

TPUG Newsletter

Views and News of Toronto Pet Users Group, Inc.

P.O. Box 48565, 3605 Lakeshore Blvd. W., Etobicoke, Ontario, M8W 4Y6

Volume 6, Number 9

Summer 2001

From the President -

The dawn of the 21 century, the new millennium has now begun. This year TPUG one of the older computer user groups will be 23 years of age. Most of the club members are die hard, nostalgic collectors of various computer platforms. I myself own several Commodore C64s, C128s, an Amiga A500, an Atari 1040ST, several Timex/Sinclair 1000s, as well as a Pentium II clone. Compared to some of the other members my collection is a small one.

Why are we still using our Commodore computers??? I use my C128 with PaperClip III to write articles and letters (like this one). The C64 I use to do draw electronic circuits, circuit boards (for myself and the companies I have worked for), TPUG business cards, make signs for the computer shows TPUG attends. The programs I use are Flexi-draw 5.5 (with a 1351 mouse) and/or Fun Graphic Machine ver.8. TPUG's treasurer uses his Pet to do the clubs bookkeeping. Another uses his C64 to entertain his grandson (because of the quality of the games) when he comes to visit. Some run their home business or compose music on them. The bottom line is "as long as the old computers can do the task why replace them?"

Apple, Atari, Commodore, Texas Instrument, Timex/Sinclair (and others) and their computers are not just old they are part of the computer's history. If it was not for the improvements made on the old computers, the computer would not have evolved into the computers we have today.

Apple and Commodore (C64) computers had icon based programs (GEOS) about a year before Windows for MS-DOS PCs came out. Good sound and graphics took a long time to be available in an IBM computer but where all ready available in some of the older computers (some still claim the Amiga is still better then MS-DOS systems). In most cases newer computers may have had to play catch-up on features but they usually were better features in the end.

Our old computers were designed for the average consumer to be able to write his/her own program with as little effort as possible. Today people can write their own programs but they are not so easy and not so clean (due to all the layers of code and size of a byte). What we use to write on our 8 bit computers in 1K (kilo) byte of memory may take Mega Bytes to write in today's computers. (kilo=1000, mega=1,000,000).

Five years ago, or better, some people using the C64 or other similar vintage computers may have done so because of their economic factors. Today that is less likely to be the case due to the very low price on MS-DOS 486 PCs. Selling a complete working C64 system (monitor, computer/keyboard, 2 disk drives, power supply, software and accessories) a person might get \$100.00 CAN in the Toronto area if they are fortunate. For about the same price you could pick up a 486 computer with a 12 inch monitor, (likely

.... continued on page 15

For users of all
Commodore Computers :

- * PET/CBM
- * SuperPet
* B-128
- * VIC 20
- * Commodore 64
* PLUS-4
* C-16
- * Commodore C 128
- * AMIGA
PC/MS-DOS

* Registered products of
Commodore Business
Machines, International
and/or their assignees.

Table of Contents

President's Comments	1
Member Information	2
TPUG News	3
Tom's World	4
ComputerFest	7
Mouse as a Joystick	7
Why Use Commodore 64	9
New Loadstar Editor	9
Amiga Updates	10
Amiga - Still viable	12
Synth Sample Author	13
Programming Languages ..	14
Vic-20 Challenge	14

Member Information

Voice Info

We have discontinued our TPUG phone listing - contact members as listed here at home phones.

e-mail : tpug@icomm.ca

Membership Rates

Canada \$25
USA US \$25
International US \$25

Board of Directors

President Tom Luff
Vice President John Easton
Secretary Ernie Chorny
Membership Sec. . . Ian McIntosh
Treasurer George Turek
Director Tom Haslehurst
Director Joe Palumbo
Director Ken Price
Director Tom Williams

Librarians

Head Librarian . . . Ian McIntosh
Amiga John Buller
C128 Tom Luff
CP/M Ernie Chorny
GEOS Joe Palumbo
C64 Joe Palumbo
C64 Education . . Bill Cumberland
Comal Ernie Chorny
Plus/4 Ernie Chorny
Vic 20 Ernie Chorny
PET/CBM/SuperPET John Easton
PET&C64 Education John Easton

Support

Mail Tom Luff
Telephone John Easton
Disk Orders Librarians
Member Records . . Ian McIntosh
Meetings Tom Luff
..... and Ernie Chorny
Shows . Tom Luff & Ernie Chorny
Webmaster Ian McIntosh
Newsletter
Editor John Easton (416) 251-1511
..... jeaston@idirect.com

Meeting Schedule

Amiga East: Second Tuesday of the month.

Contact - Ernie Chorny (905) 279-2730

7:30 pm at Videolink - 2284 Gerrard Street East, Scarborough.
phone (416) 690-1690

TPUG acknowledges the generous support of Videolink's Bruce Richardson

Westside and Amiga West: Third Thursday of the month (except summer) at Alderwood United Church, 44 Delma Drive. Delma Drive is just west of and parallel to Browns Line, south of the Queen Elizabeth Highway, north of Horner Avenue. From the west, exit QEW at Evans Avenue, east on Evans to 2nd stoplight, south on Gair to Delma Drive. From the north or east, follow signs from QEW or Hwy. 427 to Browns Line, exit right to Evans Avenue, turn south on Gair (first stoplight) to Delma.

Contact - Tim Luff (905) 812-5231 e-mail: ntluff@3web.net
or Ernie Chorny(905) 279-2730

TPUG on the Internet:

<http://www.icomm.ca/tpug>
e-mail: tpug@icomm.ca



-Ha! So you thought we'd forgotten you? Thought we were in hibernation? Thought we'd died? ... None of the above - just your ancient Editor (not to mention your ancient Editor's wife) being ornery and thinking that Spring Cleaning and painting were more important than that dumb newsletter.

Please accept my apologies (*pray tell, what else can you do?*) and rest assured that we at TPUG are still very much alive (despite the comments of wives and sweethearts) - albeit in an arguably downwardly-mobile state of commitment.

With luck, and a bit of blind determination, you might yet see the next issue of this paper before the snow flies.

Why do we still bother? Good question - but as the grateful owner of a still operational PET system (vital to maintaining his customer data base) said just the other day as he left my house with a replacement for his suddenly dead 8050 disk drive, "*this system does the job I need*" whereas the DOS machine beside it by now refuses to even think of operating - the local techies just laugh at his problem, suggesting of course that he should merely invest a few more thousand into a real machine and throw all else to the trash.

The *real* reason for remaining alive is Networking - allowing others to know that somewhere, somehow there exist folk who still support our 'technology of the heart' - read - '*that which we know will work!*'

*TPUG Newsletter is published somewhat quarterly by the Toronto Pet Users Group Inc. (TPUG). TPUG is a volunteer non-profit corporation dedicated to the service and support of owners and users of Commodore computers. All rights to material published in TPUG Newsletter are reserved by TPUG, Inc. and no material may be reprinted without written permission, except where specifically stated. When reprinting is authorized, please credit TPUG Newsletter, the issue date, and the author. (note - electronic copy *may* be available, please enquire)
Articles, letters, tips, questions, art, etc. are welcome. Send hardcopy or disks "Attn: TPUG Newsletter", or use Internet e-mail.
Advertisements are also welcome. Member's small ads are free. Commercial ads are \$100 per page with a \$10 minimum.*

Notice to new owners of SuperPet and CBM 8296 machines

TPUG has copies of the Waterloo LANGUAGE DISKS (3 in 4040 format) as supplied with the SuperPet on original purchase.
 TPUG has the EXECUDESK disk (8050 format) as supplied with the CBM 8296 on original purchase.
 These disks are an integral part of the operating systems of the above machines and since Commodore insisted on referring owners of these machines to TPUG for service, we have added these somewhat proprietary (and also virtually unobtainable) disks to our library - all part of the TPUG mandate of service to our members.
 We also will attempt to search out copies of original program disks to replace corrupted disks. In this category you will find such programs as VISICALC, WordPro, and PaperClip.

