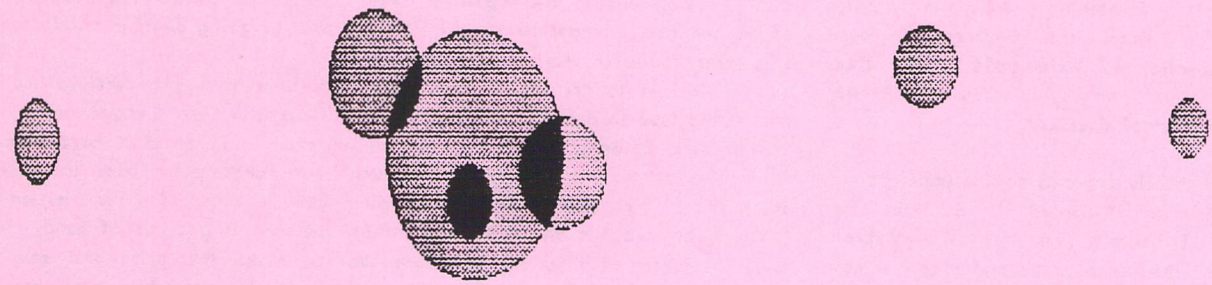
February 1993
Vol. 2 No. 2 Issue 7

$1.75 (U.S.A.)

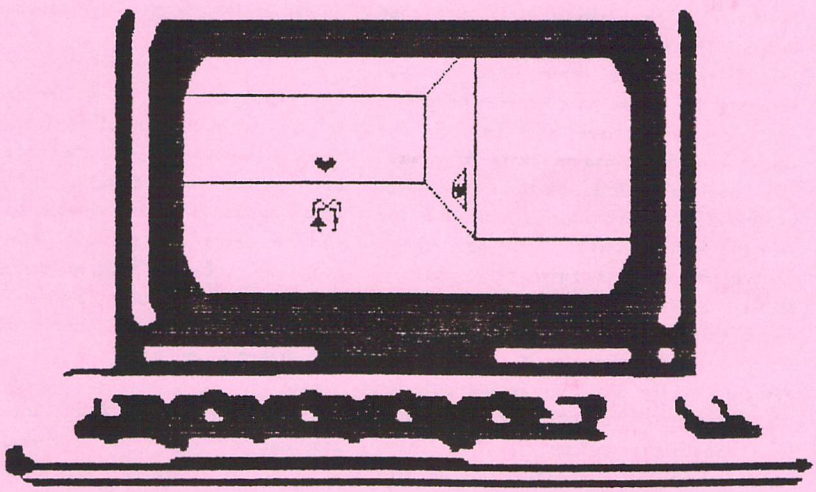
dieHard

the Flyer for commodore 8bitters

COMPUTER CIPHER



Tips!
Reviews!
Rarities!
PRG!
And much, much more...



View From The Underground
by
Brian L Crosthwaite

What is the Underground? The Underground is where I heard of Ronald Snyder when I had gotten some disks from **8 bit**. He has organized thousands of **Print Shop** art clips and converted them into several sets. Three block, two block, **GEOS**, etc. He also specializes in equipment, VIC, PET, CBM, etc. The Underground is also where **Twincities** exists. Something I have yet to check out. From the Underground come companies like **Silvasoft**, **Busy Bee**, and our own LynnCarthy Industries, publisher of **dieHard**.

I am barely exposed to the underground. I have only glanced the surface, after all, I haven't even seen **Twincities**. But I have seen some and what I've seen is impressive. A new publication has arrived called **geoVISION International**. Names like Walsh, Butterfield, and Collette, to name a few, are in both the spotlight and the Underground alike.

Some of you are wondering about the **Spinner**. The **Spinner** is a disk that has all the **PRG** programs on it. This is offered as a convenience to our readers who either hate type-in programs or just don't have the time. The **Spinner** also contains extra programs. The goal of **PRG** is to demonstrate programming techniques, using the list-and-learn concept. And as tedious as type-ins can sometimes be to enter, I learned a lot about programming as I typed things in. But that may not be a learning technique that works for everyone so the **Spinner** is available for $5 each or $45 for one year, 10 disk or 10 cassette subscription. (Idaho residents add 5% sales tax). P. O. Box 392, Boise, Idaho, 83701.

Last month's **Spinner** introduced a small, not so flashy, program that could run on either the 4, 16, 20, 64, OR 128. Most of the program consisted of character strings and printing them out, but a look into certain locations revealed what computer the program was in before it used POKEs to set the color. Although, this may not be practical for all programs or even most programs, we plan to use this technique to save time for our programmers -- namely me.

We now have a PET, actually a CBM 2001 Series! We will soon be able to offer more support for this machine. We have a few bugs to iron out in the **Spinner** realm. If anyone has information on the PET or any other computer, feel free to drop us a line. We love to see who is doing what, and would love to pass the information on. If you send us an article on disk, be sure to tell us what word processor you saved it with. We also accept hardcopies (material printed out on paper), but things go a lot faster when we can just download a file from disk. If you want the disk returned be sure to enclose $2 for postage. Speaking of downloading, we don't have a BBS, however arrangements can be made to send us files via modem in the not so distant future.

We also now have a plus/4 computer, so we can easily use the files saved off of its word processor.

I've received a lot of letters from people telling me about their systems. I love hearing about everyone's systems, it betters our understanding of where to offer support. There are a lot of multi machine users out there -- send us some programs. Every program published will receive a free issue. We can add it to your subscription. You don't have to be a subscriber to submit.

The types of things I like are music and graphics of any type. (Music must be original or classical in origin, unless written permission is given from the copyright owner.) I'd like to see an all DEMO issue of **PRG**.

Getting back to letter's. Don't feel like you've been unread, I read all the mail that comes in. Hopefully I will have time to print some of them next month. Our regular features include four major articles; **PAPSAV**, **Rarities**, **Archaic Computer** and **PRG**. Among these we would like to place reader mail.

There have been mix ups on our pricing, and yes we have downsized. First, we publish 10 times a year and subscriptions go for $15 -- 10 issues. Second we are forced down to 15 pages, this is because of print costs. The good news is this will be going down in the feature.

I have not yet received my February **Compute**, so I have yet to read the review. It couldn't have been too bad, our subscription base is growing. The issue reviewed was titled "Oops," because it got out of hand. We decided to raise the price and stay with the larger format, but that proved to be impractical. There were a lot of errors that got by because of the lack of time, many of the smaller articles missed spell checking, after all they were small enough, the proofreader would catch anything -- they went right passed proofreading, after all they were small enough to get proofread while they were being spell checked...

Any MIDIots out there? I am interested in playing my SID chip rather than programming my keyboard. What's happening with Euphony? Any SID programmers out there with programs you'd like to share?

As I ramble on, I'd like to mention our Special issue of the **Spinner**, coming in the late summer or early fall. It's going to be PD DEMOs. I have most for the 64 all ready chosen, what I need is help compiling a list for the PET, 16+4, and VIC. If anyone knows of and PD DEMOs that are out of this world, send them in! Now some of you are going PD WHATs?!? PD is public domain and demos are demonstration programs. In this case we'll define them as free programs that are loaded with color and sound. Stay tuned!

READY.

■

Oops!

We forgot the sequential file writer in the October **Archaic Computer's Q&A**. So here it is.

```
60000DOPEN#1,(N$),D0,U8,W:CMD1:LIST-59999
 :PRINT#1:DCLOSE#1:END
```

Tack this line onto the end of the 128 program you wish to copy into a sequential file and type: RUN60000. It will list all of the program except line 60000. This program should work on BASIC 4.0 machines as well.

READY.

Volume 2 - Number 2 - Issue 7

Inside

Tip of the Month

by

Brian L Crosthwaite

The disk catalogue. If you're like me, you have what seems like thousands of disks. Print out the directories of all your disks. Place them into a loose leaf binder in alphabetical order by disk name. If you have a data base program, you can make a data base that indexes the programs by name, telling what disk they are on, then place the index in the back of the notebook. It's also a good idea to have a copy of the directory stored with the disk. This will expidite locating files in the future.

READY.

Trader's Corner

Got something to trade? Need something? Try here, maybe one of our readers has just what you're looking for. Or perhaps they want what you have. Trader's Corner is free to anyone looking to trade or buy. (If you want to sell something you'll have to take out a classified, they are $5.)

Write to **dieHard**, P. O. Box 392, Boise, ID, 83701.

Wanted: commodore Plus4 joysticks. Any information appreciated. P.O. Box 392, Boise, ID, 83701.

Wanted: Books on any computer, old new, never released. P. O. Box 392, Boise, Idaho, 83701.

READY.

WP

WORD PROCESSING
by
Brian
L
Crosthwaite

Well, we are back again with some of those valuable tips on word processing. First let's take a look at backup disks.

Backups You should always make backups of everything you do -- every disk you have. In the event of disaster, you will still have the original. Making a BACKUP copy of a program is not illegal, as long as it is for your own personal use. Make a backup and use it. Stash the original away in a safe place, away from dust, magnetic fields, and moisture. When you place that backup into the drive and you accidentally format it, you'll be glad you have the original. Write protect not only the original, but the backup as well.

