'll be aware of . . . "The Catch" . . .

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 over-head 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.

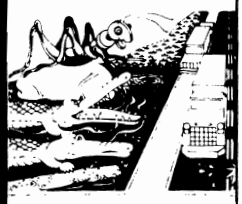
NÜFEKOP

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

VIC is a trademark of Commodore Business Machines, Inc. Games will be on tape unless you request disk. Ask for our FREE catalog!

Write For Free Vic Catalog **NEW** *Write For Free 64 Catalog*

VIC™ Software VIC™



CRICKET

CRICKET - From the company that brought you Asteroidz, Munchman and a host of other blockbusters. We now present CRICKET. This is a challenging game with a cast of characters you will love and hate. All you have to do is get Cherp from one side of the road to the river and then across the river. Not so fast though. First you have to figure out how to dodge the traffic and get to the center. Then how are you going to get across the river? Look here comes a log — even a turtle. Hitch a ride across the river and jump from one to the other. Keep a sharp eye out for Ade the Gator. He loves to have crickets for lunch. How many times can you get across the road and river. You will have to work as fast as you can. Time limit and bonus. You will find this game addictive and challenging and it will entertain you with hours of fun and enjoyment. **$14.95**

BUG BLAST

BUG BLAST - If you think Centipede was fun — look out for BUG BLAST. A new and fast action arcade game with realistic smooth action, quality hi-res graphics and trouble. Its very calm as the first wave attacks. Only a few bugs to kill. Just shoot thru the cactus and wipe them out. After a few attacks you feel you have everything under control. Now the attacks really start. Those protection areas have to go. Blast away. Will they ever stop? OK — the BUGS got me this time. Now its my turn, Just one more time — BUG BLAST — Now its your turn to get even. **$14.95**

PARATROOPER

BOMB'S AWAY - Can you stop him? The crazy bomber drops the bombs from the top of the screen. You get 3 buckets to catch them. Before you know it bombs are falling so fast you wonder when he will stop. Just when you think you have him under control your bucket gets smaller. Is your hand quicker than your eye? **Special $9.95**

PARATROOPER - You are the only one left to stop them. The sky is full of enemy choppers. Paratroopers keep dropping into your area with non-stop barrage of enemy troops. They are out to destroy you. This new game is an unbeatable blend of arcade action and quick thinking strategy. You must make every shot count — don't be to fast on the trigger. Every time you hit a chopper or paratrooper you get extra points. Wait until you see the climax of this game — you won't believe it! This is a multiple skill level game with razor-sharp graphics and sound. **$19.95**

TARGET COMMAND

MOW - Get ready for the fast and furious action of the craziest mower you have ever seen. How much grass can you cut? Joystick moves your mower around as fast as you dare. Watch out for granny's dafodils and grandpa's radio antenna. **$12.95**

MOW

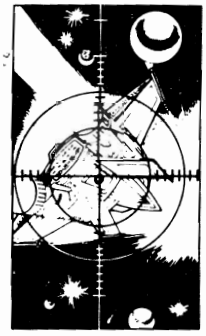
COSMIC CRUZER - Bring the coin-op game into your VIC. 3 Scenarios. Your Cruzer moves over a mountainous landscape & into a tunnel of surface - to - air missle, silos and ground - to - air weapons. If you can make it in and out of the tunnel you fly into the asteroid field. Drop bombs and fire missiles at the fuel dumps to keep your fuel supply up. If you are really good you can get to the base and try, to destroy it. We don't know of any one that has hit the base yet. Maybe you will be the 1st. Cosmic Cruzer is a fun filled magnificently rendered home video game that will last for months of challenge. Highly addicting. Hi-Res Graphics, Color & Sound. **$19.95**

COSMIC CRUZER

SPACE PAK - Can you survive? 3 space games with the sights and sounds of arcade games. The excitement builds as the action is un-ending. Blast away at everything in sight. The alien attacks will stop at nothing to destroy you. Prepare for battle, there is no escape, unless you can help. Can you survive? Hi-Res, Color, Graphic & Sound. Joystick or keyboard. **$19.95**

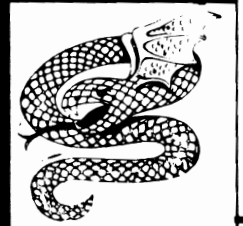
ALIEN INVASION

ALIEN INVASION - Invaders from space are attacking your home planet. Hurry and man your lasers and prepare your robot forces for the inevitable attack of the Alien Invaders. The excitment builds as you command a battery of missile bases in a bunker. Each invader has a laser aimed right at you. Will they ever stop. Only you can save the Galaxy. You can compete with 4 people in the solar system. There are 20 levels of play. If you destroy the Aliens in the correct order you will receive bonus points. Can you get the top score? **$12.95**



SPACE PAK

TARGET COMMAND - The whole West Coast is being bombarded and only you can save it. You are at the controls of the missile launcher and hold the destiny of our country in your hands. It takes a cool head, not hand and fast reflexes to zap those missiles right out of the air. Get ready to pulverize — atomize and vaporize them. Oh, my God, those warheads are heading right for our ammo dumps. They are everywhere. NO ONE CAN SAVE US — EXCEPT YOU. You must move your laser into position and fire as fast as you dare. Time limit with arcade style excitement. Protect your ammo at all costs. 10 levels of play. **$12.95**



SNAKE OUT

SNAKEOUT - Slip your snake into position and score by chomping the blocks. Watch the way you slither because your escape routes get smaller. 2 Bonus games included. **$12.95**

HEAD ON

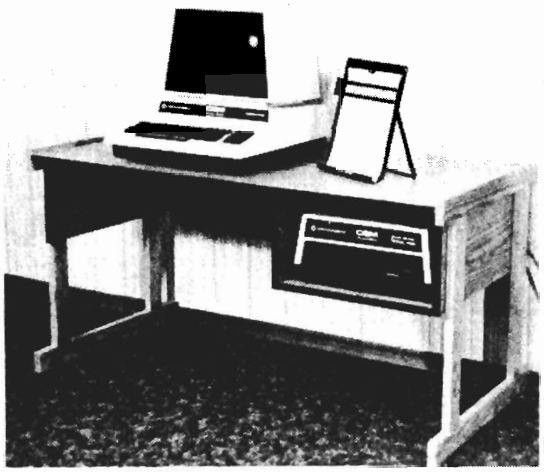
HEAD-ON - Please do not buy this game if you are the type that says "I'll play it just one more time". Players have been known to start playing HEAD ON at 8:30 p.m. and at 2 a.m., wonder where the time went? Have you ever tried to explain to someone why you played a game for five and a half hours. We know of no remedy for the addiction to HEAD ON except to beat the VIC on level 9. No one has done it, YET, will you? We think not. Move your car as fast as you can dare around the tracks. You get 3 cars and MUST avoid the computer car. Points for the most dots covered. Bonus cars, nine levels of play. **$12.95**

ComputerMat • Box 1664K • Lake Havasu City, Arizona 86403

DESKS

NEW GENERATION TOYS

- *Attractive*
- *Low Cost*
- *Sturdy*
- *Convenient*



send for your free catalog

NEW GENERATION TOYS INC.
290 Larkin Street
Buffalo, N.Y. 14210 (716) 854-1164

NEW GENERATION TOYS
Chatsworth, Ont. N0H 1G0
(519) 794-3463

RETAIL PRICES

Item	*Stock No.*	*U.S.*	*Canada*
Desk as Shown	100	$200.00	$250.00
Single Computer Desk	110	$133.00	$166.00
Printer Table	120	$120.00	$150.00
Paper Dolly	130	$ 80.00	$ 99.00

DEALER INQUIRIES INVITED

NEW

Basic Utility for the Commodore 64

POWER 64

- **easy to learn**
- **easy to use**
- **program faster and more efficiently with better results**
- **MOREPOWER included FREE**

Powerful Programmer's Utility
by Brad Templeton
Manual by Jim Butterfield

$99.95 from your local Commodore dealer.
For your nearest dealer call:

(416) 273-6350

PRO·LINE SOFTWARE

755 THE QUEENSWAY EAST, UNIT 8
MISSISSAUGA, ONTARIO L4Y 4C5

MAILORDER VIC-20 SOFTWARE
HOME FINANCE - GAMES - EDUCATIONAL

SOFTWARE HOUSE

4630 Dufferin Street, #309, Toronto, Ont. 416-663-6401

We stock a large selection of arcade style and adventure games, home finance and educational software from Abacus, Acadamy, MIS, Microspec, Nufekop, TIS, Victory Software, and many more.

Call or write for our free catalogue

DISTRIBUTORS FOR:

Eastern House Software
Rabbit (VIC & C64) - High speed cassette SAVE & LOAD. 8k in 30 sec $69.95

Intelligent Software
Word Processor - Powerful, yet easy to use $39.95*
Inventory - Inventory control system for small business $39.95*
Copycalc Electronic Spreadsheet $29.95*
Baseball Manager - Keeps team and player stats $29.95*

Coach Educational Series
Keyboard Coach - Typing Tutor and practice exercises $18.95
Capitals Coach — Learn the capitals of the World $18.95*

P.R. Software - Simulative Strategy Games
Dungeons of Kal/Star Defender - Fantasy adventure & Starship attack $22.95
Convoy Raider/Convoy Escort - Both sides of the battle of wits $22.95
At the Track/Boxer's Corner - Bet on the race & manage your boxer $22.95
Football - Select the teams from NFL/AFL roster & call the plays $19.95*
Dungeons of Kal - Expanded semi-graphic fantasy adventure $19.95*
Baseball - Select your team from league players & manage the team $16.95

+ Also available for Atari 400/800 computer
* Needs VIC + 8k min

DEALER ENQUIRIES INVITED

COMMODORE-64

Software Review

by Ravi Palepu
Sydney, Nova Scotia

FROGGEE:
Commercial DATA SYSTEMS
730 Eastwood Ave.
Regina, Saskatchewan Canada
S4N 0A2

$29.00 CDN

Some say this version is better than the Arcade. The frog really looks like a frog. The cars, trucks, racing cars and vans look like they are supposed to look. The colors make a beautiful blend. This game is easy at first, but becomes harder at later stages, but also, it can be easily mastered. It gets boring once you master it. To make 1000000 it would take approximately two hours and forty-five minutes. The frog looks good and the game doesn't put too much stress on your hand using the joystick.

RATE: From 1 to 10 - 9.2

ANNITHILLATOR: (ANNIHILATOR?-ed)
Victory Software
7 Valley Brook Road Paoli, PA 19301
U.S.A. 215-644-7227

$19.95 U.S.

This game ranks similar to Defender. Good graphics and sound. There are only two bad points about this game. These are that the aliens move too quickly and that using your joystick puts some stress on your hand. There are good graphics when your ship explodes. Each stage has a different color for the border and the ground. This game also brings a lot of tension. Highly recommended.