INSTANT 1581 DRIVE KIT (Just add a standard PC floppy drive) \$49.95
 Includes Upper & lower Shell with logic board & faceplate, a serial cord and power supply box.

POWER SUPPLY ONLY (1581/41-II) \$24.95
 UPPER CASE ONLY \$9.95
 LOWER CASE ONLY \$9.95
 1581 LOGIC BOARD ONLY \$14.95
 SERIAL CABLE ONLY \$8.95
 -OPTIONAL:
 1581 JIFFYDOS ROM....add \$32.95
 10% shipping (15% USA)
 Taxes are extra for Ontario and Canada residents (PST/GST)
 Mail cheque or Money order to:
 JP PBM Products By Mail
 Box 60515, Jane/Wilson P/O
 Downsview Ontario, M3L 1B0

Note: Dealers and User Groups Welcome!

TPUG News - Hold the press ...

Are we ready for this? A real CBM Expo (that's Commodore Business Machines to you that think the only stuff produced by Commodore had a C=number or Amiga associated with its name) - in the fall, right here in Toronto. Dates are still tentative, awaiting confirmation of the Chicago Fall Expo.

We could look forward to such nostalgia as old Commodore posters, promotions (psst, can I interest you in a *GenYewine* CBM digital watch?), banners, machines - how many of you have actually seen a C=65 in action? - or for that matter (*chuckle*) a real SuperPet with its multiple language capabilities - or, a SuperPet running under OS9?). Naturally, we should look for a word from Torontos own **Jim Butterfield**, but perhaps we might look to meeting the likes of **Maurice Randall** (of Wheels 'n all that stuff fame) or **Jeri Ellsworth** (she, who admitting to little knowledge of Commodore limitations, seems to have re-invented the C=64 as a powerhouse platform - does she call that the C=1?). How about those folks from CMD and their powerful add-on peripherals? And of course, we'll expect a gazillion swap-tables and gab-fests.

At any rate, this is your first warning - and just in case we havent yet reverted to our quarterly publication schedule by September, why not send a note to **Tony Foster** at <teknical77@sympatico.ca>, he's the originator and chief organizer of this venture. He will keep you up to date on developments.

Classified

Another member-service!
For Sale:

2 - C64s, 2 - 1541 disk drives, colour monitors, joysticks, printers, and printer interfaces.
 Call Tom Luff (905) 812-5231.

Miscellaneous Commodore Hardware and Software
 is available from :
 D.L. Johansen

Box 912, Troy, MT, 59935

COMMODORE GAZETTE

Magazine-on-Disk
 Christopher Ryan
 5296 Devonshire Rd.
 Detroit, MI, 48224-3233
 (313) 882-8128 (4thru 10 PM EST)
 <cmdrgazette@ameritech.net>
<http://headgap.com/gazette.html>
 \$25.00 for a one-year subscription.

J.P. PBM Products by Mail is the NEW Manufacturer of Super Snapshot Cartridge V5.22 - NOW SHIPPING

We are pleased to offer this cartridge regularly \$89.95. For a limited time **SAVE \$15 WITH THIS AD. UNTIL December 31/01.**

CURRENT Commodore Club MEMBERS SAVE \$5 MORE off the regular price before freight and taxes.

Mail Cheque/M.O. to:	SSv5.22 Cartridge	\$89.95
JP PBM Products by Mail	save \$15 now	<u>-\$15.00</u>
BOX 60515 JANE/WILSON P/O		\$74.95

DOWNSVIEW, ONTARIO	* C= Club Members (-\$5)	-\$
CANADA M3L 1B0 *	32K RAM add \$19	+\$
	subtotal	\$

	+10% Freight (15% USA)	\$
All Prices Are Cdn. Funds	Subtotal	\$
20% Exchange On US Funds	Ontario Res add 8% PST	+\$
Send \$2 for a catalogue	Canada Res add 7% GST	+\$
on disk (1541 format) (CDN FUNDS)	TOTAL	\$

Tom's World

Ed note: this is the continuation of Tom's Machine Language article of Winter 2000

Getting Your Program To The Computer.

There are several ways to get your code into the computer. One way is to use an Assembler to convert the assembly language program to machine language code and save it to disk (as a M L program). Another is to use basic to enter machine code to the computer, then SYSxxxx to it. A third method is to use a M L monitor program to type it into the computer.

Assemblers.

The assembler editor allows you to construct a program in assembly language using mnemonics, addressing modes, operands and some other features (which may change from one assembler to another) then save your program as a Source Code file. The source file can then be assembled, by an assembler program, into a Machine Language program (also referred to as an Object Code file) which the computer can be told to load and run. Assemblers differ from one to another so it would be difficult to get into any more detail on this subject. The one advantage here is the source code makes it easier to debug and edit the program. One advantage to using a good quality assembler is the use of labels to link branches to memory addresses. When editing source code the labels adjust the branch offsets automatically so you don't have to. It is important to read and be familiar with the operator's manual for the assembler program. A book with a tutorial structure is a great learning tool. (Jim Butterfield is one of the founders of TPUG and a author of one of the books I recommend).

A Basic Program

It is possible to use a BASIC program to deliver the machine code to the computer. There is a little more work involved because it usually means you will have to hand assemble the assembly code to machine code (if the machine code is in hexadecimal form then it will need to be changed into decimal form). From there you would write a basic program that would loop, and each time it does, the program would read the machine code stored in data statements, then using a poke command it would insert the machine code into the correct memory address. At the end of the loop the program would SYSxxxxx to the starting address of the machine code program (SYSxxxxx : x=the start address of the machine code).

On paper write out your assembly code and beside each line write the equivalent machine code in a second column. The machine codes (opcodes) can be easily obtained for any good book written for programming the 6502 microprocessor. It can be found in the books listed at the end of the article. I do find that the Commodore 64 Programmer's Reference Guide is nice to use because it lists the mnemonic, the addressing modes used by the mnemonic, followed by the opcode (machine code) and the flags affected by the instruction. In a third column write the decimal equivalent beside the hexadecimal number. This will be the numbers we use in the basic program. See the sample program at the end of the article.

M.L. Monitors

As your Commodore 64 or C128 is first turned on it is ready to accept Basic commands because the operating system uses a Basic Monitor. A basic monitor is a program in ROM which defines rules and syntax for symbols (i.e. LOAD, Run, GOTO and others) for a particular language, in this case the Basic language.

A M.L. Monitor is another program designed for easy use of Machine Language. A M.L. monitor may be in ROM, 1) built into the computer (as in the C128), or 2) in a cartridge (as in Super Snapshot). It can also be loaded in from a disk drive or tape drive (as in Supermon). Supermon64.v1 can be found on TPUG's (C)D1 Demo disk. Supermon v2.c can be found on TPUG's (C)U1 Utilities disk. Monitors usually use a common command set however some will have an expanded set.

Check your monitor's manual for more commands and syntax. For the modification of a line in the monitor the 3 monitors I mentioned above each allow you to use M to display the memory and then you can type over it. When finished that line press the RETURN key to set the changes into memory. The only difference is the line's first character. In the C128 a ">" sign is used to indicate the beginning of a line when the M command is used to dump memory to the screen. In the other two monitors a ":" is used. Then all three display the first memory address of the 8 memory to be displayed. The contents of the next 8 memory addresses are then displayed by two hexadecimal characters each with a space in between each memory value. The remainder of the line is a symbolic representation of the memory contents.

