The Magazine for true Computer Fans

Euro 5.33 USD 6.17

GO64!

Go64! commodore WORLD

COLUMN

WORLD WATCH: A GRAPHICS CARD FOR THE COMMODORE 64?

TUTORIALS

HOW TO GODOT: CONVERSION TO INTERLACED FLI

GEOS

EXCLUSIVE INTERVIEWS WITH THE HIGHLY RATED CODERS MAURICE RANDALL AND WOLFGANG GRIMM

GAMES 2000

FORTHCOMING GAMES FOR THE YEAR 2000

TITLE

INTO THE FUTURE WITH THE C64

AB IN-DEPTH REPORT ABOUT CLiPS, THE OPERATING SYSTEM OF THE FUTURE

TUTORIALS: GRAPHICS, ASSEMBLY LANGUAGE BASIC EFFECTS PLUS: NEW JPX VERSION ON THIS MONTH'S COVER DISK!

CONTENTS

- January 2000 -

10
GEOS

Will the C64 and GEOS continue well into the 21st century? Read our interview with GEOS MP3 developer Wolfgang Grimm and with Wheels and The Wave coder Maurice Randall, beginning on page 10.
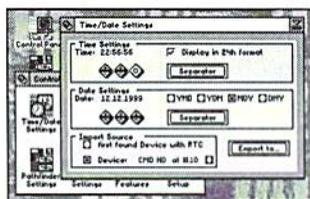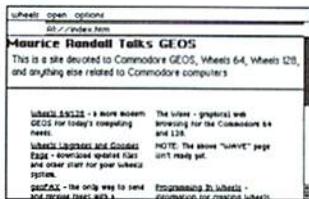
6
Into the future with CLiPS

Here it finally is. The long awaited in-depth report about the latest developments of the revolutionary operating system CLiPS, beginning on page 6. Things like 'preemptive multitasking' and 'vector fonts' will make C64 enthusiasts go wild!

14
GoDot Work-shop

This issue's main focus in our workshop will be the basics of IFLI and conversion into this special format. Arndt Dettke introduces many handy Godot modules for that purpose. Prepare to be impressed, starting on page 14.
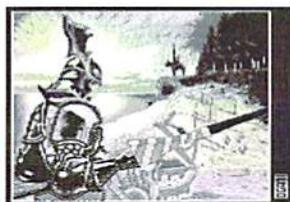
27
Graphics card for the C64?

Many rumors about a potential graphics card for the C64 have spread around in recent years. Beginning on page 27 we will look into the theorectial aspects of such a project.

A

Assembler Tutorial files
Graphic Tutorial files
How To Godot Files
SuperCPU Monitor
Scene Stuff

B

GEOS - Y2k - Patches
JPEG Viewer with
sample graphics

Hello, everybody!

Maybe I should start by introducing myself. My name is Ranjith Ekanayake Mudiyanselage, and as a translator for the GO64!/Commodore World magazine, I have the honor of writing this issue's editorial. Some may call it the 'millenium issue', others may deny this, but does this really matter? Whether a new millennium has begun or not, we should all agree that a new year has begun, which is reason enough for us to look into the future after a little glance back.

We should be content with last year's C= developments. The English version of GO64! was made available to readers worldwide. The C64 scene also seemed to be quite healthy. The major computer disaster (a.k.a. the Y2K bug) that loomed above our heads because of a common shortsightedness among commercial programmers failed to come crashing down upon us after all.

In order to convey these positive aspects, we have dedicated this issue to our computer's future. We are making an effort to show you what new things are currently being developed for our machine. Although the major commercial basis for the C64 market is long gone, there are still industrious programmers and designers who keep creating and making accessible new programs and hardware for us time after time. In this issue we report about CLiPS, the new SCPU operating system; offer an interview with Wolfgang Grimm, the relentless GEOS programmer from Germany; and show you which new games will make this year a formidable year for the C64. And a formidable year it will truly be, because we will conquer the Internet with a graphical web browser.

The big commercial firms abandoned us long ago, and the general public deemed us dead. This is why we need the strength and support of all remaining users so that we may keep growing and flourishing, being an albeit small counterbalance against the broad masses of ignorant PC users. It may be true that not all of us are able to program, but this isn't really necessary anyway. We need graphicians, musicians, designers, organizers, translators, and so on.

Actually, GO64! wants to enlarge its staff of translators, which is why we are looking for dedicated persons to help transfer the magazine into English. So that articles can be promptly transmitted to and from the editors, a prospective translator must possess a personal e-mail account. From my own experience, I can say that there is quite a good atmosphere within the GO64! team and that you can really get along very well with all the editors.

If you haven't yet discovered your talents, you can still help by remaining devoted to our machines, in order to keep up the developers' morale. As long as there are users and gamers out there looking forward to new releases, there will be such honourable programmers like Maurice Randall, who stay with us out of conviction, resisting the mass market's undertow.

My greatest wish is that this year we keep on vigorously fighting and don't allow ourselves to be overcome. The future is ours.

Yours,
Ranjith

News Flash

Web Addressbook for Germany - 2000

"Das Web-Adressbuch für Deutschland 2000" (The Web Addressbook for Germany - 2000), available this year in German bookstores, is an ideal publication for web surfers. The book can also be ordered directly on the web at **http://www.web-adressbuch.de/** The GO64! homepage, among others, is mentioned in this book. *(st)*

Elite for everyone

There's now a special website for fans of the classic C64 action game "Elite", where you can meet others interested in this game. In addition, the source code will be available, as will be ports and sequels for other systems. All this can be located at:

http://www.iancgbell.clara.net/clara.net/i/a/n/iancgbell/webspace/elite/
(st)

Advance Version of LNG Available on the Internet

Daniel Dallmann is now offering an advance version of LNG with SuperCPU support on his homepage. The LNG (Little Unix Next Generation) project is an effort to port UNIX to the C64. The preview version can be downloaded from:

http://www.heilbronn.netsurf.de/~dallmann/linux./lng.html
(st)

Matting's PD Mailing Sent Out

The Breadbox Corner site for C64's on the internet has been taken down and the PD mailing has been discontinued. As a consolation, Matting is working on complete PD diskettes on CD, in emailboxes as well as being offered on the web. *(st)*

"The Party 8" - Results

Here are the C64 results from the Danish Computer Party held Dec. 27 - 30, 1999:
Position, Contribution, Title, Coder, Points

Graphics
1. (5) "Another evil man" by Choke (328)
2. (6) "Bells jingle" by Zinc (191)
3. (4) "Smoke it !" by Cori of DAL (121)
4. (3) "Ode to BV" by JailBird of Tempest/Padua (109)
5. (7) "Renegade" by Sentinentel (107)
6. (1) "mad chicken" by Drake of Anubis/Tempest (79)
7. (2) "Reincarnated" by Exile of Anubis (78)
8. (8) "Pertrolized" by Crusader (64)
9. (9) "Nintendoh" by Eyesee (59)

Music
1. (3) "Love" by Vip of Role/WoW/Padua (463)
2. (1) "Morbital" by Rayden of Breeze/Cyberpunx (203)
3. (2) "bloch-zet" by Welli of Anubis/Axelerate (185)
4. (4) "MarioLand Cover" by JamBen and Toaster of Seminar (172)
5. (5) "Roc-Musik" by Heinmueck of TUM (158)
6. (6) "Moonfire" by Brite-Lite and Chaj of Dekadence

Demos
1. (4) "Y2K Duck & Cover" by Perff and Ghostride of No Name (673)
2. (5) "Rea-M" by Scala Production (Smash Designs) (487)
3. (2) "Rastertime" by Cyberbrain of NoName (295)
4. (3) "BRAK 3" by MZ1453 and Copy-Dan of GFB64 (129)
5. (1) "noname-v" by anonymous-bastard of Flare (80)
(st)

The RE-System

As you know, there aren't too many useful programs supporting a REU, but some time ago I stumbled over (meaning: pushed my nose into) a tool which can really make your work easier.

by Wanja Gayk

This tool allows those who don't dare burn an eprom with the regular tools to use their REU as a kind of Ramdisk system that serves this task almost as well. Well, Commodore themselves supplied a Ramdisk system on the disk that comes with the REU, but the RE-system by Johannes Schulze-Oechtering can't be compared to that, because it works in a totally different manner.

How does the RE-system work?

First of all, the RE-System loads all tools from the REU tooldisk into the REU, using a fast loader (Ultraload or Heureka Sprint, depending on your system; Heureka Sprint is about as fast as Warp*25/Epyx). After that, you use the keyboard to choose the program you want to run. When you've reset the machine, all you have to do is load a tiny program (1 disk block in size) which waits for a keystroke to load the associated program directly from the REU and run it.

This way, all the programs you need now and then (like packers, copiers, monitor, assembler, etc.) can be accessed extremely fast. The number of programs you can keep in the REU is limited by the REU's memory size and the capacity of a disk side. The system I've got at the moment is still limited to 32 programs (2 megabyte REU equals 32 banks) for reasons of safety.

In order to secure the programs in the REU, you can specify the lowest REU bank to be used, which makes sense since copiers usually occupy the lowest two banks (Copy1581-1MB even uses the first 16 banks). Naturally, storing programs in these areas would be fatal, so you may leave those banks empty. However, Novaterm still poses

a problem, because it always erases all of the REU memory.

The downside...

Sure, it sounds so good you wonder where the problem might be - the problem hidden in anything that just looks too good. Well, the largest problem is simply that you can't use an REU and a Final Cartridge, Action Replay, Nordic Power, Super Snapshot, or whatever cartridge in the same port. Although you won't miss the cartridge's fast loader as long as you use the programs in the REU, you'll instantly miss the cartridge's built-in monitor. REU-compatible cards, like Swiftlink or a SuperCPU, will cause no problems. The other flaw is the installation, which isn't exactly easy. In fact, it's quite a trial until you've configured the system; however, you just have to do this once.

For the C128

Of course, there's a special version of the RE-system for the C128. When used with 1571/81 and FD drives, this version makes use of the fast serial routines of the 128 (burst mode) that make C64 speeders, like Ultraload, look pale. Unfortunately, the 128 version can only handle programs below a size limit of 45200 Bytes (about 178 disk blocks). Once loaded into the REU, the programs can be loaded from there, just like with the C64 version by using a small program called "RE128". It works in the C64 as well as the C128-mode.

Hints & tips

There are some programs that refuse to run with the RE-System, since being started from the REU naturally isn't 100% the same as being started after loading from disk. However, you can work around the problem by freezing the program with a cartridge

(Action Replay/NordicPower) and installing the freeze in the REU instead of the original program.

The future

According to Johannes, the RE-system is already designed to run with a 16 meg REU, which might still cause some trouble due to the limit of 144 files per 1541/71 disk. In reality, however, you'll hardly reach this limit, as you can easily fill a disk with less than 144 files. To solve this problem, several loaders will be used, each of which will only load a part of the complete set of files. RamLink users will certainly be better off, but it's still the most elegant solution for people with just an REU. A SuperCPU adaptation is planned as well, and maybe the future will also bring an automated installation.

Where can I get it?

The RE-System was published by "64'er" magazine, which means we can't include it on our coverdisk for copyright reasons. It appeared in several issues of the "64'er", namely, 5/97, 12/97, 8/98, and 9/98. These issues are available from:

WEKA Computerzeitschriften-Verlag GmbH
64'er-Redaktion
Gruberstr. 46a
D-85586 Poing
Germany

CLiPS - _the operating system of the future_

Some of you may remember our last review of CLiPS in issue 7/ 1999 - in the meantime, there were some new developments we would like to report on.

by Wanja Gayk

Well, what happened? First, I'd like to fire a little bomb: CLiPS, as presented on the fair in Stuttgart in the end of October, is dead. CLiPS developer Chester Kollschen scrapped the whole thing, and rewrote the kernel after making fundamental changes to the concept.

Long live CLiPS! No more a monolithic construction like Windows, it's now based on a micro-kernel, much like QNX (http:// www.qnx.com). A tiny system core? The word "micro" does not refer to its physical size, but to its set of functions. This isn't to say the new CLiPS was less functional - on the contrary. A micro-kernel only takes care of the most elementary functions of the system, like, assigning memory and processing time to the applications, and controlling the message exchange between them.

All the other components of the operating system, namely the file system, the process manager, or the GUI (Graphical User Interface) are nothing but add-ons to the kernel, applications which are replaceable, at least in theory. Talking about assigning processing time, let's mention a feature of CLiPS that's good for a little sensation: a capability considered very difficult or almost impossible to implement on the C64, has become real: CLiPS makes use of preemptive multitasking!

