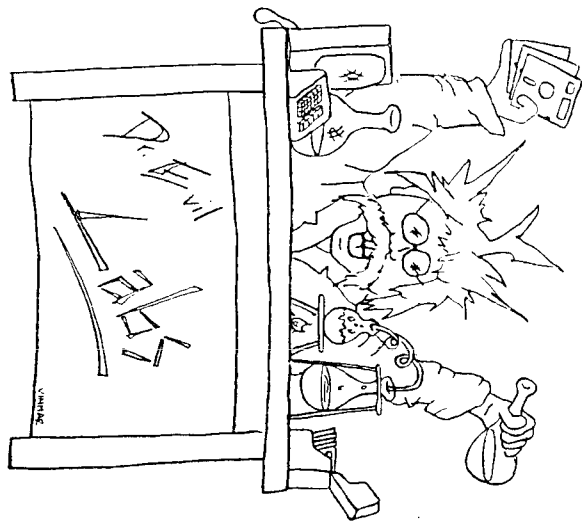


Dr. Evil Laboratories
P.O. Box 190
St. Paul, IN 47272

THE IMAGE

Volume 1, Number 1 August, 1987

SidFest '87: Memorable

by Kent Sullivan

Dan, George, and I recently attended what was certainly one of the most unique Commodore computer-related gatherings ever--SidFest '87. This event, sponsored by the Commodore Club of Central Ohio (CCCO), was a one-time gathering of people involved with and interested in the excellent Sidplayer music composing/playing system for the Commodore 64/128. SidFest was held in the Rhodes Center on the Ohio State Fairgrounds on Saturday, June 27, from 10 AM to 6 PM. Sidplayer was designed by Craig Chamberlain and Harry Bratt, and is published by Compute! Books (for a description of the Sidplayer system, please see the Enhanced Sidplayer review elsewhere in this issue).

Almost every famous programmer and/or composer known for his/her Sidplayer tunes was present at SidFest. Among those present, in no particluar order, were: Craig Chamberlain, Harry Bratt, Brian Copeland, Brian Szepatowski, John Mackey, Bob Huffman, Bob Retelle, Ellen Kauffman, Mark Dickenson, DC Star, Dr. J (Jerry Roth), Bobbye, Sysop Jon (Jon Rafalak), Jabba Hut, Kermit Woodall, Nick Zelinsky, and Diz Cop (note that some of the people in the list could only be identified by their bbs handles). Whew!! Over 200 people attended the event, half of which were from out-of-state.

The fest was divided into a small vendor section and meeting rooms for seminars and programs. The only music-related vendor at SidFest was Coyle Music of Columbus. Jerry Wade of Coyle gave an interesting demonstration of some of the educational packages he sells to schools for music education (Coyle Music, 2864 N. High St., Columbus, OH, 43202, (800) 282-0700).

After a little more than an hour's worth of socializing, Ken Crosby, president of CCCO, opened SidFest officially. After remarks by several people, John Mackey performed two incredible Sidplayer pieces he transcribed, The Magnificat by J. S. Bach and the Ninth Symphony, final movement (Ode to Joy) by Beethoven, using six C-64's simultaneously, giving a total of eighteen voices!!! The sound was incredible, to say the least! (The two pieces are available from CCCO on audio cassette for $5.00 at CCCO, P.O. 292392, Columbus, OH 43229)

Participants then broke off into groups to discuss different aspects of using Sidplayer, while others visited the displays and demonstrations. A display of the VIC-20, C-16, Plus 4, C-64, and the C-128 was interesting (I had never seen a C-16 up close), while Craig and Harry had their original Pokey Player for the Atari (on which Sidplayer is based) next to Sidplayer and Enhanced Sidplayer on a C-64 and C-128, respectively. The roots for Sidplayer could be seen clearly in Pokey Player.

Mark Dickenson demonstrated his novel SidStereo player and hardware modification. By carefully following Mark's instructions (it's not easy by a long shot), you can add a second SID chip to your C-64 or C-128 and with SidStereo produce true stereophonic tunes. A transcription of "John B. Goode" (Chuck Berry) by Jerry Roth was really rockin' in stereo! Mark is working on a cartridge version of SidStereo that will make it much easier to have SidStereo. The Image will keep you informed of his developments.

Brian Szepatowski, known for his great Sidplayer transcriptions of tunes like the themes to Star Trek and Star Trek II, demonstrated an impressive MIDI setup including synthesizers, a reverb, and a drum machine.

The party was grounded for almost two hours unfortunately by the one thing no computer convention can afford to have happen: a power outage. Rhodes Center became a eerie place without any electricity. Dan, George, Craig, and others all sat through lunch in the dark in the center's cafeteria. Conversation didn't stop just because of no power, thank goodness!

An impromptu Q & A session was hastily organized to pass the time in the dark. Most of the above-mentioned Sidplayer personalities fielded questions on many topics. The main topic of interest seemed to be the popular SidPic player, which displays a hi-res graphics screen while playing a song.

When the power finally did return, many of the participants were ready to head home. Many of the seminars were cancelled, and the door prize drawing was held early. SidFest '87 was billed as "A Celebration of Computer-Generated Music" and it fulfilled its promise-- but it was a shame the electricity was interrupted!!