Drives Some drives are kind of tempermental. I made a backup copy of my WP and it resets the computer when I try to load it off of my old 1571 disk drive. The original loads ok, but something is amiss in true 1571 mode with my backup. It may be part of the copy protection scheme; 128's are often expected to have a 1571. The scheme might be set up to reset a bootleg copy so it will not work, but when the drive is in 1541 mode, it loads fine...

If you do need to put the drive in 1541 mode just enter in direct mode OPEN15,8,15,"U0>M0" with a disk in the drive to prevent that annoying flashing green light. I don't know why the copy won't load in 1571 mode, it's on a 1541 disk. Maybe it looks for something on the other side of the disk and if it's not there it resets. I may have put the spell checker on the other side instead of the thesaurus, or lost some code in the transfer that only a 1571 could read.

Macros Busy Bee Software's **The Write Stuff** is out of this world, especially in the macros arena. Now, I'm not big on word processors that want me to type in formatting codes. That's why I use **geoWrite** -- but I've still got to type in every single word, sentence and paragraph. I can use the text scrap feature to repeat words or sentences. I can also use a text album if I have lots of repetitive stuff. But the latter takes some serious time, especially if I have to search a large text album or several text albums. **T W Stuff** handles just about as many macros as you can cram onto the boot disk. It comes with approximately one hundred predefined (also redefinable) macros. The only limit to the number of macros is the amount of memory in the computer (don't forget you'll need some memory for your document).

Macros are really easy to do. For instance, B followed by a space could print my entire name and address centered. I could write several letters and place them all in one document and never make a flub-up on my name, providing I didn't mess up when I enter the macro.

To enter a macro, simply call up the directory and load the "bb macros" file. Just enter your macro on the screen as you would any other text. You can use the others as a general guide. B=Brian, for example. The macro can be up to 250 characters long. By pressing <CTRL><1><back arrow> you can add more than one line. Be sure to end with a <RETURN>. (Don't forget to leave the three reversed arrows at the bottom of the file intact.) Save your new file under the same name, but remember to remove the prefix "-", so the program knows what file to load. To use your new macros you will have to reboot **TW Stuff**.

You can have temporary macros as well. These are defined while you are working and are gone forever when you exit the program. Enter command mode and press <SHIFT><M> then enter your macro: B=Brian. They are limited in size to 16 characters in length.

<CTRL><M> turns on macro mode. One last point, don't define words like "a" and "I" as macros or you will have to type <SHIFT><SPACE> after every "a" that you want to read "a".

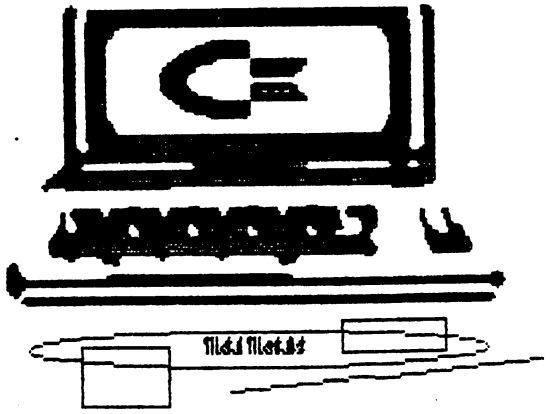
MS to CBM If you don't have a conversion program that can read MS DOS files into CBM DOS, you can still convert files back and forth between the two machines via a modem. You can connect them directly with a modem (apparently the IBM machine needs a null modem or a modem with a null mode -- for some extra-extemporaneous-mega-more-complex than necessary reason).

You can call up the other machine over the phone and upload the file to it, or you can use a BBS. To use a BBS call the BBS and upload the file then exit. Call back on the other machine and download the same file. You might want to ask a SysOp for instructions.

No Spaces Many word processors' dictionaries don't have the two letter state abbriviations in them. Simply make a document listing them all. Make sure they are correct. Spell check the document and add them if they are not recognized.

Got any great tips? An insight on word processing? Send HARD-COPY or DISK to **dieHard**, ATTN: WP, P.O. Box 992, Boise, Idaho, 83701.

READY.
■



geoTips

by

Brian L Crosthwaite

Half Line Spacing

Recently, I wrote a letter to someone. When I got the page all laid out there was just not enough room for it all to fit on one page. There were two lines on page two. To make it all fit, I highlighted the area behind the period on the bottom of the first paragraph all the way to the area just before the first word in the next paragraph. I then selected the same font, but half the original point size; the font I was using was 12 point, so I selected 6 point. That way the letters remained the same point size and the spaces between the paragraphs were half line spaced. I did all the paragraphs this way, so everything would look the same. It only gave me a little room each time, but after three or four of these half-spaced lines, I got the lines to fit.

Be sure that you start at the last character of the paragraph and end before the first character of the next paragraph, no characters should be highlighted. The font doesn't have to be the same, but the point size should be half (or a number evenly divisible into the original size).

AAAAE!!!

Hopefully you will never need these ***geotips****. Sometimes things go wrong, like a system lockup or a monitor that craps out. If this happens it is not necessarily the end of the world.*

If your system locks up, you can try to reset it with the reset button -- if you have one. If you don't, get one. If after resetting, the system returns to BASIC, reboot with **RBOOT** *or* **128RBOOT***. Make sure that you have the fixed version of* **RBOOT** *for your system. If you have an REU and the system reboots ok, then your files should be in memory.*

If your monitor shuts off, whether intermittently or forever, you can still save a ***geoWrite*** *file. Press <RETURN> repeatedly, soon you will reach the bottom of the page and the program will update to disk -- your file is on the disk isn't it?*

In ***geoPaint*** *you just hope that if it ever happens, you just happen to be moving around in the document, meaning your pointer is connected to the four arrow scrolling icon, or the box in the little window on the bottom of the screen. If so, move the mouse and press the button.*

If you are in the editor in ***geoPublish****, the <RETURN>-pressing technique is your best bet. Keyboard short cuts like <C=><+> to turn the page will update the file. Press <RETURN> in case there is not a next page.*

Having a replacement monitor would be nice, but not likely -- break out that old monochromatic job from the closet.

CAPS LOCK CRUD

Remember to release the 128's <CAPS LOCK> key before you press the <C=><V> combination and do any formatting -- some fonts simply don't like it. They may turn to garbage and fill up several pages. If you do this, don't panic, release the lock key by pressing it again. Press <C=><V> again (only if you need to rehighlight) and reselect the font that went crazy. It should come back.

Send geoTips to: **dieHard**, P O Box 392, Boise, Idaho, 83701.

READY.

■

REVIEW!

by Brian L Crosthwaite

***** Excellent
**** Great
*** Good
** Poor
* Really Really Bad

The Duel: Test Drive II

from Accolade

Overall Rating: Excellent * * * * *

0 to 60 in less than 4 seconds and hardly a squeal from the tires as the G-force pushes you into the conforming leather seat. What's that? Too late-- "Hello officer."

You have two cars to chose from, the Ferrari F40 and the Porsche 959. But you'd better make up your mind fast because the computer or the clock is racing you and there's no time to dwaddle. Quick thinking and great control are the secrets to the game.

First let's take a look at the cars of ***Test Drive II.*** *The F40 has a 2936cc, twin turbo, 32 valve, V8 power plant. With a top speed of 201 miles per hour, it can cut the quarter mile in 12 seconds at 124 miles per hour. This mid engine, rear drive, 5 speed manual goes from 0 to 60 in just under 4 seconds.*

Equally devastating is the rear engine, all wheel drive, 959. Also twin turbo, this flat 6, 2849cc, with 6 speed manual cuts a quarter in 12 seconds at 116 miles per hour. Topping out at a respectable 197 miles per hour; this puppy will take you from 0 to 60 in 3.6 seconds.

Unlike lots of games that have those awesome pictures on the box of the "actual Amiga screens," and goofy graphics for the 64, these graphics are truly awe inspiring. Game play is top notch. Even though you will be steering with a joystick, the game is highly playable. At slow speeds you have to steer or the road will turn and you'll crash, unlike other driving simulators, where at slow speeds the road turns around the car nice and safe. For those just starting, you won't crash and burn every time, because you can actually adjust the skill levels of the game from infant to expert. At lower levels the car will shift for you.

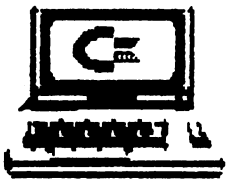
What you see is the cockpit just as if you were sitting in a real car. The image in the rear view mirror is realistic and animation is excellent. Sound effects are fairly well done too, although they didn't use as much of the SID chip's potential as they could have.

The opening music, however, is a different story. Great voicing here. The opening graphics are stupendous. The program plays a little animated sequence that is out of this world. Three cheers for the overall performance of the entire package.

The game itself is great. The setup screen allows you to select your car, preview the specs on either car, load additional scenery or cars from optional disks, and select play. When you play against the computer the first few times you'll see the computer's car pass you. But, as you get better that happens less and less. If you play the clock it won't spare you the humiliation of bad driving times, because the program will post it's comments as it reports the times for each driving segment. Comments like, "It's only a game."