True preemptive multitasking with CLiPS

Several processes at the same time

Multitasking? Several tasks at the same time? But, can a CPU execute several processes at once? No, it only seems so! In order to come as close as possible to the ideal of simultaneousness, a multitasking system successively processes small bits of the programs currently running. It's like four smiths sharing one hammer, such that each of them may do one strike on the anvil in each turn, and then has to let the other three at it. Someone who doesn't see them working will think they'd all be working simultaneously on their pieces. Actually, they work one after another. Of course, this example seems absurd in the real world, but inside computers, it's nothing unusual. The only question that might come up is, who's to say who gets the hammer, when and how long - the same question arises in the world of computers, leading to two different kinds of multitasking: cooperative and preemptive multitasking.

The difference is easy to explain: In a cooperative multitasking system, as used by the "RiscOS" of the Acorn computers, the operating system starts a program, it works for some time and then returns control to the OS, which starts the next program. This means the amount of time an application gets, depends on itself - if it gets caught in an

infinite loop, it will never pass control to the operating system, and thus lock up the whole system.

But that's not how CLiPS works! CLiPS uses preemptive multitasking, as we know it from e.g. the Amiga. In a preemptive multitasking system, the OS decides how much time an application is given - the program itself can't influence this, and also causes no harm if it gets caught in a loop. When the OS sees the time given to a program is up, it interrupts its work and switches to the next program. In CLiPS, this is realized as follows: each program may request a certain amount of processing time; if a program has no special wishes, it gets a standard amount. CLiPS now waits until the 'time slice' given to the program has passed - at the moment, a time slice equals 4000 clock cycles.

You might think a program that's in a loop waiting for a keystroke wastes a lot of time, but Chester took care of this: there are several ways to handle processing time. Instead of giving a program a fixed time, it can also be explicitly put to sleep, until it's aroused by a click on the mouse. This way, the program uses no time at all, it won't be called before the aforementioned click occurred.

The process manager

The process manager is one of the most important components in CLiPS. Similarly to the Task Manager in the Acorns' RiscOS, it lists all running processes, together with the memory requested by each, and the processing time assigned to it (its priority). You can use the process manager to let any process sleep, arouse it, give it more or less processing time, or end it. There's a hard and a soft way to end a process: the soft way, the program is sent a Close signal, just as if the user clicked on the close symbol of the program's window. However, should the program choose not to react on this signal, you can kill it the hard way: it simply gets no more processing time, and its memory is de-allocated.
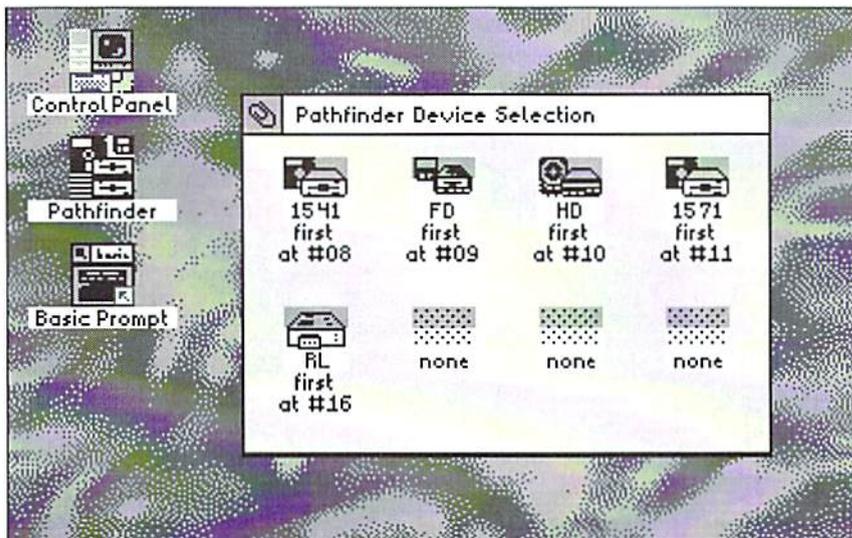
The GUI

On first sight, the graphical user interface looks like the old one, but there are new features as well. First, thanks to the micro-kernel concept, the GUI is nothing but a program, like any other application. But since it's the central control instrument, it can't simply be closed through the process manager. This change isn't very relevant to the user; the second is by far more noticeable: the "old" CLiPS didn't allow the windows to exceed the screen borders, which could be inconvenient in some situations. This is now history, as the new CLiPS draws each window into a virtual bitmap. Each time it's changed, the image on the Desktop is updated. Not only does this method save precious time, it may also be completely disabled temporarily, e.g. to allow an application to display an FLI image, or change to multi color mode, without risking other applications messing up the screen. By the way, an application doesn't even have to open a window. It's possible to write a program to play music in the background, which stays invisible on the screen and is only listed in the process manager.

Vector Fonts!

On the graphical side, CLiPS will even go one more step ahead: plans are to include a TrueType font interface similar to that in Windows, with the ability to scale fonts to any size. However, the file format for the fonts will be specific to CLiPS, and different from the TTF format used on the Mac or PCs. There have been thoughts to write a converter from TTF to the CLiPS format, but that's still far off in the distant future. And, to avoid unnecessarily slowing down the system, scalable fonts won't be used at all times, but only in such applications that benefit from them, like word processors, or graphics programs.

A joy for programmers

Chester Kollschen says, he wants to make life as easy as possible for the coders. Not purely out of charity, but also to make his own life easier. CLiPS provides the programmer with some routines that allow easy handling of windows and buttons. For instance, an application is told when, and at which position in the window, the mouse has been clicked, and if it was, say, a drag&drop operation (that's to say, an object has been dragged into the window). In case a file has been dragged inside an application's
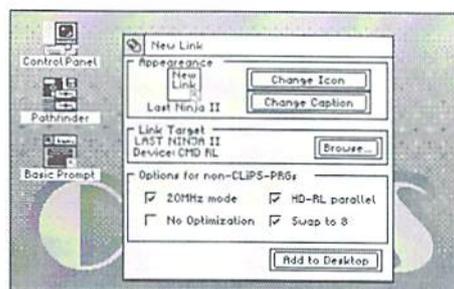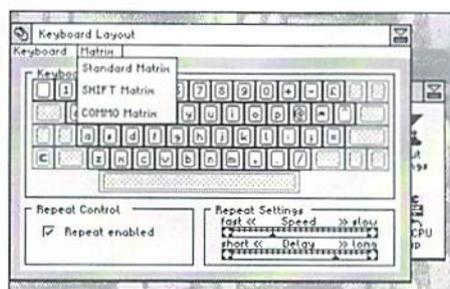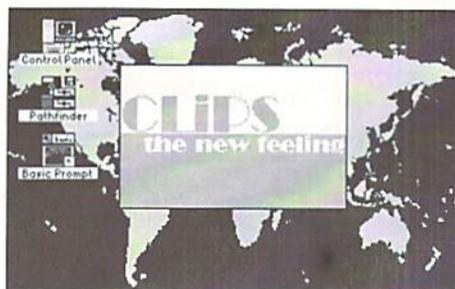
window, this application is sent an "Open" message containing the name, path and type of the file. This message is stored in a message queue which can be read byte by byte using routines provided by CLiPS.

One feature I found particularly interesting, is the fact that any application may call itself, creating a second incarnation of itself, which is located in the same memory area as the original, but due to the dynamic memory management, it can still run totally independent from the first incarnation, since it can be assigned new memory areas for its direct page (equals to the zero page) and its data bank. This technique was used for Chester Kollschen's demo of the CLiPS multitasking which displays 32 independent windows each showing a rotating paper clip - there's only a single copy in memory. The programmer won't have problems choosing a program's starting address either - like on Acorns, all programs are written for the same starting address, namely $0000 for CLiPS.
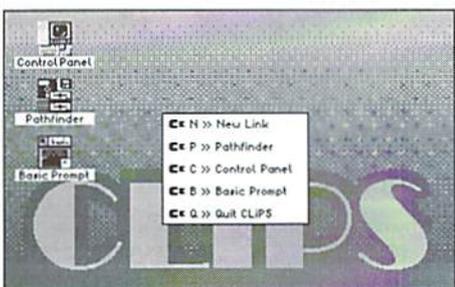
Applications planned for the future

The "Virtual Assembler 16" by Manuel Nickschas is nearly finished, and might be available when this issue is out. The "Virtual Assembler 16" will be the basis for the "Visual Assembler" which is destined to be the development platform for CLiPS. Visual Assembler will run completely under CLiPS, and offer an API library that allows easy handling of the CLiPS functions, along with several new functions.

In the meantime, Arndt Dettke is planning to write GoDot 2 for CLiPS. It's going to make use of the improved memory situation, and thus it will be able to handle larger images than before. Furthermore, a new image format using 8 or 16 bits is being considered, since it will allow for a less lossy conversion and processing of images.

As standard applications, CLiPS will contain a simple text editor similar to the Windows Notepad, and a console which may serve for debugging purposes as well.

Compatibility issues

CLiPS and BASIC harmonize better than DOS and Windows, as could be seen from the CLiPS demo shown by GO64! at the Hobby&Elektronik fair in Stuttgart. At any time, you may jump from CLiPS into the good old BASIC editor of the C64. The SCPU uses its own memory area in the SuperRam which replaces the slow memory of the C64 and allows a maximum access speed using the full 20 MHz. In order to make this data visible to the VIC, the memory is mirrored into the C64 RAM. This way, we can have an I/O area in the SuperRAM that looks like a ROM to the 64, but can be modified by changing the addresses in the SuperRAM! This means that CLiPS can provide a return address for the C64 which can't be overwritten by normal C64 programs that don't know about the SuperRAM - no matter what they do with the 64's memory. The astounding result is, you can easily leave CLiPS to play a game of Turrican or R-Type, and return to CLiPS via a SYS command. This way, even a GEOS system which is not adapted to the SCPU can be executed under CLiPS!

Speaking of GEOS... certainly, many of you are wondering what to do with your old GEOS files - the answer is: use them under CLiPS! Chester has plans to write a converter for GEOS files (photoscraps from GeoPaint, texts from GeoWrite, etc.) One limitation, though: It seems the GeoPublish file format is so weird Chester is not going to write a converter for it. But who knows? Maybe this will be taken care of by another programmer. Using data from 32 bit systems like the PC will be easy too, as all texts and characters are stored in ANSI format; this way, it shouldn't pose a problem to export/import texts from PC, Amiga, Mac and Acorn systems.

How far is CLiPS?

The kernel is complete and seems to run flawlessly. Multitasking works, the messaging between applications is constantly improved, the Pathfinder (some kind of program manager) is ready, the process manager is in the works, and the GUI is currently being adapted to the new kernel.

Internet versus Snail Mail

The times they are a-changing... nothing ever remains the same. Innovations are what people are looking for today, and the computer industry is the last, lagging behind in such matters.
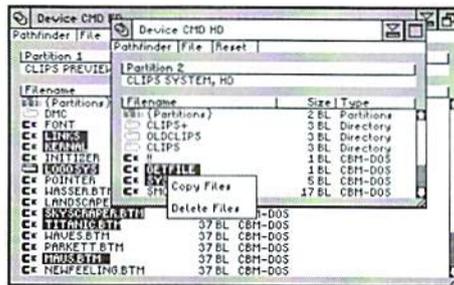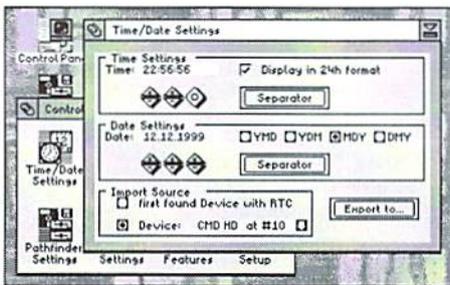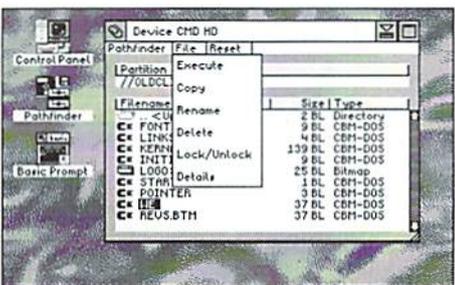
Spotlight

by Sven Tewes

Since the C64 is a cult machine that nowadays only plays a somewhat humble part amongst more powerful computers, people tend to think that technical progress has long forgotten the legendary breadbox. Of course, some minor hardware extras - like the SuperCPU for a start - have left more modern fingerprints on the C64.