RATE: From 1 to 10 - 9.6

ARCADE PAK (3 games)
COMPUTER MAT
Box 1664 Lake Havasu City
Arizona 86403 U.S.A.

$24.95 U.S.

Head On: A dot game with two racing cars, yours and the computer's. It is hard to get into the little slots using the joystick. It is not that addicting and can get boring, but is fun in later stages. Using the keyboard, the game is much easier.

Alien Invasion: A space game. The best game of the three. Each alien has a name and a different figure. A nice game with cute graphics and sound. It is a lot easier to use the keyboard instead of the joystick. When using the joystick, it is hard to move one place only right or left.

Target Command: A lesser version of Missile Command. This game has okay graphics and sound. It is hard on your hand using the joystick. Its ranks vary to the player.

RATE: From 1 to 10 - 7.5 (3 games)
Head On -- 7.5
Alien Invasion -- 7.8
Target Command -- 7.3

GRAVE ROBBERS
Victory Software
$13.95 U.S.

An adventure with graphics and also sound, very cute graphics and sound. A good adventure for any beginner up to an intermediate. Easy to get into jams-- a very good adventure to get. It also ranks differently to the user.

RATE: From 1 to 5 - 3

ADVENTURE PAK (3 adventures)
Victory Software
7 Valley Brook Rd.
Paoli, PA 19301
U.S.A.
$19.95 U.S.

Jack and the Beanstock: -- Not too boring and is an okay adventure for beginners.

Computer Adventure: A fun adventure with some addiction.

Moon Base Alpha: The hardest one of the three. A good pak of adventure, but it is overpriced.

RATE: From 1 to 5 - 2.9

ADVENTURE PACK 2 (3 Adventures)
Victory Software
$19.95 U.S.

African Escape: A good adventure for an intermediate. Easy to get into a jam.

Hospital Adventure: A fun adventure. Not too hard to finish.

Bomb Threat: A good adventure for anyone. It doesn't get boring. This pak is also overpriced.

RATE: From 1 to 5 - 3

Butterfield Box

by Jim Butterfield — Toronto, Ont.

You Can't Get Away From It...

It was the fall of 1971. Vicki and I were travelling to Istanbul on the Orient Express, and decided to stop overnight in Sofia. The city was crowded, so we stayed in a small motel on the outskirts of town, complete with strolling musicians and a dancing bear.

(Haven't you always wanted to start a story with a paragraph like that?)

In the morning, we telephoned to say hello to a chance acquaintance we had met briefly the year before. He wasn't in his office; but an hour later he pulled up at the motel with his car and considerable baggage.

"Where are you going?", he asked. When we said we were continuing on to Istanbul, he exclaimed "Come with me!", loaded our baggage into his car, and set off into open country. "We're going to Plovdiv," he explained, "To the famous World's Fair of Trade. You've heard of it, of course?"

We hadn't. In fact, I wasn't sure exactly where Plovdiv was ... or why we were on our way there.

"The fair opens tomorrow," our host happily informed us, "but I must be there to set up a display next to the U.S. Pavilion." This baffled me even more, since he was definitely not American. I asked for an explanation.

"I am with Shipka", he announced, "which is the Bureau of Representations. Bulgaria is a communist country, of course, so we don't have private enterprise companies here. Once in a while, we need foreign products, and then we appoint government employees to represent the companies that make them. I represent an American manufacturer."

I had to ask the next question: what company was that? "It is called I.B.M.", he beamed. "Perhaps you have heard of them?"

I allowed as how I had indeed heard of them. "Good," he beamed, "Their employee here is a Canadian from New York. Perhaps you know him?". I didn't recognize the name.

We got to Plovdiv, entered the exhibition grounds, and drove to the U.S. Pavilion, which was a geodesic dome. (I swear: I'm not making this up. I can't help it if it sounds unlikely). We entered the adjacent building, where the IBM exhibit was located.

Vicki and I were introduced to the U.S. co-ordinator, who was indeed Canadian (originally from Montreal) and really did live in New York. He seemed puzzled to see two Canadians arrive as guests of a Bulgarian government representative. "Are you from the embassy?", he asked. We said no. It was really impossible to explain how we got there; we didn't even try to do so.

I could tell he still suspected that we must be VIPs of some sort. Especially when he said, "I brought a bottle of Canadian Club over with me. Would you like to try a little?" I decided not to try to disillusion him; I said, "Sure!"

He was setting up a system 360 model 40 for display at the exhibition; it was being put through its installation diagnostics. Now, the 360 is a nice machine, but it doesn't have a lot of glitter and pizazz. I couldn't see it being a dramatic world's fair exhibit doing payrolls or statistics. So I asked: what would visitors see?

The IBM man almost blushed. "Ahhh... we're printing pictures on our line printer of Giorgi Dimitrov, a Bulgarian national hero ... ahh .. and, er, Lenin," he said. He looked at me speculatively. You don't think anyone back in the States would be upset by that, do you?" I didn't think so ... especially if he didn't tell them.

The day wore on, and it started to rain torrentially. We stayed in the U.S. pavilion, and sampled a little more of the Canadian Club. Then the roof started to leak. Rain started to drip, then pour into the computer area. There was a mad scramble to put plastic over everything.

The fair director was called in. Photographs were taken. It was decided to put an extra roof on our building ... on top of the existing one. The IBM man looked on glumly. "That's all I need," he said. "Now the roof will be too heavy and the whole thing will collapse."

But it didn't. As evening approached, we picked up a ride to the Plovdiv railway station, and continued our journey on the Orient Express.

And I sometimes reflect: Could I ever have imagined, when I went plunging into the heart of the Balkans, that I would spend a day drinking Canadian whiskey and watching a computer being protected against a cloudburst?

Truth is stranger than fiction.

MAKING FRIENDS WITH SID

by Paul Higginbottom Toronto, Ont.

Part II

Hello again. Last issue, we got acquainted with some of the various terms and parts of the SID chip. We're now equipped to learn some more advanced things about the synthesizer, as well as more advanced techniques to fully utilize what we have already learned, to, for example, produce more than one note simultaneously, and then, to create software that can play actual pieces of music.

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

1) Define the capabilities of SID

2) Explain some software techniques to make SID do perform.

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

In the last article, the various parameters of a voice were outlined, except for the filter in the SID. Essentially, the filter will (as is implied) filter the sound output from any of the voices in a number of ways. The actual term 'filtering' means that the sound is changed by quietening the voice to varying degrees above, below or around a given 'CUTOFF' frequency. However, don't worry about understanding this concept fully yet, since this issue won't use the filtering capabilities of the SID. I simply wanted to make you aware of this feature in the SID, so you won't be taken by surprise in the future!

