TPUG Newsletter

Views and News of Toronto Pet Users Group

c/o John Easton, 258 Lake Promenade, Etobicoke, Ontario, M8W 1B3 phone (416) 251-1511

Spring— 2011

From the President -

Winter has quickly turned into spring. The warmer weather presents a challenge for us Commodore users. You see, during the cold months we tend to hibernate. One is pretty much confined to the indoors, which gives the average Commodore user plenty of time to tinker with the 8-bit hardware. Warmer months mean more outdoor activities and our vintage Commodores are usually left very lonely until the evening hours.

As summer beckons us, I get the chance to reflect back on TPUG's past successes. World of Commodore was again a huge success. We switched venues this year to the hotel down the street and the crowds were just as big as the prior year. The main room was filled to capacity with eager clubbers and non-clubbers alike. Commodore enthusiasts from all over North America were taking in the demos and checking out the vendors. We had great demos of new products and innovative solutions for all of your Commodore needs. We had sales tables full of wares (including a recent donation of a number of PET computers) and of course the free table (which was actually on the floor this year!)

We started the year out right with our Annual General Meeting in January. The meeting was held in January to set the direction for the rest of the year and look back at the prior year. The various reports were presented, which included current finances, membership, shows, chapter meetings, website, TPUG store, TPUG library and the President's report. All together, these reports and the successful night we had show that the club is healthy and heading in the right direction.

This year at the AGM, I mentioned the importance of preservation. The TPUG library CD was an incredibly important part of the preservation of the history of TPUG. We can't stop there, however. As papers get misplaced, disks fail and memories fade, we need to take good care of the history of our great club.

Back in February, Zbigniew Stachniak from the York University Computer Museum came and spoke on the efforts that they're putting into preserving Canadian computing history. We need to do our part as well. Pre-serving TPUG's history provides us with a road map of where the club has been, what we did as a club and why we did it, and using that as a guide, like a map, it gives us an understanding of where we want to go as a club.

I'll give you an example of what I mean. Many years ago I used to play a game on the VIC-20 called Voodoo Castle. The game was a cartridge based text-adventure game where you are a curious character navigating your way through a haunted castle. Text adventure games, like Voodoo Castle and Zork, are games where you type in a text and the computer responds back. In Voodoo Castle if you typed `north', you would move north into another room and then be given a description like "you are in a large ballroom with a grandfather clock".

When playing Voodoo Castle, you get a mental picture of the layout of the castle and the various rooms and attributes of each room. Years ago, I made up a rather elaborate map of the castle on a piece of paper detailing all of the rooms, somewhat like a blueprint or a layout. Using this map, I knew all of the rooms that I went into. While playing the game years later I had difficulty navigating the castle, but I remembered the map that I sketched out long ago, so I pulled it out and was able to navigate the castle because I had preserved the map. Looking over that map and notes that I had written gave me an understanding of how I conquered the Voodoo Castle back in the day.

This year, part of TPUG's efforts to preserve its history will involve scanning our past newsletters. As you know (because you're reading this), TPUG puts out a newsletter at least a couple times a year but did you know that TPUG has put its newsletters out since the early 80's? We still have the paper copies but these need to be scanned to preserve them and fill in the holes that exist in the archive section of our website. In order to fill in the gaps we need to determine first how many newsletters TPUG put out and what year/month/season they were, then we can begin to scan and preserve this vital part of TPUG's history.

Member Information

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

Website: www.tpug.ca
e-mail: info@tpug.ca

Membership Rates

**$15 per year
else, with e-mailed Newsletter,
$10 per year**

Board of Directors

President	Greg VanLaere
Vice President	John Easton
Secretary	Ernie Chorny
Membership Sec.	Ian Colquhoun
Treasurer	Ernie Chorny
Director	Leif Bloomquist
Director	Tom Luff
Director	Ian McIntosh
Director	Joe Palumbo
Director	Tom Williams

Librarians

Head Librarian	Greg VanLaer
Amiga	Ernie Chorny
C128	Tom Luff
CP/M	Ernie Chorny
GEOS	Joe Palumbo
C64	Joe Palumbo
C64 Education	Greg VanLaer
Comal	Ernie Chorny
Plus/4	Ernie Chorny
Vic 20	Ernie Chorny
PET/CBM/SuperPET	John Easton

Support

Mail	Tom Luff
Telephone	John Easton
Disk Orders	Librarians
Member Records	Ian Colquhoun
Meetings	Leif Bloomquist
and	Ernie Chorny
Shows	Tom Luff & Ernie Chorny
Webmaster	Leif Bloomquist

Newsletter

Editor   John Easton (416) 251-1511

Meeting Schedule

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 next 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 - **Leif Bloomquist** (416) 737-2328 leif@schemafactor.com
or   Ernie Chorny (905) 279-2730 chorny@tamcotec.com

From your Editor ...