Editors ..... **Kent Sullivan**
Dan Moore
Columnist ..... **Rob Tillotson**
Artwork ..... **Vince Martin**

The Image

Published quarterly for *Imagination* members only by:

Dr. Evil Laboratories
P.O. Box 190
St. Paul, IN 47272

you can simply click on the file left by the assembler and it will create a GEOS-ready copy of your program for you. This approach has the advantage of being almost totally automatic.

Now, at long last, you're ready to program. You will find that GEOS programs have a significantly different structure than standard C-64 programs. GEOS is an "event-driven" system, which means that the GEOS Kernal automatically moves the mouse, checks for clicks, and calls the appropriate portion of your program to process the action. Your program will often consist of many subroutines, each of which is dedicated to a particular icon or menu item. It is important when designing GEOS programs to decide the external appearance of the program first, since in GEOS the user controls your program and not the other way around.

It should be noted that there are some things that you can't do in GEOS. The RS-232 port is not supported, although you can write your own routines to use it; in fact, BSW has already done so in a few of the many GEOS printer drivers. A much more significant limitation is that GEOS only supports the 1541 drive. You can use other drive types, but doing so requires bypassing GEOS temporarily. Also, GEOS controls the hi-res graphics screen, one sprite, and the IRQ interrupt, so programs that require any of these things exclusively will probably interfere with GEOS. Of course, any program that requires that much control of the system would probably not benefit much from using GEOS in the first place.

Finally, in the interest of consistency you should design your GEOS software to look and feel as much like the existing GEOS programs as you can. That is part of the appeal of GEOS; if you know how to use one application, you know how to use them all. A set of specific guidelines could fill a book (in fact, in the case of the Apple Macintosh, they DO fill a book!) but there are just a few basic principles you should follow. Keep your dialog boxes and menus in the same format as BSW does. Use the standard Text Scrap and Photo Scrap formats for data exchange. And use system defaults wherever possible. The authors of the various books on GEOS never touched on the issue of consistency, but with the recent upsurge in third-party GEOS programming I think it's about time someone does.

Well, I hope this article has provided you with enough practical information to get you started programming for GEOS. I will be writing a followup to this article in which I will cover some specific information on GEOS internals. In the meantime, feel free to write to me either on paper or electronically with your questions and comments.

(Editor's note: As a service to its customers, Dr. Evil Laboratories is offering a disk of programs to accompany this article. The disk contains assembly source code to example GEOS programs, the public-domain C-ASSM assembler that works within the C-Power environment, a set of header files containing BSW-standard label definitions suitable for use with any assembler, and several GEOS utilities. The cost of the disk is $5.00. Please make all checks payable to Dr. Evil Laboratories. Indiana residents: please add 5% sales tax.)

is very accurate, but it uses short label names (6 characters as opposed to BSW's 10 or more) and many absolute references (BSW uses labels and named constants). Also, some of the terms the author uses are different from those BSW uses for the same concepts. This document is quite useful, since it fills in the gaps in the BSW book, and it should serve as a useful overview of GEOS internals if you are hesitant to spend $20 for the BSW book.

Now you've got an assembler and some documentation. Next, you should prepare a file of label definitions for your assembler, so that you can include it into your programs. Since GEOS has so many Kernal entry points, important memory locations, and constants, you should use multiple definition files divided according to subject. The BSW book has a complete set of symbol definitions in it; but, if you use these definitions verbatim your assembler should support labels of at least 10 characters or more. Otherwise, you will have to shorten them, make up names of your own, or use the ones provided in the Boyce book.

One of the primary problems in GEOS software development is the GEOS file system. A GEOS file has extra information in its directory entry, and an extra sector of information as well. But a Commodore assembler will produce a standard program file on disk, which will not have the required extra information attached. So, before you can run your newly-assembled GEOS program, you will have to attach an information sector and extra directory information somehow. There are several possible ways to do this.

The first method is to use Berkeley Softworks' Icon Editor (from Desk Pack 1) to attach an icon to your program. This is quite easy, but it has one major disadvantage: when GEOS loads your program, it switches BASIC in and does a standard SYS to start the program. This means that you must include the necessary statements to turn off the BASIC and Kernal ROM to switch GEOS back in. It also means that you cannot develop special GEOS file types, such as desk accessories or files with VLIR overlays. This method is not recommended.

The second method is to use a standard C-64 program to create the required information using direct disk access commands. This is much better than the first method, but it is rather inefficient, and subject to the quirks of the 1541's DOS. Also, special care must be taken to put all string data to disk as true ASCII, instead of Commodore ASCII. This method works, and in fact BSW includes a program to do this in their Programmer's Reference Guide. However, it is not the most elegant method.

The third way to add an information sector to the GEOS program is to use GEOS itself. At the beginning of your assembly language file, you add a pre-constructed information block suitable for the GEOS Kernal routine which is used for writing file information. Then, you attach a loader before the actual program which switches in GEOS, saves your program using GEOS Kernal routines, writes the information sector, and exits to GEOS. If you begin your loader with a BASIC SYS line (i.e. 10 SYS 2061)

Welcome to Imagination!

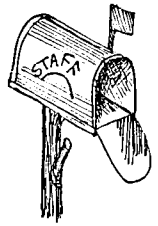
by Kent Sullivan

Welcome fellow adventurers!

This is the first issue of *The Image*, the quarterly newsletter of *Imagination*. We on the staff are working hard to bring you the best in articles, reviews, and features on many facets of Commodore computing. We will try to focus on products and events not covered elsewhere (like Imagery!) to make *The Image* a source of USEFUL information.