To begin with, 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 GOTO 80
110 POKE SID+4,32
120 GOTO 80
```

Explanation of program:

Line 10 declares the variable SID to the start location of the SID chip.

Line 20 POKE's all the SID locations with a zero to initialise 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 0.

Line 60 declares the variable KEY to the zero page memory location which holds the keyboard matrix number of the current key depressed, or 64 if no key.

Line 70 sets SID register 1 to 16, which sets the high order byte of the frequency of voice 1 (therefore, frequency of voice 1 =16*256 [see last article for explanation of 'low' and 'high' bytes]).

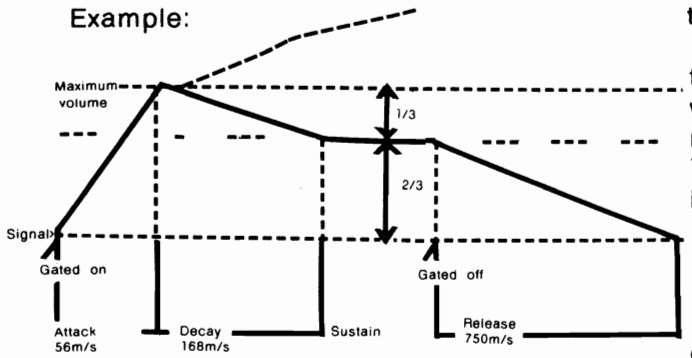
Line 80 waits for a key, by GETting a keypress from the keyboard, and IF the keypress is a null, i.e., no key has been pressed, the program will GOTO the same line and keep waiting.

Line 90 sets SID register 4 with 33, which gates voice 1 on with a triangular waveform (see last article for explanation [33 32+1]).

Line 100 checks to see if a key is still depressed (as with a piano), and if it is (i.e., location KEY is still something other than 64), the program will GOTO the same line and check again.
Line 110 sets SID register 4 with 32, which gates voice 1 off now that no key is depressed on the keyboard, still with 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).

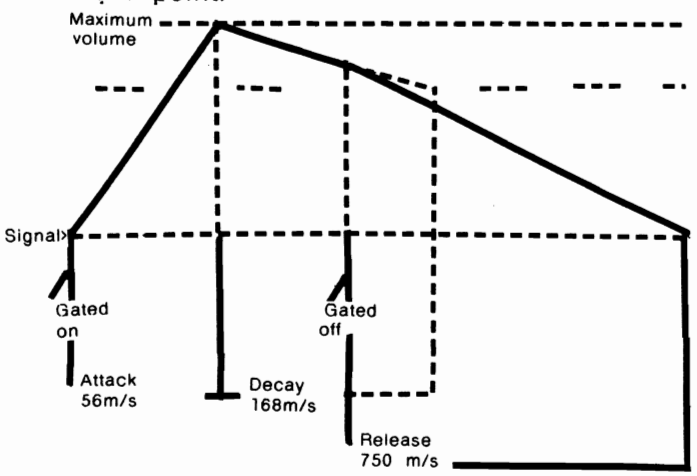
Something which ought to be understood here, is the fact that when a voice is gated off, i.e., released, the envelope RELEASEs from WHEREVER it had reached. Probably a diagram would be the best way to show this:

Example:



ATTACK =5, i.e, 56 milliseconds.
DECAY =5, i.e, 168 milliseconds.
SUSTAIN =10 i.e., two thirds of maximum volume
RELEASE =8 i.e., 750 milliseconds.

Now using the same parameters, only gating the voice off (i.e, releasing) at a different point:



Here, it can be seen that the voice was gated off (i.e, released), before the envelope had decayed to its sustained level, but that when the voice was released, it simply released from the point it had reached.

This relates to the program you just entered, because it uses this fact to simulate the 'feel' of a piano keyboard. I.e., 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), giving 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 will begin from wherever the envelope currently is (i.e., the current output volume). 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 this time and hold it down, you'll hear the volume rise (attack), and then fade some (decay) to a constant level (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, and that is because the release occured (i.e., by you releasing the key), before the envelope had reached either its maximum, or the sustained level of volume. Similarly, 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 away to, and if you keep depressing and releasing the key, and ensure that 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 be useful if we could 'play' the Commodore 64's synthesizer too, 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 12 semitones of the top octave, we could generate the values of all the lower notes by continually dividing all 12 by 2 to derive the pitch of the semitones in the next octave lower. It is not necessary to go into the math here, but if the ratio between octaves is 2:1, then the ratio between semi-tones is 2 (1/12):1.

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 former by a constant which is derived from the frequency of the internal clock in the SID chip, and the system clock.

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

Therefore:

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

Which turns out to:

Fundamental Frequency =Frequency x 16.404 (approx.)

Therefore, middle A which is 440Hz, would be:

=440x16.404
=7217 (approx.)

'A' in the next octave up would be 880Hz, or 880x16.404 =14435 in the SID chip.
The maximum value in the SID chip is 65535 (255 in both the low and high byte), therefore, by doubling again...
1760 (1760x16.404=28871)
and again...

3520 (3520x16.404=57742)

57742 is fairly near the top end of the SID frequency value range, and doubling once more would push it beyond it, so we will base our frenquency range around 3520Hz. To create a 2 dimensional array (subscripts being 'octave', and 'semitone') 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:sustain0: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)/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 k1 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 the next and subsequent articles will slowly become impossible to follow. Have fun.

COMMODORE-64

Paul Higginbottom Making Friends With SID.

C-64 Chapter Meeting News

Greetings to all you C-64 users out there. After a busy few months we have finally decided to show our faces in THE TORPET.

We now have a management team consisting of:

Mike Hyszka, C-64 General Co-ordinator 1-416-249-5805

Jerry Field, C-64 Communications Co-ordinator 1-416-284-0658

Dave Glostein, C-64 Special Services Co-ordinator 1-416-633-5220

David & Richard Bradley, C-64 Librarians 1-416-782-9252 or 1-416-782-7320

If you have any questions relating to the C-64 group feel free to phone any of these members.

Now for our meeting dates. Our main meetings are held at Earl Haig Secondary School, 100 Kenneth Ave, in the Yonge and Sheppard area.

The dates are: May 16 at 7:30
June 19 at 7:30

Summer Programme

After much pondering and soul-searching your VIC 20 and C-64 co-ordinators (Dave Simpson and Mike Hyszka) have arranged for summer meetings in July and August at York Public Library (Main Branch), located at 1745 Eglinton Ave, West (corner of Northcliffe Blvd.). The four scheduled meeting dates are:

July 4th August 2nd
July 18th August 22nd

All meetings start at 7:30 p.m. and end at 10:30 p.m.

Jerry Field, our Communications Co-ordinator is the developer of the questionnaire that was distributed at the May 10th meeting. Jerry has analyzed the information with a view to determining the meeting format for the summer sessions and for the future. The results of the questionnaire and the format of the pre-registration will be presented at the C-64 meeting on June 14th.

It is imperative that those interested in the summer sessions attend on June 14th to be acquainted with the procedures. If this is not possible members are advised to get in touch with any of the management team.

David and Richard Bradley have been busy developing our C-64 library. At the present time we have 7 complete disks. The Bradley's have indicated that there are a fair number in the process of being developed. If you have any public domain software you wish to contrubte please get it to them. The Bradley's would also appreciate any help to assist them in their project work.

Finally, an appeal--we would appreciate any technical assistance, teaching presentations and general help from the membership at large. The group is expanding and we need more support staff.

Keep on Byte-ing

Mike Hyszka
C-64 Co-ordinator

Programmers Do It In Software

by Hal Chamberlin Raleigh, NC

Part 2

Digital Audio Fundamentals

In Hi-Fi magazines today, "digital audio" is the current rage just like quadraphonics was ten years ago. A digital audio recording system is simply an analog-to-digital converter (ADC) which converts the incoming audio waveform into a string of numbers, a digital mass storage device such as a tape or disk drive, and a DAC for playing the numbers back. (see figure 2). What is actually being done in a software synthesis system is to tap into this chain in the middle somewhere with a computer. For the "real time" system that will be described, the tap is just before the DAC where the computer and suitable software generates the numbers for the DAC rather than a mass storage device. The tap can also be made **before** the mass storage device which yields a "delayed playback" system. Since a digital audio record/playback system can obviously handle **any** kind of sound, it is apparent that, at least theoretically, a software synthesis system can synthesize **any** kind of sound.

In a digital audio system, two operating parameters work together to determine the system's sound quality. Obviously the speed at which numbers are sent to the DAC determines how much detail in the reproduction of fast waveform wiggles that may be achieved. In audio terms, it determines the **high frequency response**. Theoretically, frequencies up to 1/2 of the "sample rate", which is the rate at which numbers (samples) are sent to the DAC in samples per second, may be reproduced. Thus, if 20,000 samples are sent to the DAC every second, frequencies up to 10KHz may be reproduced. You should note that just above 1/2 of the sample rate lies a considerable quantity of distortion frequencies which must be filtered out by the low pass filter mentioned earlier. It takes a very sharp filter to separate the desired signal from the distortion but this sharpness requirement is relaxed somewhat if frequencies up to only about 40 percent of the sample rate are attempted. One interesting fact is that a digital audio system has no low frequency limit (it goes down to OHz) so bass response is excellent regardless of the sample rate.

The other operating parameter is the amount of round-off error in the DAC and the samples sent to it. Such round-off error gives rise to a general background noise level at all frequencies and therefore cannot be filtered out. Usually the DAC itself sets the system's numerical precision which is measured in bits. An "8-bit DAC for example accepts 8 bit samples which have an inherent round-off error of + or - 1/512 or + or -0.2 percent. The noise level associated with this amount of error is approximately 20*LOG10(1/512)+6 decibels

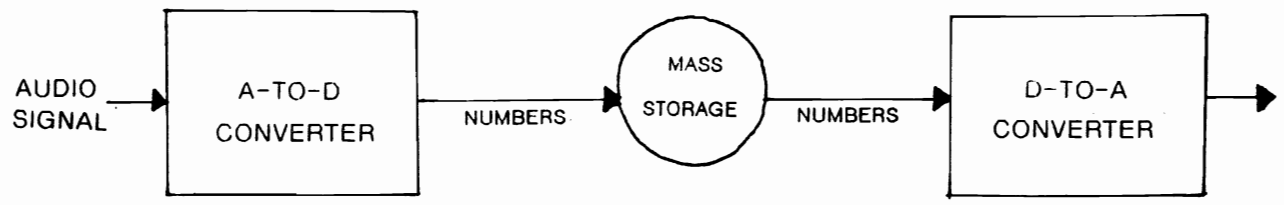
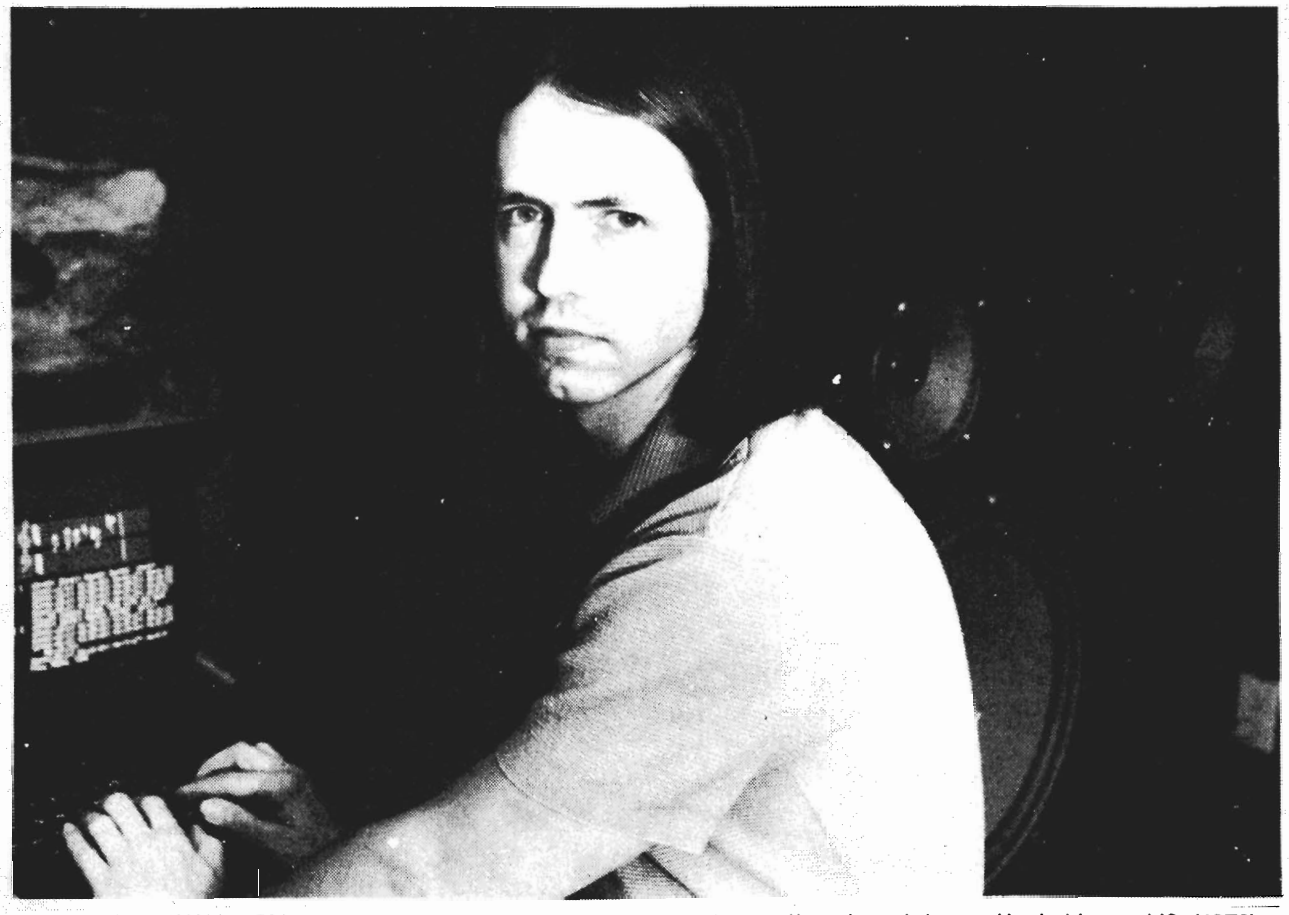


Figure 2: Digital Audio Recording System

or -48dB. A 12 bit DAC yields -72dB and a 16 bitter, which is typically used in professional digital audio applications, gives -96dB. In more familiar terms, 8 bits is about the noise level of a cheap cassette recorder, 12 bits is about the noise level of a first rate consumer audio system, and 16 bit noise is difficult to even measure. Note that while the bit precision of the DAC determines the ultimate minimum noise level, if the calculations that produce the samples are sloppy, they may introduce excess noise.

Software music systems on 1MHz 6502 microprocessors generally use 8 bit samples and sample rates between 8 and 9KHz. The DAC precision of course is set by the system's 8 bit word size while the sample rate is determined by how fast existing music programs can calculate the samples for an acceptable number of voices. The 6502 actually does this sort of thing very well; it would take a 5MHz Z-80 (with no memory wait states) to perform as well. In a delayed playback system in which samples are written to a mass storage device instead of being output immediately, the DAC precision, sample rate, and number of voices can be increased to professional levels since computation speed is no longer a limiting factor. However in such a system you may have to wait hours for the program to compute just a few minutes of sound.



HAL CHAMBERLIN: This series by Hal Chamberlin will be continued next issue. He holds an MS (1973) degree in Electrical Engineering from North Carolina State University. While in school he worked part time for IBM in speech recognition and synthesis research. He also had the opportunity during this time to use a signal processing computer for music generation experiments using his own programs.

He is an active speaker at computer shows on the topics of computer graphics and music and is the author of numerous articles on computer music synthesis and microprocessor circuit design. He has recently written a comprehensive book titled musical Applications of Microprocessors which is published by Hayden Book Company and has enjoyed uniformly favorable reviews and strong sales.

He is presently single and lives in the country near Wake Forest, North Carolina.

COMMODORE 64

SO MUCH MORE FOR YOUR 64

NEW 8.4 VERSION OF THE PET™ LOADER/EMULATOR CASSETTE

- Automatically resets colors. No poking necessary.
- Loads into upper memory in just 80 seconds.
- Runs all CBM public domain software.

Expand Your 64 Library

Run Thousands of **PET** Programs

COMMODORE 64

COMMODORE 64

The Pet Loader/Emulator reconfigures the Commodore 64 memory regardless of **any** recent ROM change to duplicate the PET internally without interfering with BASIC user memory.

SO MUCH FOR SO LITTLE

JUST $19.95 prepaid

The Education Circuit, Inc.
P.O. Box 333, Landing, NJ 07850

Dealer Inquiries Invited—201-398-6185

Pet is a trademark of Commodore Business Machines

COMMODORE 64

NEW COMMODORE 64

Software &Accessories. We now have one of the largest selections of 64 software and accessories

BUSINESS

PCS—6480 80—column board with top—rated word processor, includes data base and spreadsheet all linked together package price.....**$395.00**

Wordpac — Word processor.....**$99.95**
Inquire Pac - Data base.....**$99.95**
Calc Pac - Spreadsheet.....**$99.95**
File Pac - Mailing list.....**$79.95**
Account Pac - Home/Bus.....**$69.95**
General Ledger.....**$95.00**

EDUCATIONAL

64 BASIC Tutorial.....**$39.95**
64 Tour - overview.....**$15.95**
Happy Tutor - Typing.....**$29.95**

GAMES

Cyclons - Best seller game.....**$39.95**
Skiman - Slalom game.....**$29.95**
Sluggo — Boxing game.....**$29.95**
Casino Pac - 3 games.....**$39.95**

UTILITIES

Editor Pac - Prog. aid.....**$89.95**
Assembler Pac - Pro. aid.....**$89.95**
PET Speed 64 - Compiler.....**$199.00**
PET Speed 4000/8000 compiler.....**$249.00**

ACCESSORIES

C—64 80-Column board.....**$279.00**
RS232 Printer Intrerface.....**$129.00**
Parallel Printer Interface.....**$129.00**
Z—80 Board run 40 and 80 column CP/M.....**TO $399.00**
VIC 20/64 Interface to IEEE & RS232, totally transparent Interpod.....**$245.00**
VIC 80-Cloumn board.....**$179.00**
VIC 16K Memory Expansion.....**$129.00**
VIC/64 Switch.....**$259.00**

MODEM FOR VIC20/64

Accoustic coupled; one cable to computer supplies all signals and power; capable of 110 to 300 baud, full/half duplex - originate. Assembled & tested, with software, but without case.
.....**only $129.00**

ORDER INFORMATION

All prices in Canadian dollars, available from your Commodore dealer, or if not, send cheque or money order (include 7%sales tax, Ont. Res. Only) plus $350 for shipping.

DEALER INQURIES INVITED

COMPUTER WORKSHOPS LTD.

465 King St. E. Unit 9
Toronto, Ontario M5A 1L6

PHONE (416) 366-6192

CBM INTERFACES

The Connecting Links

Increase Your Computer's Ability

CBM PRINTER ADAPTERS
—addressable-switch selectable upper/lower, lower/upper case
—works with BASIC, WORDPRO, VISICALC and other software
—IEEE card edge connector for connecting disks and other peripherals to the PET•
—power from printer unless otherwise noted

RS-232 SERIAL ADAPTER — baud rates to 9600 — power supply included
MODEL-ADA 1450a $149.00

CENTRONICS/NEC PARALLEL ADAPTER — Centronics 36 pin ribbon connector — handles graphics
MODEL-ADA 1800..... $129.00

COMMODORE 64™ to RS-232 CABLE ADAPTER
—plugs into RS-232 port — provides voltage conversions to drive standard RS-232 printers, terminals and mainframes — 6 foot cable included — receives power from computer — uses address #2
—electronics fully enclosed — case 2¾ x 2 inches
MODEL ADA 6410F
(Female Connector)..... $79.00
MODEL ADA 6410M
(Male Connector)..... $79.00
MODEL ADA 64115
Modem Cable..... $79.00

COMMUNICATIONS ADAPTER — serial & parallel ports — true ASCII conversion — baud rates to 9600 — half or full duplex — X-ON, X-OFF — selectable carriage return delay — 32 character buffer — centronics compatible — power supply included
MODEL SADI..... $295.00

ANALOG TO DIGITAL CONVERTER — 16 channels — 0 to 5.12 volt input voltage range — resolution 20 millivolts per count — conversion time less than 100 microseconds per channel
MODEL-PETSET1 ..... $295.00

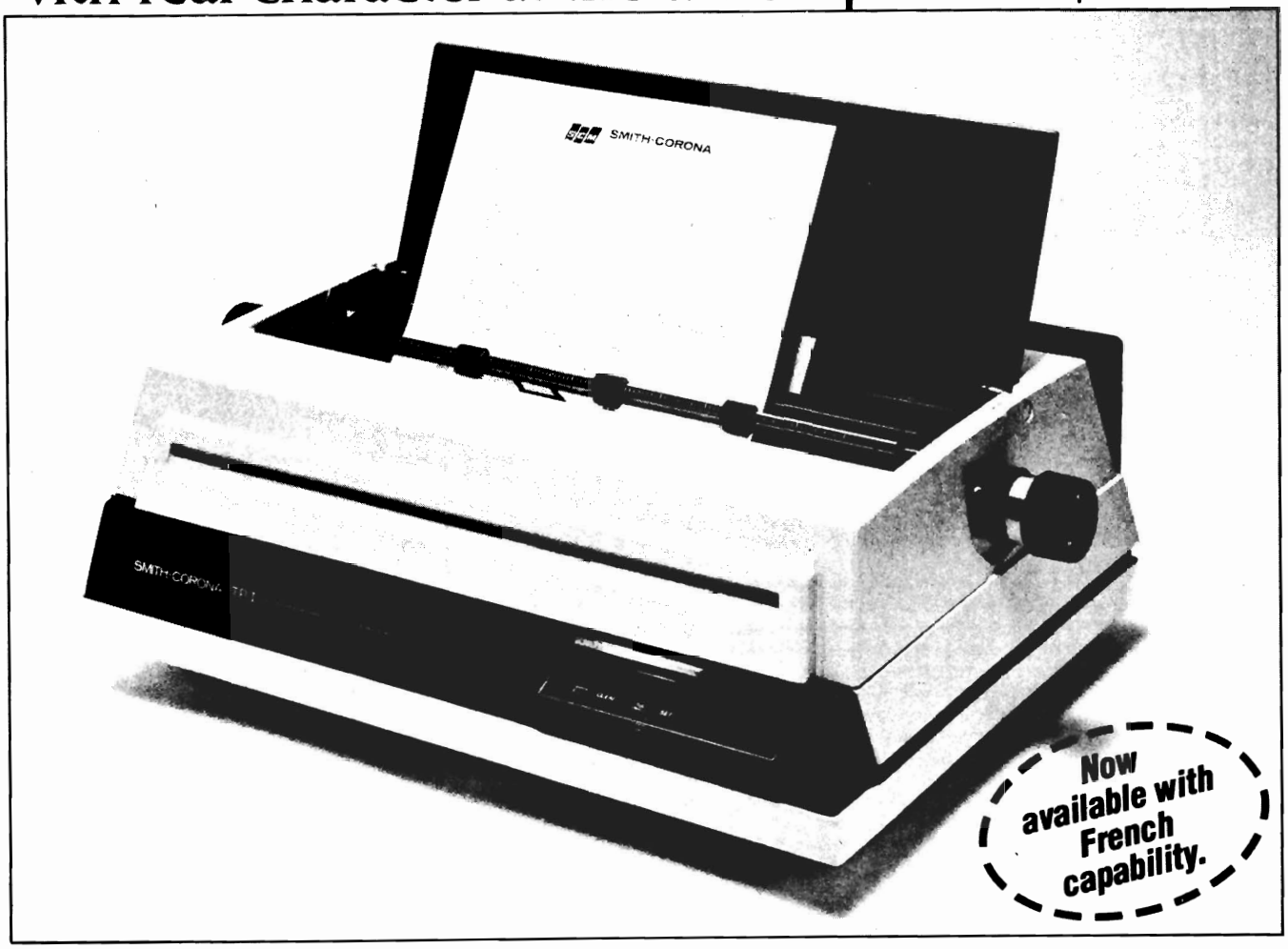
US Dollars Quoted
$5.00 Shipping & Handling
MASTERCHARGE / VISA

IN THE USA order from:
Connecticut microComputer, Inc.
36 Del Mar Drive
Brookfield, CT 06804
(203) 775-4595 TWX: 710 456-0052

IN CANADA order from:
Batteries Included, Ltd.
186 Queen Street West
F6 Toronto, Canada M5V 1S1
(416) 596-1405

Dealer Inquiries Invited

Smith-Corona introduces the first printer with real character at the unreal price of $1095.*



The Smith-Corona Daisy Wheel Printer

Until now, if you wanted to include a reasonably-priced printer as part of your computer or word processing system, you had to use a dot matrix printer. Daisy wheel printers were just too expensive.

Not anymore. Now Smith-Corona* offers a daisy wheel printer at such an incredibly low price, you can't afford *not* to include it. That means that even the smallest installation or business can now have letter quality printing capabilities at every work station.

The Smith-Corona printer operates with micro-processor-controlled daisy wheel technology, and is available with industry standard serial or parallel data interfaces.

Best of all, it produces results identical to those of our very finest office typewriters – printing with real character. So it can be used to create letters or documents that have to look perfect. As well as financial statements, inventory reports, direct mail campaigns – anything that requires quality printing.

And it's easy to use – just turn on the power, load the paper and away it goes. (It works equally beautifully with letterhead bond or fanfold paper.) There are drop-in ribbon cassettes and a choice of easy-to-change, snap-on daisy print wheels for a variety of fonts.

So why not get your hands on a real bargain: letter-perfect printing at an amazingly low price. Because, thanks to Smith-Corona, a printer with real character is no longer expensive.

Ask for it by name. **Smith-Corona**

DIVISION OF SCM® (CANADA) LIMITED

*suggested retail price

Please send me more information on the Smith-Corona daisy wheel printer.

Name_____

Title_____

Company Name_____

Business Address_____

City_____ Prov._____ Postal Code_____

Type of Business_____

Mail coupon or call:
Education Director,
Smith-Corona
29 Gervais Drive, Don Mills, Ontario M3C 1Z1
(416) 449-0164

T

Making Your VIC Programs Run With Any Memory Configuration

by Gottfried R. Walter Proton Station, Ont.

You may have noticed that many of the programs in the TPUG's VIC software library run only on VICs that have the same memory configuration that the VIC they were written on had. This occurs because, as you add more memory to your VIC, the screen and color memory locations 'move' around.

There are two memory set-ups that can occur :

(A) With the standard VIC (and with the VIC with only a 3K memory expander) the screen memory is located at 7680 ($1E00 in hex) and the color memory is located at 38400 ($9600 in hex).

(B) When an 8K (or more) memory expander is in place the screen memory moves to 4096 ($1000 in hex) and the color memory moves to 37888 ($9400 in hex).

This causes problems when a program written on a VIC with set-up (A) is loaded and run on a VIC with set-up (B) or vice versa - if any peeks or pokes to the screen memory or color memory occur.

If no peeks or pokes to the screen memory or color memory occur, and the program is entirely in BASIC, then the program should run correctly on any VIC, no matter what size of memory expander it has, if it even has one (provided of course that there is enough memory to run the program). If there were any peeks or pokes in your program, it could conceivably poke your space invaders (or whatever) into the same memory locations where your program is residing. This is not very good for your program.

Happily, it is very easy to make any programs you write for the VIC compatible with VICs with other memory set-ups. The process of making a totally BASIC program work on any VIC with enough memory is a simple one - simple if the programmer puts it in before he/she writes the program. (When the TPUG librarians get a program which will not run on every VIC with enough memory, it is unlikely that they go through the program and fix it, because they don't have the time. So it is best that you, the programmer, do this beforehand).

What you would do is assign variables to represent the start of screen memory and the start of color memory. All pokes and peeks to the screen and color memory locations would then be offsets from the variables you have set-up.

This routine is fairly short and easy to implement. This is the method that I use.