Another year, another issue … well, that is to say, we're still here and this newsletter attempts to keep us all up to date on current Commodore-related topics of interest.

We've reviews from the WoC 2010 presentations in the article by our own Ian Colquhoun about scanning documents … cheap and easy. We have a report from the local Maker Faire (geek heaven) by our own guru of geekdom … Leif Bloomquist..

And, you might notice the reference to the early chicklet keyboard PET in the included comic strip.

And, speaking of PETs, I really should include this photo, taken at the WoC 2010 … Chiron Bramberger's immaculate PET 4032 was running **Space Invaders** to the delight of the young attendees. Ah, yes, I remember it well … to make this machine game more appropriate to a Sunday School application, we re-wrote the program to change the 3-character 'Invaders' to something like SEX, SIN … whatever. And the 5-character `Mystery Ship' that zips across the top of the screen occasionally, we re-named SATAN. All in all, a fun re-work to produce the acceptable application … **Grace Invaders**. Ah, Memories ...

TPUG Logo Contest

Well now, your local executive have decided that perhaps we need a universally recognizable logo to identify this, our own familiar **Toronto Pet User's Group**, to the rest of the world. Hmmm, we've done fine so far, (not to mention the occasional e-mail from Pet (as in dogs 'n cats) Advocate groups) but just in case, we're accepting submissions of any ideas for a new **LOGO**

Contest rules:

1. Logo can be black and white or colour.

2. Design must be easily converted to black and white or monochrome.

3. No tagging of artwork. (That is, don't even think of placing your personal ID within the artwork.)

4. Multiple submissions are allowed.

5. Any format accepted, drawn or computer rendered. Computer rendered artwork should be in an easily readable file format (TIFF, JPEG, etc.).

6. Should be usable on letterheads, business cards, newsletters and signs.

7. Contest is open to all and voting is restricted to *paid up TPUG members* as of the date of the contest close.

8. All submissions become the property of TPUG.

9. The TPUG board reserves the right to exclude unsuitable entries and/or to break a tie in the voting.

Send entries to our mailing address, else, via e-mail to *contest@tpug.ca.* Contest close— Aug 31

Acceptable submissions will be published for voting in the fall TPUG newsletter

Winners will be announced at a later date

Prizes will be awarded for the best design.

1st prize: $100 Gift Certificate for Jim Brain's store *Retro Innovations*
2nd prize: TPUG library CD
3rd prize: One year TPUG membership

Joe's World 11   Searching in Memory

Here is a program to search for data and/or text in memory. Note that printers differ; this works for the printer I use.

```
0 rem "searching" by Joe Quittner
10 ? cH(144)cH(14)
20 pO 56,78:clr:rem protect 20k+
21 def fna(i)=i-256*fnb(i):def
fnb(i)=int(i/256)
22 def fnc(i)=peek(i)+256*peek(i+1)
30 fO i=820 to 1023:read a:if a>255 tH 5000
31 pO i,a:nE:stop
32 data 230,3,208,2,230,4,96:rem"Increment
memory locations 3/4
100 :
997: pO 204,0:get g$:if g$=""tH 997
998 if pE(207) tH 998
999 pO 204,1:reT
4000 rem "For the printer interface I use
4010 ?"  Printer on?":goS 997:? g$
4012 p=0:if g$<>"y" tH reT
4020 p=1:if p1 tH reT
4021 p1=1:close 5:close 7:open
5,4,5:open7,4,7
4030 fO i=1 to 10
4032 pR5,cH(27)cH(108)cH(5):rem left margin=5
4034 pR5,cH(27)cH(81)cH(77):next:rem right
margin=77
4039 ?"  Set interface to Junior":goS 997:reT
5000 :rem" SEARCH IN MEMORY
5001 goS 4000:?"  Text(t) or Numbers(n) mode?"
goS 997:? g$
5002 if g$="t" tH 5010
5003 if g$="n" tH 5005
5004 gO 5000
5005 :?" Search for which numbers? (end=999)
5006 :fO i=679 to 767:? i-678;input a:
if a=999 tH k=i-679:pO2,k-1:gO 5008
5007 pO i,a:nE:stop
5008 :?" Searching for ";:fO i=1 to k:?
pE(678+i);nE:?:gO 5020
5010 :?" Searching for which text?":input
a$:k=len(a$):pO 2,k-1
5012 :fO i=1 to k:
b$=mid$(a$,i,1):b=asc(b$+cH(0)):pO 678+i,b:nE
5014 ?" Searching for ";:fO i=1 to k
5015 ? cH(pE(678+i)): nE:?
5020 :?" In memory search from, to
(return=20000,40959)
5022 f=20000:t=40959:input f,t:t=t-k
5024 pO 3,fna(f):pO 4,fnb(f)
5026 pO 5,fna(t):pO 6,fnb(t)
5030 data 164,2, 177,3, 217,167,2, 208,8,
136, 192,0, 16,244, 162,111, 96
5032 data 165,4, 197,6, 208,9, 165,3, 197,5,
208,3, 162,222, 96
5034 data 32,52,3, 162,0, 240,217
5040 sys 827
5042 if pE(781)<>111 tH 5050
5044 a=fnc(3):? stR(a);:if p tH pR7,stR(a);
5046 b=fnc(5):if a=>b tH 5052
5048 c=a+1:pO 3,fna(c):pO 4,fnb(c):gO 5040
5050 : if pE(781)<>222 tH ?:stop
5052 : ?:?" END OF SEARCH":if p tH pR7
5054 end
5999 data 999

820:inc 3 822:bne to 826 824:inc 4 826:rts
827:(also from 864) ldy 2 829:(also from 839)
lda(3),y  831:cmp 679,y 834:bne to 844
836:dey 837:cpy#0 839:bpl to 829 841:ldx#111
843:rts 844:(from 834)lda 4 846:cmp 6 848 bne
to 859 850:lda 3 852:cmp 5 854:bne to 859
856:ldx #222 858:rts 859:(from 848 or 854)jsr
to 820 (increments 3/4) 862:ldx #0 864:beq
always to 827
```