Here are a few commands :

```
M 0800 08FF - displays the contents of memory addresses $0800 to $08ff
G 0801 - starts executing from memory address $0801
C 0800 0C00 2000 - compare memory $0800 to $0C00 to memory starting at $2000
T 0800 0C00 2000 - transfers memory ($0800 to $0C00) to a new start address
A 0800 LDA#1 - assembles the A.L. code to M.L. code starting at $0800
D 0800 0900 - disassembles from $0800 to $0900
L "filename",8,0800 - loads a file from device 8, starting at address $0800
S "filename",8,0900,0A01 - saves a file starting at $0900 to 0A00+1 to device 8
                    - must add 1 to the last address used by your program
>0800 - modify a line of code starting at address $0800 (C128 monitor)
:0C00 -modify a line of code starting at address $0C00 (the other two monitors)
; - modify the register display
R - display the registers (BR -for Super Snapshot)
X - exit the monitor
```

The first thing the monitors display upon start-up of the monitor is the contents of the registers:

PC SR AC XR YR SP
:XXXX XX XX XX XX XX

PC - contents of the program counter, points to the current memory address.

SR - is the contents of the status register, where the flag bits are located.

AC - is the contents of the accumulator register.

XR - is the contents of the X register.

YR - is the contents of the Y register.

SP - is the contents of the stack pointer.

This is also displayed when a BRK instruction or an R command is encountered. The register display is handy for debugging programs.

To enter a M.L. monitor see your manuals. For Supermon just load and run the program. If using Super Snapshot, press the cartridge button to access the sub-menu. From there select option 5) Monitors, followed by option 1) M.L. Monitor. For the C128 depress the F8 key.

Your First ML Program

A 033C	A9 93	LDA #\$93	169 147	-load a clearsreen code to Accumulator.
A 033E	20 D2 FF	JSR \$FFD2	32 210 255	-chROUT sends Accum.to output device.
A 0341	A9 00	LDA #\$00	169 0	-will store the colour black to the
A 0343	8D 86 02	STA \$0286	141 134 2	cursor colour register.
A 0346	A9 01	LDA #01	169 1	-will store the colour white to the
A 0348	8D 21 D0	STA \$D021	141 33 208	raster register.
A 034B	20 D2 FF	JSR \$FFD2	32 210 255	-tells output device to take info.
A 034E	A9 48	LDA #\$48	169 72	-puts "H" in the accumulator.
A 0350	20 D2 FF	JSR \$FFD2	32 210 255	-chROUT puts it to output dev.(screen)
A 0353	A9 45	LDA #\$45	169 69	-puts "E" to the screen next to "H"
A 0355	20 D2 FF	JSR \$FFD2	32 210 255	on the screen.
A 0358	A9 4C	LDA #\$4C	169 76	-puts "L" to the screen.
A 035A	20 D2 FF	JSR \$FFD2	32 210 255	
A 035D	A9 4C	LDA #\$4C	169 76	-puts "L" to the screen.
A 035F	20 D2 FF	JSR \$FFD2	32 210 255	
A 0362	A9 4F	LDA #\$4F	169 79	-puts "O" to the screen.
A 0364	20 D2 FF	JSR \$FFD2	32 210 255	
A 0367	60	RTS	96	-returns control to Basic.
A 0368	00	BRK	00	

(comment)
() Decimal Machine code
() Assembly Language
() Hexadecimal Machine code
() Memory address-in hexadecimal
* -Monitor command

Note: all addresses in this program are listed in a low byte, high byte order. This is the way the C64 and the C128 require address notation.

134 02 = 646.....decimal low byte, high byte = full decimal address
33 208 = 53281 210 255 = 65490

The same is true for the hexadecimal addresses...D2 FF = \$FFD2

The program above has a lot of information but do not be intimidated. It is easier than it looks. The program is a result of assembling each line in a monitor. Use the "A" command in the first column, then the address (in hexadecimal), followed by the mnemonics then press the return key. (The first line of the program will look this- "A 033C LDA #93"). The monitor (in this case Super Snapshot) alters the line by inserting the hexadecimal code between the address and the mnemonics (A 033C A9 93 LDA #\$93) and inserted the \$ in front of the 93. The decimal column, I added to show the connection between the it and the hexadecimal column. Someone using basic can use the decimal column to write their program.

The program uses the chROUT kernel routine by calling it through a vector (\$FFD2) in a look-up table. This table is the same in all C64s no matter which upgraded version. The chROUT kernel routine takes the contents of the accumulator and sends it to the output device, which can be the screen (usually), the disk drive, the printer or something else (chosen by another kernel routine). The pointer or cursor is position by another routine. The character in the accumulator register will be placed where ever the cursor points to, then the cursor will then advance by one position. It seems the chROUT routine also affects the video registers such as raster, border and cursor colours.

In the above program we start at memory address \$033C (or 828 decimal) and in the first line we load the clear screen

code \$93 (or 147 decimal) into the accumulator register. The second line calls for the chROUT kernel routine using the vector \$FFD2 (or 65490 decimal) by using the Jump to SubRoutine instruction. The JSR \$FFD2 instruction is repeated each time a code needs to be sent to the screen. Lines \$0341 and \$0346 load a colour code to the accumulator and lines \$0343 and \$0348 store the contents of the accumulator in the cursor colour register \$0286 (or 646) and the raster colour register \$D021 (or 53281). Lines \$034E, \$0343, \$0358, \$035D and \$0362 each load a letter ("H", "E", "L", "L", "O"), while each next line sends them to the screen. The last line \$0367 returns (RTS) control to basic. After typing the program in, save it then use the "X" command to exit to basic. In basic type in "SYS828" then press the return key. The screen will clear and change colour then the word "HELLO" will be spelled out in the upper left corner of the raster.

Special Note

If you are using a C128 computer in its 40 column native mode then use a starting address of \$0B00 (or 2816 decimal). Using the starting address as above will over-write code in that area. Also the C64 uses address \$0286 (646 decimal) for the cursor colour but the C128 uses address \$00F1 (241 decimal).

Make the following changes when using the C128 :

First line of the ML program-
A 0B00 A9 93 LDA #\$93 169 147
Fourth line of the ML program-
A 0B07 8D F1 00 STA \$00F8 241 00
-the address line is automatically incremented in the monitor.

In the basic program (below)-

Line 10 change the "POKE 827+X,A" to "POKE 2815+X,A"
Line 20 change "134,2" to "241,0"
Line 50 change "SYS828" to "SYS2816"

BASIC Program

```
10 FOR X=1 TO 44:READ A:POKE 827+X,A:NEXTX
20 DATA 169,147,32,210,255,169,0,141,134,2,
169,1,141,33,208,32,210
30 DATA 255,169,72,32,210,255,169,69,32,210,
255,169,76,32,210,255
40 DATA 169,76,32,210,255,169,79,32,210,255,96
50 SYS828
```

Save, then run this program. It is the same program as above but it loads and runs in basic. The program when it runs, will load the machine code into memory (line 10) one byte to each loop, then it will turn control over to the machine language program (line 50).

Final Note

Although it is easier to use basic to accomplish the task of this program, the ML program does run faster. In a more complicated program, such as a game, where you need a smoother flow on the screen ML will do a better job of it than basic could do.

The purpose of these articles is to pique your interest. There is a lot more to Machine Language than touched in these articles. The advantage to our older computers to the MS-DOS computer is if we make a mistake all we do is turn off the computer then turn it back on and start all over again with no harm done to our hardware. Like any large new learning task (i.e. complicated programmes like GEOS) you should try a little then build on what you know each time you go back to it. I find a trial and error method to learning something is the best way to get to know and understand what you are learning.