My favorite car is the Ferrari. I think it handles better. Yes, handles better, your mind will be blown by the realism in handling. Turn power slides through twisting mountain roads. Don't run into that car in front of you, and watch out for the on-coming traffic! If you like simulations, you'll love ***The Duel.***

READY.
■

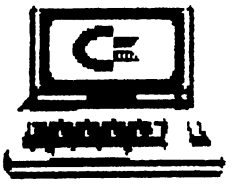


Archaic Computer

The Computer Store of the Past

by

Brian L Crosthwaite



Where were you in 1983? Did you own a computer? Had you ever used a computer? 1983 was a special time for me. That's when I got my first **commodore 64**. I already had a **Timex Sinclair 1000** and I was taking BASIC at Boise State University to learn how to use it. I remember driving my 1970 bug to the stadium parking lot, hopping on my skateboard and rolling across the campus to what was known then as the Math/Geology building.

HP 3000. My pass word was Umbria. I was logged on when I heard of the 64. Well, I didn't hear about it on the HP, I just happened to be working when someone spoke to me about it. It must have been a good two weeks before I got my 64.

My first commercially packaged game was Sierra Vision's **Crossfire**. My first Utility was **Simons' BASIC**. My first magazine was **Power Play**. All the programs seemed to be for the VIC. Once in a while, I'd run into a cool looking program that had some great graphics, but I could never run them because they were for the 64 with the **Super Expander** cartridge.

Time went by, and I finally got the **Super Expander**, of course I could never find those programs. Let us zip into the nineties. The capabilities of this cartridge are outrageous -- even by today's standards! **AC** takes a look at this oldy, but goody.

Super Expander 64

from Commodore Electronics Ltd 1983

• • • • Great

The **Super Expander Cartridge** adds 32 statements and functions for sound and graphics, much like BASIC 7 on the 128. In fact, if you want to convert programs from one to the other, you can. The graphics conversions are the easiest.

The BASIC tokens are not the same, so if you want to run a 128 program that is already on disk, you'll have to list it on a 128 and REM out all the statements (that is type REM after the line number) and resave it to disk. Then load the program into the 64 with the SE cart inserted. Remove the REMs. Some commands, like the sound commands will have to be rewritten. Others, like the joystick JOY, will have to be changed. (If anyone is interested we could do an in depth article on the conversion from 128 to SE 64 or SE 20 and visa versa.)

The graphic statements include:

GRAPHIC to call up a graphic screen or text screen.
BOX for squares and rectangles.
CIRCLE does circles, ellipsis, and polygons.
DRAW will set a dot or draw a line or a set of lines.
PAINT will fill a shape on the screen to a given color.

Hires and Multicolor graphics are supported. Text can easily be placed onto the hires screen. Shapes can be placed into and recalled from a string. All areas of graphics are supported, including sprites. In fact, there is a sprite designer just like the 128's.

Music can be played by PRINT and special strings. Although composing music seems easy in this manner, I don't like it. The greatest addition to any **commodore** BASIC was the SOUND statement, something they didn't add to the **SE**.

The function keys are programmable for use in direct mode. KEY keynum,"string" will define the key of your choice, making programming easier. I program <F1> to save my program then verify it, all with only one key stroke.

The manual is well written and well laid out. In the front is a command summary for quick reference. In the back is an extended version that gets into more detail. And if that doesn't do the trick, you can always look up the command in the table of contents or index to get to the complete story. There are programming notes in the first appendix explaining how memory is used, I/O, error handling, bit map displays and sprite and collision interrupts. Also in the back as well as throughout, there are example programs, making this a rather complete package.

My only complaints are the sound and tokens. There are probably numerous reasons that the tokens are different from the 128's, but it would have been nice to have them be the same. Not only from the 64 and 128, but from the VIC as well, especially since both the VIC and the 64 have the same BASIC in them. But this has little to do with what most people want from an extention language. I'd liked to have seen the SOUND command implemented.

The **64 Super Expander** is available from TEMEX Computer Express, 56800 Magnetic Drive, Mishawaka, IN, 46545.

READY.
■

Archaic Computer Q&A

by

Brian L Crosthwaite

Q: I am thinking of purchasing a cartridge port expander for my 128D. All my cartridges are for the 64 mode, except my 1750 REU which I can use in both modes. What I'm concerned with is will my 1750 work with one and will it work in the 128 mode. Also I've seen two advertised, the Navarone and the Aprospand, which do you recommend?

A: I have tested both of these units with an REU and have had no problem in either mode of my 128D. In fact, my 128D has a voice cartridge in slot one, the **Super Expander** in 2, **Simons' BASIC** in 3 and an REU in slot 4. While the first three carts are for the 64 only, the REU works fine in both modes. Some cartridges don't like

other cartridges, for instance my **Final Cartridge** works fine in place of the **Super Expander**, but my GEORAM will not show up upon booting **GEOS**. The trick is to experiment until you find what slot the cartridge works in and with what cartridges. You may find the first slot is the place most utility-fast load carts work best.

While I have not used the 1750 with either, both boards are said to work with Commodore REUs. I have heard people say that the REUs on their machines only work in the first slot. As far as the differences between the two go, the Navarone costs less, only has three ports, and you can only turn one cartridge on at a time. Both have reset buttons which is nice for 64 owners. The Aprospand has four cartridges slots that you can turn on in any order. You can turn more than one on at one time. Although I've never seen it, some cartridges will work together. It also has a fuse to protect your cartridges and computer. The idea behind the reset button is so you don't have to power down your computer to turn on a cartridge, but I recommend you do; it's just a good idea. Also take note, some manufacturers of the cartridges may recommend that the cartridge never be used with a port expander.

They are both good quality equipment, but if you can afford it I'd get the Aprospand, it gives you four slots and protection.

Q: I have a C64 and am thinking of getting a 64C, what is the difference between these two machines and will my peripherals work with it? I have a 1541 disk drive and 1802 monitor.

A: The ONLY difference is the case and the color of the keys. Some say the slopped design makes typing easier. I have never noticed a difference. Someone makes a case that looks similar to the 64C to turn your C64 into a 64C, but I've seen one and it is ugly compared to the real thing. They ARE THE SAME COMPUTER. However, if you have a really old 64, some of your chips may have some different characteristics. My first 64 had to have the screen cleared after the character color was set in order to see characters POKEd onto the screen or the color had to be POKEd as well. If you don't have any of the quarky chips from way back when, you might try typing on one to see if you really want to get one. All ports are the same and all peripherals will work with the 64C.

Q: I once saw a program that turned the Commodore 64's screen display upside down. I have an Atari 800XL, is there any way I can convert the program to run on it?

A: It may be possible, but not without a vast knowledge of both machines. You can, however display text upside down and backwards with a simple POKE. I think this was left over from game tables that displayed graphics one way for one player then the whole screen would flip over for the other player. POKE 755,5 turns the character set upside down and backwards. POKE755,1 turns it back. The only problem is that the screen will not be readable upside down, because the characters are backwards. But now anything you print on the screen will be inside out. If you POKE 755,4 you get the same effect, but the cursor will not flash and may be invisible.

READY.
■

PRG

by
Brian L Crosthwaite

Last month the **Spinner** had a program for the VIC 20 called **Alpha Count**. AC is a game for children of all ages. The program starts by asking if you want to play a game. The VIC version does this in a nonshalant way, as if someone has mysteriously appeared in the computer. This version starts similar, but the text color is black. Load the program in the computer and run it. Leave the computer on for the next unsuspecting person!

After some simple input from the user(s) the screen fills with color and the game begins. Instructions are included. While the premise is rather simple it will provoke thought on the behalf of the user. **Alpha Count** is listing one and runs on the 4, 16, 64 and 128.

Want an easy way to go from spite design to sprite on screen? **Ez-sprite 64** does just that. It demonstrates how the sprite can be designed right in the data statements using "•"s. Just type it in and run it. Press space if the computer is just sitting there and it will list out the data area. See listing 2.

For our 128'er out there, there is a simple cartoon for the holiday. Listing 3 runs only on the 128, but should be easily converted to the 64 with a **Super Expander**. The SOUND command will have to changed. This PRG has many things marked for clarification with REMarks.

For the VIC 20 there is **Stargate** and **Swirl III**, listings 4 and 5 respectively -- both mathamatical art. These are hires graphics on the unexpanded VIC. They will not run on an expanded VIC without modification. They will, however, run on a VIC with a **Super Expander**. Also notice it is **Swirl III -- I** and **II** are only on the **Spinner**.

Listing 6 is the same formula as it's VIC 20 counter part, **Swirl III** for the plus/4 and 16 is expanded slightly and the out-come's difference is startling. This PRG take well over an hour to finish.

Listing 7 is a pseudo-wedge for the plus/4 and 16 written entirely in BASIC 3.5.

Some parts of these programs are different on the **Spinner**, as we sometimes have to remove some flash to get things done. The **Spinner** also contains other programs not found in the **Flyer**. Readers are invited to share their work. We specialize in modification programs which means BASIC -- but we will accept machine code programs as well. I am very interested in mathematical art. Submitters who get published will receive a free issue of **dieHard the Flyer**. Send us your PRGs on disk or cassette, with a hardcopy of the listing and explanation of programs. One thing we do like are programs that demonstrate how to do a particular thing, like backwards scrolling or sprite use, that sort of thing. We love the strange. **dieHard**, P O Box 392, Boise, Idaho, 83701.