```
10SC=PEEK(648)*256:PRINT"[home]";:
CO=PEEK(244)*256:LL=22:NL=23:NC=NL*LL
```

Naturally, it is necessary to make sure that these variables are not used elsewhere in your program. If they are, you must change either these set-up variables or the variables in your program.

In case you do not immediately see how I set each variable up with the correct value I will explain exactly what is being done.

SC =start of screen memory

Location 648 holds the high byte of the start of the screen memory location, so to get the start of screen memory, I just multiply the contents of location 648 by 256.

CO =start of color memory

The ' ?"[home]"; ' puts the cursor at the top left of the screen memory. It also puts the current color memory location pointer to the top left of the color memory. This pointer is in locations 243-244 (low byte, high byte order). To get the start of color memory I just have to multiply the contents of location 244 (the high byte of the start of color memory) by 256.

LL =line length

This is set to the VIC default of 22 characters per line.

NL =number of lines per screen

This is set to the VIC default of 23

lines per screen.

NC =number of characters in one screen

This is set to the VIC default of 506 (NL*LL) characters per screen.

The reason that I include line length, number of lines per screen, and number of characters per screen in my set-up routine is a rational one. There are expanders out that will give the VIC 40 or even 80 columns, instead of the normal 22 columns. This means that checks should also be put into this routine that would see if the 40/80 column expander board for the VIC is in place and change the variables NL,LL and NC accordingly.

Since I do not have access to them I could not come up with a set-up routine that would handle them. (If any one out there has one of these expansion units and knows how to find out the number of lines per screen and the line length from within a program I would greatly appreciate this information being sent to me).

Here is an example program using my routine which will demonstrate how these variables are used.