The Image is not only designed to be for *Imagination* members, but BY *Imagination* members. For *The Image* and *Imagination* to survive and grow, we need to hear from each of our members! If you have an opinion about a program, a machine, a policy, WHATEVER, please write us! We will need your input to make *The Image* the best that it can be. Please send us any articles you write for the next issue of *The Image*. The club atmosphere we are trying to create is very dependent on the interaction of all *Imagination* members. We're waiting to hear from you!

Note: Because of the time we have spent in getting the first issue of *The Image* ready, we are extending everyone's *Imagination* membership through December, 1988. We thought it was only fair considering your generous support and patience.



MAIL CALL

Dear Dr. Evil:

I was reading a letter published in the *Transactor*. It was pessimistic and interesting reading.

The letter said there were about five thousand software development companies, plus individuals, to compete with.

One large entertainment software publisher said that of the few programs they accept, only one percent does well in sales.

One source said that the average reader response for advertising in a computer magazine is from .1% to .5%.

I have been taking a good look at the Amiga line. I think it is the only computer that can take the programmer and buyer to his/her limit.

Sincerely,

Chuck Slotter

Readers, what do you think? Is it worth the possibility of failure to bother trying to publish your own programs? And what about the Commodore Amiga? How good is it? Send us your thoughts!

Dear Dr. Evil:

I have put together a collection of solutions of some popular adventure games. The solutions are arranged in a menu format so that a player can receive help on just the sections of the adventure that he/she is having problems with. There are solutions to ten games in the first collection I have assembled. This "first aid kit" for adventurers is called *Soluware Volume I*. It is available from me at the address given below. Included in Soluware Vol. I are: *Alpine Encounter*, *Buckaroo Banzai*, *Mindwheel*, *Moonmist*, *The Neverending Story*, *Nine Princes in Amber*, *The Pawn*, *Tass Times in Tonetown*, *Tracer Sanction*, and *A View to a Kill*.

Sincerely,

Carl Kukkonen, III
5467 La Forest Dr.
La Canada, CA 91011
(CA residents: please add CA state sales tax)

Dear Carl,

Thanks for the info! Readers, we have seen Soluware Vol. I *and it is nicely done.*

Summer CES Worthwhile

by Kent Sullivan

The 1987 Summer Consumer Electronics Show was held in Chicago, IL from Saturday, May 30, through Tuesday, June 2, 1987. CES draws every imaginable kind of electronics company-- although this year most of the major computer manufacturers decided to go to the all-computer COMDEX in Atlanta. For some reason COMDEX and CES were held at the same time this year. I hope this practice won't continue because it is very hard to cover two simultaneous shows!

This article will focus on a few of the many vendors at CES, noting some things that the more complete reviews in magazines such as *Compute!'s Gazette* may not have covered.

One of the main points of interest for me at CES was the Berkeley Softworks booth. BSW (as the company is commonly known) is the manufacturer of GEOS (Graphics Environment Operating System) for the C-64 (and soon, the C-128... read on). BSW had a very nice private room and demonstrated several upcoming/now available products: *GeoFile*, *GeoCalc*, *GeoSpell*, *GeoPublish*, and *GEOS 128*. They also announced the upcoming *GeoProgrammer* and *GeoBasic*. I will describe each of these products briefly with a note or two about each.

GeoFile is the only of the products mentioned here that is actually available now. *GeoFile* is a competent small to medium-sized database suitable for client records, invoices, inventories, recipe files, and hobbiest uses. *GeoFile* features custom form designs that may be modified after some data has been

GEOS CORNER

Column
Graphic Environment Operating System

by Rob Tillotson

The Graphics Environment Operating System (GEOS) is a significant departure from the familiar environment of the C-64 Kernal and BASIC 2.0. In addition to the obvious external differences between GEOS and the BASIC/Kernal environment, there are many internal differences as well which make GEOS very attractive to the programmer. In this article, I will attempt to describe those differences and give you some guidelines on how to start writing programs for GEOS.

Currently, programming under GEOS requires that you use machine language. There are no high-level languages available for the GEOS system, although Berkeley Softworks (BSW) plans to produce a version of BASIC for GEOS. Also, you must develop programs using the normal C-64 operating system, since there are no assemblers available which run under GEOS. A system called "GeoProgrammer" from BSW will probably be available this fall, however (Ed. Note: See the CES Report in this issue for some specs on GeoProgrammer).

You can use any standard C-64 or C-128 assembler to develop GEOS programs; but there are some features which, if your assembler has them, will be very helpful. First of all, your assembler should be able to handle large numbers of symbol definitions, since GEOS has many entry points and variables you will want to refer to by name. Second, it should be able to output strings in true ASCII, since GEOS does not use Commodore ASCII (PETSCII). Finally, it should allow you to use labels of at least 10-12 characters.

Once you have a suitable assembler, you will need some programming documentation. Currently, there are only two sources of this sort of information: *The Official GEOS Programmer's Reference Guide* by Berkeley Softworks, and the *GEOS Programmer's Reference Guide* by Alexander Donald Boyce. There are several other books on GEOS with programming information in them, but none are as complete as either of the documents just mentioned. Personally, I recommend that you use both of them. However, if you must choose one, each has certain advantages and disadvantages you should consider.

The BSW book is the official word on GEOS internals. For the most part, it is the only source you can really trust for accurate information. However, the presentation of that information is extremely poor. The book is full of typographical and formatting errors, and some important tables are incorrect. Also, some subjects, such as the structure of font files and how to program desk accessories, are not covered at all. Even with all of those problems, though, the BSW book is a valuable source of information.