D64 instead of C64?

C64 fans experience a stronger influence from the Internet than many people are conscious of. In the mid-eighties, people from the scene and other C64 users exchanged diskettes, whereas nowadays the Internet has taken over this communicative function. So-called D64 files that are created with a PC can be transmitted effortlessly from PC to floppy 1541 by means of a X1541 cable. Production costs of such X1541 cables lie between 5 to 10 marks, and the number of users interested in what the cables offer is rising constantly.

The effects are obvious; the more people use the new cable, the fewer who rely on good, old, snail mail. "So what's the problem?" some of you might now ask.

In earlier years the standard way to communicate with computers was to exchange C64 disks and disk notes (disks with private messages). Now this form of communication has become an absolute rarity. Even at the beginning of the 1990s when the Amiga began to pull the carpet from under the C64's feet, the scene still had its 'megaswappers', people who had the job in their group to spread the latest productions far and wide among the other scene members as fast as possible. These megaswappers had hundreds of contacts, i.e., people to exchange diskettes with - sometimes the number of contacts exceeding 300 or more people.

Mailing started many, new acquaintances

As a result, you received mail every day from different people all over the world. The C64 scene was a true melting pot of cultures. Many people now prefer to communicate via the Internet, debate for hours on end in one of the C64 chat channels, and get information and news from different groups' homepages that often are hopelessly outdated. This concentration on the Internet has estranged many people from the C64, and the scene has become less alive now that everybody overuses new, high-tech computers.

What's the incentive for people to give up on using snail mail and instead organize everything solely through the Internet? The advantages are evident. E-mail messages reach the receiver considerably faster than letters via snail mail, cost less, and do not run the risk of getting lost in sorting offices. Mankind clearly has a preference for the easiest way of doing things. Doubtlessly, this has become a characteristic feature of modern life. Rather than copying disks, writing addresses on envelopes, and pasting on enough stamps, clicking a 'Send' button seems a logical step in this quest for the least trouble possibly attainable.

It would be nice to keep old habits instead of getting rid of them, at least to a certain degree. Only a very small part of today's scene still relies on the good, old, snail mail - mostly users who do not have private access to the Internet. Just maybe, some people from the scene will feel the loss, and for nostalgic reasons and remembrance of the old days, some snail mail will once more go out to the world. That would be a great idea, if only to recall the slightly stale whiff of what was once a cult.

GEOS 2000

A Look Ahead

Wolfgang Grimm

Anybody can read about what has happened in the fields of GEOS last year. But what will the future be like?

by Nikolaus Metz

When it comes to GEOS, I'm not just curious - I want to know everything. Therefore, I have interviewed Wolfgang Grimm, a very active GEOS programmer, in order to get the answers.

GO64!: Hi Wolfgang, after a quite good year for GEOS - MegaPatch64/128, Wheels128, the Finder, The Wave - what are you expecting for 2000?

WG: It will probably continue that way for GEOS in the year 2000, if I only have enough time to realize all my ideas. For quite some time now, I have been thinking about a new GEOWrite, but MegaPatch has to be perfected first.

GO64!: Can you see cooperating with other programmers, the way Maurice does?

WG: Sure, why not? This has worked fine so far with Markus Kanet. Currently, I am working together with Roy Bachmann in

order to adjust his Epson printer driver to work with MP3 and 64net. Ever since the internet and e-mail have become affordable, the cooperation runs much more smoothly, because you can exchange source codes whenever needed.

GO64!: What will be the future like for MegaPatch? What will be improved and expanded?

WG: First thing will be completing WinDesk. At the same time, Markus Kanet will be finishing the new GeoDOS V3. The next item to be approached will be adjusting 64net. Then we will see what else the users need. WinDesk has already evolved very far, and will probably be the finishing touch to MegaPatch. Currently, I am focusing most of my efforts on 64net and the HP printer drivers. My intent is that 64net be completely integrated into MegaPatch. The new HP printer drivers are almost finished, and are finally able to print GEOPaint pictures in color (all of the C64's 16 colors). In addition,

you now also have the ability to use control codes in GEOWrite documents, allowing for different styles, colors, and so on, in its text mode as well.

GO64!: Will 64net be available again, now that PPE have ceased their activities in the C64 field?

WG: 64net is already available again, since I have already made a distribution contract with Paul Gardner from Australia. And the price has also been significantly reduced. Until 01/31/2000, the complete set, including the cable, is available for the introductory price of 79 German Marks. Thereafter, it will be 99 Marks, cable included. The 64net programs on the C64/C128 side have also been upgraded. I am offering a new 64net-Setup and a German version of 64net-Move, for instance. In addition, reading the PC's RealTIme Clock has been integrated into MP3's Mega.Editor, and 64net cooperates with MP3 and a SCPU at 20MHz, making it nearly as fast as a RAMdisk. Completey integrating the device driver for 64net into MP3 is only a matter of time now.

GO64!: Are you working on a C128/80-column version of GeoDos?

WG: Definitely not, that would too much of a hassle. GeoDos will remain a specialized GEOS64/GEOS128 (40 columns) program. Both MP3 64 and MP3 128 will get the new WinDesk, which will probably possess functions similar to those of GeoDos. Converting files, including DOS-CBM and vice versa, will certainly remain the province of GeoDos, though. WinDesk is supposed to become a new, comfortable environment which allows the user to quickly and easily start, copy, and work on files. GeoDos, on the other hand, is going to be a powerful tool, but also can funtion as a DeskTop.

GO64!: What's the difference between these two DeskTops?

WG: The concepts are different on a fundamental level: WinDesk is supposed to use up as little memory as possible (the maximum is 64KB, so that it can fit into a single bank of a RAM expansion unit), whereas GeoDos, in its latest version, will probably require at least 2MB of external memory. Both programs will make heavy use of context menus (right mouse button), menus automatically popping up when the pointer is moved over them, and some other things many PC users have long since grown accustomed to. Yes, the C64/C128 can also be used as comfortably as a PC. In addition, the 'Internet Controls' will be introduced into GEOS, which introduces features like using a single mouse click in order to start programs etc.

GO64!: Our German readers have a special interest in the area of online, of course. What news do you have?

WG: BTX is bound to die out sooner or later, as the Deutsche Telekom has no interest in the obsolete CEPT structure. Therefore, there will be no more developing the MegaOnline-Decoder. The only sensible thing should be a web browser, as the one being developed for Wheels in America. This is supposed to be freeware, and for that matter, open source, so that adapting it to MegaPatch should be possible. Once the time has come, all efforts will probably be directed at adjusting that program to MegaPatch.

GO64!: Do I have to look for a new Online service then, or will this also work with T-Online, or the BTX access used thus far?

WG: No, if there will be a web browser, it will have to support TCP/IP (Transmission Control Protocol/Internet Protocol) so that all access works just as with the PC.

GO64!: ... and my BTX access can also be used by the T-Online program on the PC. But now must I stop keeping you from your work! We wish you a good time programming, and thank you for extensively answering my questions!

WG: You are welcome...

Another Look Ahead

Not only MegaCom in Germany is developing products for the new millenium. Let's have a look at the news from Click-Here-Software.

by Nikolaus Metz

In the last months of 1999 we've read some interesting messages which made us really curious, so we contacted the well-known creator of Wheels 64/128, Maurice Randall, to gain some first hand information.

GO64!: A turbulent GEOS year passed by - MegaPatch3 got ready, Wheels128 came out, you published the Finder and finally you told us about continuing work on The Wave - what are your expectations for 2000 ?

MR: This year we finally get to see some actual Internet software development. And it's not coming from just myself. Of course, I'm taking on the job of creating a graphical web browser and it has some pretty stiff requirements as far as the hardware goes. But then again, that's not all bad either. It requires a SuperCPU with SuperRAM and there's nothing wrong with having equipment like that!

GO64!: Do you want to force all the other users to upgrade their system with a SCPU(128) and a RAMCard ?

MR: There are a few others doing Internet development as well. Some of them require only a stock 64 setup, but in order to do this, they are mainly text based applications. It takes a lot of horsepower to do this graphically.

GO64!: The Wheels users would like to know what's to come in the future and perhaps how soon we'll see it .

MR: Of course, I've already mentioned The Wave, the graphical browser package. But I think we'll see some other interesting projects, also. The Wave is really the first BIG application being developed for the SuperCPU. This is likely to spur further development. Software sells hardware and there will be many more SuperCPUs sold because of The Wave. And with more SuperCPUs out there, there will be more encouragement to do more software for it. So, I think you'll see the major developments from now on being aimed at the SuperCPU.

GO64!: The Wave will break down the belief that an old system can't take part in the Internet. What will be implemented? When will we get a version we can go online with?

MR: We should be seeing the first version capable of accessing the Internet real soon, if not already by the time you read this. The
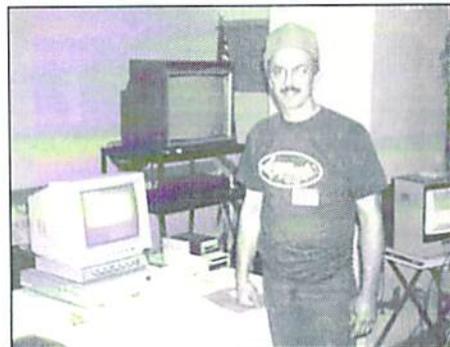
Maurice Randall

Wave is not really the browser, but actually the whole package. It contains a graphical browser, a VT-100/ANSI terminal, and the terminal will also have telnet capability. The Wave will be made available for free downloading. It will develop more and more as time goes on. All the necessary stuff will be added such as an Email application, FTP, etc.

GO64!: How will all those Wave users get their updates?

MR: Getting updates to The Wave will be simple. Getting the first few copies will not be so simple for some people. The Wave will be made available on my web site (http://people.delphi.com/arca93/) as well as my BBS (517-322-2386). As I create upgraded versions, these will be placed on the web site and the BBS. So, users will be able to download the new versions and install them. Eventually, I'll add a feature in The Wave that will allow the user to check the web site for new updates and do the whole upgrade automatically. The user won't have to figure out what files to download or where to copy them to, the auto-installer will do it all.

Maurice Randall Talks GEOS

This is a site devoted to Commodore GEOS, Wheels 64, Wheels 128, and anything else related to Commodore computers

Wheels 64/123 - a more modern GEOS for today's computing needs.

geoFAX - the only way to send and receive faxes with a Commodore 64 or 128.

geoSHELL - a versatile command line interface for GEOS and Wheels.

Flight Simulator II - take to the skies quicker using your SuperCPU and your CMD storage devices.

The Wave - graphical web browsing for the Commodore 64 and 128.

Programming In Wheels - information for creating Wheels applications.

Software ordering information - prices and other info on Wheels, geoFAX, geoSHELL, and others.

Stock Car R[...] ventures.

Recent updates to this site - check out what's changed since your last visit.

Want to contact me? My email address is: arca93@delphi.com

This site is WAVE friendly!

Left: Homepage of Maurice, rendert with Netscape

Below: The amazing result of the same page using The Wave under Wheels.

wheels   open   options

A1://index.htm

Maurice Randall Talks GEOS

This is a site devoted to Commodore GEOS, Wheels 64, Wheels 128, and anything else related to Commodore computers

Wheels 64/128 - a more modern GEOS for today's computing needs.

Wheels Upgrades and Goodies Page - download updated files and other stuff for your Wheels system.

geoFAX - the only way to send and receive faxes with a

The Wave - graphical web browsing for the Commodore 64 and 128.

NOTE: The above "WAVE" page isn't ready yet.

Programming In Wheels - information for creating Wheels

wheels   open   options

A1://index.htm

line interface for GEOS and Wheels.

Flight Simulator II - take to the skies quicker using your SuperCPU and your CMD storage devices.

prices and other info on Wheels, geoFAX, geoSHELL, and others.

Stock Car Racing - news about my own racing ventures.

Recent updates to this site - check out what's changed since your last visit.
Want to contact me? My email address is: arca93@delphi.com

This site is WAVE friendly!

GO64!: *You started programming The Wave in 1996. Did you have to create your new OS, Wheels, first before you could go on with The Wave?*

MR: If you want to know the truth...