```
10 SC=PEEK(648)*256:PRINT"(home)"::
   CO=PEEK(244)*256:LL=22:NL=23:
     NC=NL*LL
20 PRINT "[clr]":: REM CLEAR SCREEN
30 FOR A=0 TO LL-1 STEP 1
40 POKE SC+A,160 : REM POKES REVERSE
   SPACE TO TOP LINE OF SCREEN
50 POKE CO+A,6 : REM MAKES SPACE BLUE
60 POKE SC+A+(NL-1)*LL,160 : REM POKES
REVERSE SPACE TO BOTTOM LINE OF SCREEN
70 POKE CO+A+(NL-1)*LL,6 : REM MAKES
   SPACE BLUE
80 NEXT A
90 FOR A=0 TO (NL-1)*LL STEP LL
100 POKE SC+A,160 : REM POKES REVERSE
SPACE TO RIGHT COLUMN OF SCREEN
110 POKE CO+A,6 : REM MAKES SPACE BLUE
120 POKE SC+A+LL-1,160 : REM POKES
REVERSE SPACE TO LEFT COLUMN OF SCREEN
130 POKE CO+A+LL-1,6 : REM MAKES
    SPACE BLUE
140 NEXT A
150 FOR A=1 TO NC/3
160 OFFSET =RND(1)*NC: REM CALCULATE
THE OFFSET FROM SCREEN BEGINNING
170 IF PEEK(SC+OFFSET)<>32 THEN 160 :
    REM THERE IS SOMETHING THERE
180 POKE SC+OFFSET,65 : POKE
    CL+OFFSET,2 : REM RED SPADES
190 NEXT A
200 END
```

What this does is it first sets up the beginning of the screen memory, color memory, line length, number of lines, and number of characters per screen. Then, after it has cleared the screen, it puts an outline around the screen using blue bars. Then, using the variable NC (number of characters per screen), it fills 1/3 of the screen with randomly positioned red spades.

This shows you how you would do offsets that would let you poke to the screen and color memory. The offset is (naturaly) from the beginning of the screen, (ie. to poke a [shift A =spade character] into the fourth column of the second row the offset would be : 3+(1*NL*LL). (ie. (column#-1) + (row#-1) x NL x LL). This would then be added to SC to get the actual screen memory address and to CO to get the actual color memory address).

I hope this article has been of assistance to you and that you will now be able to write your programs so that they will run on every VIC.

Generating Random Numbers in Machine Language

by Vince Sorensen Regina , Sask.

One of the most difficult problems to tackle is to find how random, or illogical, numbers can be generated in a totally logical machine. Commodore BASIC solves this by making lists of "random" numbers using logarithms and subtraction. Since this is not a completely random process, the same list is generated each time the computer is turned on. Thus if you PRINT RND(1) after turning on your computer today, it will show the same number it did last time you did this. Try doing this. Write down the number, turn your computer off and on, wait a while, and PRINT RND(1).

Since I have a VIC, I was able to try plugging in extra memory to see if that made any difference. It didn't, so I was able to conclude that the process of generating random numbers is independent from the amount of memory, and the timer (because of the previous test). Random numbers are therefore found using logic. However, each machine has its own list, different from any other's, but the same as its own. Compare your first RND(1) with mine: .185564016. A RND(1) is a serial number for your VIC, PET, or C-64.

At this point, you may say that if there are no true random numbers, then how can there be events that will take a user by surprise in SPACE KONG? There are ways of getting a more "random" random number, and easily. The key is the argument for RND, the argument being the number in brackets after RND. If the number is positive, BASIC will reference another list if asked. If the number is negative, BASIC will rescramble the lists. The most random number is a RND(RND(-TI), because TI is always changing, giving a different base for the lists to be scrambled on. With it, random numbers are based upon logic, and the timer, which is now randomly referenced to.

Now, the reason we'd like to use random numbers is so that things are not too predictable. Where would arcade games be if the fifth invader from the left always fired when it was above the second bunker? They'd be boring! Random numbers give the Commodore computer that element of unexpectedness that makes us humans so interesting.

The question is: How do you combine machine language speed with random numbers? The easy way is to use BASIC's own routine. BASIC is only slow in translation, so it will still be fast.

The RND routine is located at $E094. At the start of this routine, there are checks for the sign of the argument, and which list is referenced. I'll separate these subroutines for our use right now. JSR $E09B will generate a random number between 128 and 255, and place it in location $62(see Program 1). JSR $EOBB will do the same thing, except it will assume you want the lists scrambled first, thus giving you a more "random" random. It simulates what happens when a negative argument is found.

Now that you have a random number, you'll probably want to generate odds, such as a one in four chance of a bomb dropping from your COSMIC EAGLE. After you get your random number, AND it with a bit pattern, and compare to get the results. In program 2, there are four possible outcomes, after the AND. They are the numbers zero to three. Comparing the outcome with one of the possibilities will give you the odds one in four or three in four.

The odds will always be calculated this way: You have X chances in 2^n, where X is the number of comparisons, and N is the number of bits on in your bit pattern mentioned above. Some examples for finding N:

AND #$02 =%0010 ...there are 2^1 or 2 possibilities.
AND #$06 =%0110 ...there are 2^2 or 4 possibilities.
AND #$OE =%1110 ...there are 2^3 or 8 possibilities.
Your next question is probably: What if you want odds out of a number that is not a power of 2? Program 3 finds four possibilities, rejects one, leaving you with odds out of three.

If you want odds higher than one in 128, just store the first random number, and generate another. If the first is say 128, and the second meets further conditions like the ones outlined in previous paragraphs, then you will have your "one in 129-256" possibility. Usually, odds between 1 and 128 will be enough, but using this technique, you can get odds so high that your longshot horse will come in once in a billion years.

I hope I've helped you with a problem that has puzzled me for quite a while. Here are the example programs: (in BL format, for the VIC)

PROGRAM ONE

