April 1995

Commodore Users Group of Saskatchewan

3617 29th Avenue ● Regina, SK ● S4S 2P8 ● Tel: (306)584-1736 ● BBS: (306)565-8562

the MONITOR

The Monitor is published monthly by the **Commodore Users' Group of Saskatchewan** (CUGS). Meetings are held on the first Wednesday of every month in **Room 173** of Miller High School unless otherwise noted. The next meeting will be held on **May 3**, 1995, from **7:30 to 9:30 P.M.**

CUGS is a nonprofit organization comprised of 64 and 128 users interested in sharing ideas, programs, knowledge, problems, and solutions with each other. Membership dues ($15) are pro-rated, based on a January to December year.

Anyone interested in computing is welcome to attend any meeting. Members are encouraged to submit **public domain** and **shareware** software for inclusion in the **CUGS Disk Library**. These programs are made available to members at $3.00 each (discounted prices when buying bulk). Since some programs on the disks are from magazines, individual members are responsible for deleting any program that they are not entitled to by law (you must be the owner of the magazine in which the original program was printed). To the best of our knowledge, all such programs are identified in their listings.

Other benefits of club membership include access to our disk copying service to make backups of copy-protected software, and any members who own a modem and wish to call our BBS will receive increased access. The board operates 300-2400 baud, 24 hours a day. The number is **565-8562**.

Editorial

We open this month's newsletter on a depressing note -- according to our club secretary/treasurer, Lyndon Soerensen, we will have run out of funds to finance our bulletin board system within a month or two from now. As Lyndon has pointed out on a message he recently left on the BBS, the club has kept its membership fee pretty much constant since its inauguration over ten years ago, while membership has somewhat dwindled. This, combined with the increasing amount of money needed to maintain a bulletin board and the frivolous spending of club funds on various BBS software and phone services may spell the end of the CUGS BBS. This does not have be so, however. With the current state of the club, it will be up to the entire membership, not just the oligarchical executive, to decide what, if anything, is to be done to preserve the board, and to decide exactly in what fashion the BBS will continue after May, if at all. The board has served the membership well in the past. Now we will have to decide if the board is still being used to its maximum potential, or if support for it has subsided to such an extent that it is no longer an essential facet of the club.

Certainly, the future of Commodore bulletin boards in the city does not look bleak. We have always maintained a healthy population of at least two or three systems in the past few years -- presently, the Commodore BBSes in the city include CUGS, Fortress, Digital Underground, and Arboria. Losing CUGS BBS will not mean the end of Commodore BBSing in Regina, then. With three other systems to serve the 64 and 128 users in the city, plus our club itself, we can rest assured that there is no immediate threat of a lack of 64/128 files or information in our city. On that note, I bid you enjoy this month's sizeable newsletter.

3617 29th Avenue ● Regina, SK ● S4S 2P8 ● Tel: (306)584-1736 ● BBS: (306)565-8562

Gems at the Library

Stan Mustatia

This month I would like to briefly introduce you to a wonderful game from the CUGS library. Mah Jongg is an ancient Chinese game introduced to the Commodore by way of RUN Magazine in their January/February 1992 issue. The game can be found in the club library on Graphic Games 01,02.

The word to remember with Mah Jongg is *strategy*. If you use a little, skill is sure to follow. This can be a very addictive game. I know, because I've spent many a late night playing this game instead of doing reports for work, for making up disks for the library. As a matter of fact, I decided to refresh my memory on the game before writing this article. That was an hour ago.

Mah Jongg is not a hard game to play, but it can be very hard to win. When you load the game, you are presented with an opening screen of 132 tiles, stacked in four layers like a pyramid. The object of the game is to match any two tiles from any level. A joystick is used to move around the playing surface and select the individual tiles, which are highlighted as they are selected with a click of the fire button. To select a tile, it must first have a free edge on either the right or the left. Each level of the pyramid is coloured differently, making it easier to tell where the free edges are on each level. If you select a tile that is incompatible, nothing will happen and the clock will continue to run. Oh yeah! Did I forget to tell you? There are only 60 seconds in which to make a pair of selections. Early on in the game, this is no big deal, but you may have to scramble late in the game. Don't worry, though -- if you get stuck and time is running out, just press *H* for *HELP*. That will cost you 255 points and might not show the best selection to make, anyway. It could at the very least keep the game going a while longer, or inform you that there are no moves left to make. At this point, you can press *Q* for *QUIT*. You will be asked if you want to play again, which, in most cases, will be *YES*, I can assure you.