GO64!: *Sure!*

MR: I didn't realize what I was getting myself into at the time! I had big ideas at the time, and I failed to realize how large a project (The Wave) could actually be. Plus, our computers were fairly limited on speed and memory. Of course, the SuperCPU has changed all that. At the time, I also wanted to simply experiment with doing a terminal program. I had already proven that fast communication was possible in GEOS with geoFAX, but I wanted a terminal program that could run in GEOS and make use of the SwiftLink and a fast modem. So, The Wave was originally nothing more than a very simple terminal. It couldn't upload or download, but it could at least dial out to a BBS. I used it quite frequently to dial out to Genie and Delphi, also.

GO64!: *And this changed as you continued working on Wheels ...*

MR: As Wheels evolved, I realized that I now had an operating system that could better handle something like The Wave, and I had the SuperCPU for both the 64 and the 128. I've now got something better to work with in both hardware and software. If I had kept on

trying to make a web browser 4 years ago, it would have been a big flop. I probably wouldn't have finished it anyway.

GO64!: *Now we know all about this one, but are there any other new projects?*

MR: An ongoing project during 1999 will continue to be developed. This project is called "PostPrint II" (Laser Lover Disk - see below). It's designed to take the place of geoPubLaser and geoLaser, but is really much, much more. It has a layout mode that

lets you combine geoPublish documents with JPEG images, currently. And you can import any page from any geoPublish document into your layout. With one layout, you can import as many as 127 geoPublish pages. Printing is very versatile. You can print to a PostScript printer through a serial interface, a geoCable, a SwiftLink/T232, a userport RS-232 adaptor, or to a disk file. There are many more features and I'm sure you'll see this application talked about in the near future.

GO64!: *Do we have to own geoPublish or is this a stand alone application?*

MR: PostPrint II is a stand alone application. You can use it with or without geoPublish. If all you want to do is be able to print JPEG images, you can use it for that purpose. Of course, the program is really geared up to cater to geoPublish users. With PostPrint II, you can treat geoPublish as an assembly tool, but PostPrint II can be the final layout and printing tool. I use it now for all my geoPublish layout and printing functions.

GO64!: *You'll have a lot of work the next time, are there any further projects which are worth mentioning?*

MR: I also see Wheels and The Wave possibly being combined someday into a newer Super Wheels operating system. Currently, this is merely an idea, though.

GO64!: *A new 'SuperWheels'? The 'Wheels64/128' are super! Or will there be much more than this? No, no, I'm not just*

curious - I want to know everything!

MR: Like I said, this is just an idea at this time, but SuperWheels, or whatever I might call it when I do it will be a version of Wheels that requires the SuperCPU. The current version of Wheels runs with a SuperCPU, but it also runs without it. So, it can't really take advantage of all the SuperCPU has to offer. The Wave is a step in that direction because when you launch The Wave, it becomes your current desktop. You can run applications and desk accessories while you have the browser running.

GO64!: *So The Wave will replace the Dashboard. Why did you choose this route?*

MR: The Wave contains a lot of routines for handling the memory in the SuperCPU, moreso than Wheels itself. And it can run routines from any available free bank in the SuperRAM. Much of the code in The Wave will eventually find its way into the SuperWheels OS. SuperWheels will also be able to run any current application since it

will emulate the current version of Wheels, so to speak. New applications written specifically for SuperWheels will have much more capability, though. But like I said, this is in the "idea" stage at this point.

GO64!: *Thanks a lot for answering our questions and good luck for your further work.*

Note: The Laser Lover Disk is only available from

Dale Sidebottom, P. O. Box 303, New Albany IN 47151-0303, USA,

luckykds@iglou.com,
$25 US plus shipping

Picture: Sebastian Bach

how to GoDot

by Arndt Dettke

The Interlace Mode's Peculiarities

Interlace with the C64? That exists only on the C128 with its VDC, doesn't it? You are right and wrong! The C128 VDC interlace is an interlace in the most literal sense, a hardware interlace. With it, the video image is fed to the monitor in small morsels, so to speak. Thus, you see one half of the picture first and then the other one. Some of you may probably say, "I haven't noticed that ever before". Hmmm, but isn't it the upper half of the screen that's shown first, then the lower one, and so on? Even a cow would notice that. No, things are a bit more subtle. The monitor shows the first half by displaying only every other raster line, and during the second pass, it fills up the lines not shown yet. It does so by switching back and forth, but fortunately so fast that the eye doesn't really notice the flickering. By the way, every t.v. set is based on the same principle.

Interlace on the C64 is a software interlace. It switches back and forth between two pictures as well, not on a line by line basis but rather column by column. With the hardware interlace, the second half of the screen is one raster line below the first one (in other words, at a different location in vertical direction). With the software interlace, it is one dot to the right (at a different location in horizontal direction). And there is another difference... During software interlace, there are really two complete pictures exchanging places (without any holes every other column). Unfortunately, this makes the flickering quite easily visible, a significant disadvantage for this mode.

However, the following advantage certainly outweighs the problem. If the single dots in the two picture halves have different colors, they are mixed by our slow eyes into a new

color which doesn't really exist! Yellow and white become light yellow, light red and yellow become light orange, and so on. Together with the 16 original colors (if two dots of the same color are exchanging places), you get a palette of up to 136 different colors! Now if that ain't something!

Some combinations are unbearable, though - black and white, for instance. Their flickering really makes you sick. This holds true for all combinations whose components differ strongly (see box). That is why GoDot only has a selection of 55 colors that make flickering less vivid.

Now enough with the basics! We are able to cut down the amount of flickering by cleverly placing the dots, and in order to do so, GoDot comes with an entire battery of modules: **mod.OddSwap, mod.RasterIFLI, mod.FlickerFixer, mod.-PixelEdit, mod.Scroll, mod.ClipWorks, mod.ColorCount, mod..Histogram,** and **mod..HowMany.** Some of these (the latter three) only give information in order to help evaluate the picture correctly while the others change it.

Because there is no MCI saver available yet, we will show how it is done using IFLI as an example. Having some color gradients in blue in the background, our example picture is something quite problematic for the C64. First, the picture gets a FLI treatment as shown in our last workshop:

```
Load: GIF
Inst: DecodeGIF
Load "fighter.gif"
Load GIF
Display

Inst: Scroll
Execute
Execute (Settings are already
correct)
Leave
Inst: ClipWorks
Hor: 3
ClrClp
(Set:) black
Inside
Leave
Cancel
Display
```

This set of actions has loaded the picture and cleared the 24 dots of the left border the FLI mode can't display. We can now see it on the screen and find out that it looks green rather than blue. Let's use mod.PixelEdit to see what has happened to the blue dots:
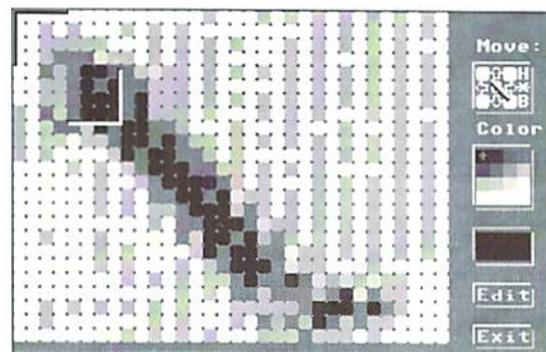

```
Inst: PixelEdit
Execute
Move O
```

Let's click on one of the green areas of the graphic screen which has become visible again, for example, the light green next to the cruise missile just fired. We can see two things: a) the blue dots are there, and b) the picture seems to be streaky! What has happened?

Well, what this has impressively shown us is the way GoDot saves the two interlaced picture halves. GoDot really switches the picture halves column by column! The left one of two dots belongs to the first picture and the right one to the second picture. This has a lot of advantages. GoDot can transfer the picture into the standard multicolor format more easily, for instance. The other advantages are related to today's topic. After we save the picture and others that will be developed later on, we can compare them, thus illustrating how GoDot renders these different images.

```
Exit
Save: FunPaintII
Save "normal"
```

We will have another look at it later in the IFLI mode, because we want to see which of the following processes brings forth the best result, in effect the least flickering. You need to know that placing the dots in a checkered pattern distributes the contrasts in a better fashion between the two picture halves. Those contrasts are to blame for the flickering, and with this distribution, both pictures have the same amount of dots rich
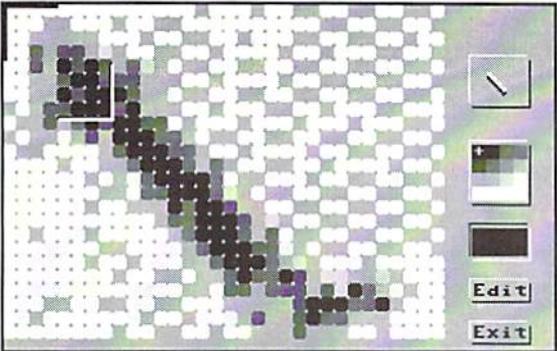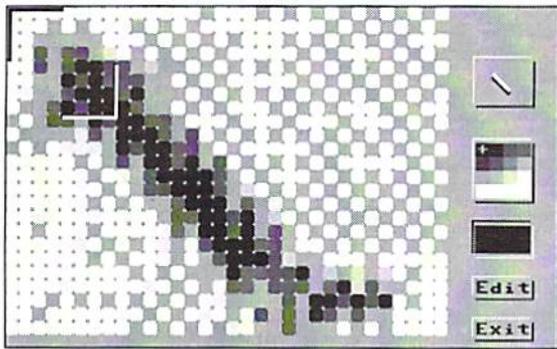
and not so rich in contrast. This reduces the IFLI picture's flickering significantly. We get a checkered pattern by doing this:

```
Inst: OddSwap
Execute
```

After "Display" (which shows us a picture with horizontal stripes and changed background color) and "PixelEdit", we can see that our missile's surroundings look much more even. We save this picture as well, this time using the name "oddswap" so that we can later discern which module yielded which success. For the next shot at it, we need to undo OddSwap:

```
(Inst: OddSwap)
Execute
Inst: RasterIFLI
Execute
Display
```

Wow! This picture looks quite good! What we just did was optimize an IFLI picture for multicolor. This created a screened multicolor graphic. In zoom mode, it looks like our third illustration. The dot distribution is still a checkered pattern, but now it seems to be doubled dots, that is, multicolor dots. Those taking a closer look will realize that this is not quite the case, though. At the magnification's right border, you can see how the dots are really organized. One pair of dots is still made up by the dots from the two semi-pictures, but they are exchanged line by line. This is good for multicolor, but is it also good for IFLI? Let's give it a try by first saving the picture with the name "rasterifli" (what else?).
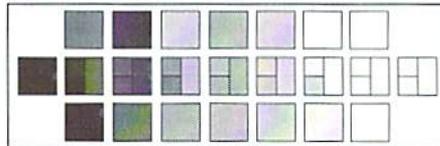
Good. Now let's leave GoDot and have a look at the graphics created in the IFLI mode:

```
Exit
Exit
load "IFLI-diashow",8
run
```

If the IFLI pictures are saved on drive 9, we add to the LOAD line :"u=9" (with quotation marks and colons!) before pressing <RETURN>. Then the viewer will automatically search the correct disk drive. American readers are asked to use the Jeffrey Jones' program (with IFLI capabilities by Todd Elliott), FileMaster, from Loadstar #176. And now, which result looks best?

The original one ("^^normal") is hopeless; it flickers disastrously. The other two pictures are good, but you can still see the pattern quite strongly in "^^rasterifli". This can be quite adavantageous with some other motifs, though, so always give it a try! Maybe with last issue's sample picture?

And next time something completely different. We take care of problems when loading GIF images!

Suitable Color Combinations

Combinations of colors that never flicker are those whose components are adjacent to each other in GoDot's palette. This is due to these colors having the same brightness on the hardware side. You can see this directly by tuning the monitor's colors down while viewing the palette window. With the exception of black and white, which stay the same, every two color boxes become one rectangle. Nine different shades of grey remain. Therefore, the following are good combinations:

blue - brown
dark grey - red
purple - orange
medium grey - light blue
light red - green
light grey - cyan
yellow - light green

Combinations from two adjacent levels are also well suited:

black - blue - brown
blue - brown - dark grey - red
dark grey - red - purple - orange
purple - orange - medium grey - light grey
medium grey - light blue - light red - green
light red - green - light grey - cyan
light grey - cyan - yellow - light green
yellow - light green - white

The more the brightness differs, the stronger the combination flickers. The worst pair is black and white. Therefore, those two colors can be used for brightening or darkening only with very few colors.