```
LO1 JSR $E09B  Get a random number,
               have it placed in $62
LO2 LDA #$00   Print out as an integer
               between 128 and 255.
LO3 LDX $62    See TORPET No. 17, "Non-
               Kernal Routines in the
               VIC-20" by Thomas Henry.
LO4 JSR $DDCD
LO5 LDA #$20 Load accumulator with ASCII
               value of a space
LO6 JSR $FFD2  Print it.
LO7 RTS        Finished.
```

PROGRAM TWO

```
LO1 JSR $E09B  Get a random number,
               have it placed in $62
LO2 LDA $62 Load the accumulator with the
               random number
LO3 AND #$03 AND it with three (%0011)
LO4 CMP #$03   Is it the possibility #3?
LO5 BEQ LO7    Yes...
LO6 RTS No:    We're done.
LO7 TAX        Print out the three,
LO8 LDA #$00 using the PRLINE routine
               explained in TORPET #17.
LO9 JSR $DDCD
L10 RTS        We're done.
```

PROGRAM THREE

```
LO1 JSR $E09B  Get a random number,
               placed in $62 again.
LO2 LDA $62    Put it in the accumulator.
LO3 AND #$03   Four possibilities.
LO4 CMP #$03 Is it the fourth possibility?
LO5 BEQ L1 Yes:Get another random number
LO6 TAX It's one of three possibilities, so
               print out which one it is
LO7 LDA #$00   using PRLINE at $DDCD
LO8 JSR $DDCD  again.
LO9 RTS        We're done.
```

PROGRAM 4

```
LO1 JSR $E098 Commodore VIC Random
              Generation routine
LO2 LDA $62 Get number produce by
              above routine
LO3 AND #$1F Next highest exponent minus
              one(than below number)