Here are the reference books I used to research these articles:

Commodore 64 Programmer's Reference Guide by Commodore Business Machines, INC

Machine Language For The Commodore 64, 128 and Other Commodore Computers: by Jim Butterfield (a TPUG FOUNDING MEMBER)

Commodore 64/128 Assembly Language Programming: by Mark Andrews

Commodore 128 Reference Guide For Programmers: by David L. Heiserman

More information on machine and assembly languages can be found on the following TPUG disks:

(C)E1 - C64 Tutorials

(C)E4 - 6510 Opcodes

Tom Luff

Switchless C=1541 8/9 select

From C=HUG's Gazette, September, 2000

Connecting a switch to a C=1541 disk drive to allow manual selection of device number 8 or 9 is a well-known procedure. There is also a way to create a switchable drive without installing a switch.

A close look at the little half-moon spots on the two jumpers reveals a circuit trace connecting one spot of each jumper to the other.

Solder a wire to the half of jumper 1 that is NOT connected to the other jumper. Connect the other end of the wire to the fourth pin of the 15-pin plug that contains a wire, usually orange (the fourth pin from the other end contains no wire).

The drive will now turn on normally as drive 8 unless it is turned on with a disk half inserted, then it will be drive 9.

ComputerFest:

A dose of nostalgia at the Commodore booth. - Story and photo by Andrew Moore-Crispin

Among the high-tech vendors at ComputerFest and Mac Expo last weekend was a booth offering a big dose of nostalgia for long-time computer users, and a valuable resource for die-hard Commodore fanatics.

The Toronto PET (Personal Electronics Transactor) Users' Group Inc. (TPUG) is a club that donates Commodore systems and software to other non-profit organizations, as well as selling Commodore software and hardware at computer shows to cover operating costs and promote public interest.

TPUG's main reason for renting a booth at Computer Fest was to recruit new members, director Joseph Palumbo said. Palumbo uses the most widely recognized Commodore system, the Commodore 64, to run a new-and-used Commodore products mail order business, complete with business cards printed on the 64.

Club president Tom Luff said TPUG doesn't go to computer shows to sell Commodore systems or software. "Basically, we use this venue as a

means to communicate with the public," Luff said, "to let them know that we're available."

Most of TPUG's income is from membership fees (\$25 a year), Luff said. "Normally we don't make enough money to pay for the booth, but some people are just die-hard Commodore users."

Commodore went bankrupt in May 1994, due to bad marketing and mismanagement. "Commodore basically screwed up," Luff said.

Upon the release of the Amiga computer, the company stopped paying attention to the 64, which was undoubtedly the most popular of Commodore's line of computers, Luff said. The Amiga didn't take off and the company faltered.

At the time, Commodore was working on a faster, more powerful 64, called the Commodore 65, Luff said. The 65 was supposed to bridge the gap between the 64 and the Amiga, but never made it to market.

TPUG takes donations of Commodore systems. There is a glut of the computers themselves, but Commodore monitors are in short supply, Luff said. "One of the things that people are

realizing is that Commodore monitors are ideal to hook up to a VCR," he said. "Consequently, when we get donations, they're without monitors."

While Luff uses the 64 for all his correspondence as well as to create his business cards, Commodore computing remains largely a hobby. TPUG's Web site and before that BBS (Bulletin Board Service) were both authored and run on a PC, he said.

(From the torontObserver, February 12, 2000)



Toronto Pet Users Group President Tom Luff (left) and one of the club's directors Joseph Palumbo at Computer Fest and Mac Expo in the automotive building at Exhibition Place Feb. 9-11. Palumbo is wearing a button which reads "I adore my 64."

Using Your Mouse As A Joystick On The C64

If you follow the instructions from Commodore, you would always turn the computer off whenever you needed to plug or unplug a cable or device. So changing from a mouse to a joystick you would turn the computer off, followed by unplugging the mouse, then plug the joystick in, turn the computer back on, load the program and run it. Even if you use the Commodore 1351 Mouse as the joystick you must turn off the computer, hold down the mouse's right button while you turn the computer back on. This changes the mouse (proportional mode) to a joystick (8 directional mode).

The C64 first came out around 1982 and were no longer manufactured by about 1990, this makes these great machines about 10 years old (or older). When the power switch is flipped on or off there is a power spike (a sharp change in the voltage levels) which over time can result in damaged components and the computer may not work again. To reduce the possibility of this premature demise of the computer it would be beneficial to reduce the number of times you turn on the computer to a minimum. Please note I

am NOT recommending you leave the computer on all the time and never shut it off, this could lead to other problems of premature demise. What I do suggest is reduce the number of times you turn on the computer to a daily minimum.

One way to accomplish this is to use the 1351 mouse as both mouse and joystick combined with this Mouse To Joystick project. A second way to accomplish this is to install a computer reset switch which would reset the computer back to its start-up screen. The Mouse To Joystick project is basically an extension cable with an interrupt switch in the +5 volt conductor. The interrupt switch is a normally closed (N.C.) momentary push button and when pressed it will remove the +5 volts from the mouse, just as if you had turned off the computer.

After building the project, turn the computer off and plug the project into game port 1 (closest port to the front of the keyboard), then plug the mouse's cable into the project's male connector. Turn the computer on and run your mouse

program as usual, this is to test to see that the mouse is working. Once the mouse is tested, change the program to one which requires a joystick in port 1. To change the mouse to joystick mode, hold down the right mouse button and depress the project's button. NOTE: the mouse's right button must be the last button to be released. The mouse should now operate like a joystick. When you need the mouse as a mouse simply depress the project's button only. Keep in mind it is the program which dictates which input device will be required.

PROJECT CONSTRUCTION

For Parts List, see box below -

Start with the monitor cable (item 1), cut the cable 12 inches from both ends. Use a marker and put a mark 6 inches from each 9 pin connector.

Take the Project box (item 3), and on opposite ends drill a hole just large enough to insert the monitor cable into. On one side of the box, perpendicular to the ends drilled for the cables, drill a hole large enough to mount the N.C. momentary pushbutton (item 2). Open the project box and insert, from the outside of the box, the two cables into the holes on either end of the box.

Install the N.C. momentary pushbutton (item 2) into the hole in the project box drilled for it and tighten its nut.

Pull the two cables into the box until the 9 pin connectors are against the box. On each cable, locate the 6 inch mark and tie a simple knot in the cable at that mark. Make sure the knots are tight, then about 1/4 inch from each knot remove the outer coverings from the rest of the cable (be careful NOT to damage the inside wires). Cut all the wires to a length of 3 inches.

On each wire strip about 3/4 of an inch of insulation and twist the strands of the wire together. Use an ohm meter and locate the wire from pin 7 of each connector and solder them each to their own solder lug on the pushbutton (item 2), by placing the insulation of the wire close to the lug and wrapping the wire around the lug twice, then soldering. Cut off any excess length of bare wire and solder. Note : There should be no short between the lugs. Pin 7 is the +5 volt

conductor, by breaking this conductor the power will no longer go to the mouse. This will allow us to change the mode of the mouse. Put a probe to both pin 7s and the meter will show a short, then when you press down on the push-button the meter will show an open circuit, this is what we want.

For the rest of the wires they need to be wired 1 to 1 as follows. Pin 1 is to be wired to the other pin 1, by placing the ends of insulation from both wires side by side and twisting the bare wires together then tinning the twist wire (solder the wires). Cut the tinned wire to about 1/4 inch then fold the tinned wire flat against one wire and wrap the connection with electrical tape to prevent accidental shorts. Do exactly the same to the wires from pin 2 to pin 2, then pin 3 to pin 3, etc ... When you are finished you should have 8 connections done the same way. The diagram which follows shows pin locations, DO NOT confuse the two connectors or you will have problems.