The programs are written in lower-case. Type them in with your machine in upper-case mode. Where you see a capitol, that means that key is used in conjunction with another key, ie: {shift T} don't type the braces, just type <SHIFT><T> to get the graphic on the right front or top of the key. ctrl means CONTROL and C= is the **commodore** key. If you see c= (lower-case c) that means type: c=, c is often used as a counter variable, so you might see c=c+1 or something like that. None of this month's PRGs use lower case. Type carefully and enjoy! Listings start on page 11.

READY.
■

DOS & Don'ts

DOS and DON'TS originally appeared in the premier issue of **LOADSTAR** and **LOADSTARs** there after. Because the information is as vital today as it was back then, we are running the complete series. The **Complete DOS and Don'ts** is available on 1541 disk from Softdisk for $9.95 +$4.50 shipping for 2nd day delivery. Softdisk, P. O. Box 30008, Shreveport, LA, 71130.

The NEW Command

by: Joel Ellis Rea

In this installment, we will discuss the basics of how the disk holds information, and how to do the four most important operations on a disk:

1. Format a NEW disk so that information can be organized on the surface of the disk.
2. SAVE a program onto the disk.
3. LOAD a program from the disk.
4. Make a BACKUP of the information from one disk to another in case Murphy visits!

We will not discuss the why's and wherefore's of the particular commands and syntaxes and such this time. We will just show the commands.

When you got your disk drive, you more than likely received a disk entitled "1541 TEST/DEMO". If you have no such disk, go out and buy a copy of the "1541 Bonus Pack" disk from a local dealer. It is inexpensive and very useful!

With the "1541 TEST/DEMO" disk in your drive, type:

```
LOAD "C-64 WEDGE",8:
```

(at this point hold down the SHIFT key and press the RUN/STOP key. DO NOT hit RETURN!)

This will load and run DOS Manager Version 5.1, also known as the DOS Support Program or the DOS Wedge. You should load the DOS Manager (hereafter called the Manager) every time you turn on your computer, except when you are just going to run a pre-packaged program like Loadstar.

If you have the Disk Bonus Pack instead of the 1541 TEST/DEMO disk, you will have to insert it instead and type:

```
LOAD "DOS 5.1",8,1
```

Wait for the drive to stop and the word "READY." to appear, then type:

```
SYS 52224
```

That will also activate the Manager for you.

The Manager makes it easier to use the disk drive. You can use the drive without it, but it is much more tedious. We will not discuss non-Manager commands this time.

Before you can use a new, blank disk, the disk must first be prepared for use by a 1541 drive. This preparation is called "formatting". This needs to be done only once per disk unless the disk is physically erased, partially or completely. You should format all of your blank disks as soon as you get them home!

To format a disk, type:

```
@NEW0:disk name,id
```

The "disk name" can be any name of your choice up to 16 characters long. It cannot contain a comma, colon, asterisk, question mark or quote. The "id" is a Disk Identifier. It must be exactly two characters long, with the same restrictions on legal characters as the "disk name". WARNING! The disk "id" MUST be DIFFERENT on EVERY DISK YOU OWN!!! Some examples:

```
@NEW0:JUNQUE
          DISQUE,JD
@NEW0:MY LETTERS
              #1,L1
@NEW0:ACCOUNTING
              5,A5
```

Some examples of illegal format commands:

```
@NEW0MY FAVORITE
          ,MF
```
(No colon)

```
@NEW0:"GOOFY" STUFF
     ,GS
```
(Quotes in disk name)

```
@NEW0:LOADSTAR
   PROGRAMS,LP
```
(Disk name too long)

```
@NEW0:DISK #1,1
```
(Id too short)

```
@NEW0:DATA DISK
```
(No id. This is legal, but the disk will not be formatted -- just erased and renamed. Use this to re-use an already formatted disk that has nothing on it you want to keep.)

Saving and Loading

Okay, now you have a formatted disk and you want to put stuff on it. With the Manager, to save the BASIC program you currently have in your computer, just type:

```
(arrow back)file
             name
```

where "file name" is a name for the file. It can be up to 16 characters long, with the same restrictions on legal characters as for the disk name. When the save is completed, the system will print the Disk Error Status IMMEDIATELY after the file name. If all went well, that should be

```
"00, OK,00,00"
```

An example:

```
(arrow back)STAR
      TREK
```
(saves the current program under the name STAR TREK. When done, the display should read:)

```
(arrow back)STAR
TREK
 SAVING STAR
      TREK00,OK,00,00
 READY.
```

To LOAD a program from the disk, you have three choices:

```
1.  /file name
2.  ↑file name
3.  %file name
```

Form 1 just loads a BASIC program into the normal BASIC memory space, erasing any program that might have been in memory before.

Form 2 does the same thing, but then RUNs the program as soon as it is LOADed.

Form 3 loads a machine language program or other memory image file at the same place it was SAVEd from. It is used for LOADing a program that you would otherwise have to use

LOAD"name",8,1 on.

Here are some examples:

```
/STAR      (Loads "STAR")
↑STAR      (Loads and runs
                "STAR")
%ROUTINES  (Loads the
    machine language "ROUTINES")
READY.
■
```

PAPSAW

by

Brian L Crosthwaite

Plotting Characters On Any Screen

On a commodore 64, you can draw circles on the screen using POKE and some math. It is relatively easy because you know the first screen code address (if you don't, look in your user's guide under **Screen and Color Memory Maps**). The VIC (Video Interface Controller) chip is a very versatile, highly programmable chip. The VDC (Video Device Controller) chip is also a very versatile, highly programmable chip. If you know how. I, for one, do not know the location of the first cell on the screen. I do know that there are two SYStems that control reading and writing to the chip, 52698 and 52684, respectively.

Within these SYStems is the ability to control various aspects of the VDC. But all I want to do is draw a circle. To whoever the responsible party is for the invention of the cursor key movements within a PRINT statement much praise is due. I'm talking of the reverse video characters you see when you enter quote mode.

Type:

```
"(cursor down)
```

See that reversed "Q" on the screen? That's what we are going to use, with the help of some of her friends. The following program will draw a circle on an 80 column screen. This will work on any **commodore** with 80 columns that supports the cursor control characters, even a 64 with an 80 column card such as Video Pak 80.

```
1000 PRINT"(CLR)"
  :XM=79:YM=24:XC=XM/2
    :YC=YM/2:RX=1/3*XM
                :RY=1/3*YM
1010 FOR N=1 TO 360
                    STEP 10
1020 X=INT(XC+RX*SIN(
          N/180*(pi)))
1030 Y=INT(YC-RY*COS(
          N/180*(pi)))
1040 PRINT"(HOME)";
        :FOR I=0 TO X
         :PRINT"(crsr
        right)";:NEXT
1050 FOR I=0 TO Y
  :PRINT"(crsr down)";
                 :NEXT
1060 PRINT"(shift q)"
1070 NEXT
1080 GET E$:IF E$=""
              THEN 1080
2000 PRINT"(CLR)"
                 :LIST
```

If you change the XM to 39 it will work with any 40 column screen. To run this program on the VIC, change XM to 21 and YM to 22.

These two variables, XM and YM are the deciding factors on screen size and are all that need to be changed for running this short program on any **commodore** computer.

The next step would be to decide how our program will discover what computer it is running on. One way would be user input:

```
100 INPUT"Enter
     maximum number of
 columns and rows";XM
                  ,YM
```

or

```
100 INPUT"ENTER
   COMPUTER TYPE":CT$
110 IF CT$="VIC20"
     THEN XM=21:YM=22
120 IF CT$="C64" THEN
          XM=39:YM=24
et cetera
```

This one could get hairy. What if the user enters "COMMODORE VIC-20" or "VIC" or "20." What we need is not to rely on the users input choice, but rather to give the user a choice of input.

```
100 PRINT"(CLR)
 ENTER COMPUTER TYPE
      (1, 2, 3 OR 4)"
110 PRINT"VIC
   20.............1"
120 PRINT"PLUS4 OR
        16........2"
130 PRINT"C-64......
          .........3"
140 PRINT"C-128 40
        COL.......4"
150 PRINT"C-128 80
        COL.......5"
160 GET CT$:IF CT$
>"5" OR CT$<"1" THEN
                160
et cetera
```

A menu cleans input up enormously. Line 160 allows only 1, 2, 3, 4 or 5 to be accepted as input. Line 170 might be something like:

```
170 ON VAL(CT$)
    GOSUB10000,20000
  ,30000,40000,50000
```

This would then execute a subroutine that could set up all the necessary variables such as start addresses for screens, pointers to execute certain routines and ignore others. In this case, the routines might look like this:

```
10000 REM VIC
10010 XM=21:YM=22
10020 RETURN
20000 REM 16+4
20010 XM=39:YM=24
20020 RETURN
30000 REM 64
30010 XM=39:YM=24
30020 RETURN
40000 REM 128 40 COL
40010 XM=39:YM=24
40020 RETURN
50000 REM 128 80 COL
50010 XM=79:YM=24
50020 RETURN
```

Of course, if you only needed to set the XM and YM variables, five subroutines would not be necessary since the +4, 16, 64 and 40 column 128 screens are usually the same size. If you needed to set up other things such as colors and sound, separate routines may be necessary.