LO4 CMP #$15 Number of random inte-
              gers you want (A)
LO5 BCS L01 Reject extra numbers
LO6 ADC #$3A Add lowest number wanted
              (B)
LO7 RTS       Routine finished
```

In brackets after certain lines above, a letter appears. It represents the same number that it would represent in the BASIC formula: R=INT(A*RND(1)+B)

The explanation for the program is as follows:

In Program 4, a random number between 58 and 79 is generated. In BASIC, you would ask for INT(21*RND(1)+58). These numbers are used in the example routine, to load the accumulator with the desired random number. The program is self-explantory, but I'll enlarge on line 3. The number of random possibilities you want is 21. 2 4 is 16, so it isn't large enough, but 2 5 is 32, and is large enough to contain at least 21 possibilities. We therefore generate 32 numbers, and reject those above and equal to 21 (giving us the numbers between 0 and 20 inclusive). Then we add the 58, and VOILA! we have our random number between 58 and 79.

The Line-Number Speed Fallacy

by David Williams Toronto, Ont.

Almost anyone who has become reasonably fluent in programming in Commodore BASIC has also learned a few tricks which are supposed to increase the speed with which programs will run. We all know, for example, that integer variables are (surprisingly) slower to process than real-value ones, that extensive string manipulations are liable to lead to a process called "garbage collection", which is (except in BASIC 4.0) very time-consuming, and that subroutines are usually accessed faster if they are placed near the beginning of a program than if they are at its end.

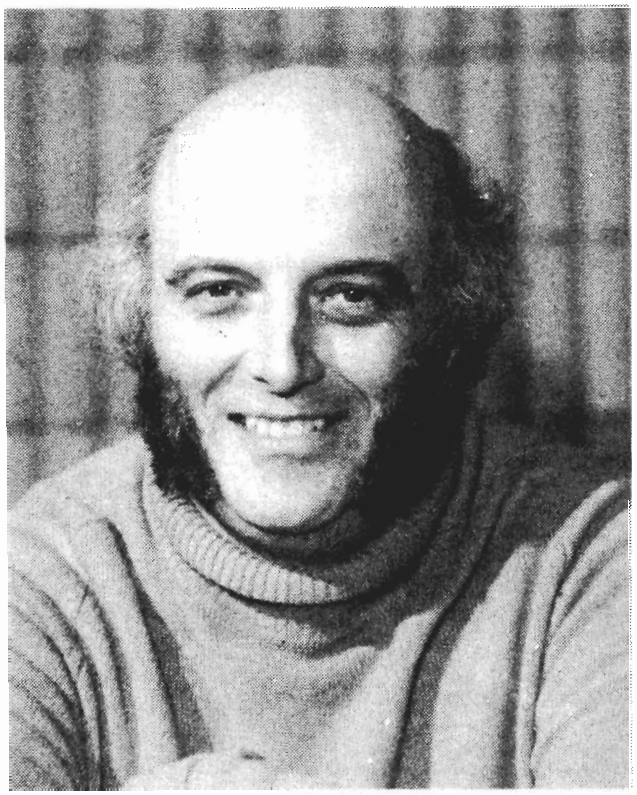
We also know - or think we know - that programs will run faster if they are renumbered so that their line numbers are as low as possible. This is because instructions such as GOTO and GOSUB are stored in program memory with the destination line number in the form of a string of ASCII digits. This has to be converted into the computer's internal binary notation before the machine can proceed with the instruction. A five-digit line number, for example, takes longer to convert than a two-digit one, so programs with low line numbers should run faster than those with high numbers.

This piece of programming lore has one major defect. It isn't true! If you have access to a line-renumbering utility, try the following little experiment. Take any BASIC program which takes a noticeable amount of time to process information, as opposed to waiting for user INPUT, etc. Add a couple of lines to it so that it will print out the time it takes to do this processing. The timing variable, TI, can, of course, be used for this. Renumber the program, starting at line zero and incrementing by one. This gives it the lowest possible set of line numbers. Run the program, making a note of any inputs you may have to give it, and observe the time it takes. Now renumber it again, starting at line zero and incrementing by 300 (yes, three hundred). If, by any chance, this leads to the end of the program having illegally high line numbers (64000 or more), use the highest increment which will keep the numbers within bounds. Run the program again, giving it exactly the same inputs as you did before. Almost certainly you will find, as I have done with this experiment, that the speed of the program is increased a few percent by giving it the higher line numbers!

It is true that high line numbers take longer to translate into binary notation than do low ones, and it is possible to write programs which will demonstrate this by running faster with low line numbers than with high ones. But there is another effect, which I will describe below, which acts in the opposite direction. In most "normal" programs, which have not been deliberately set up to demonstrate one effect or the other, speed is optimized by using a wide "spread" of line numbers rather than by minimizing them.

Translating the number of the target line into binary notation is only the first step by which a GOTO (or similar) instruction is executed. The computer then has to search through the program to find the line. Because of the way programs are stored in memory, it is not practicable (it would be possible but very slow) to search "backwards" to find a line which is earlier in the program than the one which is currently being executed. All searches are done "forwards", starting either at the current line, or from the start of the program.

There is a curious flaw in Commodore BASIC which affects the decision as to whether to search from the current line or from the first line of the program. Logically, one might expect that the numbers of the target and current lines would be compared. If the target line is later in the



David Williams

program than the current one, the search would proceed from the current line. If the target line is earlier than the current one, the search would begin at the start of the program. In essence, this is indeed what is done. However, the two line numbers are not compared exactly! When they are expressed in binary notation, each of them occupies two bytes of memory. One, known as the high-byte, contains the integer quotient which would be obtained by dividing the line number by 256 (2 to the power 8). The other, low-byte, contains the remainder from this division. To compare the two line numbers exactly, both the bytes would have to be compared, which would be easy enough to do. However, only the high-bytes are compared. If the high-byte of the line number of the target line is greater than the high-byte of the number of the current line, the search for the target line starts from the current one. In ALL other cases, BASIC goes back to the beginning of the program to search for the new line.

This little approximation in the BASIC interpreter cannot be called a real "bug", since it never causes programs to misbehave, however, it can have significant effects on their execution times. Consider, for example, what happens if a program which has 256 lines or fewer is renumbered to start at line zero and increment by one. In binary notation, every line number has a high-byte of zero. Every time a GOTO or similar instruction is executed, the comparison of the high-bytes fails to find the target-line high-byte greater than that of the current line. Thus every search for a new line has to start at the beginning of the program, even in cases when the target-line is later than the current line in the program. This often leads to many lines being searched through unnecessarily, wasting time.

By way of contrast, consider a program which is numbered in increments of 256 or more (this is why I suggested 300 earlier). In this case each line has a unique high-byte, different from all others. Comparing the high-bytes of two line numbers in this program can determine with certainty which line is later than the other. Thus forward GOTO's, GOSUB's, IF ... THEN (line number) instructions etc. always work in the most econimical way, without searching through the earlier part of the program. This can save a significant amount of time in the execution of a program.

CHAPUT!

They are usually very user friendly, but they do have their limits!

CLUB ACTIVITIES

TPUG LIBRARY NAMING CONVENTIONS APR/83
by
Mike Donegan, Toronto, Canada

Anyone who has been involved with microcomputers in the last year or two, has experienced the confusing problem of getting a disk of programs from a friend or club and trying to decide what he got. Frequently the name of the program is an obscure reference to the original application, that only the author knows. By the time you receive the copy, it has been modified many times and is very cryptic.

I have lived with this problem for a few years. I lived in western Canada and regularly traded, mooched, begged, and borrowed programs from all I met. My personal collection soon grew to many disks. The problem was, that when I went to get a program, I spent many hours sorting through unrelated versions. Either I couldn't remember the special name that it had, or it was mixed up with versions written for a configuration that I didn't own.

NEW LIBRARIANS

As one of the new librarians for TPUG the problem has escalated. The club has 68 category disks, more than 20 monthly disks and the 50 Commodore Education disks. The club decided that there was too much work for one librarian. TPUG now has four librarians. Mike Donegan for the PET/CBM and SuperPET; David and Richard Bradley for the Commodore 64; and Craig Bonner for the VIC 20.

LIBRARY CODES

At our first meeting we decided to use a number of conventions to help sort the programs into logical groups. The first, is to divide all programs into the appropriate library. These libraries are:

"C" Commodore 64 **"V" VIC-20**
"P" PET/CBM **"S" SuperPET**
"O" Old TPUG prior to March/83

To categorize disks within each library group, we will be using the disk name and disk ID in a special way. Both of these fields are created when we 'NEW' or 'HEADER' a disk. The disk name can be up to 16 characters long, while the ID is only 2 characters long. To indicate to which LIBRARY the disk belongs, we append to the disk name a period and one of the LIBRARY codes (C, V, P, S or O).

To indicate which CATEGORY the disk is in, we use both characters of the disk ID. The first character is the CATEGORY code (B for business, U for utility, etc). The second character is used to make each CATEGORY disk have a unique ID. The numbers1 to 9 are used first. When there are more than 9 disks within a CATEGORY, we then start using the letters A to Z. Therefore, B1 is the first business disk, B9 the ninth and BA the tenth. The CATEGORY codes are:

CATEGORY CODES

A Assembler/Machine Language
B Business
C Communications
E Education
G Games
L Languages
N Mathematics/Science
S Music
T TPUG Monthly
U Utility
X Best of Series
Z Miscellaneous

To make it easy to describe each disk, we use the LIBRARY code plus the disk ID in the following way: (C)B3. The LIBRARY code (C) for Commodore 64 is put in parenthesis and is followed by the disk ID (B3) Business 3. These three characters will allow us to uniquely describe every disk in the TPUG library.

PROGRAM CODES

To further document programs on disk, we have come up with a PROGRAM CODE. To use this, a period plus the PROGRAM CODE is appended to the end of each program name on the disk. These PROGRAM CODES include:

.B B series (available soon)
.C C64 programs
.D Data or Sequential files
.F Fat Forty, 40 column 12" screen
.L List-Me file
.P All PET/CBM (not VIC or C64)
.S SUPER PET/S9000
.V VIC 20 programs
.W Word processing files
.Z All Commodore machines or undefined
.4 40 column PET/CBM, 9" screen
.8 80 column CBM

ADDITIONAL DOCUMENTATION

Since the program name is not sufficient information when trying to determine what a program does, we have decided to add a BASIC program called LIST-ME. This program contains only REM statements. Each REM statement contains a program name and about 55 characters of description for that program. There would be one REM statement for each of the programs on the disk. The reason for choosing this format is that it provides a simple method of documenting a disk that is compatible with all machine types. The LIST-ME program is named with the three letter LIBRARY and CATEGORY code added to the end.

If anybody has any ideas to improve the suggestions made here, I will be happy to listen to them. BUT, the overiding idea is 'KISS' Keep It Simple Stupid. The method has to be easy to remember and implement

by Doris Bradley,
Assistant Business Manager

Do you have anything for this column? The three headings are: **(1) Helpful Hints (2) Who's Got the Answer?**, and **(3) PET Pals Wanted**. Just send your contributions (including answers to the above questions, where a full address is not provided) to the TPUG office, 1912A Avenue Rd., Ste. 1, Toronto M5M 4A1. Please mark on the envelope "Dept. Help" and let us know if you wish your full address published.

HELPFUL HINTS

C-64 Connection

Let me tell about how to resolve potential trouble when connecting a C-64 to a monochrome monitor such as a BMC 12" green (12AU model); The standard manual pin-out instruction calls for a "ground" and "video out" connection; Don't do it! You'll get a grid of lines through which you'll barely read the screen output; Use the "luminence" pin instead of "video out". It clears it up in no time!

Jack Goldstone
New York City