Picture: Sebastian Bach

BASIC Effects

Neon Lights
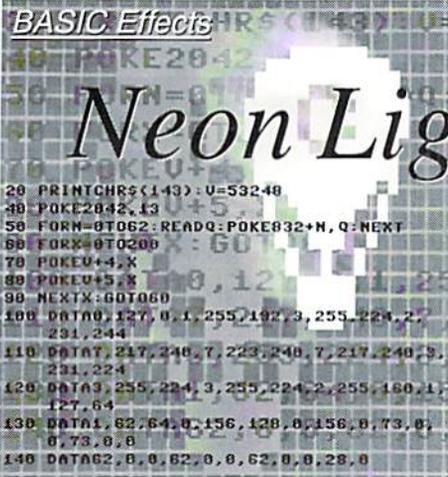
Once more I'd like to demonstrate to you how to program breathtaking effects in BASIC.

by Marcus Hinzmann aka Silver Fox/Satovia/Sanity

Great-looking effects that bring movement on the whole screen have always impressed observers. And who wants to be bothered with effects that only show in one small window? Effects that take up more space on the screen simply are more sensational. To have a large effect, however, it is not necessary to enter miles of BASIC lines. The effect we're going to try today generates giant vertical bars of colors that roll across the screen and look very nearly like Las Vegas neon signs. These are the memory cells and ROM routines we need for our effect:

Memory Cell 648 ($0288)

I'm sure you still remember this memory cell from the November issue, because we already used it for the effect then. Normally set to 1024, it defines the screen memory start for the C64's operating system. Because the number 1024 cannot be represented in one byte, we take the number of the memory page (one page equals 256 bytes) where the screen memory starts. That's why memory cell 648 contains the value 4, i.e. 4 x 256 = 1024. Those of you who don't know how to work with this memory cell should return to the November issue of GO64!.

Memory Cell 53272 ($d018)

This memory cell defines the location of the screen memory for the C64's hardware. Moreover, you can find information on the character set used at present in cell 53272. In normal mode this memory cell contains the value of 21. It is easy to explain why; 16 is the standard value for the start of the screen memory from above, and it is now increased by 4 plus 1. 16 defines the screen start (16 x 64 =

1024). It's a 4 here, because we want to use the regular character set from the C64's ROM. This character set can be found (in VIC memory bank 0 which represents the first 16 KB) at the position 4096 (4 x 1024 = 4096). The C64 automatically adds 1 to the number for reasons that need not be explained here. We already had this memory cell in the last issue, too, so this short summary should be enough to refresh your memory.

Memory Cell 56576 ($dd00)

This memory cell is also an important one in respect to the effect. It defines which of the four VIC areas is to be used. In the C64's standard mode, it has the value of 151 (the fixed value of 148 plus 3 to define the bank. The value of the bank has to be indicated as inverted, so 3 means bank 0). For our effect we need bank 2 and therefore have to enter the value of 149 (148 plus 1). Those of you who find this difficult to understand should reread the article from the November issue.

Memory Cell 53625 ($d011)

This is one of the VIC's control registers. It gives you the possibility to set many things, e.g. if the screen should have only 24 or 25 lines, or if the computer should switch to graphics mode. Another possibility is to just shut down the screen. It will then turn to one color, i.e. the one of the screen border. To stop the VIC generating a screen, we just have to POKE this memory cell with 0. To reverse that command, POKE the memory cell with 27, which is the default value.

Memory Cell 198 ($c6)

This is a very useful memory cell which - by means of an inquiry with PEEK - indicates the number of keys pressed which has been placed in the keyboard buffer. If you want to wait until the user has pressed 4 keys, you have to include the command WAIT 198, 4 in the program.

System Routine at 58640 ($e510)

With this routine you can place the cursor on the screen. POKE memory cell 211 with the number of the column (0 - 39) where you want the cursor to be, and POKE memory cell 214 with the line number (0 - 24). If you now execute SYS 58640, the cursor automatically is at the position defined in 211 and 214.

Neon Lights

Basically this effect does nothing more but continually PRINT one predefined string that consists only of color codes onto the screen. Here's little side remark concerning the way the BASIC editor works. Surely, you all noticed that the C64's screen automatically scrolls up one line if you place the cursor in the last line and then repeat "Cursor Down". There's no difference there whatsoever in the way the BASIC editor behaves if you list a long BASIC program. Every line is continually scrolled up by one and makes room in the last line for yet another one. This way it is possible to display the whole program. So what's the connection to our effect? There is a very strong connection! If you take a string of colors and go on PRINTing it various times, the colors PRINTed on the screen will scroll up, too. The result is a continuous movement on the screen in all of its areas. This is only logical, because every time one more line is added at the bottom, all other lines scroll up by one. This effect is exactly what we want. Let's have a little test program:

```
0 a$="{Rvson} {orange}Shift+Q{red}Shift+
Q{lightred}Shift+Q{yellow}Shift+Q{white}
2xShift+Q{yellow}Shift+Q{lightred}Shift+
Q{red}Shift+Q{orange}4xShift+Q{Rvs off}"
1 PRINTa$;:GOTO 1
```

If you start the program, you'll get a picture that's really unbelievable! The screen slowly fills with large vertical bars that unroll from right to left. However, this little program still has some flaws - tiny but visible. If you look more closely, you will notice that something is happening at the bottom of the screen, leaving the impression of hectic activity. Moreover, there's no possibility to color in only certain areas of the screen. Instead, the whole screen is colored, and the last flickering line is easy to detect.

The Program

So what do we have to do in order to make visible only a certain part of the screen? If you think back to the last issue, you won't find this a particularly difficult problem. Using the memory cells 648 and 53272, it is possible to set the screen start, which normally is at 1024, to two different locations so that now one of these two screens can be filled with characters as usual while the other is reserved for the color bars. The procedure continues as follows; you PRINT any kind of character on the screen in those places where later on you want to have colors. Those chars are going to light up in the colors that you defined for the color string in the second screen. If you now set the parameters of memory cells 53272 and 56576 to the correct values, the only thing left to do is to continually go on PRINTing the color string you defined at the beginning of the program. The colors will scroll up, because they move in the first screen (1024 to 2023). On the other

hand, the characters are neither deleted by the color bars nor do they scroll up as they do in the second screen that should be defined from 40960 ($a000) to 41959 (that's at 8192 bytes from the start of the corresponding VIC bank).

Let's write and start a little program that you can call up with GOSUB 1000. With it, observers of the demo can stop the colored bars by pressing any one key. In order to be able to do this, you have to define the following variables in the first BASIC line:

```
Z as 53272    (memory cell for case shift
              and hardware location of
              the screen memory)
F as 56576    (memory cell for changing
              the VIC bank)
D as 53280    (border color)
E as 53281    (background color)
P as 648      (memory cell for changing
              the OS's screen memory
              location)
C as 58640    (sets the cursor position
              with the values of X from
              211 and Y from 214
              respectively)
K as 198      (number of keys pressed)
H as 53265    (graphics chip control
              register)
```

Next you have to define the color string a$. You can define and change the color code in any way you want to create your own color mixtures. It is important that you don't change the number of spaces in the string, because this is the only combination of characters that produces the image of rolling bars. Now you should shut down the screen. POKE 0 into memory cell 53265. You should do this so that the build-up of the screen cannot be observed, because that would seem a little unprofessional. Later on, when the bars have taken on the right color and everything has begun to move like it should, you can reopen the screen. However, everything that happens in the beginning remains hidden to the observer.

First the screen start is changed to 40960 (POKE the number of the memory page for 40960, which is 160, into 648) and set the border and background colors to black (0). Now you are working in a new screen, which you first have to clear (PRINT "{clr}").

In the third line you define the chars that later on are going to show the colors. For this, you need the program loop T with numbers ranging from 2 to 22. T represents the lines of the screen, of course. Then position the cursor by setting the column to 2 (POKE 211, 2) and the line to T. T takes every line number from 2 to 22. You can change the numbers in any way you want, but none of them should be higher than 23. Start the cursor control routine with SYS 58640. The cursor is now located in column 2, line 2 to 22, depending on the value T at a certain point in time during the program loop.

Secondly, write another loop that defines the

value of the variable J from a basic value of 1 up to the last value of 35. Within this loop a simple command PRINTs the reverse, shifted Q on the screen for running 35 times (because of the semicolon behind the PRINT command), i.e. the screen fills with reverse circles (see CHR$ codes in the manual). Every time one line is filled with 35 circles the cursor jumps to the next line of the screen. Thus all lines from 2 to 22 are filled with circles.

The program now sets memory cell 53272 to a value of 133 (128 + 4 + 1: 128 + 4 represents the position of the screen memory in VIC bank 2 plus 4 x 1024 for the character set plus a 1 that cannot be influenced from the outside), so that now you're able to see the circles you just PRINTed on the transferred screen.

The following fourth BASIC line (1003) enables working on the transferred screen in VIC bank 2 by entering POKE 149 (148 + 1) into 56576. In order to be able to PRINT the colors on the standard screen starting at 1024 in the main part of the routine, you should now reset 648 back to 4. There PRINT again the color string a$ for 70 times using the variable T as a loop counter just as you did before. Now the colors for the effect fill the screen without any blanks, because you just PRINTed all of them (by means of the semicolon). Because memory cell 53272 is still set so that pages are displayed from 40960 on (POKE 53272, 133), you only see the colors in those lines that you filled with inverted circles before.

And now the time has come to reopen the screen for our effect by entering 27 into memory cell 53272 by the command POKE.

The fifth BASIC line (1004) is the beginning of the main routine. It PRINTs the color string a$. The colors begin to scroll up with the screen, and the vertical bars start moving. PEEK (198) then checks the keyboard buffer for the number of keys pressed. If no key is pressed, the routine jumps back to its beginning and restarts from there (...THEN 1004), giving out the next string of colors. If a key is pressed, the program continues with the succeeding line.

With this last BASIC line (1005), the C64 is reset to its normal settings. Thus the character set is returned to the normal mode with the value of 21 (16 plus 4 plus an automatical 1), and the computer stops working on VIC bank 2 (POKE 56576, 148 + 3). Last but not least, the program leaves the routine by RETURN.

This is the complete source code in BASIC 2.0 format (please type it without line breaks; I just chose the format in lines because it's clearer):

```
1000 z=53272:
     f=56576:
     d=53280:
     e=53281:
```

```
     p=648:
     c=58640:
     k=198:
     h=53265
1001 a$ ="{red} space {orange} space
{lightred} space {yellow} space
{white} 2xspace {yellow} space
{lightred} space {orange} space {red}
4xspace":
     POKE h,0:
     POKE p,160:
     POKE d,0:
     POKE e,0:
     PRINT"{clr}"
1002 FOR t = 2 TO 22:
     POKE 211,2:
     POKE 214,t:
     SYS c:
     FOR j = 1 TO 35:
     PRINT"{RvsOn}shift+q{RvsOff}";:
     NEXT:
     NEXT:
     POKE 211,2
     POKE z,133
1003 POKE f,149:
     POKE p,4:
     FOR t = 1 TO 70:
     PRINT a$;:
     NEXT:
     POKE h,27
1004 PRINT a$;:
     IF PEEK (k)<>1 THEN 1004
1005 POKE z,21:
     POKE f,151:
     PRINT"{clr}{white}":
     RETURN
```

All you have to do to activate the effect is enter GOSUB 1000, so it is easy to integrate this little program into a bigger demo. If you have problems with the program, try values different from those that I gave you here. This effect makes it easy to see that BASIC can be used for so much more than editing texts only. So use your knowledge to get the most out of BASIC. It's really worth it.

Try it and have fun.
Marcus Hinzmann aka Silver Fox
(ad)

Overview of the memory cells used

648 - start of the screen memory (number of page)
53265 - VIC control register
53272 - basic addresses for screen memory and character set
53280 - frame color
53281 - background color
56576 - VIC bank control
198 - number of keys pressed (keyboard buffer)

System routines used
58640 - cursor positioning
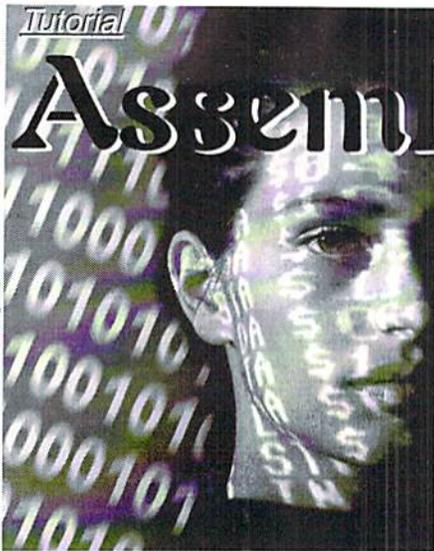
Assembly Language

by Wanja Gayk

Part 10

I can really imagine you sitting in front of the glowing screens waiting to learn some more about programming a C64. So, hesitate no longer! This issue will cover Multicolor and Extended Background Color modes - each of them quite handy guys!