Another way of going about this is to have the program check what computer it is on and set up from there. Sort of one of those this-is-what-computers-are-all-about kind of things. Last month the **Spinner** had one such program on it. **Alpha Slate**, as boring as it was, demonstrated well how this can be done. The subroutine at the beginning of the program called **COMPUTER CIPHER** looks into the BASIC tokenizer routine addresses and PEEKs a couple values out and converts them into single value, looks for a match and sets a flag to let the program know what to do to set up the color for the computer it is on. The color setup could have been done when the flag was set instead of setting the flag altogether, but it is done this was to accommodate a wide variety of set ups that may occur in the future. Here is that subroutine:

```
100 REM * COMPUTER CIPHER *
110 COMPUTER=PEEK(772)+256*PEEK(773)
120 IF COMPUTER=50556 THEN FLAG=20
130 IF COMPUTER=42364 THEN FLAG=64
140 IF COMPUTER=35158 THEN FLAG=+4/16
150 IF COMPUTER=17165 THEN FLAG=128:IF RGR(0)=5 THEN rem 80 column
160 IF COMPUTER=62580 THEN FLAG=2001
```

Now the program knows what computer it is running on. Checking FL (FLAG) will tell you what computer it is on. On the +4/16 FL will be .25 if you type PRINT FL, however, you can use "IF FLAG=+4/16 THEN" to make things clearer in your program.

Let's put it all together, with some downsizing by removing repetitive code. Take a close look at this simplified print routine (lines 1040-1050). I removed the cursor key movement characters and replaced them with PRINT and TAB statements. HOME can be replaced with CHR$(147) if you like.

```
100 REM ***** COMPUTER CIPHER *****
110 COMPUTER=PEEK(772)+256*PEEK(773)
120 IF COMPUTER=50556 THEN XM=21:YM=22:GOTO 1000:REM FLAG=20
130 XM=39:YM=24
140 IF COMPUTER=17165 AND RGR(0)=5 THEN XM=79:REM 128 80 COL
1000 ***** PLOTTER ROUTINE *****
1001 PRINT"{CLR}":XC=XM/2:YC=YM/2:RX=1/3*XM:RY=1/3*YM
1010 FOR N=1 TO 360 STEP 10
1020 X=INT(XC+RX*SIN(N/180*(pi)))
1030 Y=INT(YC-RY*COS(N/180*(pi)))
1040 PRINT"{HOME}";
1050 FOR I=0 TO Y:PRINT:NEXT
1060 PRINT TAB(X)"{shift q}"
1070 NEXT
1080 GET E$:IF E$=""THEN1080
2000 PRINT"{CLR}":LIST

READY.
■
```

This program should run on any computer that is not using an additional graphics card.

Rarities

by
Brian L Crosthwaite

This month's list has some rather unique entries indeed.

The President of the **Colorado Commodore Computer Club**, Ronald Snyder specializes in used equipment. VIC-20, C-64, C-128, PET, **GEOS**, **Printshop** and MS-DOS. He also has about 10,000 PD disks. He must have a warehouse or something. For more information write him at: Ronald Snyder, 1192 S. Nome Suite B, Aurora, CO, 80012. He has listings that fall into the above categories, they are $2 each.

Now that **RUN** magazine is gone, what of **RUN** Special Products? Never fear, TechMedia Special Products is here. The TechMedia Ancillary Products Manager is Tim Walsh and the catalogue has all of the products found formerly in **RUN**, including the advertising from other marketeers. They have included all of the ads that would have been in the January issue had it been a reality. Their address is TechMedia Special Products, 80 Elm Street, Peterborough, NH, 03458.

geoVISION International is a bi-monthly magazine that supports **GEOS** and its users. There is a disk also available that is a connection from the Nets, such as **GEnie** and **Quantum Link**, as well as from around the world. For more information: **geoVISION International**, 815 S. E. Polk Street, Camas, Washington, 98607-2240.

Send any info on **commodore** support to: **dieHard**, P.O. Box 392, Boise, ID, 83701.

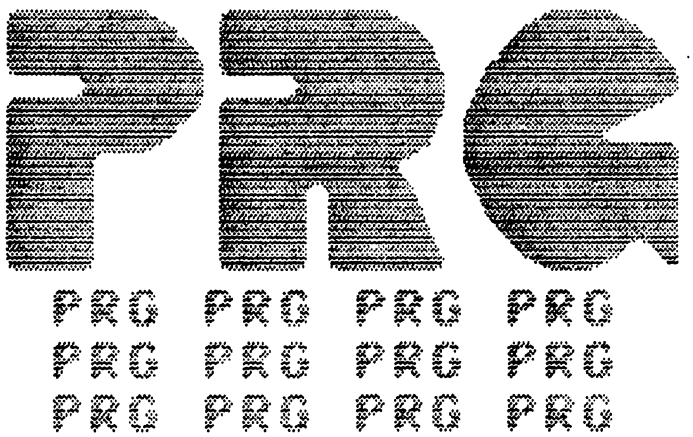
READY.
■

The programs of **PRG** are available on the disk known as the **Spinner**. The **Spinner** also has other programs in it not found anywhere in the **Flyer**. The **Spinner** is $5 each or $45 for a 10 disk subscription (Idaho residents add 5% sales tax). **dieHard**, The **Spinner**, P.O. Box 392, Boise, Idaho, 83701.

PRG now accepts Shareware prgrams! The shareware claim must be within the program or a file that appears when the program is run. LynnCarthy Industies will neither accept nor forward shareware donations, all fees are to be sent to the prorgammer.