Play Mah Jongg in your spare time, but be sure that you have plenty. One game only takes a few minutes to play, but as the potato chip commercial says, "I bet you you can't just have one."

GEOS Gems

GEOS is a program that, when understood, can make your Commodore 64 or 128 really sing. Many people hat GEOS and don't use it at all. Some people use it only a little and plug through it grudgingly until the finally get the results they wanted with much trial and error, or they simply give up altogether and use a completely different program to get the job done. There are others who use GEOS whenever they can and actually love the GEOS graphic environment. I am one of the latter. I was also both of the former two for quite some time, I might add. I inherited GEOS v1.2 when I bought my C64 second hand eleven years ago. For the most part, I didn't understand it at all. I found the manual incomplete and hard to understand. Besides, I just wanted a game machine. That was until one day my nephew from Calgary opened my eyes for me.

I had one drive, keyboard, two old Atari joysticks, a TV, and no printer. Pretty basic stuff, right? Well, he had a monitor, mouse, two 1541 drives and a nine pin printer. He also had Geos v1.5. When he started to show me what he could do with what he had, I was floored. Maybe this wasn't just a game machine after all. Alas, he was one of those people in the middle who used GEOS sometimes but didn't understand it much, or didn't have many other programs compatible with GEOS. I was determined to learn all I could about this wonderful program. This is where CUGS comes into the picture.

Commodore Users Group of Saskatchewan

3617 29th Avenue ● Regina, SK ● S4S 2P8 ● Tel: (306)584-1736 ● BBS: (306)565-8562

It was actually at least two years later that I noticed an ad in the Leader Post about a club for Commodore users that met once a month down at the Northwest Leisure Center every month on the first Wednesday. If anyone could be able to tell me about GEOS, this bunch would. I wasn't disappointed. First, I proceeded to buy all the GEOS disks in the library. The club had regular sessions on the GEOS environment. I even went out and bought version 2.0.

Now I have a mouse, a 128, RAM expansion, two 1571 drives, a printer, a monitor, both the C64 and C128 versions of GEOS, and hoards of programs gleaned from magazines, flea markets, BBSes, mail order -- everywhere. Our club library has got some of the best there is to offer. Many of them are my favourites. Now, you don't need any of them to travel in the GEOS world, but they sure make the trip a whole lot easier. Here are some of my all time favs and a little of what they can do:

Convert: Converts file format for uploading to a BBS and back again.

WrongIsWrite: This is another conversion program. It converts different text files into Geo format.

Icon Edit: Create your icons to personalize your files.

Scrapeek: Look at photo albums created with Photo Manager.

Solo Poker: Something to relax with when I have been pounding away at the keys for a long time.

Combiner: Combines multiple GeoWrite docs into one.

Bigclipper: For cutting those huge photos out for use in GeoPublish.

Getgraphic: Converts PrintShop graphics to Geo format. (the library has thousands of PrintShop graphics)

GeoAlbum: Saves the screen currently displayed into a photo album.

Quickview: Read a text file without entering GeoWrite.

Scrapit!: Convert other graphic formats, such as Doodle or Blazing Paddles, to GEOS.

Multilabel: Create labels with graphics included. (I make all the new club labels with this program)

Iconget: I use this program to import icon images as photoscraps onto my disk labels.

Dualtop: Look at and use files from two different drives at one time and use three drives at one time without having to switch. (a more detailed description is in an earlier *Monitor*)

Paintview: Look at GeoPaint files quickly and easily.

There are scores of font files and printer drivers, as well as many, many more accessory programs just sitting in the club library just waiting for you to come along and make your 8-bit Commodore sing. Try a few, and don't give up on GEOS. There is always someone out there who has the answer to your problems. There was for me and now I'm a convert.

A Look Into the C128 Video Display Controller — Part Two

E. Carl Reilly

The BASIC Connection
==========

While BASIC is not as functional as Assembler, we can still access the VDC with it. We must use the SYS and RREG command to do it. With BASIC 7.0, arguments can follow a SYS command that let us access the Accumulator and X & Y- Pointers, respectively. The RREG command will let us read a value from a particular address in memory.

Even with BASIC 7.0 commands, we are still going to have to use a little bit of Machine Language to accomplish our task. Fortunately, there are two subroutines in ROM that we can SYS to. Address $CDCC will let us write to a register (52684) and address $CDD8 will let us read from a register (52696). We are going to use these addresses and the BASIC 7.0 XOR command to make the 80 column display toggle off and on.