Carefully fold the wires into the project box (item 2) and close the box making sure no wires get pinched then screw the box closed.

Make sure your computer is turned off. Plug the female connector into port 1. Plug the mouse's connector into the male connector of our project. Turn the computer on. Load a program which requires a 1351 mouse then run it. Check to see that the mouse is responding correctly. If everything functions alright then the project was built correctly, if not you will have to pull the project apart and find your error.

Once the mouse is working OK, exit the mouse program and load the joystick program (which uses port 1) and run it. Test the mouse it will not work. Now hold down the right mouse button, then depress the project's pushbutton, then release the mouse's button. The mouse's button must be the last button to be released to put the mouse into joystick mode. Try the mouse now it should work with the joystick program.

To put the mouse back into mouse mode simply depress the project's button.

Tom Luff

PROJECT CONSTRUCTION - Parts List				
Item	Qty.	Description	Radio Shack Part No.	Cost
*1	1	DB9 male to female 6' Monitor cable	2600117	\$12.99
2	1	N.C. momentary pushbutton	275 - 1548	4/\$4.99
3	1	Project box	270 - 1801	\$3.79
4	4	A roll of electrical tape	64-8026	\$.99
*NOTE : Any substitute for item 1 must have 9 conductors (wires).				

Why A Million People Still Use Commodore 64

From the August, 2000 KEYWORDS newsletter via various sources

Happily, we still see active Commodore User Groups, newsletters, magazines, conventions, weekly on-line discussions and web sites. That raises a question: Why do a million people still use Commodore 64?

For some, it is a fondness for their first computers, a love for the old, like vintage cars. There are those who have even set up miniature computer museums in their homes. Indeed, not a few think that this nearly 18-year old 1 MHz machine with 64K RAM is on the verge of becoming a collectors' item. Remember, 20 million units sold from 1982 to 1992, making it the most popular single computer of all time.

For others though, it's a matter of actually using the machine. It isn't that

they are unable to acquire a more powerful PC. Nor is it a matter of not knowing how to use them. In fact, a number of 64 users own newer machines.

But there are some very good word processing, spreadsheet and data base software available for the 64. For that reason, many see no need to add a bigger PC. GEOS is available to those who prefer the windows environment. And a Commodore can fax via modems, access the Internet and utilize laser printers. Furthermore, RAM packs, hard drives and speed up devices can be added to increase the 64's capabilities.

However, there is more to it than just productivity. A lot of game players stay with Commodore because the older games, without all the fancy graphics

and sound effects, are focused more on game play. The result, it is argued, are games which rise above their technological plainness.

And Commodore 64 users are not frustrated by the rapid obsolescence of newer machines. Neither, are they bothered by viruses. Then there is the simplicity factor.

Those who wrote programs for the 64 were forced by its limits to make the software fit the machine rather than increase the hardware requirements to do the task. As a result, Commodores aren't as prone to crash and they don't break down as often. It's not surprising, then, to hear previous users say they miss the simplicity of the Commodore.

New Loadstar Editor

From LOADSTAR, <http://www.loadstar.com>

Dave Moorman began programming on July 3, 1979 when his new TRS-80 arrived.

In 1986, after earning a Masters in Divinity degree and beginning a full time pastorate, Dave got a Commodore 128. By 1989, he had moved almost entirely to the C-64 platform.

"All the computer I really need"

Dave loves computing challenges. In 1992 he saw a game called Railroad Baron on his sister's 386 and immediately got to work creating 'something like it' on the C-64. The result, Sea to Sea, looked good enough to publish. So he sent it off the LOADSTAR -- and quietly awaited a rejection notice.

What he got was a phone call from Fender Tucker.

"Nothing beats the excitement of getting a complete stranger to give you money for something you have created."

After that first sale, Dave produced over a dozen unique games and amusements for the C-64. In 1999, he tried out

the various emulators that were then available for Pentium computers, and got a crazy idea.

He called Fender: "What if we were to put together a LOADSTAR-like magazine for Windows people?"

Fender replied, "Go for it!" And in January 2000, eLOADSTAR shipped issue 001 to a few interested people. Over the year, Dave improved the look, feel, and robustness of the new magazine.

Meanwhile, Fender and Judi were ready to set aside the monthly demands of LOADSTAR. Dave and Sheri talked it over and offered to serve the Commodore Community by continuing the Longest Running Disk Magazine at least through issue 211, December 2001. They also will continue eLOADSTAR for the Pentium crowd.

J&F Publishing agreed to license the Moormans' marketing company to use the thousands of great programs in the LOADSTAR Library.

Dave is dedicated to keeping the Mighty LOADSTAR Tower growing, plus seeking new markets among those who never heard of LOADSTAR or used a C-64.

Amiga Executive Update -

St. Louis and Beyond February 06, 2001

Greetings to the Amiga Family:

In Melbourne this last year, I was able to surprise many of you with several announcements that were unexpected, and we delivered on each of those.

Now we have a new show coming up in St. Louis. In a few weeks we will all learn...

Where is the AmigaOne? Will Cringley finally eat his words about a new Amiga? What about AmigaDE on PDA's, Set Top Boxes, Internet Appliances, or even desktops? Will Tom and Nicole stay together or is it over? Is there life for the AmigaOS after 3.9?

Contrary to the rumors and what many people are saying about our demise, or that we are going to have the same fate of the others before us, we have a big surprise for all of them.

I look forward to visiting with all of you in St. Louis, and I look forward to taking over the Hotel, and the surrounding area with the Amiga Spirit, and Community.

I hope to see many of you there.

There is still time, and in 6 weeks, it will all be known.

Bill McEwen, and the rest of the Amiga Team

From the Amiga Inc. website

(www.amiga.com) October, 2000

The Next Computing Revolution

The world is going through dramatic changes. Consumer Electronics companies are adding traditional computer features to their products, and computer companies are adding traditional Consumer Electronics features to their products. This leaves the consumer confused and stuck with no clear choice. This chasm of confusion needs to be crossed and Amiga is the bridge that will cross it and bring consumers, Consumer Electronics Companies, and traditional Computer companies together and heading in the right direction.

Amiga - The Beginning of The Future

In 1985 the Commodore Amiga A1000 was introduced to the world and nothing like it had been seen before. People were amazed. Over the next 15 years individuals, families, and companies purchased millions of Amigas and were introduced to a whole new world of computing. Since then Amiga users have changed computing for everyone with revolutionary ideas, creative implementations and innovative advancements, all brought about by the elegance and simplicity of the Amiga computer.

It has taken the rest of the industry 15 years to catch up to Amiga. Now they never will.

The New Amiga

Computing today should not be hardware specific, but content specific. A computer should work the way you think, not the other way around. It should be easy to use and scalable, meeting everyone's needs. There should be easy access to what is important to the user without the user having to worry about the constraints or compatibilities of the operating system.

The current strategy of offering different operating systems for different devices is a flawed strategy and cannot succeed. There must be a cohesive, scalable operating system that provides brilliant multimedia performance on all supported products, from cell phones to multi-processor servers.

Amiga, Inc. with our partner The Tao Group, Inc., is developing a new breed of operating system that is so radical and revolutionary that it requires a new name: The Digital Environment (DE).

It is time for a scalable, easy-to-use Digital Environment that will foster the creation of content and provide convenient, intuitive features and functions.

Features of the Amiga Digital Environment

* "Write Once- Run Everywhere" -- This statement has been the mantra for

tired and weary developers. The new Amiga is being designed to make good on this promise and present developers with unprecedented options and flexibility.