The other guide, by Alexander Boyce, comes as text files on disk, and prints out to 89 pages. It is shareware, which means that the author requests a donation from everyone who uses it. The author created it by examining GEOS in great detail. This book

To contact Kent by electronic mail:

UUCP: ihnp4!pur-ee!corvair
Q-Link: Mail to CorvairKid

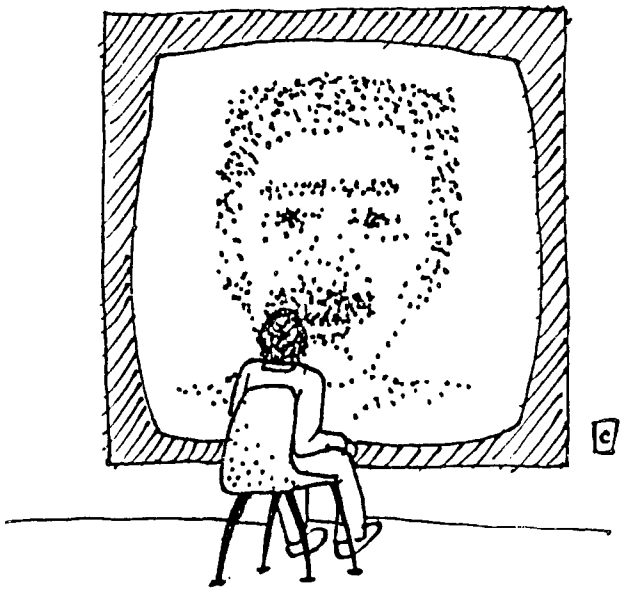
Rob Tillotson:

Rob met up with the Lab last year at Purdue and hasn't been able to get away yet! Rob likes adventure games but doesn't work with them very often.
Rob will be writing the regular column for *The Image* on GEOS. GEOS is one of Rob's main programming interests. Rob is also our contact on the grapevine for news on Commodore products as he monitors many major bulletin boards across the U.S.

Rob will be a sophomore at Purdue this fall where he is majoring in Computer Science. His hobbies include reading science fiction and playing role-playing games, and music (favorite artist: Rush).

To contact Rob by electronic mail:

UUCP: ihnp4!killer!sentinel
FIDO: Net 1, Node 203



entered, with forms that can be up to a full page long. *Geofile* has an automatic sort, although only by one field at a time, a keyword search feature, and a selective print feature that can allow more than one form per printed page. *Geofile* can also accept graphics as part of the form as well as data from *GeoMerge*. Besides lists, *GeoFile* can also be set up to print labels or tags. At $49.95, *GeoFile* appears to be a reasonably-princed database. Note: From some discussions I saw recently on Quantum Link, *GeoFile* apparently has a few bugs that BSW is working to iron out. One might think about waiting a bit to see if an updated *GeoFile* is released before buying...

GeoCalc is a graphics-oriented spreadsheet that will be released very soon. You may have seen ads for it already in some of the larger computer magazines. *GeoCalc* will be useful for chores like cost estimating, mortgage analysis, and tax planning. *GeoCalc* will support 112 rows by 256 columns with arithmetic to 12 decimal places and several niceties including trigonometric, logarithmic, and exponential functions. Other features include sum and average calculation and the ability to see two different portions of the spreadsheet simultaneously. *GeoCalc* will retail for $49.95.

GeoSpell, the stand-alone spelling checker for *GeoWrite*, is still in development. The beta version BSW was demonstrating looked very impressive and has some features I haven't seen in other spell checkers for the C-64. When *GeoSpell* finds a word it doesn't know, it displays near where it would be in the *GeoSpell* dictionary. Adding, correcting, or choosing a word from the dictionary are all easy to accomplish. *GeoSpell* will also present pre- and suf- fixes that you may click on with the mouse to add to a root word that you are entering to the dictionary. In other words, if you want to enter the word "fubar", you can also easily add "fubared" and other combinations. (Is that a word? I doubt it!)

GeoPublish brings desktop publishing to the C-64. With *GeoPublish* you can create your own newsletter or flyer easily. *GeoPublish* will feature multi columns/pages, auto reformatting of text/graphics, support of up to 48 point fonts, preview and zoom options, *Post Script* laser printer output, and master pages. A master page will define the documenst and can be saved in a library for future use. *GeoPublish* is slated for a fall release at $69.95. Not only was it not finished, but a rep crashed it for me to prove it! I liked his honesty-- very few companies would dare to do that for the critical buyers/press people always present at CES.

GeoProgrammer will, if it lives up to its claims, be truly incredible. This comprehensive assembler package for the GEOS environment will consist of *GeoAssembler*, *GeoLinker*, and *GeoDebugger*. *GeoAssembler* will be a full macro assembler capable of over 1,000 labels per source file. It will assembler documents created with *GeoWrite* and can contain graphics! This means no more manually creating ".byte" statements for graphics for you programmers out there-- *GeoAssembler* will convert the graphics automatically! External labels will also be supported. *GeoLinker* will be capable of creating both SEQ and VLIR GEOS files, and will be able to assemble conditionally with memory segment allocation. *GeoDebugger* will turn the 17XX RAM expander into a full monitor for debugging large GEOS programs. Its command set will include Step, Subroutine Step, Loop, Next, and Execute. All text will be shown in a special overlaid window

that will not destroy any underlying graphics. The price is scheduled to be $69.95 with fall availability.