CLUB ACTIVITIES

Machine Language

To master machine language you need both knowledge of the 6502 chip (there are numerous books on this) and knowledge of the architecture of the PET, VIC or 64,(there is nothing available on this). One possibility is the book, Programming the PET/CBM by Raeto Collin West and put out by Compute Books.

Jim Butterfield
Toronto, Ontario

C-64 ROM Versions

Old ROMs - 1. Clearing the screen fills colour memory with white (like the VIC)

2. During cassette load, will print **FOUND NAME** and wait forever for you to push key.

New ROMs - 1. Clearing screen fills colour memory with screen colour.

2. Waits about 8 seconds after FOUND message then loads program, if no key pushed.

This first problem turned up while trying out disk D4 at a friend's place. In one game both your man and the target were invisible. My friend has the new ROM and the programmer has the old ROM. So on his screen they both appeared in white. In future I hope programmers make programs that work on all machines.

Greg Erker
Saskatoon, Saskatchewan

WHO'S GOT THE ANSWER?

"Has anyone come up with a fix for RF interference on standard TV's when hooking up the C-64? Besides going to a monitor, how do I get a clean picture?"

I am especially interested in Genealogy, Family History, Ancestor Research. Do you know of any programs for these subjects?

W. P. Farrington #6585
New Orleans, Louisiana

I am the Mathematics Head at Centennial S.S. in Windsor and we are using a NEC PC-8023A-C printer (ADA1600 interface) with our class set of 12 PET 4032 computers. We are delighted with the quality of the output and the mechanical quality of the printer. We are not satisfied that we can utilize this printer to anywhere near its potential with a PET. Any Help that you can give us will be warmly appreciated.

Keith A. Gallie #6274
Windsor, Ontario

We're interested in forming a network for the Commodore 64 machines in our school. Has anyone any helpful hints?

Oracle Middle School
P.O. Box 588
Oracle, Arizona 85623

My equipment includes C-64, 1541 single drive, Microworld MW-302 Interface(sw 4 on), GEMINI-10 Printer (sw 4on) and Name/Word Machine. I have been studying the Name/Word Machine listings to attempt to revise them to use more features of the printer. Could you put me in contact with other members who are interested in working this out?

H. Gary Docherty
San Gabriel, California

I teach Microbiology at a community college here in Southern California. Perhaps, it might be possible for some of your members who are Biology teachers to give me some advice and assistance.

Lee W. Rhodes
Mission Viejo, CA

Am interested in hearing about anyone who has used the RAMAX advertised in Compute.

Ken Clybor
Des Plaines, Illinois

Participation House, a residence for handicapped persons is looking for old copies of "Power Play"--can you help?

10 Bell Lane
Brantford ON N3T 6A6

I recently purchased a Gemini 10 printer with a Micro World MW 302 Interface to use with my VIC, and a C-64. I also have a disk drive and a cassette.

I hope you can help with some suggestions on how and where to get started. The manual that comes with the printer is almost useless. I even tried to use an Epson 80 manual but to no avail. I know the printer works because I have gotten it to do a few things by using some of the commands from a VIC printer book. I bought it 170 miles away from where I live and they have been of no real help since I purchased it.

Grace Packer
1800 Laguna Parker, AZ 5344 U.S.A.

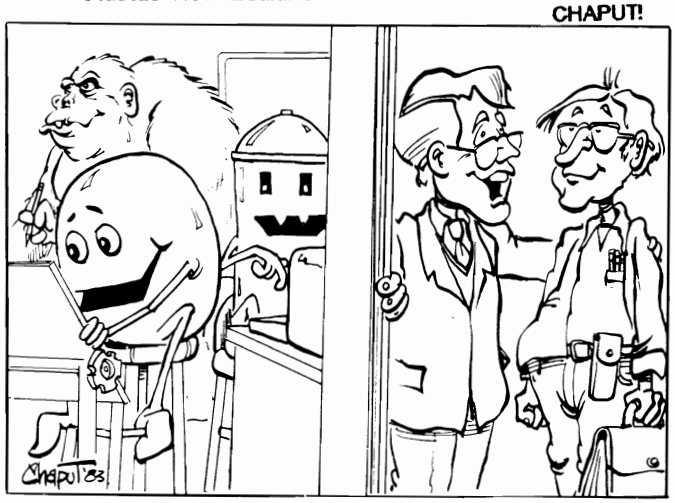
PET PALS WANTED

Correspondence to the TPUG office indicates a number of people with similar backgrounds and interests amongst our members (besides computers). If you would like to correspond with another sailor, farmer, HAM operator, quadraplegic etc. who has a computer, let us know and we'll see what we can do.

Have you got a member who would care to correspond with a 53 year old sheep and beef farmer who likes to play around with a computer in his spare time?

J. G. McKenzie
Eastern Bush
No. 1 R.D.
Otautau New Zealand

CHAPUT!



Now I want to introduce you to the other members of our Games Development Team.

TORPET CLASSIFIED
P.O. Box 100
Station "O" Toronto, Ontario
Canada M5M 4L6

5 Cents per word,
with $1.00 Minimum.
Payment in advance

WANTED

NEW 64 owner needs help, willing to pay (Mississauga area). Russell Thomas, Bus. 828-1600, Home 278-0996.

CREATIVE PROGRAMMERS! Original VIC-20 programs required by new software company. Recreational tape programs may be submitted with documentation and stamped self-addressed mailer to: The Cintechs Company, P.O. Box 2220, Stn. A, London, Ontario, N6A 4C3. All submissions will receive a response.

FOR SALE

COMMODORE 4022-P Bidirectional printer, used only two months, $825.00. Phone 519-524-9520 anytime. Darryl Carpenter, 36 Angleser St., Goderick, Ontario. N7A 1T9.

4016 COMMODORE PET. Includes Commodore cassette drive with footage counter, dust cover, connector and speaker for CB2 sound and manuals. All only 18 months old. $950.00. Call 416-459-9694 after 7 p.m.

$70.00 -- Arrow ROM chip for cassette system PETS. Features hyperspeed load, Save, Verify and Append (6-7 times faster), plus several other features. Imported from Supersoft in Great Britain and distributed by William G. McConnell, 92 Kensington Place, Chatham, Ontario, N7M 2X9.

PET JOYSTICK INTERFACE. Now any PET or CBM can use joysticks and game paddles. Accepts popular Atari- and Apple-style joysticks/game paddles. No assembly required. Ready to plug into User Port. Sample software provided. Only $69.95. Send cheque, money order, VISA/MC (please include exp. date) or specify C.O.D. to J. Systems Corp., 1 Edmund Pl., Ann Arbor, MI 48103, U.S.A. Credit card orders, call 313-662-4714 collect. (1)

ATTENTION VIC-20 and Commodore 64 users! Rubik Cube for C-64 and VIC-20 (16K necessary) for $14.00(tape). Do you want to traduct your software in "French" for the Quebec French speaking people, or for France? Have you software to sell or to distribute? Other offers considered. Write: Logimicro Inc., 138 Deslauriers, Neufchatel, P.Q. Canada G2B 3P4 (1)

The HELPING HAND is a digitizing stylus with attached 14"x10" drawing board. Connects to the game port of your VIC-20 or 64. Use it to draw or to trace from paper directly onto the screen. A friendly alternative to the keyboard, THE HELPING HAND is more versatile than a joystick or paddles. It's a powerful mate for the Super Expander. It's a low-cost "mouse".
Two programmable function keys are mounted directly on the drawing board, so you can carry out additional operations like clearing the screen without using the keyboard. Includes several demonstration programs, instructions, and is shipped in easy-to-assemble form (no soldering). $39.95 US plus $6.00 postage and handling from Persimmon Peripherals, Route 2, Box 2306A, Clayton, GA 30525.

FOR TRADE OR SALE

I am a pin and button collector and would enjoy trading with others that share my hobby. Please write to: David Bradley, 147 Roe Avenue, Toronto, Ontario, Canada M5M 2H8

CATALOGS

PET/CBM ADDS-ON FREE CATALOG. ECX COMPUTER COMPANY has over 20 new add on circuits and software for your PET/CBM computer and peripherals. For a FREE CATALOG send a self-addressed stamped envelope to:
ECX COMPUTER COMPANY, 2678 North Main St., Walnut Creek, California, 94596 (2)

WORLD'S WORST COMPUTER PUN

by John Ylimaki

1. One has to be a sailor of the I.C.'s to fathom the VIC-64. It's even named after a navy man, a COMMODORE. He's in charge of a whole fleet of CHIPS.

- the 6510 CHIP is sort of a c.p.U--Boat. Its maneuvers are called SUBroutines.

- the 6566 Video Chip carries the fleet's colours. It's a SPRITE for sore eyes!

New Additions to the TPUG Library

VIC

(V)TS
TPUG MARCH 83.V

VIC SLOTS
V TAX 82 ON V1.0
V TAX PART 2
V TAX PART 3
V 8K TAX 82
V RHINO
V 8K-LOAD
V 8K VICAB1
V 8K VICAB3
V 8K VICAB4
V 8K VICAB5
V BOMBER PILOT
V PAINT BY PEN
V CHINESE C'BOOK
V THUNDERBIRD
V ARITH CHALLENG
V NOTONE
V DRUM MANIA
V VICAB2
V BUSINESS DEMO
----LIST ME----

(V)TT
TPUG APR 83.V

2 JOYSTICK VIC.V
AIR GUNNERS.V
??????QQQQ
SCROLLING INST.V
SCROLLING.V
LABEL MAKER 8K.V
VIC DT
MAKE-A-SKETCH.V
HIDDENMAZE JOY.V
SNAKE.V
AUTO LINE#.V
MATH SKILLS.V
LOAN PROJECT.V
LOTTO.V
CAR COSTS.V
CALENDER.V
NIM.V

(V)X2 - BEST MUSIC.V

-LIST-ME (V) X2-
OVER THE R'BOW.V
GREENSLEEVES.V
ZIPPITYDOO-DA.V
VIC ORGAN.V
V DRUM MANIA
VICMUSIC$1201
V 76TROMBONES
V ENTERTAINER
V WONDERLAND
MERRY VIC-MAS
FRERE JACQUES
VIC KEY
VIC SIL. NITE
BUMBLEBEE
PIANO
USA SONG

(V)X1 - BEST UTIL.V

-LIST-ME (V) X1-
VIC AID4.REL.V
HIRES INSTR.V
HIRES HRDCOPY1.V
HIRES HRDCOPY2V
HIRES H/C DEMO.V
CUSTOM CARDS.V
TURTLE BOOT DISK
TURTLE BOOT TAPE
PLOT ML
TURTLE PROTO
TL]FOTT
TL]STAR
FUNCTION KEY.V
TERM 5K INST.V
TERMINAL 5K.V
VIC DT
CATALOG.V
VICWORD
JOYSTICK TEST.V
LABEL MAKER 8K.V
TINYMON1 FOR VIC
TINYMON INST
VIC DIS1
VIC DIS2
VIC DIS3
DISASM
SUPER VICMON2

(V)TU TPUG MAY/83.V

LIST-ME VTU.V
DR DEMENTIA IN.V
DR DEMENTIA.V
MORTGAGE.Z
PRINT USING.Z
ASTRO WARS.V
DYNAMITE!.V
BRAIN WARP.V
GLOBE QUIZ.V
VIC HANGMAN.V
VIC X/O'S 8K.V
VIC EDITYPE 8K.V
TINY PLAN 8K.V
R.B. SPEEDWAY.V
VIC G.I.R. INS.V
VIC G.I.R.V
CAR RACE(T)3K.V
ALPHA. COMMAND.V
VIC PILOT 3K.V
OVER THE R'BOW.V
GREENSLEEVES.V
ZIPPITYDOO-DA.V
VIC ORGAN.V
GRUNGY TOWERS 8K
GOLDRUSH.V

(V)E1 - EDUCATION.V

-LIST-ME (V) E1-
VICAB1 8K.V
VICAB2 8K.V
VICAB3 8K.V
VICAB4 8K.V
VICAB5 8K.V
ARITH CHALLENG.V
MATH SKILLS.V
GLOBE QUIZ.V
VIC HANGMAN.V
ALPHA. COMMAND.V

C-64

(C)TS
TPUG MARCH 83.C

LIST-ME CG1
MONTANA.64
MONOPOLE.64
LABYRINTH.64
LIST-ME CU1
PIANO.64
DISKVIEW.64
SPRITE-BOOT.64
+SCROLL.64
+SPRITE ED.64
DOS.BOOT.64
+DOS 5.1.64
+DOS.INST.64
COPY-ALL.64
1541 BACKUP.64
SUPERMON V1.1.64
BOOT.CLYDE.64
+DEMO.GUTS1.64
+DEMO.C000.64
+DEMO13.64
SPRITE MANIP.64
TERMINAL.64
TERM.64

(C)TT
TPUG APRIL/83

LIST ME CTT.L
PONZO TUTOR-1.C
PONZO TUTOR-2.C
PONZO TUTOR-3.C
PONZO TUTOR-4.C
PROG CONVERT.C
PADDL TEST.C
PRNT PADDLES.C
TERMINAL DOC.C
LISTER.C
1525 CHAR.EDIT.C
KAT $ MOUSE.C
CLIFFY.C
MIN2INS.C
MINOTON 2.C
TIME VEN INST.C
TIM VEN SETUP.C
TIME ADVENTURE.C

TPUG's This and That

Friday May 6th we registered member **#7000**. Welcome to Reiner Schubert of Duren, West Germany! That means 1000 new members in 36 calendar days--no wonder we're a little behind with the mail!!

Renewals

Please be sure to mark clearly that you are sending a renewal, and include your membership number.

Stamped Self-Addressed Envelopes

A number of members send these which is fine for those living in Canada. If you live outside Canada and wish to send return postage, please get an IRC at your post office and include it with your correspondence.

Packet Radio

HAMS interested in getting involved in digital packet communications should get in contact with Bruce Cowan VE3 GBC. Currently operating on 2 metres with a simplex digital store and forward repeater soon to be on the air. (Oshawa-Toronto-Brampton-Hamilton area) Bruce Cowan, (H) 243-9164, (W) 867-7979

Tape Prices

There are a number of growing pains that go along with taking over mastering and shipping tapes out of this office. Please note that TAPES FOR THE C-64 ARE $6 EACH, rather than $12 (i.e. D1 to D4 and the monthly tapes (C)TS (C)TT etc.)

Disk/Tape Orders

When placing an order please indicate what computer equipment the order is to be used on. We do try to warn you of compatibility problems.

Doris Bradley,
Assistant Business Manager

PET/CBM/COMMODORE 64

PaperClip

Professional Word Processor at a Breakthrough Price

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

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

Price:

$150.00 CDN.

BATTERIES INCLUDED

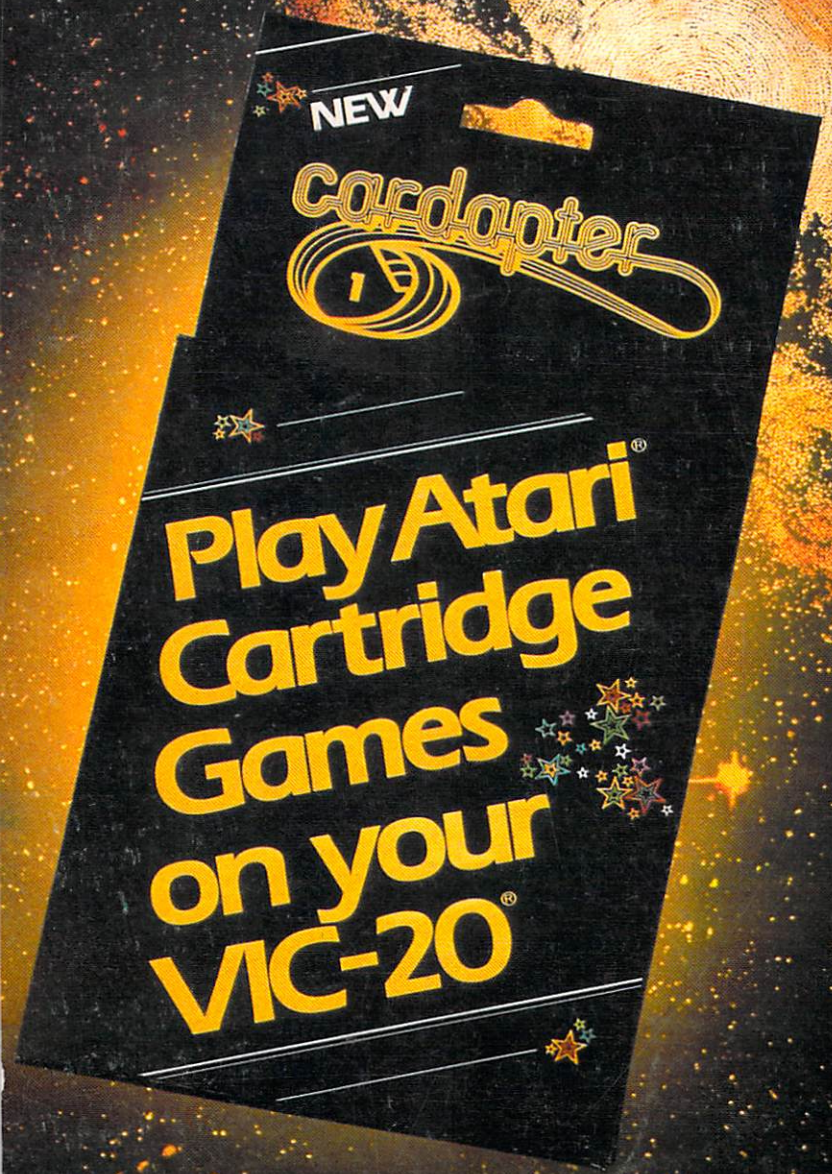
186 Queen Street
Toronto, Ontario
Canada M5V 1S1
(416) 596-1405

NEW

Introducing...

cardapter 1

EXPLORE A NEW GALAXY OF ENTERTAINMENT PLEASURE WITH YOUR VIC-20® COMPUTER.



Designed by: The Computer Works
Manufactured and Distributed by:

cardco, inc.
313 Mathewson Ave • Wichita, KS

VIC-20® is a registered trademark of Commodore Intl'
Atari® is a registered trademark of the Atari Corp.

Dealer inquiries invited.

United States: Cardco, Inc. • 313 Mathewson • Wichita, KS 67214 • (316) 267-6525
West Canada: LSI Distributors Ltd. • Attn: Mr. Cheng • #163-810 W. Broadway • Vancouver, BC. CA V5Z 4C9 • (604) 733-0211
England & Europe: Audiogenic • Martin Manary • 34-36 Crown St. • Reading, Berkshire England • (0734) 595647
East Canada: Hobby Craft Canada • 24 Ronson Drive • Rexdols Ontario M9W1B4 • (416) 241-2661