* *Scalability* - Because the new Amiga DE is less than 5 megabytes, Amiga is scalable, and can fit on a cell phone and scale up to multi-processor servers. All content created for Amiga is 100% binary compatible. This means that developers are able to create content and applications and deal with the markets they want to target.

-- For the developer, the new Amiga is 100% binary compatible across all platforms supported. This means that an application can truly be written one time and can operate in hosted environments (running on top of or through another OS) such as Windows, Windows NT, Linux, Windows CE, Palm, Epoc, OS/9, iTron, QNX, VXWorks, and others. This fact alone has the potential to redefine the development of computer games. Amiga can also operate the entire system with native versions of the DE for PPC, x.86, StrongARM, ARM, Hitachi SH, NEC 85x, M*Core, Mips, and others.

* *Amiga developers are able to use Java, C, C++, or Amiga VP Assembler in creating their applications.* This flexibility is already gaining new market acceptance as over 15,000 Amiga Software Development Kits

(right) have been sold, and with the new additions of 3D, Sound, and User Experience API's it is sure to gain even more ground as the developers platform of choice.

Benefits to Hardware OEMs

For the hardware OEM, Amiga is an operating system that provides a scalable solution and the ability to select and utilize their hardware of choice. These choices include a minimum of 25 different CPU architectures that encompass at least 10 different operating system environments. Of course, offers multi-processor support, too.

Additionally, the Amiga DE can be ported to new CPU architectures in as little as 2 to 3 months as opposed to the massive 6 to 18 month effort it would take to port a traditional operating system to a new CPU architecture. This allows great cost savings and flexibility to the hardware OEM.

Another benefit for the hardware OEM is content. As new devices enter the market, and new markets are being developed it is in the area of content that the Amiga can really enhance an OEM's offering. By using the Amiga DE hardware OEMs can offer much greater value added products than those who don't use the Amiga DE. Just imagine a set-top box manufacturer who can offer not only full Internet and home entertainment features on their system, but stunning games, Java applications, video editing capabilities, and programming tools, too.

The Amiga DE will allow the OEM to do all these things with one cohesive, scalable operating environment that communicates between itself and any other device connected to the system that uses the Amiga DE.

Amiga Community

Amiga, Inc. is not the typical emerging company: It is already a worldwide recognized brand name that is synonymous with ease of use and multimedia. The Amiga development community numbers in the tens of thousands and it releases software and hardware on a near daily basis.

Many current AAA gaming companies, such as Psygnosis, Sierra, Westwood, Lucas Arts, Team 17, and others, received their start on the Amiga platform and are excited by the prospect of the new Amiga. Developers remember fondly the heyday of the Classic Amiga and look forward to a day when software development will be as simple and elegant as it was on the Classic Amiga. Many users remember the way their Amiga rarely crashed, booted quickly, presented seamless animation and made efficient use of resources. The new Amiga will bring all this back.

With the release of the Amiga SDK, developers around the world have begun creating applications. In addition to the 45,000+ titles that are already available for the Amiga - many of which are likely candidates for porting to the new Amiga--the products listed below are in the process of being ported directly to the new Amiga Digital Environment.

Amiga DE - Committed Applications as of 9/14/00

Video / Imaging

Art Effect
Motion Studio
Candy Factory
Fantastic Dreams
Taifun
Art Studio NG

Personal Productivity

Burn It Pro
Amiga Writer
Espial Assistant
Papyrus
Star Office

Desktop Publishing

Pagestream
Repro Studio NG

Internet / IM / E-mail

Messenger Force
Ebox
IMAP Force
Escape

Development Tools

Storm C
Rebol

Database Tools

Pointbase Embedded Database
ThinAccess Realtime DB Access

Games

Soldier of Fortune
Sin
Heretic 2
Freespace
Alien Nations
Alien Nations Mission
Shogo
Worms Armageddon
Battle for the 3rd Moon
Evils Doom
Exodus: the Last War
The Feeble Files
Earth 2140
Simon the Sorcerer
Gorky 17
Foundations Edge
Operation Counterstrike
Magick
Dweebs
Die Volkor
Gilbert Goodmate
Bubble Heroes
Earth 2150
Dafel: Bloodline
Tales of Tamar
Aqua
Alien Nations 2
Majesty
Pacific Wars
Air Kit Dogfight
Sudden Strike
Jagged Alliance 2
Jagged Alliance Mission
Nomads
Wizzardy
Echelon: The Storm
Claws of the Devil
Blood 2
Descent: Freespace
Homeland
Payback
Wipeout 2097
Dark Soul
Last Patriot
Scavengers
Hell Squad
Escape Towards Unknown
Virtual Ball Fighters
Eat the Whistle
Command and Conquer
Great Nations

Lambda Sector
Moonbases
Salvage
Phoenix
Mario 64
Dead Walk
Foundation: Directors Cut
Holy Trinity
Final Fantasy 5
Recovery 2190
Grand Theft Auto

The AmigaONE

Very soon the world will meet the next Amiga (the AmigaONE) and it will amaze users with its simplicity and power. The world will finally have an operating system (or Digital Environment) that gives the worldwide digital revolution the means to address all users.

Software developers will be able to write once and run their software on dozens of different operating systems. The Amiga DE can be scaled to fit the needs of any application, from cell phone messaging systems to full-scale university databases.

Users and developers need only become familiar with one operating system that works on everything from their car information center to their workplace desktop to their home entertainment system. No more cumbersome software porting, no more conflicting standards, no more compatibility problems. It will work on your home computer, your work computer, your school computer and your handheld computer, regardless of the OS the device currently uses.

The Amiga will bring the power of the silicon chip into the hands of the masses because it will be easy and painless to use. It will be something even technophobes can understand. It is about simplicity and giving power to the users who are tired of having to think like a computer to use one. Amiga will make the computer think and act like we do-effortlessly-and make computing and using computers an easy, coherent and enjoyable experience.

We hope you'll come along for the ride.

Bill McEwen and the Rest of the Amiga Team

The Amiga - Still a good choice

by Eric W. Schwartz, AmiTech User Group, Dayton, OH

I was reading a recent issue of "Wired" (yes, yes, I know) for a specific article. The article talked about what they called "retroactivists" -- people who prefer one or more certain computer platforms, supposedly left behind in the rush of Windows upgrades and CPU after CPU. Not surprisingly, the Amiga was generously mentioned, and in a reasonably favorable light, too. The article starts off with a software engineer, who advocates users to find a good, capable, and stable computer operating system and software, learn how to use it well, and stick with it for ten years. While that may be a little extreme, I can understand the sentiment behind it. Especially in the Windows world, there's a climate where constant updates, upgrades, and spending are viewed as necessary, whether it actually is or not. More people are living on the bleeding edge, with all the instability and outlay of cash it entails. It all got me thinking about my own views on the subject.

I am an Amiga user. With the exception of a game console or two and a Palm handheld device, Amigas are the only computers in the house. I'm not so

much of a fanatic that I would turn down a Windows box or I-Mac if it were offered to me, but I'm certainly in no hurry to get one. Through experiences with all platforms, I prefer the Amiga, and I'm used to how (and how well) it performs. If there was something I required that the Amiga could not provide, then I would get a different machine, but that need has not sprung up yet -- and I do a lot with my Amigas. I do graphics for screen, print and Internet. I create and edit animation and video on my hard drive. I surf the Internet, I burn my own CD-ROMs, I play a few games, and I publish this newsletter. Your average 666Mhz Pentium user might be lucky to get that much use out of their machine. I am not someone who stands still in the technology race -- I have upgraded my Amigas with bigger and better CPUs, drives, and the new 3.5 OS. I view it as a more enlightened approach, as it's more a matter of what use I can get out of it, as opposed to just going faster, so that new game can run that tiny bit smoother.