We will accomplish this toggle in the same way we did while in Assembler. First, we are going to declare our equates like so:

```
10 COLUMNS = 80        :REM The total number of columns
20 ROWS = 25           :REM The total number of rows
30 VR = DEC("CDD8") :REM VDC Read subroutine
40 VW = DEC("CDCC"):REM VDC Write subroutine
50 DIM RE, A, RS       :REM Dimension variables for register number, data, toggle value
                          respectively
```

With our equates declared, we can now tell the VDC what to do. Like the Assembler version of our program, we'll start with reading register 1 and toggle it.

```
55 GETKEY A$ :REM Wait for keystroke from User
60 RE = 1              :REM Declare register 1
70 SYS VR,,RE  :REM Access register 1 to read
80 RREG A              :REM Read current register
```

Line 70 has two commas in the SYS function. Be sure to include those two commas! BASIC 7.0 was nice enough to include an extension to the regular SYS command of BASIC 2.0. This extension lets us access the Accumulator, X pointer, and Y pointer, respectively.

Now we can XOR the value in "A" with that in our equate "COLUMNS" and write it back to the VDC.

```
90 SYS VW,XOR(A,COLUMNS),RE
```

A lot is happenning at line 90. During the BASIC SYS function, we are performing an XOR function with our equate COLUMNS and our variable A. After BASIC has determined the result of the XOR,

Commodore Users Group of Saskatchewan

3617 29th Avenue ● Regina, SK ● S4S 2P8 ● Tel: (306)584-1736 ● BBS: (306)565-8562

it copies the result into Accumulator A and copies variable RE into the X pointer. Finally, BASIC let's the ML routine at the address pointed to by our equate VW take over.

Finally, we can perform the same code for register 6.

```
100 RE = 6
110 SYS VR,,RE
120 RREG A
130 SYS VW,XOR(A,ROWS),RE
```

The last line of our BASIC program will go back up to line 55 and wait for another keystroke from the User.

```
140 GOTO 55
```

Putting all this code together it should look like the following:

```
10 COLUMNS = 80                 :REM Number of columns
20 ROWS = 25                    :REM Number of Rows
30 VR = DEC("CDDA")             :REM Address for the VDC Read
                                     function
40 VW = DEC("CDCC")             :REM Address for the VDC Write
                                     function
50 DIM RE, A, RS                :REM Dimension variables
55 GETKEY A$                    :REM Wait for keystroke
60 RE = 1                       :REM Set variable to point to
                                     VDC register 1
70 SYS VR,,RE                   :REM Turn control over to ML
                                     routine to read VDC
                                     register 1
80 RREG A                       :REM Copy value of VDC
                                     register 1 into variable
                                    A
90 SYS VW,XOR(A,COLUMNS),RE     :REM Turn control over to ML
                                     routine to write the
                                     result of the XOR
                                    function
                                     back to VDC register 1
100 RE = 6                      :REM Set variable to point to
                                     VDC register 6
110 SYS VR,,RE                  :REM Turn control over to ML
                                     routine to read VDC
                                     register 6
120 RREG A                      :REM Copy current value of VDC
                                     register 6 into variable
                                    A
```

```
130 SYS VW,XOR(A,ROWS),RE       :REM Turn control over to ML
                                     routine to write the
                                     result of the XOR
                                     function back to VDC
                                     register 6
140 GOTO 55                     :REM Go back to line 55 and
                                     wait for another
                                     keystroke from User
```

If we RUN this program it will wait for us to press any key (except RUN/STOP and/or RESTORE) before it will toggle the display off and on. It may seem that the BASIC program was easier for us to toggle the screen with BASIC 7.0 than the Assembler version. So, let's fix that! With Assembler, we can access the C128's main code to produce some pretty nice effects. We are going to have to decide whether to have any key access the toggle subroutine, a combination, or just one. To keep it simple, we are going to let one key on the keyboard toggle. We'll designate that key to be the RESTORE key. After all, how many times have you actually used the RESTORE key by itself?

To access the RESTORE key, we would enter the monitor by typing:

MONITOR <RETURN>
>318 00 13 <RETURN>

What we just did was tell the C128's Operating System to go to our Assembler program at address $1300 everytime the RESTORE key is pressed.

There you go. You now have a screen saver for your C128. Add to some of your programs for a little more User Friendly software. You might want to add a few lines to the Assembler code and have a timer count down when a key is not being pressed to have an automated 80 column screen saver.

- E. Carl Reilly
#5 - 3845 Robinson Street
Regina, SK
S4S 3C5