Line 5001: If you are searching for 111 in text mode, then the program will look for 49 49 49. But if you are in numbers mode, then the program will look for 111.

Lines 21 and 22 are definitions of functions a, b and c.

Line 30: Machine Language (m/l) will be stored in the area reserved for tape storage; I only use disks. Data statements can be anywhere in the program and need not be consecutive. See also line 5999.

Line 31: The data numbers will be stored as long as A is not greater than 255.

Line 5030: 164=LDY from page zero, 177=LDA (indirect),Y, 217=CMP absolute,Y, 208=BNE, 136=DEY, 192=CPY immediate, 16=BPL, 162=LDX immediate, and 96=return to BASIC.

Line 5032: 165=LDA from page zero, 197=CMP with zero page.

Line 5034: 32=goto start of a subroutine, here to increment (add 1 to content) 3/4

Line 32: 230=INC on page zero. Byte 3 is incremented, and if that results in a zero (255+1=0) then byte 4 is also incremented; this is the equivalent of a carry.

The search via machine language starts with memory location 827, the .Y register (.Y in 782) is loaded with whatever is in 2, actually the length of the search item minus 1, then 829 loads the accumulator (.A in 780) indirectly with (content of 3/4 plus .Y). In 831 .A is compared with (content of 679 plus .Y). If these are not equal, which is most of the time, then the next memory byte will be looked at from 844, otherwise, in 836 .Y is decremented (1 is subtracted from it). In 837 the new .Y is compared with zero. 0 is a plus number too, and, if plus, then the program jumps back to 829, searching for another match. But if .Y is already 255 (=negative) then the searched-for item has been found, in 841 the signpost .X (=781) is set to 111, and the program returns to BASIC, following sys 827, line

5042. In that line the signpost .X is looked at. If .X is not 111 then the program will go to 5050. If there is 111 in .X then line 5044 sets a=the content of 3/4, converts the number `a' into a string in order to get rid of one space per number, shows it on the screen, and puts it on paper too if the printer is on. There may be more places in memory where the same search item is located, therefore, in line 5046 `a' is compared with the end search byte contained in 5/6. If `a' is already at `b', or even higher, then that's the end of the search. Otherwise, c=a+1, c in line 5048 is put into 3/4, and the search for some more hits, if any, continues.

If the comparison in 834 shows that the two are not the same, then from line 844 at first the content of byte 4 is compared with the content of byte 6, and in case these are equal then also 3 is compared with 5. If both comparisons result in equality then this indicates the end of the search, the program sets the signpost .X to 222 in 856, then goes to BASIC lines 5042, 5050 and 5052. There it

puts on screen "END OF SEARCH" and the program ends.

If either one of the comparisons in 846 and 852 shows inequality, then the subroutine starting at 820 increments 3/4 and the search continues from, again, 827.

Because machine language programs are much faster they should be used in cases of repetative operations, such as searching, but otherwise BASIC programs are safer and easier to use. Basic has many error-checking procedures built in that help considerably in the building and running of programs, but all these use extra time.