Listing 1. Alpha-Count For the 128, 64, 16 & +4

```
0 rem
2 rem copyright 1993
3 rem lynncarthy ind
4 rem all rights reserved
100 rem *** computer cipher ***
110 computer=peek(772)+256*peek(773)
120 if computer=35159 then c=1
130 if computer=50556 then c=2
140 if computer=42364 then c=3
150 if computer=17165 then c=4
160 bank64,bank128:poke dc,113:s$="
    {9 crsr right}"
500 print"{2 HOME}{CLR}
    {ctrl 1}{10 crsr down}hello, do
     you want to play a game?"
510 geta$:if a$=""then510
520 if a$<>"y"thenprint"{CLR}":end
530 print"{CLR}{10 crsr down}
    {15 crsr right}oh, goody!"
    :fort=0to1900:next
534 input"{CLR}{10 crsr down}
    how many
players";pnif n<=0then
    n=1:elseif n>9then534
540 print"{CLR}{10 crsr down}
    my name is "cn$","
550 dimn$(n),w$(n),no(n)
    :forp=1ton:print"what is player
     number"P"{crsr left}'s name
     ?":inputn$(p)
560 next
900 print"{CLR}":gosub7000
1000 pokebc,2:pokesc,0:prints$"
   {2 crsr down}{2 crsr right}
   {ctrl 8}{shift U}{14 shift D}
    {shift I}"
1010 prints$"{3 crsr right}{shift B}{ctrl 7}
    lynncarthy ind{ctrl 8}{shift H}"
1020 prints$"{3 crsr right}{shift J}{14 shift F}
    {shift K}"
1030 prints$"{3 crsr down}{6 crsr right}{ctrl 2}
    {shift U}{8 shift *}{shift I}"
1040 prints$"{6 crsr right}{shift -}{shift S}
    presents{shift Z}{shift -}"
1050 prints$"{6 crsr right}{shift J}{8 shift *}
    {shift K}
1060 prints$"{3 crsr down}{4 crsr right}{ctrl 7}
    {shift N}{12 C= T}{shift M}"
1070 prints$"{4 crsr right}{C= G}{ctrl 5}a{ctrl 8}
    l{ctrl 7}p{ctrl 3}h{ctrl 2}a{2 space}{ctrl 5}
    c{ctrl 4}o{ctrl 7}u{ctrl 8}n{ctrl 5}{ctrl 7}
    {C= M}"
1080 prints$"{4 crsr right}{shift M}{12 ctrl 0}
    {shift N}"
1090 fort=0to2200:next
1120 gosub2000
1130 gosub3000
1140 gosub4000
1150 gosub5000
1160 gosub6000
1170 goto1130
1180 rem finalize and quit option
1999 end
2000 print"{HOME}"s$"{8 crsr down}{5 crsr
right}{C= A}{13 shift *}{C= S}
2002 prints$"{5 crsr right}{shift -} do you need
{shift -}
2004 prints$"{5 crsr right}{shift
-}instructions?{shift -}
2006 prints$"{5 crsr right}{C= Z}{13 shift *}{C=
X}
2010 geta$:if a$=""then2010
2012 if a$<>"y"then2999
2014 print"{CLR}
2020 prints$"{ctrl 2}the alphabet will be"
2030 prints$"displayed.{2 space}under each"
2040 prints$"letter is a number."
2050 prints$"spell words with the"
2060 prints$"highest numbercount."
2070 prints$"the player with the"
2080 prints$"highest count wins."
2092 print:print:prints$"press any key to start"
2900 geta$:if a$=""then2900
2999 return
```




```
3000 rem display game
3010 pokebc,6:pokesc,113:print"{CLR}{2 crsr down}"
3020 print"{3 space}{ctrl 1}a{2 space}{ctrl 3}b
    {2 space}{ctrl 4}c{2 space}{ctrl 5}d{2 space}
    {ctrl 6}e{2 space}{ctrl 7}f{2 space}{ctrl 8}g
    {2 space}{ctrl 1}h{2 space}{ctrl 3}i{2 space}
    {ctrl 4}j{2 space}{ctrl 5}k{2 space}{ctrl 6}l"
3030 print"{3 space}{ctrl 1}{2 space}{ctrl 3}2
    {2 space}{ctrl 4}3{2 space}{ctrl 5}4{2 space}
    {ctrl 6}5{2 space}{ctrl 7}6{2 space}{ctrl 8}7
    {2 space}{ctrl 1}8{2 space}{ctrl 3}9 {ctrl 4}
    10 {ctrl 5}11 {ctrl 6}12":print
3040 print"{3 space}{ctrl 7}m{2 space}{ctrl 8}n
    {2 space}{ctrl 1}o{2 space}{ctrl 3}p{2 space}
    {ctrl 4}q{2 space}{ctrl 5}r{2 space}{ctrl 6}s
    {2 space}{ctrl 7}t{2 space}{ctrl 8}u{2 space}
    {ctrl 1}v{2 space}{ctrl 3}w{2 space}{ctrl 4}x"
3050 print"{2 space}{ctrl 7}13 {ctrl 8}14 {ctrl 1}
    15 {ctrl 3}16 {ctrl 4}17 {ctrl 5}18 {ctrl 6}
    19 {ctrl 7}20 {ctrl 8}21 {ctrl 1}22 {ctrl 3}
    23 {ctrl 4}24":print
3060 print"{18 space}{ctrl 5}y{2 space}{ctrl 6}z
    {25 space}{10 space}{ctrl 5}25 {ctrl 6}26"
3999 return
4000 rem input & play
4010 print"{3 crsr down}":forp=1ton:print"
    {2 crsr up}{{ctrl 6}player:{ctrl 5}"n$(p)"{ctrl 1}"
4020 inputw$(p)
4022 print"{2 crsr up}{21 space}":print"{20 space}"
4030 forl=1tolen(w$(p))
4040 no(p)=no(p)+(asc(mid$(w$(p),l,1))-64)
4050 next:next
4999 return
5000 rem select winner
5010 w=0:w$="":forp=1ton
5020 ifno(p)>wthenw=no(p):w$=n$(p)
5030 next:p=n:print"{CLR}{3 crsr down}"
5034 forl=1top:cc=int(rnd(ti)*9)+1:cc$=mid$("
    {ctrl 3}{ctrl 4}{ctrl 5}{ctrl 6}{ctrl 7}{ctrl 8}
    {C= 1}{C= 2}{C= 3}",cc,cc)
5039 prints$cc$n$(l)"' count is"no(l):next
5040 prints$"{3 crsr down}{6 crsr right}{ctrl 3}
    "w$:prints$"{2 crsr right}{ctrl 5}is the
     winner!"
5050 fort=0to3900:next
5999 return
6000 rem play again
6010 print"{CLR}{10 crsr down}{ctrl 1}would you
    like to play again{20 space}":geta$:if a$=""then6020
6020 if a$<>"y"thenprint"{CLR}
    {10 crsr down}thank you very
     much for playing --
    {6 space}goodbye!":goto6050
6040 return
6050 forp=1ton
6060 printn$(p)" "::if p>1andp<nthen
    print" and"
6070 next
6080 fort=0tod900:next:print
    "{CLR}":end
7000 if c=4thenfast
7010 forx=0to39:fory=0to24
7020 r=int(rnd(0)*255)+1
    :r=r-(int(r/16)*16=r)
7030 pokesa+x+d0*y,81:poke
    ca+x+d0*y,r:pokebc,r
7040 next:next:if c=4thenslow
7050 return
60020 rem *** vic 20 ***
60030 print"{CLR}not for vic!":end
60063 end
60064 rem ***** 64 *****
60070 cn$="noesis"
60128 rem **** c128 ****
60130 bc=53280:sc=53281:sa=1024:ca=55296
60140 if c=4thengraphic0:cn$="edgar"
60400 return
60415 end
60416 rem * plus4 & 16 *
60420 bc=65305:sc=65301:sa=3072:ca=2048
    :cn$="geist"
60500 return
60600 end
```

Listing 2. Ez-sprite 64 For the 64.

```
0 rem
1000 rem{3 space}reading sprites from pictures
    {11 space}drawn in the data statements
1010 print"{CLR}{crsr down}{15 space}{ctrl 3}
    ez-sprite"
1012 print"{crsr down}draw sprites directly in data
     statements
1020 print"{2 crsr down}{18 space}{ctrl 4}by"
    :poke53280,1:poke53281,7
1030 print"{3 crsr down}{10 space}{ctrl 7}
    {19 C= 0}
1040 print"{10 space}{ctrl 7}{rvs on}
    brian l crosthwaite
1050 print
1060 print"{8 space}{ctrl 6}copyright 1993
     lynncarthy ind.
1070 print"{9 space}{ctrl 6}all rights reserved
1080 print
1090 print
1100 print"{8 space}{rvs on}{ctrl 2}{shift british
pound}{C= 8}{22 space}{C= 5}{C= *}{rvs off}
1110 print"{8 space}{rvs on}{C= 8} {C= 5}press
    <space> to begin{C= 4} {rvs off}
```

```
1120 print"{2 space}{C= 5}{C= 8}{rus on}{C= 4}
   {22 space}{rus off}{ctrl 1}
   {shift british pound}
1190 rem nu=number of sprites
1200 cz=0:sa=53248:sc=53287:nu=1:dim$$(20*nu)
   :rem nu can be up to 8 sprites
1210 get$:if$=""then1210
1220 gosub2000:rem read data
1230 gosub3000:rem set up sprites
1240 gosub4000:rem convert data
1250 a$="":get$:if$=""then1250
1260 gosub5000:list3050-3250
1999 end
2000 rem read data into strings
2010 for i=0to20*number of sprites:read$$(i):next
2020 return
2999 end
3000 rem set up sprites
3004 sn=nu-1
3010 pokesc+sn,0:rem sprite color 0
3020 poke53277,peek(53277)and(255-2^sn):
   {17 space}rem unexpand x
3030 poke53271,peek(53271)and(255-2^sn):
   {17 space}rem unexpand y
3040 pokesa+sn,255:pokesa+sn+1,100{5 space}:
   {17 space}rem position sprite
3050 poke53269,peek(53269)or(2^sn){5 space}:
   {17 space}rem turn sprite on
3300 return
3999 end
4000 rem convert strings into sprites
4002 poke2040,192:cu=7
4010 print"{CLR}{2 crsr down}{crsr left}"
   :for i=0to20*nu
4020 for j=1to24
4026 p=-(asc(mid$(a$$(i),j,1))=42)
4030 printmid$(str$(p),2,1);
4032 ifp=0andcu=0thenpu=puor0:gotod038
4034 pu=puor((2^-(p=1))^cu)
4038 cu=cu-1:ifcu<0thencu=7:poke192*64+cz,pu
   :pu=0:cz=cz+1
4040 next:print
4050 next:cz=cz+1
4300 return
4999 end
5000 rem end
5004 sn=nu-1
5010 poke53269,peek(53269)and(255-2^sn)
5020 print"{CLR}"
5300 return
5999 end
6000 rem{2 space}sprite data
6001 rem{12 space}111111111112222
6002 rem{2 space}012345678901234567890123
6003 rem{2 space}--------------------------
6004 rem{2 space}765432107654321076543210
6005 rem{2 space}--------------------------
6010 rem{2 space}1{7 space}1{7 space}1
6020 rem{2 space}2631{4 space}2631
   {4 space}2631
6030 rem{2 space}842684218426842184268421
6040 rem{2 space}--------------------------
6050 data"{24 space}":0
6060 data"{24 space}":1
6070 data"{10 space}*****{10 space}":2
6080 data"{7 space}*****{4 space}*****
   {7 space}":3
6090 data"{6 space}*{10 space}*{6 space}":4
6100 data"{6 space}*******{4 space}*******
   {6 space}":5
6110 data"{6 space}*{3 space}*******{3 space}
   *{6 space}":6
6120 data"{6 space}*{10 space}*{6 space}":7
6130 data"{6 space}*{10 space}*{6 space}":8
6140 data"{4 space}*****{10 space}*****
   {4 space}":9
6150 data"{3 space}* *{12 space}* *
   {3 space}":10
6160 data"{2 space}*{2 space}*{12 space}*
   {2 space}*{2 space}":11
6170 data"{2 space}*{2 space}***{10 space}***
   {2 space}*{2 space}":12
6180 data"{2 space}*{4 space}*****{4 space}
   *****{4 space}*{2 space}":13
6190 data"{3 space}*{6 space}*******{6 space}
   *{3 space}":14
6200 data"{4 space}***{12 space}***
   {4 space}":15
6210 data"{6 space}*****{6 space}*****
   {6 space}":16
6220 data"{3 space}*********{3 space}":17
6230 data"{24 space}":18
6240 data"{24 space}":19
6250 data"{24 space}":20
```