GeoBasic is still far from completion but promises to be interesting and differerent dialect of BASIC. *GeoBasic* will ad many useful commands that will be helpful in the GEOS environment. One problem I can see immediately is a lack of programming space-- there is only 32K or so free space with the GEOS Kernal in memory. I didn't get enough details to say much, but I'm sure we'll all hear more as development progresses.

GEOS 128 really transforms the C-128. I was impressed at the speed and clarity of the 80-column graphics (yes, *GEOS-128* requires an 80-column RGB monitor). It looks very much the same as GEOS for the C-64, but has a hardware-produced 80-columns and several touches to make it superior. Look for an in-depth review in our next issue.

Xetec was at CES with several new products including the *Lt. Kernal* hard drive, *Super Graphix Gold* printer interface, and the *Printer Enhancer*. The *Lt. Kernal* 20 megabyte hard drive is available for both the C-64 and C-128. A 40 megabyte version is also available. Up to seven drives may be daisy chained, and a multiplexer is available to allow up to four computers to share one drive. The *Lt. Kernal* plugs into the cartridge port and requires some computer modification for burst mode compatibility for the C-128. The *Lt. Kernal* is very fast-- it's capable of transferring 16 kilobytes of data per second, which is on the order of 50-60 times 1541 speed! The *Lt. Kernal* has its own expanded DOS with many features. The price is around $1000.00.

The *Super Graphix Gold* will be an updated interface with a 32K buffer with burst mode support for the C-128. It will have a variety of screens dumps and command options as well. The price is slated to be $119.95.

The *Printer Enhancer* is a controller for two different printers (a switch, in other words) with some bells and whistles: eight internal fonts, 64K buffer, and multiple copy capability. The price is expected to be $249.95.

Many other companies showed new products, including Inkwell Systems, manufacturers of the *Flexidraw* lightpen. Inkwell is now selling version 5.5 of their *Flexidraw* software, which now supports the light pen, *Koala Pad*, joystick, and mouse. They also have the *Flexifont* font pack, *Graphics Galleria* clip art packs, and *Graphics Integrator I & II* conversion utilities available.

Microprose had a novel way of demonstrating their popular *Gunship* game: they had a working simulator of an Apache Gunship attached to monitors so bystanders could see what the "pilot" in the simulator was doing. As the pilot played Gunship, we could see what he was seeing on his screen. It was very well done.

As I mentioned above, many of the large manufacturers were missing at CES, including Commodore, but their were quite a few other companies that made it worth the trip. Please drop us a line if you have a question or comment about CES or the products/companies mentioned in this article.

Ray will be a junior at Purdue next fall where he is majoring in Physics. He is also on the staff of the Purdue Computing Center, and is knowledgable about many operating systems and languages. Ray's hobbies include reading, watching Dr. Who, swimming, and music (favorite artist: Peter Gabriel).

To contact Ray by electronic mail:

Arpanet: ray@j.cc.purdue.edu
UUCP: ihnp4!pur-ee!j.cc.purdue.edu!aij
Bitnet: moody@purccum.BITNET

Dan Moore:

Dan became a part of the Lab through Kent. Dan found *Imagery!* to be a good way to express his creativity for writing/designing adventures.

Both Dan and Kent take care of the day-to-day business of the Lab. Dan also writes for and is co-editor of *The Image*. Several of Dan's ideas will probably see fruition in *Imagery!* adventures.

Dan will be entering the U.S. Army this fall. His hobbies include writing, electronics, reading, and music (favorite group: The Blues Brothers).

Roy Riggs:

Roy is one of the founders of Dr. Evil Labs. The name Dr. Evil was his nickname in high school (but that's another story). Roy is proud of the *Imagery!* system and says he probably wouldn't have bothered creating it if he had known in advance how much work it was to involve!

Roy prefers to stay clear of the business end of the Lab and instead concentrate on programming. Several of Roy's other programs will be released by the Lab in the near future.

Roy will begin his fourth semester at Purdue next fall where he is majoring in Computer Science. His hobbies include reading, watching Dr. Who, and music (favorite artist: Peter Gabriel).

Kent Sullivan:

Kent, along with Roy, conceived the basic ideas and plans that eventually produced Dr. Evil Labs and Imagery! in December, 1985. Kent was familiar with the Eamon system for the Apple and showed it to Roy... and they were both hooked on doing a similar system for the C-64/128.

Kent's responsibilities for the Lab include managing the business end of the company as well as writing doucmentation for all Dr. Evil Labs' programs and producing *The Image*.

Kent is currently between his sophomore and junior years at Purdue where he is majoring in Professional (Technical) Writing with a focus on the computer industry. His hobbies include reading, bicycling, music (favorite group: Led Zeppelin) and collecting/restoring Corvair automobiles.

VT-52 emulation and has both 40 and 80-column displays. Support has also been included for the Commodore 128 80-column screen (while operating in 64 mode!) as well as the Batteries Included BI-80 80-column adapter.

Kermit v2.0 is an excellent terminal emulator and is a valuable telecommunications tool even if you have no use for the Kermit protocol. And since it is public domain, it's free!

Kermit is available on many networks and from user groups across the U.S. Dr. Evil Labs also offers *Kermit* on disk for $5.00 including disk and postage (this is the price established a few years ago by Columbia U. for private distribution).
If you have any questions about *Kermit*, please contact us here at the Lab.