Before running any new machine-language program be sure to first save everything just in case there is an error in the program which may require switching off the computer, and then back on again, with the resulting loss of everything that has not been saved.

Joe Quittner, April 17, 2008

Joe's World 12 .... Sorting

Note that printers differ, these programs work for the printer I use.

Sorting Numbers:

```
0 rem" sort numbers" by Joe Quittner
10 ? cH(144)cH(14):dim c(1000):?" Enter Num-
bers":gO 1100
997 pO 204,0:get g$:if g$=""tH 997
998 if pE(207) tH 998
999 pO 204,1:reT
1100 ? "  To end input, enter 99999
1110 fO i=1 to 1000:? i;:input c:if c=99999
tH 1120
1112 c(i)-c:nE:stop
1120 :?:n=i-1:? n"numbers entered":gO 5100
4000 rem" for the printer and interface I
use:
4010 ?" Printer on? ";goS 997:? g$
4012 p=0:if g$<>"y" tH reT
4020 p=1:if p1 tH reT
4022 p1=1:clO5:clO7:open 5.4,5:open 7,4,7
4030 fO i=1 to 10
4032 pR5,cH(27)cH(108)cH(5):rem left mar-
gin=5
4033 pR5,cH(27)cH(81)cH(77):nE:rem right
margin=77
4039 ?" Set interface to Junior":goS 997:reT
5100 :rem " Sort numbers:
5110 fO i=1 to n-1
5112 :a=c(i):b=c(i+1):if a=<b tH nE:gO 5120
5114 c(i)=b:c(i+1)=a:i=i-1:if i<1 tH i=1
5116 gO 5112
```

```
5120 :fO i=1 to n:? stR(c(i));:if p tH
pR7,stR(c(i));
5122 nE:?:if p tH pR7
5123 end
```

Sorting Strings:

Here is a program to enter strings into memory, then uses machine language to fast sort them, leaving the result in memory, then prints the sorted list of strings.

```
0 rem "sort strings" by Joe Quittner
10 ? cH(144)cH(14)
20 pO 56,78:clr:rem protect 20k+
21 def fna(i)=i-256*fnb(i):def
fnb(i)=int(i/256)
22 def fnc(i)=pE(i)+256*pE(i+1)
30 goS 4000:for i=820 to 1023:read a:if
a>255 tH 1000
31 pO i,a:nE:stop:rem "Machine Language into
820+
997 pO 204,0:get g$:if g$=""tH 997
998 if pE(207) tH 998
999 pO 204,1:reT
1000 rem " Type into Memory:
1002 ? " Enter Strings: End=999":m=20000-1
1004 :input g$:if g$="999"tH m=m+1:pO
m,255:gO 5000
1006 k=len(g$):for i=1 to k:m=m+1:pO
m,asc(mid$(g$,i,1)):nE
1008 m=m+1:pO m,13:gO 1004
2000 rem" Print sorted list of strings
2002 :fO i=20000 to m-1:?cH(pE(i));
2004 if p tH pR7, cH(pE(i));
```

```
2006 nE:?:if p tH pR7
2008 end
4000 rem " Printer: it works for the one I
use
4010 ?" Printer on? ";:goS 997:? g$
4012 p=0:if g$<>"y" tH reT
4020 p=1:if p1 tH reT
4021 p1=1:clO 5:clO7:open 5,4,5:open 7,4,7
4030 for i=1 to 10
4032 pR5,cH(27)cH(108)cH(5)
4034 pR5,cH(27)cH(81)cH(77):nE
4039 ?"  Set Interface to Junior":goS
997:reT
5000 :rem " Sort items
5010 sys 820
5012 if pE(781)=1 tH ?" End of String Sort-
ing":gO 2000
5013 if pE(781)=2 tH ?" y=255; check
data":stop
5019 ?"END 5019":stop
5020 dA 169,32, 133,251, 169,78, 133,252
5021 dA 160,0, 177,251, 201,255, 208,3,
162,1, 96
5022 dA 160,255, 200, 192,255, 208,3, 162,2,
96, 177,251, 201,13, 208,242
5023 dA 200, 24, 152, 101,251, 133,253,
169,0, 101,252, 133,254:rem into 253/4
5024 dA 160,0, 177,251, 209,253, 176,11
5026 dA 165,253, 133,251, 165,254, 133,252,
76,60,3 :rem next pair
5030 dA 208,3, 76,184,3
5032 dA 160,255, 200, 177,251, 153,0,159,
201,13, 208,246:rem parking
5034 dA 160,255, 200, 177,253, 145,251,
201,13, 208,247
5036 dA 160,255, 200, 177,251, 201,13,
208,249
5038 dA 200, 24, 152, 101,251, 133,253,
169,0, 101,252, 133,254
5040 dA 160,255, 200, 185,0,159, 145,253,
201,13, 208,246:rem retrieving
5042 dA 76,52,3
5050 dA 200, 177,251, 201,13, 208,3,
76,108,3:rem if pairs are equal
5052 dA 177,253, 201,13, 208,3, 76,108,3
5054 dA 76,102,3:rem compare again
5999 dA 999
```

Line 5020: 20000 is put into 251/2.