Listing 3. The Trip to St. Valentine's Day For 128. ___}**

```
130 rem **february 1993 diehard spinner**
140 rem **{5 space}brian l crosthwaite{5 space}**
150 rem **{6 space}happy valentine's!{6 space}**
160 rem **{12 space}enjoy{12 space}**
170 rem {30 *}
1000 rem ** title screen & sprite data **
1100 gosub4000
1200 rem **diehard spinner**
140 rem **{5 space}brian l crosthwaite{5 space}**
150 rem **{6 space}happy valentine's!{6 space}**
160 rem **{12 space}enjoy{12 space}**
170 rem {30 *}
1000 rem ** title screen & sprite data **
1100 gosub4000
1200 rem ***** clear screen *****
1300 print"{CLS}"
1400 rem **** plains ****
1500 gosub5000
1600 rem ***** house *****
1700 gosub3000
1800 rem ***** turn off sprites ****
1999 gosub4303:print"{2 HOME}{CLR}":end
2000 rem *** clear sound routine ***
2010 sa=54272:forj=satosa+24:poke j,0:next
2020 rem{2 space}forzn=0to83:readhz:next
2300 return
2999 end
3000 rem *** house ***
3002 vol15:sound1,900,600,2,1200,3000,1
3011 print"{2 HOME}{CLR}{ctrl 2}{crsr down}
   {2 crsr right}you search for that one
    true love..."
3012 sprite2,1,1,0,1,0,0
3014 print"{HOME}{11 crsr down}"chr$(27)"t
   {ctrl 7}{crsr down}{crsr left}{10 crsr down}
   "chr$(27)"b";:color0,10
3020 for i=0to299:print"{2 C= 8}{2 C= 5}
   {2 shift R}{2 shift F}{2 shift C}{2 shift D}
   {2 shift E}{2 C= 7}{2 C= 7}";
3030 next
3040 print"{HOME}{ctrl 8}{crsr down}
   {2 crsr right}{2 space}you run out of gas and
    decide that"
3044 print"{4 space}you must walk.":sleep5
3052 sprite2,0:sprite3,0
3060 print"{2 HOME}{CLR}";:color0,2
3070 print"{C= 4}{19 space}{shift M} {C= M}
   {12 space}";
3080 print"{18 space}{shift M}{2 space}{C= M}
   {12 space}";
3090 print"{17 space}{shift M}{3 space}{C= M}
   {12 space}";
3100 print"{16 space}{shift M}{4 space}{C= M}
   {12 space}";
3110 print"{15 C= V}{shift P}{5 space}{C= M}
   {12 space}";
3120 print"{15 space}{C= N}{5 space}{C= M}
   {12 space}";
3130 print"{15 space}{C= N}{5 space}{C= M}
   {12 space}";
3140 print"{15 space}{C= N}{5 space}{C= M}
   {12 space}";
3150 print"{15 space}{C= N}{5 space}{C= M}
   {12 space}";
3160 print"{15 C= P}{shift B}{5 space}{C= M}
   {12 space}";
3170 print"{16 space}{shift M}{4 space}{C= M}
   {12 space}";
3180 print"{17 space}{shift M}{3 space}{C= M}
   {12 space}";
3190 print"{18 space}{shift M}{2 space}{C= M}
   {12 space}";
3200 print"{19 space}{shift M} {C= M}
   {12 space}";
3210 print"{20 space}{shift M}{C= M}{18 space}";
3220 print"{21 space}{19 C= V}";
3230 print"{25 space}{15 space}";
3240 print"{3 crsr down}{2 crsr right}{ctrl 6}
   several hours later you find yourself
   {3 space}in a strange house...";
3250 sprite7,1,1:sleep6
3260 print"{2 crsr left} looking at a{3 space}
   painting."
3270 char1,12,10,"{C= 1}{C= M}{crsr up}{shift M}
   {C= M}{crsr down}{crsr left}{C= M}
   {crsr down}{crsr left}{C= M}{2 crsr left}
   {shift M}{crsr up}{crsr left}{shift S}"
3280 sleep6
3290 print:print"{4 crsr down}"chr$(27)"t{CLR}"
3300 print"{2 crsr right}suddenly, she sees you
    and yet you{6 space}remain unaware."
3310 sprite6,1,1
3320 sleep3:for i=0to9:a=a+1:if a>1then a=0
3322 for d=0to299:next:sound1,200,1
3326 sprite4,a,3:next
3330 print"{CLR}{ctrl 1}{9 crsr right}the next
   thing you know{13 space}you have a
   mortgage!"
3340 sprite6,0:sprite7,0:sprite2,1,2:color0,2
   :sprite5,1,3:sound1,2200,20,0,1200,100,2,4000
3350 envelope7:tempo3:play"ha q$a $r g f d"
3390 sleep9:return
3999 end
4000 rem *** title routine ***
4010 for c=1to5:colorc,c+1:next:color0,2
4020 print:print" {ctrl 5}the trip to {ctrl 2}
   saint {ctrl 6}valentine's {ctrl 7}day"{25
space}
   {2 space}";
4030 for z=1to20:print:next
4210 print"{20 space}{ctrl 4}by"
4220 print"{20 space}{ctrl 5}brian l
   crosthwaite ";
4230 print"{ctrl 2}{40 C= @}";
4240 print"{ctrl 2}{rus on}(c)1993 lynncarthy
   ind{2 space}{rus off}";
4260 poke2022,132:poke56295,1
4272 rem *** clear sound ****
4274 gosub2000
4280 rem ***** init sprites *******
4290 u=dec("0600")
4300 for i=u+a032step64:forp=itoi+62:if i=uthen
   poke p,255:goto4303
4304 read d:poke p,d
4305 next p,i
4309 rem sprite 1{14 space}block
4310 sprite1,0,1,0,0,0,0
4311 rem sprite 2{14 space}car
4312 sprite2,0,1,0,0,0,0
4313 rem sprite 3{14 space}sun
4320 sprite3,0,2,0,0,0,0
4321 rem sprite 4{14 space}heart
4322 sprite4,0,3,0,0,0,0
4323 rem sprite 5{2 space}heart w/arrow
4330 sprite5,0,5,0,0,0,0
4331 rem sprite 6{14 space}girl
4332 sprite6,0,1,0,0,0,0
4333 rem sprite 7{14 space}boy
4340 sprite7,0,1,0,0,0,0
4341 rem sprite 8{14 space}kiss
4342 sprite8,0,2,0,0,0,0
4350 gosub3000
4999 return
5000 rem ***** rolling plains *****
5002 vol15:sound1,900,0670,2,1700,3000,1
   :sprite2,1,8
5010 print"{2 HOME}{CLR}{7 crsr down}
   {2 crsr right}{ctrl 2}{32 C= @}"
5011 print"{HOME}{ctrl 1}{2 crsr down}
   {crsr down}the rolling plains out the window
    of{4 space}your motor car."
5012 print"{HOME}{ctrl 1}{crsr down}{crsr right}
   {rus on}{C= J}{repeat the following key
   strokes 24 times total (don't type this)
   {crsr down}{crsr left}{C= J}}{rus off}";
5014 print"{HOME}{5 crsr down}{4 crsr right}
   {ctrl 8}{shift Q}{5 crsr left}{2 crsr right}
   {6 crsr down}"chr$(27)"t{ctrl 7}";:color0,10
5020 for i=0to299:print"{2 C= 8}{2 C= P}
   {2 shift R}{2 shift F}{2 shift C}{2 shift D}
   {2 shift E}{2 C= V}{2 C= T}";
5030 next
5999 return
9000 rem *** reset sprites ***
9010 mouspr1,170,220:rem block
9020 mouspr2,170,220:rem car
```

```
9030 mouspr3,045,030:rem sun
9040 mouspr4,060,143:rem heart
9050 mouspr5,090,120:rem heart w/arrow
9060 mouspr6,060,140:rem girl
9070 mouspr7,110,140:rem boy
9080 mouspr2,090,140:rem kiss
9090 for i=1to3:sprite i,0:next
9999 return
59999 end
60000 rem ******* sound *******
60090 return
60999 rem ******* sprite data *******
60000 data 0,0,0,0,0,0,0,0
60010 data 0,0,0,0,0,0,0,0
60020 data 0,0,0,0,0,0,0,0
60030 data 3,240,0,3,12,0,3,74
60040 data 0,39,73,0,63,255,252,32
60050 data 132,2,96,196,2,60,132,123
60060 data 66,120,132,25,255,48,36,0
60070 data 72,36,0,72,24,0,48,0
60080 data 0,0,0,0,0,0,0,0
60090 data 0,0,255,0,3,255,192,7
60100 data 255,224,15,255,240,31,255,248
60110 data 31,255,248,63,255,252,63,255
60120 data 252,63,255,252,63,255,252,31
60130 data 255,248,31,255,248,15,255,240
60140 data 7,255,224,3,255,192,0,255
60150 data 0,0,0,0,0,0,0,0
60160 data 0,0,0,0,0,0,0,0
60170 data 0,3,199,128,7,239,192,15
60180 data 255,224,15,255,224,31,255,240
60190 data 31,255,240,15,255,224,15,255
60200 data 224,7,255,192,3,255,128,1
60210 data 255,0,0,254,0,0,124,0
60220 data 0,56,0,0,16,0,0,0
60230 data 0,0,0,0,0,0,0,0
60240 data 64,0,0,208,0,0,48,0
60250 data 0,115,199,128,15,239,192,11
60260 data 255,224,12,255,224,30,255,240
60270 data 31,127,240,15,255,224,15,255
60280 data 224,7,255,192,3,255,128,1
60290 data 255,0,0,254,128,0,124,64
60300 data 0,56,40,0,16,24,0,0
60310 data 56,0,0,0,0,0,0,0
60320 data 0,0,0,0,0,0,0,0
60330 data 0,0,0,0,0,0,0,0
60340 data 0,0,0,224,0,1,176,0
60350 data 1,136,0,1,80,0,1,32
60360 data 0,0,48,0,0,56,0,0
60370 data 48,0,0,48,0,0,48,0
60380 data 0,128,0,0,252,0,0,32
60390 data 0,0,32,0,0,48,0,0
60400 data 0,0,0,0,0,0,0,0
60410 data 0,0,0,0,0,0,0,0
60420 data 0,0,0,224,0,1,176,0
60430 data 1,136,0,1,80,0,0,32
60440 data 0,0,48,0,0,48,0,0
60450 data 48,0,0,48,0,0,112,0
60460 data 0,112,0,0,48,0,0,32
60470 data 0,0,32,0,0,48,0,0
60500 data 0,0,0,0,0,0,0,0
60510 data 0,0,0,0,0,0,0,0
60520 data 0,0,0,225,192,1,179,96
60530 data 1,140,96,1,82,160,1,33
60540 data 0,0,51,0,0,59,0,0
60550 data 51,0,0,51,0,0,51,128
60560 data 0,123,128,0,255,0,0,33
60570 data 0,0,33,0,0,51,0,0
```