The extended background color mode is so simple that you can create great effects even in BASIC. Let's start with multicolor mode, which is a bit more complex. As usual, you won't have to do without programmed effects.

The Multicolor Mode

We know multicolor not only from pictures created with Koalapainter, Amica Paint, or Advanced OCP Art Studio, but also from many logos in demos and crack intros. This mode allows us to increase the color density on screen but for the price of half of its resolution. It works both in text and in graphics mode. Until now, our graphics or text screen always had two dominant colors. In bitmap mode (graphics), every cursor block could have its own foreground and background color. The charset mode (text) allowed for one foreground color per cursor block while the background color ($d021) was the same for the entire screen.

We know that in both the charset and in a bitmap, a set bit represents a dot of foreground color, while a cleared bit defines a dot of the background color. If these were the only two options, you may wonder how the color density would be increased then. Up to now, we had nothing but register $d021 for the text mode background color and the color RAM at $d800-$dbe7, providing a single foreground color for every cursor block in text mode. In bitmap mode, we also had nothing but two colors per cursor block, as given in the video RAM's high and low nibbles (which could be located at $0400, for instance).

In multicolor mode, two adjacent dots are turned into one of double width. The first result of this is the resolution's reduction from 320x200 dots to 160x200. The other effect is that we now have four possible settings for each dot, because it is now described by two bits instead of one! The way a bitmap or a charset is represented in memory doesn't differ, though. You could happily switch back and forth between multicolor and hi-res, if it weren't for the colors going wrong. In multicolor mode, a byte no longer represents eight but merely four dots - that's it.

Multicolor in Bitmap Mode

In this mode you have one global background color ($d021) and three more color options for every multicolor dot in a cursor block, depending on the combination of individual bits this pixel consists of.

Bit combination	Source for color
00	$d021 low nibble
01	video RAM high nibble
10	video RAM low nibble
11	color RAM low nibble

Now you may wonder whether it wouldn't be better if the VIC could fetch the cursor block's background color from the color RAM's high nibble. Unfortunately, this option has either been forgotten or rejected for monetary reasons. However, it would have been a nice feature, but it is impossible due to hardware limitations.

Multicolor in Charset Mode

The charset mode suffers from some limitations when set to multicolor, but it also offers one advantage the bitmap mode does not provide. Since the 64's video RAM contains the characters' screen code values, you can't use it as a color memory like in bitmap mode. For that reason, the VIC can only get its colors from color RAM, the background color register ($d021), and two additional color registers ($d022 and $d023). You effectively suffer from the disadvantage of three colors being strictly global, while only the fourth color may change from char to char.

In text mode the nice option of displaying hi-res and multicolor at the same time remains, though. Bit 3 (the low nibble's most significant bit) in color RAM chooses which mode to use. This also means that the color RAM's low nibble can only take on eight different values for the color instead of 16, which are colors $00 - $07. To make things obvious, have a look at the following tables:

Bit combination	name	color value from address:
00	background color	$d021 low nibble
01	multicolor 1	$d022
10	multicolor 2	$d023
11	foreground color	color RAM, least significant 3 bits

Color RAM value	char's foreground color
$00	hi-res, black
$01	hi-res, white
$02	hi-res, red
$03	hi-res, cyan
$04	hi-res, purple
$05	hi-res, green
$06	hi-res, blue
$07	hi-res, yellow
$08	multicolor, black
$09	multicolor, white
$0a	multicolor, red
$0b	multicolor, cyan
$0c	multicolor, purple
$0d	multicolor, green
$0e	multicolor, blue
$0f	multicolor, yellow

As you can see, only the first eight colors are available as a foreground color if you want to use multicolor in charset mode. On the other hand, you can easily embed hi-res patterns right into a multicolor graphic (as in the background of the flight level of "Turrican II" shoot 'em up).

How Do I Activate Multicolor Mode?

Now this is really a simple thing to do. Bit 4 in register $d016 says whether the multicolor mode should be active or not. If it is set, multicolor mode is activated. You should set and clear this bit using ORA and AND, so as not to influence the remaining bits and thereby mess up a scroll effect, for example. Programmers who know what they're doing can also precompute the $d016 values and simply put them in there, which saves them four cycles (equalling about 0.000004 seconds, if I am not mistaken).

Activate multicolor:

```
LDA $d016
ORA #$10 ; binary %00010000
STA $d016
36%STA $d016
```

Deactivate multicolor:

```
LDA $d016
AND #$ef ; binary %11101111
STA $d016
```

The Extended Background Color Mode (EBC Mode)

The EBC mode is a mode you may have come across very often but may have not noticed at first sight. Those of you who have looked at innumerable disk notemaker programs have certainly witnessed it. You have some text on your screen, and the background color of the chars flashes red, then blue, or whatever. You then think that's simple, because of course it is possible to use reversed letters using CTRL+RVS ON to turn the background color to white and then change the char color in color RAM to achieve this effect.

When you watch different color chars flash, you wonder how the background color of some can flash, though you're using a single color ($d021) as their foreground color (the chars are reversed). This doesn't work out. This effect is certainly possible in hi-res bitmap mode, because we have an individual foreground and background color for every char there. However, this would be quite a hassle to program, because you would need to convert the text for every screen into a bitmap graphic. No, there is a shrewder way! You only need to know five memory addresses and one further bit in order to master this mode. In EBC mode, the VIC fetches a char's background color from four color registers: $d021, $d022, $d023, and $d024. Which register the VIC takes the color from depends on the two most significant bits of the screen code to be displayed. Two bits allow for four different bit combinations meaning four different background colors.

These are defined in the aforementioned registers. This way of choosing a background color does have a decisive disadvantage. Since the screen code is an 8-bit value, it can show 256 different chars out of the charset. In standard ROM charset, these 256 chars are divided in either 128 lower case/upper case letters (normal) and 128 reverse lower case/ upper case letters or as 128 upper case letters/ graphic chars (normal) and 128 reverse upper case letters/graphic chars.

If we then use the uppermost two bits of every screen code for designating a background color, the screen code, now being shortened to 6 bits, can only display the first 64 chars of the charset. The C64's charset is reduced to just 64 different chars under EBC mode. When you work with the ROM charset, 26 of these are needed for the alphabet, an additional ten for digits, and the rest for sentence markers and special chars. Because of this, we only have either upper or lower case letters, no graphic chars, no mixed upper/lower case writing.

It is distinctly easy to write in this mode. What makes a difference is whether you write with SHIFT locked, or with CTRL+RVS ON activated, or both at the same time. Best thing to do is have a look at the following table:

Screen code (binary, hex)	background color register	style
00xxxxxx $00-$3f	$d021	plain
01xxxxxx $40-$7f	$d022	SHIFT
10xxxxxx $80-$bf	$d023	RVS ON
11xxxxxx $c0-$ff	$d024	SHIFT+RVS ON

The program added to this issue shows how this mode works and what it looks like. I will cover the way you activate EBC mode right away, though. It is turned on and off using bit 6 of register $d011. If that bit is set, the mode is activated; otherwise it is deactivated. Since register $d011 also has further functions - for example, turning the bitmap mode on and off - you should clear and set this bit using AND and ORA like before.

Some more - EBC mode does not work together with hi-res or multicolor bitmap mode; it only allows for standard text mode! In multicolor text mode, registers $d022 and $d023 are occupied multiple times, and in bitmap mode the video RAM is used twice. Both at the same time is not allowed, and in either case the screen simply blanks. For this reason, in the following code I deactivate either multicolor and bitmap mode. Within a real program, you may as well omit this safety guard.
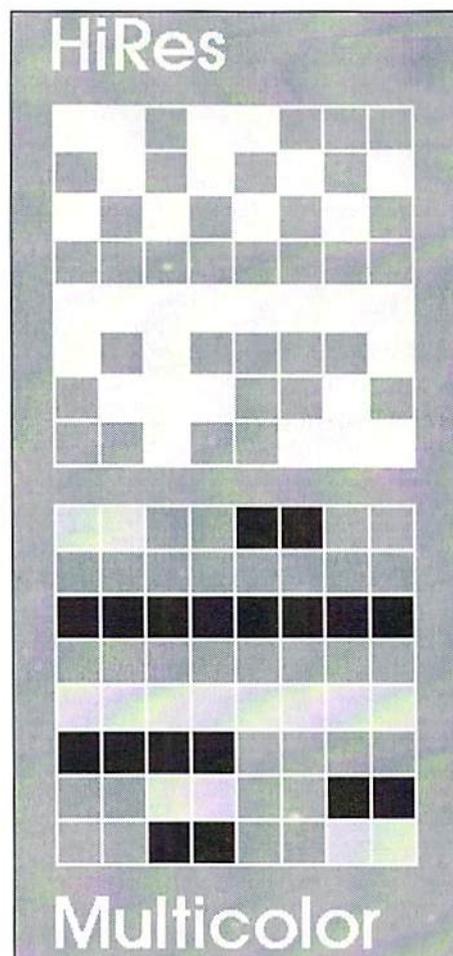
Activate EBC mode (safe method):

```
LDA $d016
AND #$ef ; multicolor mode off
STA $d016
LDA $d011
AND #$df ; bitmap mode off
ORA #$40 ; EBC mode on
STA $d011
```

Activate EBC mode (without deactivating other modes):

```
LDA $d011
ORA #$40
STA $d011
```

Deactivate EBC mode:

```
LDA $d011
AND #$BF
STA $d011
```

Finally, I'd like to show you how the described modes work and how you make use of them. In order to get a better understanding, you can try removing the multicolor or EBC mode's activation from the programs, reassemble them, and have another look at the graphics. The source codes are in Turbo Assembler format

and can be converted to other assembler systems quite easily.