Line 5021: .A (780) is loaded indirectly from 251/2. Is it 255, the end search indicator? If yes, then the signpost (.X 781) is set to 1 and the program returns to BASIC line 5012.

Line 5022: .Y (782) is incremented to zero from 255, and checked for 255. In case .Y is 255 then something is wrong, the signpost .X is set to 2, the program returns to BASIC lines 5012 and 5013. If .Y is not 255, then .A is loaded indirectly from 251/2 again, with .Y in-

cremented, and again until the end of string indicator, 13, is found. Then .Y contains the number of characters in that string.

Line 5023: .Y+1 is added to the contents of 251/2, initially 20000, to get the starting byte of the next string. That number is put into 253/4.

Line 5024: The first characters of these two strings are compared; they may need no change (876), or need reversing (892), or are equal (952). If equal, then the two 2nd (or, if again equal, even more) characters of both strings will be compared.

Line 5026: 253/4 is put into 251/252, and the next pair will be compared.

Line 5030: If reversal is needed, then to 5032; if characters are equal, then to 5050.

Line 5032: The 1st string is copied into 40704+.Y for later use.

Lines 5034 and 5036: The 2nd string is made the 1st string, overwriting the bytes that contained the 1st string.

Line 5038: The starting byte is located, where to put the string that is to be retrieved from parking in 40704+.Y.

Lines 5040 and 5042: The reversal is completed.

Lines 5050 and 5052: The next characters of both strings are compared.

```
820:lda#32 822:sta251 824:lda#78 826:sta252
828:ldy#0 830:lda(251),y 832:cmp#255 834:bne
to 839 836:ldx#1 838 rts

839:ldy#255 841:iny 842:cpy#255 844:bne to
849 846:ldx#2 848:rts

849:lda(251),y 851:cmp#13 853:bne to 841
855:iny 856:clc 857:tya 858:adc251 860
sta253 862:lda#0 864:adc252 866:sta254
868:ldy#0 870:lda(251),y 872:cmp(253),y
874:bcs to 887 876:lda253 878:sta251
880:lda254 882:sta252 884:jmp to 828 887:bne
to 892 889:jmp to 952 892:ldy#255 894:iny
895:lda(251),y 897sta40704,y 900:cmp#13
902:bne to 894 904:ldy#255 906:iny
907:lda(253),y 909:sta(251),y 911:cmp#13
913:bne to 906 915:ldy#255 917:iny
918:lda(251),y 920:cmp#13 922:bne to 917
924:iny 925:clc 926:tya 927:adc251
929:sta253 931:lda#0 933:adc252 935:sta254
937:ldy#255 939:iny 940:lda40704,y
943:sta(253),y 945:cmp#13 947:bne to 939
949:jmp to 820 952:iny 953:lda(251),y
955:cmp#13 957:bne to 962 959:jmp to 876
962:lda(253),y 964:cmp#13 966:bne to 971
968:jmp to 876 971:jmp to 870
```

Joe Quittner, TPUG, May 15, 2008

Toronto Mini Maker Faire®

I recently had the pleasure of taking part in the Toronto Maker Faire (*http://makerfairetoronto.ca/*) on the weekend of May 6, 7, 8 2011 right here in Toronto. It was held at the Evergreen Brickworks, a fascinating reclaimed industrial space at the DVP near Pottery Road.

Maker Faires are part of a renewed movement in Do-It-Yourself, hands-on craftsmanship. They attract artisans, tinkerers, hackers, costumers, geeks, gamers, and anyone who has a strong interest in any kind of creative outlet. The emphasis is not so much on the end product, but the creation - the tools, techniques, technology, materials, and so on. Although the end results are, of course, often offered for sale.

There a growing number of Maker "spaces" devoted to these activities -Site 3 (*www.site3.ca*) and KWARTZLAB (*www.kwartzlab.ca*) are two local examples. But the Faires serve to bring these groups, individual makers, and the general public together.

Technically, I was there to demonstrate my new **Arduino** and accelerometer-based wireless MIDI (Musical Instrument Digital Interface) controller, as demonstrated at a recent TPUG meeting.