Listing 4 Stargate For the VIC 20.

```
0 rem 127*127
2 poke52,20:poke56,20:clr
10 gosub1000
20 xm=127: ym=127
22 x=xm/2:y=ym/2
30 xf=ym/xm:yf=xm/ym
100 for x=0to xm:y=0:gosub2000:y=ym:gosub2000
    :next
120 for y=0to ym:x=0:gosub2000:x=xm:gosub2000
    :next
420 for r=4to xm:for a=1to360step10
440 x=int(x+r*xf*sin(n/180*(shift ^3)))
450 y=int(y+r*yf*cos(n/180*(shift ^3)))
460 gosub2000:remplot
470 next:next
480 goto480
999 end
1000 poke36269,253
1010 for i=5120to7679:poke i,0:next
1012 rem for i=5120to7679:poke i,peek(i+27648):next
1020 poke36879,8:print chr$(147)
1030 for i=7680to8185:poke i,160:next
1040 for i=0to15:for m=0to15
1050 poke 7749+m+22*i,i*16+m
1060 next:next
1070 return
2000 rem plot x,y
2002 if x>xmorx<0ory>ymory<0then2050
2003 ch=int(x/8)*16+int(y/8)
2010 ro=(y/8-int(y/8))*8
2020 by=5120+8*ch+ro
2030 bi=7-(x-(int(x/8)*8))
2040 poke by,peek(by)or(2^bi)
2050 return
```

Listing 5 Spiral III For the VIC 20.

```
2 poke52,28:poke56,28:clr
10 gosub1000
20 xm=127: ym=127
22 x=xm/2:y=ym/2
30 xf=ym/xm:yf=xm/ym
100 for x=0to xm:y=0:gosub2000:next
120 for y=0to ym:x=0:gosub2000:x=xm:gosub2000
    :next
420 for r=1to9143step1:r=r+.01
440 x=int(x+r*xf*sin(n/1210*(shift ^3)))
450 y=int(y+-r*yf*cos(n/2000*(shift ^3)))
460 gosub2000:remplot
470 next
480 goto480
999 end
1000 poke36869,253
1010 for i=5120to7679:poke i,0:next
1012 rem for i=5120to7679:poke i,peek(i+27648):next
1020 poke36879,8:print chr$(147)
1030 for i=7680to8185:poke i,160:next
1040 for i=0to15:for m=0to15
1050 poke7749+m+22*i,i*16+m
1060 next:next
1070 return
2000 rem plot x,y
2002 if x>xmorx<0ory>ymory<0then2050
2003 ch=int(x/8)*16+int(y/8)
2010 ro=(y/8-int(y/8))*8
2020 by=5120+8*ch+ro
2030 bi=7-(x-(int(x/8)*8))
2040 poke by,peek(by)or(2^bi)
2050 return
```

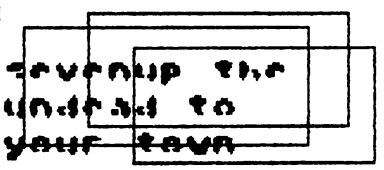
Listing 6 Spiral III For the 16 and Plus/4.

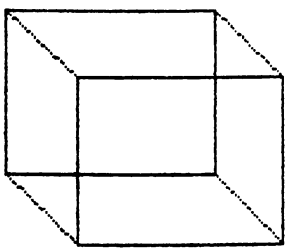
```
10 gosub1000
20 xm=319: ym=199
22 x=xm/2:y=ym/2
30 xf=ym/xm:yf=xm/ym
100 for x=0to xm:y=0:gosub2000:y=ym:gosub2000
    :next
120 for y=0to ym:x=0:gosub2000:x=xm:gosub2000
    :next
420 for r=1to9143step1:r=r+.01
440 x=int(x+r*xf*sin(n/1210*(shift ^3)))
450 y=int(y+-r*yf*cos(n/2000*(shift ^3)))
460 gosub2000:remplot
470 next
480 goto480

999 end
1000 graphic1,1
1070 return
2000 rem plot x,y
2002 if x>xmorx<0ory>ymory<0then2050
2003 draw1,x,y
2050 return
4000 rem(27 *)
4010 rem*copyright 1993 lynncarthy*
4020 rem*(3 space)all rights reserved(3 space)*
4022 rem* diehard the spinner 2/93*
4030 rem(27 *)
```

Listing 7 GeistWedge For the 16 and plus/4.

```
0 remember copyright 1993 lynncarthy ind.
    (14 space)all rights reserved
2 poke1344,0:graphic clr:print"(2 HOME)(CLR)";
3 color1,7,0:color4,7,0:color0,14,5
4 c$="(HOME)(crsr right)(3 crsr down)
    "+chr$(27)+"t(15crsr down)"
5 c$=c$+"(37 crsr right)"+chr$(27)+"b"
10 print"(C= A)(38 shift *)(C= S)"
20 print"(shift -) ";:color1,1
21 print"geist ";:color1,2
22 print"system 164 ";:color1,13,2
24 print"microsoft ";:color1,12,0
26 print"basic 3.5";:color1,7,0
28 print" (shift -)"
30 print"(C= Q)(shift *)(C= W)"
40 print"(shift -) f1=graphic(3 space)
    f4=scratch(3 space)(shift -)"
50 print"(shift -) f2=dload(10 space)
    f5=save/verify(5 space)(shift -)"
60 print"(shift -) f3=directory(6 space)f6=run
    (13 space)(shift -)"
67 print"(shift -) help=help(9 space)f7=list
    (12 space)(shift -)"
70 print"(C= Q)(shift *)(C= W)"
80 for i=0to15
84 print"(shift -)(38 space)(shift -)"
88 next
99 print"(C= Z)(shift *)(C= X)";
100 print c$:poked ec("0:00")+999,125
110 print"(CLR)(3 space)**** geist
    (2 space)system ****"
120 print"(7 space)****"fre(0)"bytes
    free ****"
130 print"(10 space)***** leosoft *****"
140 key1,"graphic"
150 key2,"dload(13 crsr right):"+chr$(13)
160 key3,"(CLR)directory"+chr$(13)
170 key4,chr$(143)+chr$(143)+"scratch
    (13 crsr right):"+chr$(13)
180 key5,":save$,8:ve8$,8"+chr$(13)
190 key6,"(CLR)run"+chr$(13)
200 key7,"(CLR)list"+chr$(13)
210 key8,"help"+chr$(13)
220 new:end
```

READY.





Oh, why not, put me down for a subscription to **dieHard**. Enclosed is $___.___ (Idaho residents add 5% sales tax) for:

_The Flyer $15.00 one year ten issue subscription
_Single issue price $1.75

_The Spinner $45.00 one year ten issue subscription
_Single issue price $5.00

BACK ISSUES!

__Issue # 1 HardCore May 1992 .....$2.50
__Issue # 2 WPC June 1992 .....$2.50
__Issue # 3 Oops! July 1992.....$3.50
__Issue # 4 Pumpkin October 1992.....$2.50
__Issue # 5 geoIssue! November/December 1992..$2.50
__Issue # 6 History In 3001 January 1993.....$2.50

Name:_____
Address:_____
City:_____State:____Zip:_____

Mail to: **dieHard**, P O Box 392, Boise, Idaho, 83701.

Check or Money order on U.S. Funds only please.

---

dieHard
P O Box 392
Boise ID 83701