Outsiders often view Amiga users as dinosaurs -- as fanatical throwbacks who refuse to accept the world has left them behind. I don't see it that way. Amiga users have seen how good per-

sonal computer technology can be when used well, and they know there's better out there than the likes of Windows. They know there's no excuse for using 100 percent of a computer's RAM, hard drive, and CPU, only because it's available to use. They know crashes, corruption, and recurring re-installation do not have to be a way of life. They know that if their Amiga provides what they need now, it's not necessary to downgrade just to get some extra speed, games, and access to websites that were written with a specific browser in mind.

In the meantime, we sit, and hope against hope that the new Amiga Inc. will strike gold again, and bring us the newest, bestest platform that will make the Windows PCs of today look as clumsy and dated as the IBM PCs did against the Amiga 1000 back in 1985. That may be a little unrealistic in today's environment, but it's all a matter of a little magic. It can potentially be captured again. If it looks like a platform we can keep using for the next ten years, we know we'll really have something, whether the world at large realizes it or not.

Synth Sample Author Revealed

The following quotation is from information posted in October, 1998 by Georg Feil in the comp.sys.cbm newsgroup in response to a question seeking the author of Synth Sample.

"Hi... As others have already said that was "Synth Sample". I wrote it around '84 or '85 as a demo for my music editor called "Synth". I was going to University of Waterloo at the time and just gave it to a friend of mine ... didn't even bother to put my name on the thing. Anyway it soon started cropping up all over the place.

I never made any sequels with music, although there was something with annoying sound effects done in the same style as Synth Sample that I posted in the beginning of 1994. So any music you see that looks like a sequel or has graphics added was done by someone else.

Anyway the computer bust of 85/86(?) hit and Synth was never marketed. I had been negotiating with a software company but they went out of business. By that time I was working on a new version of Synth that was a sequencer program for Midi instruments (called MSS), and had joined my first band. I've been in a couple other bands since then, the last one was called SugarPush and we came close to getting signed (but then broke up last year... see <http://www.sgl.crestech.ca/sugarpush>). So I guess you can credit that little program with launching my musical career...

Feel free to copy Synth Sample as much as you please. If someone twists my arm enough I might even get up the energy to post the actual Synth program and/or MSS (Midi Switching System) on the Web. To be honest there are much more user-friendly C64 music editors out there, although few are ultimately as flexible. Synth is also ideal for adding music and sound effects to C64 games, and can automatically attach its player module plus songs/sounds to your game (that's how Synth Sample was built in the first place).

For those interested in trivia I've attached a list of the songs that were in the

original Synth Sample, and how they got there.

Thanks, **Georg Feil**
<http://www.sgl.crestech.ca/~georg>

Space Geodynamics Laboratory
E-mail: georg@sgl.crestech.ca
CRESTech
Phone: (416) 665-5458
4850 Keele St. North York Ont. Canada
M3J 3K1 Fax: (416) 665-1815

Synth Sample songs:

1. **Theme from Stationary Ark**, a nature show on PBS.

Sorry, I have no additional info on this one. It appears on Synth Sample as interpreted by a friend of mine. I didn't know what it was myself until I happened to see Stationary Ark one time. This song does not play properly on some C-64's where the SID chip filter is calibrated differently.

2. **Saturdays in Silezia**, by Rational Youth.

A fluffy pop song I taped off the radio. Seemed simple enough to render on the C-64.

3. **Spiral**, by Vangelis, from his Spiral album.

4. **Tubular Bells**, Mike Oldfield, from his Tubular Bells album.

I just thought this was ultra cool. Used as the music to The Exorcist, unfortunately.

5. **Theme from Magic Shadows**, written by Harry Forbes.

Magic Shadows was a half-hour movie show on TV Ontario, sort of the Canadian equivalent of PBS. They'd show old movies in half-hour installments. I have no idea if there's an album. I taped it off the TV.

6. **Theme from A Clockwork Orange**, by Wendy Carlos, on the Clockwork Orange Soundtrack. This is a synthesizer adaptation of Prelude for Flutte Trumpets from Henry Purcell's opera "Liber-tine". Apparently there are two versions of the soundtrack album, one with more of Carlos' synthesizer stuff on it than the other.

7. **Oxygene II**, by Jean-Michel Jarre, from his Oxygene album.

Great stuff, much better in the original form than the cheesy Synth Sample rendering. I recommend Jarre's first three albums: Oxygene, Equinoxe, and Magnetic Fields, in that order. His more recent ones largely suck (IMHO). (Actually instead of spending your money on old JMJ why not get something more up-to-date, like almost anything by the Orb, older Future Sound of London (e.g. ISDN), Funki Porcini's first album, stuff off the Ninja Tunes label, or anything by Kruder & Dorfmeister...)

8. **Canon in D minor**, by Pachelbel.

This is a very popular classical hit. I transcribed the notes from a Transactor or Compute! article (this is the only Synth Sample piece not transcribed by ear).

9. **Enola Gay**, by Orchestral Manoeuvres in the Dark.

From one of their earlier albums -- not the first, maybe the second or third.

Good luck and happy listening.
Georg."

The original Synth Sample appears on TPUG disk (C)TB in September, 1984.

The original Synth Sample was combined with a graphics program called Swish and the resultant graphics/music demo program appeared on 4 TPUG disks where it was called either Swinth or Laser Show. As Georg states in his newsgroup mail, he is not identified on Synth Sample nor is he identified on Swinth/Laser Show although the author of Swish, the person who combined the two programs and the person who uploaded Synth Sample to Compu-Serve are all given credit in the opening screen of Swinth/Laser Show.

Swinth/Laser Show is available on:

(C)AAO - September, 1987

(C)I18 - November, 1987/ February, 1988

(C)WOC V87

(C)EXPO88

All the TPUG disks mentioned are still available from the TPUG library.

Selecting a Programming Language Made Easy

Daniel Salomon &
David Rosenblueth

Department of Computer Science,
University of Waterloo, Ontario

With such a large selection of programming languages it can be difficult to choose one for a particular project. Reading the manuals to evaluate the languages is a time consuming process. On the other hand, most people already have a fairly good idea of how various automobiles compare. So in order to assist those trying to choose a language, we have prepared a chart that matches programming languages with comparable automobiles.

ASSEMBLER

A Formula I race car. Very fast, but difficult to drive and expensive to maintain.

FORTRAN II

A model T Ford. Once it was king of the road.

FORTRAN IV

A Model A Ford.

FORTRAN 77

A six cylinder Ford Fairlane with standard transmission and no seat belts.

COBOL

A delivery van. It's bulky and ugly, but

it does the work.

BASIC

A second-hand Rambler with a rebuilt engine and patched upholstery. Your dad bought it for you to you to learn to drive. You'll ditch the car as soon as you can afford a new one.

PL/I

A Cadillac convertible with automatic transmission, a two tone paint job, white wall tires, chrome exhaust pipes, and fuzzy dice hanging in the windshield.

C

A black Firebird, the all-macho car. Comes with optional seat belts (lint) and optional fuzz buster (escape to assembler).

ALGOL 60

An Austin Mini. Boy, that's a small car!

Pascal

A Volkswagen Beetle. It's small but sturdy. Was once popular with intellectuals.

MODULA II

A Volkswagen Rabbit with a trailer hitch.

ALGOL 68

An Aston Martin. An impressive car, but not just anyone can drive it.

LISP

An electric car. It's simple but slow. Seat belts are not available.

PROLOG/LUCID

Prototype concept-cars.

Maple/MACSYMA

All-terrain vehicles.