Arduino is a fully open-source hardware and software combination with an emphasis on low-cost, easy to use controllers. Commodore hobbyists would feel right at home using the Arduino - it has a modest amount of memory (39K for the Uno model), is very easy to program, and easy to connect to digital and analog inputs and outputs.

As an aside though, one of the organizers asked if I could also demonstrate my modified Commodore 64C, as he also began his programming career on a Commodore. I readily agreed, and also brought along several of the new peripherals that the worldwide Commodore hobby community has created, using many of the same tools and techniques that the Makers use.

IDE64 - Designed by Tomas Pribyl, Josef Soucek, and Kajtar Zsolt in Poland and Czech Republic
Xpander3 - Designed by Jim Brain in the Unites States
ZoomFloppy - Designed by Nate Lawson and Jim Brain in the Unites States
RR-Net - Designed by Jens Schönfeld in Germany
Dualsid - Designed by Tomi Malinen in Finland
Joystick Port Switcher - Designed by Vimfuego in Australia

I had my 64C set up all weekend and...sure enough, it eclipsed my music project! Makers and the public alike walked by in droves, then did a double-take when they saw the 64 set up. Dozens of people stopped to talk at length about their Commodore experiences. Some had even been TPUG members back in the 1980s and were thrilled to hear TPUG was still going strong. Most were stunned to hear about the variety of new hobby-designed Commodore peripherals out there. I handed out several TPUG membership flyers as well.

The 64 was particularly popular with kids. They were drawn into the classic games - Paradroid, Fort Apocalypse, Ghostbusters, and Bruce Lee were particular favorites. Some stayed for several hours! My son was with me and made several friends, they had a great time playing two-player games together.

There were literally hundreds of makers showing their projects off at the Faire, I couldn't possibly describe them all or show all the pictures (check the website though!). Some of the highlights:

-A video game chair that moves and rotates in time to the action on screen
(*http://www.dyadgame.com/TheMachine/*)
-A tele-operated robot that could pick up nearly anything
(*http://web.goodrobot.com/blog/*)
-A laser cutter
-A guitar that used ultrasonic sensors instead of strings
-A teletype from the 1950s connected to Twitter
(trumping my 64C as the oldest piece of technology there!)

It was an exhausting weekend, but also invigorating to spend so much time talking about my hobbies (music, art, and Commodore) with so many people.   And there will be many more Maker Faires - the next one in Canada is in Vancouver on June 25th
(*http://vancouver.makerfaire.ca/*), and I'm looking forward to next year's Toronto event!

Leif Bloomquist

Build Your Own Book Scanner The Cheap and Easy Way

Report by Ian Colquhoun

Back at the World of Commodore show in December of 2010 I demonstrated a method of building a book scanning device using bits and pieces you might already have at home, other garbage, and one or two things you actually might have to buy. The idea is to get as good a result as possible without a tremendous outlay of cash or time. Extensive information on this subject can be found on the web at *http://diybookscanner.org/* and I do recommend you peruse that website before getting underway with your own project.

Things you'll need:

- a digital camera of decent quality

- a bright light

- a tripod

- a cardboard box (with the biggest side being just a touch larger than the biggest book you expect to be scanning)

- a piece of clear glass or plastic to hold your pages flat

- a knife

- Tape

- a table of some sort

Software required:

- Scan Tailor - *http://scantailor.sourceforge.net/* works in Windows or Linux

- PERL - *http://www.perl.org/* will almost certainly be installed already in Linux for Windows try - http://strawberryperl.com/

- PERL script to sort your raw photos into a naming structure usable by Scan Tailor *http://is.gd/iaNc2*

Let's Build:

1. Take your knife and cut the cardboard box diagonally in half lengthwise. Be careful because the parts of the box will likely fall apart during this process. Liberal amounts of tape will be required. What you are looking for is to triangular type pieces that can be put together to form a cradle to hold your open book.

2. Tape your cradle firmly to your table so that it won't move out of place.

3. Place your book open in the cradle and place your piece of glass over the first page to be photographed to hold it flat. You will photograph all the right hand side pages in your book first (including the cover if you want that), then flip the book over and photograph the left side pages (including the rear cover).

4. Place your light so that it floods the page to be photographed with as even and as bright a light as possible. The trick is to arrange the lighting so that when you look down on the page under the glass from square on top that there is no glare or reflection from the light. It's possible but can be frustrating, don't give up!

5. Attach your camera to your tripod and set it up so that the face of the lens is exactly parallel to the face of the glass. You must also try to frame the picture so that the book page is taking up most of the frame. ie. you don't want too much cardboard box in your frame. If you don't get things quite square you will end up with weird skewing in your text that software processing will not be able to sufficiently correct.