Happy working and experimenting - 'til next time!

```
;--------- Example Programs ----------

Program 1: Multicolor Mode with
Bitmaps

 lda $d011
 ora #$20 ; activate bitmap mode
 sta $d011
 lda $d016
 ora #$10 ; activate multicolor
 sta $d016
 lda #$18 ; bitmap = $2000
 sta $d018 ; video RAM = $0400

 ldx #$00
 stx $d020 ; background = black
 stx $d021
color lda #$0d ; 11 bit combination
= light green
 sta $d800,x ; color into
 sta $d900,x ; color RAM
 sta $da00,x
 sta $db00,x
 lda #$e6 ; 01 bit combination =
light blue
 sta $0400,x ; 10 bit combination =
blue
 sta $0500,x ; colors into
 sta $0600,x ; video RAM
 sta $0700,x
 inx
 bne color

 ldx #$00 ; bitmap address
 ldy #$20 ; $2000
 stx $fa ; into vector $fa/$fb
 sty $fb

 lda #$d8 ; binary = %11011000
 ; (dot pattern)
 ldy #$00
pixels sta ($fa),y ; into (vector
$fa+y)
 iny ; addr.(low) plus 1
 bne pixels ; if <> $00
 inc $fb ; addr.(high) plus 1
 ldx $fb ; load addr.(high)
 cpx #$40 ; address = $4000 ?
 bne pixels ; no, then loop
 rts ; else: finished

Program 2: Multicolor and Hi-res in
the Charset Mode

 lda #$00 ; color = schwarz
 sta $0286 ; into cursor color,
 sta $d020 ; border color,
 sta $d021 ; and background color
 jsr $e544 ; CLR routine (ROM)
 lda #$18 ; video RAM: $0400
 sta $d018 ; charset: $2000
 lda $d016 ;
 ora #$10 ; multicolor on
```

```
 sta $d016 ;
 lda #$0e ; color light blue
 sta $d022 ; as multicolor 1
 lda #$0f ; color light grey
 sta $d023 ; as multicolor 2

 ldx #$11 ; 16 values (+1)
copy lda charset,x ; from table
 sta $1fff,x ; into chars
et-1
 dex ; (screen codes 0 and 1)
 bne copy

 ldx #$00
screen lda #$00 ; screen code 0
 sta $0400,x
 lda #$01 ; screen code 1
 sta $0401,x
 lda #$0e ; color 6 (+ bit 3)
 sta $d800,x ; into color RAM
 lda #$06 ; color 6 (no bit 3)
 sta $d801,x ; into color RAM
 inx
 inx
 cpx #$28 ; already 40 chars?
 bne screen ; branch if not
 rts ; end

;-- Charset Table --

charset .byte 0

 .byte %11011011 ;hex = $db
 .byte %00110000 ;hex = $30
 .byte %11011011 ;hex = $db
 .byte %00110000 ;hex = $30
 .byte %11100111 ;hex = $e7
 .byte %00001100 ;hex = $0c
 .byte %11100111 ;hex = $e7

 .byte %00001100 ;hex = $0c
 .byte %00110101 ;hex = $35
 .byte %01100000 ;hex = $60
 .byte %11000111 ;hex = $c7
 .byte %10001100 ;hex = $8c
 .byte %01001100 ;hex = $4c
 .byte %10000100 ;hex = $84
 .byte %01100110 ;hex = $66
 .byte %00111100 ;hex = $3c

Program 3: Extended Background Color
Mode

 sei ; bar interrupts
 ; (because of timing)
 lda $d016
 and #$ef ; multicolor off
 sta $d016
 lda $d011
 and #$df ; bitmap mode off
 ora #$40 ; EBC mode on
 sta $d011

 ldx #$00 ; all chars and
loop txa ; all colors
 sta $0400,x ; available onto
 sta $d800,x ; the screen
 inx
 bne loop

 stx index1 ; initialize
```

```
 stx index2 ; table pointer
 stx index3 ; (set to 0)

blinker

 ldx index1 ; load pointer 1
getcol1 lda colors1,x ; color from
table
 bne noinit1 ; if value = 0
 tax ; otherwise to index
 beq getcol1 ; and fetch again
noinit1 sta $d022 ; color into $d022
 inx ; increase counter 1
 stx index1 ; and store it

 ldx index2 ; see above, but now
getcol2 lda colors2,x ; for counter
2,
 bne noinit2 ; color 2 and
 tax ; register $d023
 beq getcol2
noinit2 sta $d023
 inx
 stx index2

 ldx index3 ; see above, but now
getcol3 lda colors3,x ; for counter
3,
 bne noinit3 ; color 3 and
 tax ; register $d024
 beq getcol3
noinit3 sta $d024
 inx
 stx index3

 lda #$fa ; wait for raster
delay cmp $d012 ; beam
 bne delay ; for delay
 jmp blinker ; get next colors

index1 .byte 0 ; counter 1
index2 .byte 0 ; counter 2
index3 .byte 0 ; counter 3

 ;-- Color Tables: --

colors1 .byte
$10,$06,$04,$0e,$03,$0d
 .byte $01,$0d,$03,$0e,$04,$06
 .byte $00

colors2 .byte
$10,$09,$02,$08,$0a,$07
 .byte $0a,$08,$02,$09
 .byte $00

colors3 .byte
$10,$0b,$05,$05,$03,$03
 .byte $0d,$0d,$03,$02,$05,$05
 .byte $0b
 .byte $00

 ;End !
```
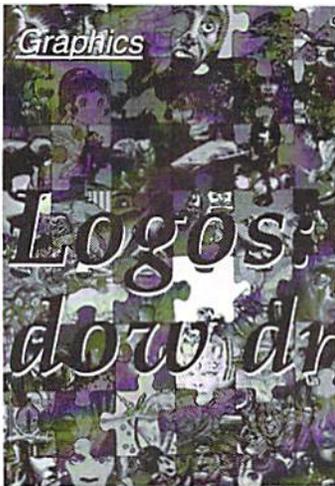
Logos: just window dressing?

Part 4

The fourth part of our graphics course is completely dedicated to drawing logo graphics, like those we often see in demos. For this purpose, we've gotten scene graphician, Andri Seeliger (also known under the name "Brain$masher"), as a guest author. He works for the groups Civitas, Axicon, Banshee, and some others, and his logos are featured in publications, such as Tiger Disk and Premium magazine. Here he's going to tell us a bit about his experiences:

by André Seeliger aka Brain$masher

So this is our try to write a logo-painting course for beginners. I'm calling it a try, mainly because there's no recipe to create an acceptable logo. The only really important requirement is being eager to experiment and to use lots of imagination. Another important aspect, of course, is to keep the logo readable. As you can see - if you've observed the scene - some painters have rather neglected this point during recent months; instead of a decent logo, you see some cryptic symbols that only make sense to those who know the meaning anyway. But never mind... surely there are some people who like a little puzzle every now and then.

In order to illustrate the process, I've provided some pictures that should make it easy to reconstruct the development phases that a design goes through. By the way, the logo I'm showing here was just finished some weeks ago. It's to be used in the intro of a forthcoming co-op demo release by the groups, "CommoBam" and "Axicon".

Remember that this course is only meant to inspire you; you can always try totally different ways in creating a logo than the one described in this article.

From size and choice of colors to a design

Full-screen logos are quite rare; usually you're limited to a certain area of the screen. The design must not exceed a predetermined size limit, or it can't be integrated later. For our example logo, I use the whole screen width of 160 pixels, but height is limited to a maximum of 40 pixels. First of all, I mark this limit with a line so that I never go beyond it.

The next step is to choose the basic color to use for our image. This is the color that will dominate the logo. You might think choosing one color out of 16 is not too difficult, but it has to be thought over. I've decided to use juicy C64-yellow. The background color has to be chosen, too; in most cases, it will be black, which gives a good contrast with brighter paint colors.

The real work is about to begin: design the lettering using the basic color, starting with the outlines. This is best done in zoom mode, even though you might find it tedious to work pixel by pixel. With some practice, you'll become fairly quick.

Slowly, our lettering is shaping up. Always keep the screen borders and the "border line" in mind when you're painting; there's nothing

as annoying as starting to draw a logo just to find the last two letters don't fit onto the screen. You might find it very helpful to use a simple paper sketch before starting your work.

In our example, you can see right away that we have a problem; lots of letters have to fit on the screen. To solve this, I designed the logo in a typical graffiti style - the letters merging into each other smoothly, covering each other, etc.. This way I use the space efficiently (fig. 1). Due to this method, readability suffers a bit, even in my own image. (sigh)

Edges and Light Effects

Your first result will often look very rough, and you'll recognize the so-called "staircase effect", which is due to the 64's very low graphic resolution. In multicolor mode, the pixels are just too large to draw well-shaped edges; you'll have to make some cosmetic improvements, namely, do some anti-aliasing. This method was already introduced in an earlier part of the graphics course; thus if you use it, you can smooth all sloping edges. Additionally, I tried to put a little light effect into the logo (figs. 2 and 3); I created it by putting some white pixels into the upper left half of the letters.

Slowly, you'll work through the whole logo. Practice will teach you which colors to use and which to avoid. You don't have to cling to a particular concept; you might just as well highlight some letters using completely different colors. Use your imagination.

Getting colorful

By painting pixel by pixel, you might like trying to paint an area with a two-colored checkered pattern, alternating between color 1 and 2. This gives a nice effect, provided you use the right combination of colors. I've used that method in our example logo. For example, look at the lower half of the "C" in "Commobam", to the right of the plus sign (fig. 4).

With some practice, you can use more than two colors in the pattern and create beautiful gradients from bright to dark colors. For a nice example, look at the lower half of the first letter "M" (between the two angular "O"s).

Naturally, it'll take some time until the whole logo is done (fig. 5 and 6). Don't forget to save frequently; if you screw up something,
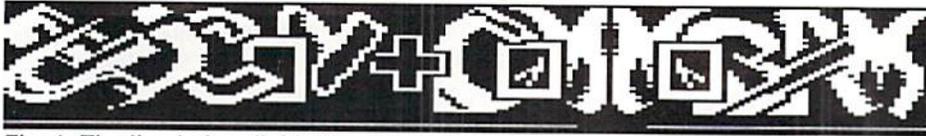
Fig. 1: The line helps divide the screen

Fig. 2: Lights from above and the first colors

Fig. 3: The first half is done...

Fig. 4: ... and the second part ...

Fig. 5: ... is developing

Fig. 6: The lettering is complete

Fig. 7: Now fill the black areas with a contrasting background

you'll be glad to have an earlier version to go on with; also, some of the C64 graphics programs are quite prone to crash.

When the logo's done, check it for small disfigurations; are there any pixels that annoy you? Is there anything else you can improve? Just take your time to work on your design.

Some more background

As a final touch, you can add a background that fits your logo (see fig. 7). Always remember that it has to contrast the colors you've used for the lettering. Since I made the logo look rather bright, I've used darker colors, like blue and purple, for the background. Here you can also use a checkered pattern to paint a gradient into the actual background color (which is black).

Of course, you can try to draw small motifs in the background, like mountains, thunderbolts, and cloudy skies. I did without them, because I didn't have the space. Our logo is sort of finished now. You can try to further improve it, if you think that's necessary, but it's usually not recommended to do any large corrections at this stage. Experience tells me that you'll seldomly be really content with your work. Even if you think it has turned out perfect, look at it again some time later, and you'll surely find two or three bad spots that you have to improve. So, better decide it's finished now, instead of sitting on it forever.

Well, that's it already. I hope I did well enough with this part of the course to prepare you for your own first masterpieces.

Have a nice day,

Brain$masher

(vr)

Supplement to GEOS Millenium Patch

GEOS users can find information about the patch files for GEOS in the Readers section of the coverdisk. This information deals with fixes for the German TextPrint V3.2 and V3.3, as well as the German GeosLQ V2 and Updater V1.1, and there is additional information about geoLaser V2.1 and geoPublish V1.0 (German and US versions). If you've already paid the shareware fee for the software from issue 11/1999, you don't have to pay again; otherwise the fee is USD 5 or DM 10. *(me)*

GO64! Bazar Form

Please insert the following advertisement in the next issue of GO64!

[grid of empty boxes for writing — 4 lines]

My advertisement is

☐ private/non-commercial (four (4) lines, 40 characters wide.)

US Dollar 3 (Euro 2.50) are ☐ included in cash ☐ included as cheque ☐ chargeable from my credit card (VISA, AMEX, Master Card)

The item(s) I am advertising about are related to:

☐ commercial (any reasonably sized ad, US Dollar 7 (Euro 6) plus taxes

☐ C64   ☐ C128   ☐ Software   ☐ Accesories   ☐ Misc

Address
CSW Verlag
Goethestraße 22
D-71364 Winnenden
Germany

From:

Last Name/First Name

Street

ZIP/Town/State

I assure that I own all rights concerning my offers.

Credit Card no./expire date/holder

Date/Signature

GO64! Bazar

Where can one buy or sell second-hand hardware, accessories, literature and software? Is there a convenient solution? Yes, there is! Check out our Classified Advertisements section, the GO64! Bazar! Above, you'll find a small registration form. Just clip it out, fill in the blanks, and send it to the publisher. For every advertisement you submit. For personal/private (non-commercial) advertisements, you may have up to four lines of text, each up to 40 characters in length. Please see the coupon for pricing and other details. We will make every effort to publish your ad in within the next issue of GO64!. If your ad doesn't make it by the next issue, it will appear in the issue following that.

For Free - Contact Corner

We're going to offer a free service to people who are looking for other C64 enthusiasts. Just send a contact advertisement in the form of a letter, FAX, or email to our editorial staff, care of: Contact Corner. The editorial staff reserves the right to shorten parts of letters which are not immediately important, in order to save space.

Computer Flohmarkt

The End of the last Multi-User Magazine

The end of the advertiser 'Computer-Flohmarkt' ('Computer Flea Market') with its two-monthly issues does not only mean that the German-speaking C64 scene has lost its last print medium available at the kiosk but also that it has lost an important forum of discussion.

by Volker Rust

By the end of the eighties Thomas Eberle, the owner of a small publishing company, had a brilliant idea: Why wasn't there any national advertising magazine for computers and everything that goes along? He started working on his idea at once: In 1989 the first issue of the 'Computer-Flohmarkt' was published. The design was poor, there were many short articles and the font size nearly made reading with a magnifying glass an obligation, but those have been the hallmarks of the magazine ever since.

A happy hunting ground for freaks

Computer freaks, however, could not be deterred by the magazine's strict functionalism and the 'Computer-Flohmarkt', usually nicknamed 'CF', kept on growing to be a happy hunting ground for selling and exchanging components, information etc.