Who's Who at The Lab

Dr. Evil Labs is based in the small Indiana town of St. Paul. St. Paul has a population of 850 (counting all of the dogs!) and is located about 35 miles southeast of Indianapolis on I-74. Why is Dr. Evil Labs based there? Well, both Kent and Dan live in St. Paul (Kent when not in school), so it seemed the natural choice.

We thought you-all out there might like to hear a little bit about each of us so we don't remain just "names."

Vince Martin:

Vince expressed an interest in helping with any artwork for the Lab last year and we quickly excepted his offer! Vince was amazed that people he knew were actually writing a program that others would buy (snicker, snicker).

Vince has conceived and drawn just about every piece of artwork that the Lab has needed from the *Imagery!* title screen to the symbol on our letterhead. Vince has also expressed an interest in creating his own *Imagery!* adventure.

Vince will be a junior next fall at Purdue University where he is majoring in Environmental Design. His hobbies include drawing and playing basketball. Vince's favorite music group is U2.

Ray Moody:

Ray became involved with Dr. Evil Labs shortly after its inception and has enjoyed every minute of the time he has spent. Ray wrote the Save/restore game option for *Imagery!*

Ray prefers to leave the business dealings to Kent and Dan and spend his time programming. Ray has written his own assembler and is currently working on improvements to the C-64 version of the *Kermit* public domain telecommunications program.

Brief Overview Given

by Kent Sullivan

The *Imagery! Adventure Designer* is the centerpiece of the *Imagery!* system- but, unfortunately, none of you prospective adventure designers have it yet! Roy and all of us here at Dr. Evil Labs would like to apologize for it not being ready on time. We had run into some unexpected difficulties. The problems are mostly solved now, so the designer is nearing completion. We will be notifying all *Imagination* members by mail when the designer is finished.

I will briefly outline some of the features of the designer for those of you anxiously awaiting its arrival! For those of you who have not yet ordered the designer, it is $10.00 (Indiana residents please add 5% sales tax). The designer is shareware as is all of the *Imagery!* system.

The designer is completely menu-driven and can be controlled with a joystick and keyboard. We have put a lot of time into making it easy for the first-time as well as experienced adventure creator to construct his/her own *Imagery!* adventure. You may concentrate on one area of designing at a time (such as creatingall the monsters) or you can switch between any of the areas quickly. The designer allows for free form or patterned designing-- whatever your mood may be. Editing an adventure is also easy-- nothing need ever be permanent until you as "master" are satisfied.

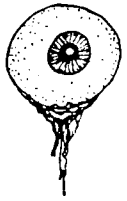
Within the menu-driven approach we have implemented full on-screen editing. For example, after selecting the option to edit a monster, you would then quickly cycle through a list of available monsters and then be presented with a full screen of information about that monster. All items about the monster, including stats, text description, items carried, and location may be changed by simply moving the cursor to the option, pressing <RETURN> or <FIRE> and then entering a new value. Editing could hardly be simpler!

Accompanying the designer is the *Adventure Designer's Manual* which explains theconcept of designing and *Imagery!* adventure as well as how to use the designer. Also included is a complete list of all the variables used in an *Imagery!* adventure and what they represent, and a line-by-line disassembly of the base adventure program. These two documents will be very handy to understanding and/or modifying *Imagery!*

Well, that's it for this time. This column will feature a regular discussion of the designer and tips for creating your own adventures starting with the next issue of *The Image*. See you then!

THE EVIL EYE

The rating scale used in The Evil Eye varies from between 1 and 5 stars (*), with the scale being:



5 = Superior
4 = Very Good
3 = Good
2 = Average
1 = Poor
0 = Trash

Every effort is made to provide accurate information for each product reviewed. Any errors or omissions will gladly be corrected.

Review #1: *Enhanced Sidplayer*

by Dan Moore

Lyricists, musicians, and crooners lend an ear to (or at least take note of) *Compute!'s Music System for the Commodore 128 and 64*. This multi-purpose software, also known as *Enhanced Sidplayer*, allows the user to not only enter, edit, and play music on their C-128 or 64, but also add pictures and words. The *Enhanced Sidplayer* package is a book & disk combo including several sample pieces of music.

When I first began reading the *Enhanced Sidplayer* book, I read a couple of chapters on music theory, sound principles, etc., and I kept thinking that a lot of this information seemed like a compilation of 8th grade science and music classes. This work goes well beyond the basics, however. *Enhanced Sidplayer's* real accomplishment is that it creates a workable medium for composing music for those of us who are more interested in creating than programming.

The *Enhanced Sidplayer* system is composed of an editor and a player. The editor's most obvious advantage over most other music programs is that the composer uses simple music notation when entering a song. Note entry is accomplished with either a joystick or the keyboard. Only a minimal knowledge of how the SID chip works is required to begin composing. With defaults given for almost every sound parameter, you can enter a simple tune to get your "feet wet" right away. But, the power is there for those those who want it.

Enhanced Sidplayer has many powerful features not found in any other music system. Most of these more advanced features are accessed on the Command screen. With *Enhanced Sidplayer* you have very powerful synchronization and ring modulation commands, some of which use special software-generated waveforms. Although complex in nature, all of the features of the editor are easily understood due to the clear explanations and many examples in the text.