6. Tape down your tripod once you are happy with the positioning so it won't move during the photographing.

Now you're ready to take pictures! Turn the flash on your camera off - your light should be plenty of light for the photos. Now, start with the front cover under your piece of glass, snap a picture (it helps if your camera has a focus lock - mine doesn't and it slows things down considerably), lift your glass, turn the page, put the glass back down, snap another picture. Lather, rinse repeat. When you get to the end of the book, turn the book upside down and start photographing with the inside front cover and working once again towards the back of the book - these will all be upside down, but don't worry, Scan Tailor will fix them.

Copy your complete set of photos of your book to a directory/folder on your PC. Copy the fix_order.pl script into this directory. There should be nothing else in there except this script and the photos of your book. If you're in Windows, double click on the fix_order.pl script, or in Linux open up a shell and run the script in that directory. This will rename everything so that Scan Tailor doesn't get confused.

Once the rename is complete you're ready to open up Scan Tailor and start processing. Scan Tailor is quite a robust processor for book scans and is way beyond the scope of this article. To get started I would suggest visiting the Scan Tailor website mentioned above, click the Documentation link, and choose the Video Tutorial link. It provides an excellent guide to getting started and will allow you to get good results right away.

So get out there, dig out your rare Commodore manuals, books, magazines, etc. and get scanning! Don't forget to share!

See *http://pipesup.ca/2010/12/diy-bookscanner-from-world-of-commodore/* for the PowerPoint presentation from WoC 2010.

Ian Colquhoun

And now you shall know the truth....

A revelation with an Incredibly Big Message (IBM)

Well, you might have thought that you knew how the Internet started, but here's the TRUE story.

In ancient Israel, it came to pass that a trader by the name of Abraham Com did take unto himself a young wife by the name of Dot.

And Dot Com was a comely woman, broad of shoulder and long of leg. Indeed, she was often called Amazon Dot Com. And she said unto Abraham, her husband: "Why dost thou travel so far from town to town with thy goods when thou canst trade without ever leaving thy tent?"

And Abraham did look at her as though she were several saddle bags short of a camel load but simply said: "How, dear?"

And Dot replied: "I will place drums in all the towns and drums in between to send messages saying what you have for sale, and they will reply telling you who hath the best price. And the sale can be made on the drums and delivery made by Uriah's Pony Stable (**UPS**)." Abraham thought long and decided he would let Dot have her way with the drums. And the drums rang out and were an immediate success. Abraham sold all the goods he had at the top price, without ever having to move from his tent. To prevent neighbouring countries from overhearing what the drums were saying, Dot devised a system that only she and the drummers knew. It was called Must Send Drum Over Sound (**MSDOS**), and she also developed a language to transmit ideas and pictures - Hebrew To The People (**HTTP**). But this success did arouse envy. A man named Maccabia did secrete himself inside Abraham's drum and began to siphon off some of

Abraham's business. But he was soon discovered, arrested and prosecuted - for insider trading.

And the young men did take to Dot Com's trading as doth the greedy horsefly take to camel dung.

They were called Nomadic Ecclesiastical Rich Dominican Sybarites, or **NERDS**. And lo, the land was so feverish with joy at the new riches and the deafening sound of drums that no one noticed that the real riches were going to that enterprising drum dealer, Brother William of Gates, who bought off every drum maker in the land.. And indeed did insist on drums to be made that would work only with Brother Gates' drumheads and drumsticks.

And Dot did say: "Oh, Abraham, what we have started is being taken over by others."

And Abraham looked out over the Bay of Ezekiel, or **eBay** as it came to be known. He said: "We need a name that reflects what we are."

And Dot replied: "Young Ambitious Hebrew Owner Operators." "**YAHOO**", said Abraham. And because it was Dot's idea, they named it YAHOO Dot Com. Abraham's cousin, Joshua, being the young Gregarious Energetic Educated Kid (**GEEK**) that he was, soon started using Dot's drums to locate things around the countryside. It soon became known as God's Own Official Guide to Locating Everything (**GOOGLE**).And that is how it all began.

So now you know.

Mind in a Box Releases Commodore 64 Tribute Album

Mind.In.A.Box's past three albums have garnered them a lot of respect with fans and critics alike with their hard to categorize style of electronic music. Their newest album, R.E.T.R.O., takes a slightly different approach to their songwriting. **An homage to the era of the Commodore 64** and 1980's video games, the album will even include a few cover versions of cult classics – all done in MIAB's signature style.

However, do not take these adaptations lightly. This is not just another cut and paste cover album. This is Mind.In.A.Box re-inventing the past for an advanced future. Released on March 9th, 2010.