FORTH

A go-cart.

LOGO

A kiddie's replica of a Rolls Royce. Comes with a real engine and working horn.

APL

A double-decker bus. It takes rows and column of passengers to the same place all at the same time. But, it drives only in reverse gear, and is instrumented in Greek.

ADA

An army-green Mercedes-Benz staff car. Power steering, power brakes and automatic transmission are all standard. No other colors or options are available. If it's good enough for the generals, it's good enough for you. Manufacturing delays due to difficulties reading the design specifications are starting to clear up.

2001 Commodore VIC-20 Programming Challenge

Note - though just past the 'deadline, we post this item trusting that Bruce Clarke is still open to suggestions.

I will award two prizes to the best two original programs for the Commodore VIC-20 computer that I receive by e-mail before midnight June 15, 2001:

1st prize winner - \$50 US

2nd prize runner-up - \$25 US

All entries will be in the public domain. I'm doing this to encourage the use of the VIC-20 and simply because I think the VIC-20 has a lot of potential that was rarely tapped by most programmers when the VIC-20 was in common use.

Background: The Commodore VIC-20 was released in 1980 and was the first mass-marketed home computer sold in North America for under \$500. This was at a time when an Apple II with 48 KB and a single floppy drive was considered a very powerful machine.

The VIC-20 cost about \$499 initially, though the price dropped pretty quickly to around \$299. For another \$100

you could buy a tape drive and some extra RAM, and have yourself a true computer. Maybe the VIC-20 wasn't a very powerful computer, but you could play games, word process, print crude graphics, maintain some small databases, etc. For many teenagers the VIC-20 was our introduction to computers and programming.

The VIC-20 had a 6502 8-bit processor (back then fairly powerful), came with a whopping 5 kilobytes of RAM, and had a colourful display of about 23 by 22 characters (wow). Although most of the games for it were crude character-tile based maze games (the VIC did not have sprites), there were a few real gems written for it. One in particular was Omega Race - I played that game for hours as a teenager. There were also some decent conversions of games like Pacman, Dig Dug, Defender, Centipede, Donkey Kong, etc. Commodore stopped production of the VIC-20 in early 1985 to concentrate on their Commodore 64 computer.

There are good emulators available for a variety of platforms: I recommend the VICE emulator at this site (though

there are many others that work well):
<http://www.cs.cmu.edu/~dsladic/vice/vice.html>

Various utilities and other software are available at:
<http://ftp.funet.fi/pub/cbm/vic20>

Over the last few years, the emulation scene has witnessed a 'rebirth' of many old 'classic' video games and computers. Enthusiasts have recently written some surprisingly good games for systems such as the Atari 2600, Vectrex, Colecovision, etc.

I'm announcing this contest to develop new original software for the Commodore VIC-20. There will be a prize of \$50 US awarded to the entry that I judge best, and \$25 US to the second best runner-up. There are no specific criteria - other than that the winning entries impress me more than any other entry.

Rules:

The contest is open as of December 11, 2000. All entries must be received by midnight Pacific Time, June 15, 2001. Entries will be accepted as e-mail attachments sent to: brclarke@islandnet.com

I'll be providing the prize money, and therefore I think it only fair that I be the sole judge. I have an EBay ID: brclarke@islandnet.com that you can view for my feedback. I can be counted on to award the prize fairly and promptly after June 15th.

The entry can be any kind of program: animation, game, educational, utility, graphics/sound demonstration, etc. If the entry is a demo it must have a maximum run time of 5 minutes. (IE. if you write a ray-tracing or fractal graphics demo I don't want to have to wait all night for it to run!)

The entry can be written in any language easily found for the VIC-20: BASIC, assembler, C, FORTH, etc. If written with an add-on language such as one of the many 'expanded' BASICs, it must still run on an 'unexpanded' VIC-20.

Entries must run on a majority of all available VIC-20 emulators and an actual VIC-20 with no 'unusual' modifications other than RAM expansion.

All entries (including source code) must be in the public domain, and may be copied and distributed, as long as the contribution is in its original form. No entries (including bug fixes) can be accepted June 15, 2001. Entries must be sent in via Internet along with information regarding programmer name, source code, and program instructions. Contributions with inflammatory or derogatory content (such as racist) will not be accepted.

Any questions can be addressed to brclarke@islandnet.com

This page will be updated on a regular basis until the end of the contest.

<http://www.islandnet.com/~brclarke/vic20.htm>

Subject: Re: VIC-20 Programming Contest
Date: Wed, 13 Dec 2000 11:51:07 -0800 (PST)

From: brclarke@islandnet.com (Bruce Clarke)

Reply To: chorny@ilap.com (chorny)

The fourth paragraph under 'rules' states that the program 'must still run on an 'unexpanded' VIC-20' while the fifth paragraph states that the program must run on a standard VIC-20 with no 'unusual' modifications except for RAM expansion.

Must the VIC be unexpanded or is memory expansion allowed?

Sorry - I will fix that tonight.

I meant a VIC-20 with a RAM expansion, but no other 'modifications' such as an expanded version of BASIC, etc. Its very easy to track down a standard 3K/8K/16K RAM expander for the VIC, and emulators can emulate a RAM expansion, but an expanded BASIC might be very hard to track down.

As you probably know, back in the early 80s RAM expansions for the VIC were fairly pricey, and so a lot of programmers were forced to work with very tight space limitations. Now RAM is much cheaper and easier to provide for the VIC, so hopefully we'll see some good programs come out of this...

Can we reprint your write-up in our club newsletter (the TPUG Newsletter)?

Absolutely - the more publicity the better!

Bruce Clarke
brclarke@islandnet.com Victoria, BC Canada

... continued from page 1

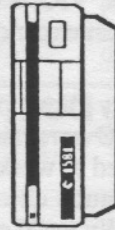
with a reasonable speed modem) which would be closer to what a person will likely encounter on the job. Because of this reason I myself would recommend the MS-DOS PC. However I would STRONGLY recommend the C64 (or similar computer) for anyone wanting to do more then run pre-packaged programs, such as doing their own programming, and/or wish to connect their computers to unusual devices such as robotic experiments.

I am a tinkering type of person. I am not afraid to take things apart to see how they work, or to see how I may be able to improve upon them. My only regret is that sometimes (not often) they may not work again when I re-assemble them. I have used my C64 to control projects that I have built, such as a burglar alarm system, a 3 dimensional table, and robotic experiments. Do not get me wrong, I feel that because these computers are part of our past, part of our history, it is well worth our time, effort, and money to use and keep alive the spirit of the old computers. If you are pondering over which computer you are going to buy, then you alone must answer the questions. Why do I want this computer? Will it fill my needs?... and of course ... Is the price right?????

Tom Luff

For Sale From TPUG


(Your Membership Dollars At Work)




New Commodore 1581

Floppy Disk Drives
Members.....~~\$200.00~~ \$ 50.00

(complete with power supply, manual, demo disk, & cable)
Non-Members add membership fee
at our Disk Order Form, found at the rear panel of News Letter

 5.25" Blank Disks (DSDD)
SOLD OUT

 3.5" Blank Disks (DSDD)
10 disks, \$3.50_{per} (limit of 30 disks)

PageSetter.....\$5.00 (each 100 p.p.s. for the Group)

Commodore 1581 CP/M disk ver3.0
\$13.50

128 Tutorial & Demo Disk \$5.00

Quantities are limited - Sales restricted to Members

Please add 10% for shipping and handling-Int. add PST

All prices listed in Canadian funds

Toronto Pet Users Group
P.O. Box 48565
3605 Lakeshore Blvd. W.
Etobicoke, Ontario
M8W 4Y6

Volume 6, Number 9