When you wish to listen to your composition, you can just sit back and enjoy it via the "Play" command or make use of the Display screen. The Display screen displays every parameter for all three voices. As the song is playing, you can watch the sound parameters change-- very helpful for locating a mistake that you can hear but not find in your music.

origin can also be found. *Disk Editor* also includes a super-quick byte search function. You may enter a pattern of bytes for which to search, and *Disk Editor* will scan the entire disk (or just the tracks/sectors you wish) and report any occurences of the pattern. *Disk Editor* can scan a complete disk in under 2 minutes! *Disk Editor* contains many other features too numerous to list here.

Flexi-Copy is a file copier that is truly flexible- it will copy from almost any device to almost any device. You can copy files from a 1541 or compatible disk drive to an IEEE drive such as a 4040 or SFD-1001. You can even copy a file to the screen! And, *Flexi-Copy* will copy relative files, a filetype that many other files copiers can't handle.

Utility Pack II should be available for $10.00 immediately following the release of *Utility Pack I*.

Kermit Announced

by Kent Sullivan

Dr. Evil Labs is now distributing the popular public-domain *Kermit* telecommunications program. *Kermit* is packed with many features not found in any other terminal program for the C-64, and best of all, it costs $5.00 at most.

Kermit has a long history stretching back to its roots in the Apple version of *Kermit* which David Dermott ported to the C-64 in March, 1984. For those of you not familiar with what *Kermit* is, here is a short history. Kermit is an acronym for KL-10 Error-free Reciprocal Micro-Interface Transfer. Kermit, in other words, is a protocol or set of rules for transferring files from one computer to another. Kermit was developed at Columbia University in New York several years ago. The headquarters for Kermit distribution and news is also located there.

Kermit was designed with the maximum amount of flexibility in mind so that almost ANY computer can use the Kermit protocol. Unlike, for example, the Punter protocol (designed by Steve Punter) for the C-64 specifically, *Kermit* is currently available for over 100 computers.

What can be confusing is that not only is the name of the protocol Kermit, but many of the terminal programs that feature the Kermit protocol (most usually the programs in the public domain). The official name for the current C-64 version of *Kermit* is "Kermit-65, Commodore 64 version 2.0" which signifies that this is *Kermit* written for the 65XX (6502/6510) microprocessors found in computers such as the Apple and the C-64.

Ray Moody and I became interested in *Kermit* while at Purdue last year. The Kermit protocol is used by many universities around the world. The version of *Kermit* we obtained was v1.5 (and later, v1.7). After using *Kermit* extensively, Ray and I decided to improve it in several ways.

We have taken on the responsibility of improving *Kermit* and maintaining it. *Kermit v2.0* now features FULL DEC VT-100 and

LAB NOTES

New Products "Foretold"

Dr. Evil Labs will be releasing, in addition to the long-awaited *Imagery! Adventure Designer*, several other useful programs. Below is a short description of some of our upcoming products, including the projected price and availibility date. And as always, all of these products are shareware!

Utility Pack I

Utility Pack I consists of three programming aids sorely needed by most BASIC programmers: *Directory!*, *Renumber*, and *Super Trace*. *Directory!* is a transparent disk directory utility that hides beneath the BASIC ROM so it doesn't use any valuable programming space. *Directory!* is activated with a simple keystroke, and instantly displays a window on the screen with the disk directory. You may stop the list of filenames temporarily for closer inspection, or quit before the entire directory has been displayed. *Directory!* restores the screen to its previous display when you exit, making it a truly transparent utility.

Renumber does just what it says: it renumbers any BASIC program in memory to open up room for additional line numbers or to reformat it to make it more readable. You may choose the starting line number, the increment, and the first and last line numbers you wish to change. *Renumber* is fast and efficient, a utility you won't want to be without!

Super Trace is a valuable tool for finding logic and/or syntax errors in your BASIC programs. With *Super Trace* you may step through your BASIC program statement by statement, allowing you to see exactly what commands are being executed when. *Super Trace* displays the line being executed in a scrolling window at the bottom of the screen. You can also turn off the display to see anything that is being printed to that line on the screen. With *Super Trace*, you can see both the internals and the displays your BASIC program creates!

Utility Pack I should be available in September or October for $10.00.

Utility Pack II

Utility Pack II is composed of two powerful disk-related programs: *Disk Editor* and *Flexi-Copy*. *Disk Editor* is a comprehensive package with features no other disk editor currently has. With *Disk Editor* you may jump both forward and backward on track/sector links. You can also trace a file and obtain a graphic display of its organization. Any sector's

Word files may be created with the editor to produce "Singalong" songs. With Singalong songs, words are presented in time with the music. Word files can be created using *Easy Script*, *Speedscript* or a compatible word processor.

The player is a stand-alone module that allows you many options for listening to music. You can listen to all of the works on a disk or just the ones you wish. You can watch the keyboard and see which notes are being played, and with Singalong songs, also see the words. Another nice feature of the player, although not directly supported by the editor, is the ability to add Koala or Doodle picture files to make a "SidPic." The player will display the graphic screen you have chosen while the music is playing.

In Chapter 15 of the *Enhanced Sidplayer* book there is a section "More Than Three Voices" which mentions the serious limitation of the SID chip only supporting three voices. This situation was addressed by suggesting that notes be cut when transcribing a work of music to Sidplayer. Naturally, a preferred solution would be to find a way of expanding the SID chip's capabilities. As mentioned in the "SidFest '87" article elsewhere in this issue, a "SidStereo" player is available as free public-domain software. Unfortunately, to be able to fully appreciate the stereo Sidplayer one would need a second SID chip. Another solution exists, however. *Enhanced Sidplayer* supports the synchronized playing of more than one computer-- making possible such feats as John Mackey's 18-voice extravaganza at SidFest '87. One song on the *Enhanced Sidplayer* disk, "Dawn River," will run on two computers.