01. The Last Ninja III
02. Lightforce
03. The Last V8
04. Supremacy
05. Shades
06. 8 Bits
07. Mindkiller
08. The Last Ninja
09. I Love 64
10. We Cannot Go Back To The Past
11. Whatever Mattered

http://www.mindinabox.com/

Review from *"Storming the Base"*

From the President ... *from page 1*

Well, whether you fancy several rounds of Voodoo Castle on the VIC-20 or Zork on the Commodore 64, the summer months will be filled with plenty of vintage 8-bit action. The Commodore show season is about to begin, starting off with the Cincinnati Commodore Computer Convention (CCCC) in Cincinnati (June 4), Commodore Vegas Expo (Commvex) in Las Vegas (July 23-34), and looking into the fall season, the Emergency Commodore Computer Convention (ECCC) in Chicago (Sept. 24-25) and of course World of Commodore.

TPUG is always on the lookout for new venues to attend and showcase our vintage hardware. So this summer, at the time of this writing, we are making plans to attend the FanExpo show, held in Toronto Ontario from Aug 25 to Aug 28. FanExpo is a very large show that bills itself as the largest Pop Culture event in Canada. Alongside the comic collectors, movie buffs, television aficionados, and autograph seekers we will be setting up a booth to show-case vintage Commodore hardware and software to fans of 8-bit computing. Last year's event was attended by 64,000 (!) people, a perfect venue for showcasing the Commodore 64, VIC-20, Amiga and PET computers. To be honest, I miss the Computerfest shows where TPUG was exposed to a larger audience outside of a World of Commodore show. It will be fun again to see people stroll by our booth, stop, point, and exclaim "Wow, a Commodore 64!"

There are no chapter meetings in the months of July and August. We will start up again in September. The last chapter meeting before the summer break will be in June. This is typically our repair night where you can bring your broken 64's, drives, etc and we'll make the attempt to diagnose and fix the problem. It's my 'look and learn' night!

Whatever your summer activities are, hope you have a safe one. And don't leave your Commodore lonely!

Happy Computing!
Greg Van Laere

Computer Hard and Software

From: "Joseph Palumbo"
I liked it...Lucky for me my 7.0 started test runs of 1.0 and has now been removed before the upgrade could happen...I know some of you weren't so lucky.

Dear Tech Support:

Last year I upgraded from Girlfriend 7.0 to Wife 1.0. I soon noticed that the new program began unexpected child processing that took up a lot of space and valuable resources. In addition, Wife 1.0 installed itself into all other programs and now monitors all other system>activity. Applications such as Poker Night 10.3, Football 5.0, Hunting and Fishing 7.5, and Racing 3.6 no longer run, crashing the system whenever selected.

I can't seem to keep Wife 1.0 in the background while attempting to run my favorite applications. I'm thinking about going back to Girlfriend 7.0, but the uninstall doesn't work on Wife 1.0. Please help!

Thanks,
A Troubled User

Reply:

Dear Troubled User:

This is a very common problem that men complain about. Many people upgrade from Girlfriend 7.0 to Wife 1.0, thinking that it is just a Utilities and Entertainment program. Wife 1.0 is an OPERATING SYSTEM and is designed by its Creator to run EVERYTHING!!! It is also impossible to delete Wife 1.0 and to return to Girlfriend 7.0. It is impossible to uninstall, or purge the program files from the system once installed.

You cannot go back to Girlfriend 7.0 because Wife 1.0 is designed to not allow this. Look in your Wife 1.0 manual under Warnings,-Alimony,-Child Support. I recommend that you keep Wife1.0 and work on improving the situation. I suggest installing the background application "Yes Dear" to alleviate software augmentation. The best course of action is to enter the command C:\APOLOGIZE because ultimately you will have to give the APOLOGIZE command before the system will return to normal anyway.

Wife 1.0 is a great program, but it tends to be very high maintenance. Wife 1.0 comes with several support programs, such as Clean and Sweep 3.0, Cook It 1.5 and Do Bills 4.2. However, be very careful how you use these programs. Improper use will cause the system to launch the program Nag, Nag 9.5. Once this happens, the only way to improve the performance of Wife 1.0 is to purchase additional software. I recommend Flowers 2.1 and Diamonds 5.0! WARNING!!! DO NOT, under any circumstances, install Secretary With Short Skirt 3.3. This application is not supported by Wife 1.0 and will cause irreversible damage to the operating system.

Best of luck,
Tech Support

TINA'S GROOVE — BY RINA PICCOLO

TORONTO PET USERS GROUP
C/O JOHN EASTON
258 LAKE PROMENADE
ETOBICOKE, ONTARIO M8W 1B3

— SPRING 2011 —

We're on the web

www.tpug.ca