Enhanced Sidplayer looks even more impressive when compared to its forerunner, *Sidplayer*, found in Compute!'s *All About the Commodore 64 Volume Two*. *Sidplayer* introduced thousands of people to making music on their Commodore 64's, while *Enhanced Sidplayer* has truly made music creation on the Commodore 64/128 come of age.

Product name: *Compute!'s Music System for the Commodore 128 & 64* (The Enhanced Sidplayer) *****
Author: Craig Chamberlain
Medium: Book, 274 pps. with Disk (2 sides, 1 side C-64 & 1 side C-128)
System: C-64 or C-128 and disk drive (printer optional)
Price: $24.95
Copy protection: None
Publisher: Compute! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 (919) 275-9809
ISBN: 0-87455-074-2

Review #2: *Inside Commodore DOS*

by Kent Sullivan

The Commodore 64 has been in production long enough that many useful programs and/or books that were once popular to now be virtually unknown or outdated. One book that has stood the test of time as being an essential part of any C-64 programmer's library is *Inside Commodore DOS*.

ICD (as I shall refer to it) explains thoroughly the intricacies of the Commodore 1541 disk drive, detailing all of its features and including a wealth of sample programs written in machine language and BASIC. Each of the examples is explained, and even the source code is provided for the ML routines. Also included are many useful utilities.

ICD begins with an introduction and gradually explains the essential parts of the 1541, with each topic being a little more advanced than the next. Among the things covered in detail is the format of a 1541 disk (the layout of tracks and sectors); diskette organization, including the Block Availibilty Map (BAM) and directory entries; direct access programming (Memory-Read, Memory-Write, etc.); copy protection; and concluding with a technical description of how the 1541 operates on a component level. And this is not even half the book-- The real show-stealer is the detailed 1541 ROM disassembly that occupies over 200 pages.

I have referred to *ICD* again and again as a reference for all of the command formats I can never seem to remember. The ROM disassembly has also proved invaluable when writing programs that execute inside the 1541.

I have only one minor complaint: the section on copy protection is woefully incomplete. The authors cover the early protection schemes very well but no updates have been included for understanding the newer, more advanced protection schemes. Such an "update" might well be the subject of a separate book, however, as the copy protection war rages on. Along the same vein, I would like to see a similiar book for both the 1571 and 1581 drives-- they are even harder to understand than the 1541!

ICD is an indispensable tool for users who wish to understand the powerful yet complicated Commodore 1541 disk drive. I highly recommend it.

Product name: *Inside Commodore DOS* ****
Authors: Dr. Richard Immers & Dr. Gerald Neufeld
Medium: Book, 505 pps. (and optional disk with all the programming examples from the book)
System: N/A
Price: Book- $19.95; Disk- $24.95
Copy protection: None
Publisher: Datamost, 20660 Nordhoff St., Chatsworth, CA 91311-6152 (818) 709-1202
ISBN: 0-8359-3091-2

Review #3: *Labyrinth*

by Dan Moore

This 1986 product of Activision Entertainment Software copyrighted by Henson Associates is a LucasFilms game (a trademark of LucasFilms Ltd.). With all of these big names from television and film one would expect an entertaining piece of software, and that it is!

Having seen the movie *Labyrinth* I had expected a text adventure of long, drawn-out descriptions. However, thanks to surprisingly good graphics, anyone can "see" that Ludo is the "big" "hairy" "brown" "creature" of little vocabulary. Also commendable for this creative endeavor is the "music and Commodore version sound" work of David M. Martin, Jr. Until this game I had never actually heard an elephant being "adumberated."

Seeing how the target audience of the movie was a bit young to be proficient computer users, I thought the idea of a cursor-controlled option list fared well. This idea also alleviates the embarassment the computer "feels" for using such phrases as "I don't know that word" or "Huh? I don't understand" since only valid commands are available.

This game certainly lives up to its claim of playability in not only keeping track of your name, sex, and favorite color but reacting to these variables as well. Also, you are held accountable for all the things you do and don't do.

On the negative side of *Labyrinth* there is a valid argument that text adventures in general better create a more realistic fantasy by directly accessing the mind's eye with words of vivid inspiration (Infocom especially promotes this argument). Of course the flip side of this debate is that graphics visually create and interact with the implied illusion. This theory of effective "graphics interaction" is supported by the soon-to-be-released *Habitat* software for Quantum Link. *Habitat* looks much like *Labyrinth* as it is also being developed by LucasFilms.

One other minor concern is the format of the documentation. The manual is very clearly written and informative, but it is written with instructions for both the Apple and C-64 versions. The player must read through sections not pertaining to his/her system.

Product name: *Labyrinth* ****
Medium: Disk
System: C-64 or C-128 (in 64 mode) and 1541 (or compatible) disk drive
Price: $35.95
Copy-protection: Heavy, third party parameters are available to allow backups
Manufacturer: Activision Entertainment Software, 2350 Bayshore Pkwy., Mountain View, CA 94043

(Questions about *Labyrinth* and other Activision products may be directed to: Consumer Relations, Activision, Inc., P.O. 7287, Mountain View, CA 94039 or call <in CA> (415) 940-6044/5 <outside CA> (800) 227-9759)