Then, discussion forums were founded which made it possible to discuss in written form with other users or talk about the latest developments. In brief: A new medium had come into creation and gave readers the opportunity to start contact with other readers and work on projects together long before the Internet triumphed in Germany, too. It didn't take the 'CF' scene much time to respond: Different magazines like the disc mags 'Digital Talk', 'X-Dome' and 'Tigerdisk' developed on the basis of the CF or from its background, using 'CF' as a way of advertisement. Scene groups were formed via 'CF' and presented themselves to a larger public in the magazine.

Talk Show

The numbers and themes of sections were amplified or even completely changed over the years. The section on non-computer media was turned into a magazine of its own called 'Musik-Flohmarkt', which is still available. Other very popular sections like the one on love, sex and partnership were very much frequented and held as high entertainment and education value but were later on dismissed from the magazine because of a lack of space. By that time, the idea hung in the air to start a magazine specially for the discussion field below the belt but for unfathomable reasons the idea was never put into practice. Some sections like 'Swindlers and con men' existed for years, for the simple reason that free trade naturally comes also with negative experiences and that there were some people who tried to make some money with fake advertisements or who were not willing to pay in time etc. In 'Ludwig's special question time' (which later was called: 'Questions to the editor') readers could have a direct discussion with the editor-in-chief and ask questions.

But the most important feature were the discussions amongst the users. Every reader was allowed to choose a pseudonym with which to publish his/her texts. During the various stages 'CF' went through there were some pseudonyms (that is, of course, their owners) that dominated the content of the mag in particular and thus managed to reach a certain fame. Audax Oratio, Mr. X and Little John were especially prominent in their time. But there was a problem to which no satisfactory solution could be found: the measure of truth in the texts was not provable,

COMPUTER Flohmarkt 6/99

Nov./Dez. DM 7,- ÖS 55,- SFr 7,-

4 398668 407009

KOSTENLOSE KLEINANZEIGEN UND VIELE DISKUSSIONSFOREN FÜR ALLE SYSTEME

BROTKASTEN 2

the result of this being that sometimes risky statements made their way into the mag and could only be corrected by other readers two months later. During those two months rumors spread and grew, and it is easy to imagine the doubtful kinds of 'truths' that suddenly invaded the scene.

Yours sincerely, Mr. Gravenreuth

One of the most spectacular campaigns of 'CF' was connected to the problems in dealing with unauthorized copies and software pirates: fake catch letters supposedly written by young girls asking for copies of games were sent out. In order to make the letters seem more realistic unclear photographs of the girls were included. The famous lawyer Guenther von Gravenreuth from Munich ended up writing with a pseudonym of his own in the different sections, preferably the section on 'Copyright and Warnings'. This section gave room to very interesting discussions between the camps where everyone could participate.

Breadbox live

>From 1994 on Thomas Eberle even dared to publish a separate C64 magazine that was only for discussions and was called 'Brotkasten live' ('Breadbox live'). Its outstanding characteristic was that it didn't contain any advertisements so that it wouldn't compete with 'CF', and it centered on information exchange and on making new contacts. Even though the readers appreciated the idea and made a lively contribution the market was no longer strong enough to make it possible for the magazine to survive for very long. Thus the publication of 'Brotkasten live' had to be ceased after only five issues.

The End

And now, five years later, the time has come for 'Computer-Flohmarkt'. Many users still suffer from shock because this early end was unforeseen. The amount of readers' contributions had dropped a little over the past few months and the focus of the magazine had been changed a bit but such trends had always been common with 'CF' and did not necessarily point to an imminent end. Thomas Eberle's motive for ceasing the publication is the fact that the magazine's concept makes it no longer supportable economically. One reason for this may be the long interval between the various issues of 'CF' as an interval of two months often meant that an answer to advertisements e.g. took four months or more. The attempt to set up a monthly issue of 'CF' had failed some years ago but to the present day the market has changed considerably. The Internet as a form of mass communication with its newsgroups, mailing lists and auction houses offers possibilities that have in the course of time left 'CF' high and dry. Moreover, the sections on some other systems such as e.g. Atari computers have dwindled enough so that readers belonging to those interest groups may have been lost completely.

Computer Bargain?

No sooner than 'Computer-Flohmarkt' has disappeared from the scene that the Thomas Eberle Verlag is offering a new advertiser called 'Computer-Schnaeppchen' ('Computer Bargain'). But in contrast to 'CF' only 'Searching/Offering' ads are published and there are no discussion forums. Whether this new concept is going to work will have to be proved by the forces of market. Perhaps it will be possible, if the magazine is a success, to include sections dealing with information exchange. The only alternative to the discussion forums is until now the Internet where readers of 'CF' at the speed of lightning created a 'Computer-Flohmarkt' mailing list. Fans of C64s can also participate in the GO64!'s mailing list which serves as a medium to exchange information and experiences and has been very active lately.

The end of 'CF' is synonymous with the end of another chapter of C64 history and no other computer magazine offers an opportunity similar to 'CF' to expand one's own horizon. However, it should not be forgotten that the decay of 'CF' was not at all due to a lack of interest on the part of the users of the Breadbox because their section in 'CF' was still very active. The former readers now have to move on with the times if they do not want to lose track. Progress doesn't stop even for users of the Breadbox (thank Heavens!). In this case, progress means using mailing lists and web forums of the Internet. Some people may tell you that you need a PC for this type of progress, but this is simply not true. You may find it hard at first to (involuntarily) kick old habits, but the advantages of the Internet are obvious: You don't have to wait for months, and all the information is basically free if you don't include the cost for your Internet access.

Mailing lists:

'Computer-Flohmarkt':

Mailing list:
http://www.onelist.com/community/ computer-flohmarkt

Web forum:
http://www.ich2.de/cgi-bin/cfml/forum.cgi

GO64! mailing list:

Send mail to

listserver@giga.or.at

with the following content

subscribe go64-list Your Name

WORLD WATCH

by Rainer Buchty

This month's focus: **C64 Graphics card?**

HELLO WORLD,

if you turn to 64 users who have a SCPU (which offers them high speed and large memory capacities), and ask them what other peripherals they'd like to have before all else, one of their dearest desires will be a graphics card that displays any desired color. Of course, there are some programming tricks (IFLI, for instance), which are capable of squeezing graphics out of the VIC-II, which its designers wouldn't have dared to dream about - but somehow, the 296x200 resolution and the (apparent) 136 colors seem a bit outdated. And, although a screen refresh rate around 50(60) Hz was quite acceptable some years ago, the new ergonomics standards decree that your eyes really need scan rates beyond 70 Hz.

As a matter of fact, there are all kinds of peripheral devices for the C64/128: SCSI interfaces, memory expansions, fast serial interfaces... but there's one thing we haven't seen yet: serious graphics cards. From the beginning to the middle of the 80s, there were some 80-column video cards, but they vanished quite soon, as there was no software supporting them, except for some word processors. These cards were primarily text-only devices.

Memory usage

Ah, the old times, when a HiRes image occupied a mere 8 kilobytes... On today's platforms, image sizes exceed such values by far, and without efficient compression methods, any PC's hard disk would burst with only a few images on it. The memory size needed for a picture can be calculated as follows:

file size (in bytes) = x_resolution * y_resolution * color_depth (in bits) / 8

So, a 640x400 pixel image with 256 colors

(that's a color depth of 8 bits per pixel) already takes a whole 250 KB, and a 1280x1024 truecolor image (16.7 million colors, or 24-bit color depth - nothing special today) occupies 4 megs - for a single image!

Although these numbers seem gigantic given the nature of the C64, there are peripherals that can provide such large storage. Unfortunately, the CPUs of the 65xx family can only address 64 KB of memory at a time, which means that such large amounts of memory are usually handled by dividing them into regions (banks) which are mapped into a reserved address space in the 64. And that's where we get a serious problem with...

The Bandwidth

The C64 has been designed to run at 1 MHz, and the C128 at 2 MHz, which was quite impressive in the beginning of the 80s, when these computers where conceived (although even then, they weren't the fastest). But today, this speed is a real skid, since it doesn't only limit the clock frequency of the CPU, but also the maximum data rate for accessing the RAM - however, this holds only under certain circumstances (Attention, rocket science coming up):

- The 64/128's CPU can access the bus only during the high phase of each Phi2 clock cycle, the other phase being used by the VIC. Sometimes, however, the VIC needs even more cycle time, which it simply "steals" from the CPU - we're talking about the beloved badlines that occur every 8 raster lines. An external DMA device (like the graphics card we'd like to have) has to know about this particularity, and how to handle it - otherwise, we'd have to disable the VIC each time we start a DMA transfer. Due to this restriction, the full 1 MHz (a bit more or less, depending on whether your C64 is a PAL or an NTSC model) is not available to us.

- Assume we'd disable the VIC to allow for a full-speed DMA. Then, the transfer speed will be limited by the RAM itself, which typically has an access time of 150 ns, which

equals a maximum data rate of roughly 6.7 MHz; if we want to do DMA's to a peripheral chip (e.g. the SID), this rate will drop to 2 MHz.

- DMA is a great idea, since it allows for an "intelligent" data transfer that doesn't need the CPU's cooperation. However, a bus is only a "one way street", because even though the CPU doesn't have to do the data transfer, it's blocked during the DMA, unable to do anything. At worst, the CPU could do nothing at all, because an external device keeps on pumping data into the 64 via DMA.

For real application, this means the CPU should not be interrupted by DMA requests too frequently, if it is to do any serious work between them. To see how grave this can get, take FLI images as an example: to obtain an FLI display, a badline is generated in every scan line. On the one hand, this method allows for display modes not known before; on the other, the computer becomes extremely slow. (For our historians: displaying software-hires on the ZX81 was similar, generating a video interrupt in every scan line instead of every eighth - by doing so, we got a resolution of 256x192 pixels, at the expense of computing speed, which is cut to about one-eighth normal speed.)

Bandwidth eater

Remember, an image with a 640x400 resolution and 256 colors already uses 250 KB, which means at a 1 MHz bandwidth, it takes the quarter of a second to transfer this image into an external frame buffer - which is a well-noticeable timespan for man: if we watched a film running at 4 frames per second, it would rather look like a series of stills than an animation. For a smooth gameplay, a frame rate of at least 10 frames per second would be recommendable, and 24 frames/second is the usual frame rate at which movies are played. Well, you can easily verify that we simply don't have such bandwidth. Is it possible by any other means?

Alternatives

Surely, the most simple solution would be to use an external graphics adapter as a mere frame buffer (a buffer for single images). This would suffice for all those who just want to use such a graphics card to display stills at high resolution with the "correct" colors, instead of scrolling FLI images with dithered colors. For this kind of use, it would be acceptable if transferring an image took several seconds, especially since most of the time would be needed to load the image from external storage anyway.

A further possibility is to disable the VIC, allowing for DMA transfers at full speed: data rates up to 6.67 MB/second for direct RAM transfers would be in reach. For this purpose, the graphics card would have to be equipped with a clock multiplier that generates a suitably higher clock rate from the Phi2 clock. This method would surely give us an advantage in that we could transfer six times the data during each DMA cycle - however, it's debatable if it really made sense: your standard external RAM expansion can only transfer up to 1 MB per second, which means that only a small portion of the expansion currently visible to the 64 could be transferred at maximum speed. From this point of view, the whole effort would be useless. Also, with this solution, we couldn't use two screens at the same time.

The "old school" method

At this point, it seems there will never be a useful graphics adapter for the C64/128; however, there's a possibility we haven't regarded yet, although it's at hand, if we think of the Commodore history:

What do the Commodore floppy drives, a SCSI interface and the SCPU have in common? As a matter of fact, all of them are complete computer systems of their own that take work off the 64 or make their computing capacities available to it. So, why not apply this idea to a graphics card?

This idea isn't new at all: Professional DTP systems use the so-called Display PostScript where PC gamers have their counterpart with OpenGL. With these standards, an image is not sent to the graphics hardware pixel by pixel, but as a textual description of the contents on screen. This would certainly be the solution for the C64/128, since there would be no more need to transfer large amounts of data. For instance, drawing a line would not require drawing it in a bitmap and

transferring the bitmap; instead, a simple L(0,0,639,399) command to the graphics card would produce a diagonal line across the screen (which would be four times as large as on a standard 64).

There are no limits to creativity: if, for instance, the external graphics processor used could display sprites, we wouldn't need to hack registers to influence them (VDC programmers know about the joys of indirect access, right?), instead we could simply implement a sprite movement command in the communications protocol. And should we need to transfer an entire image once in a while, this could also be done with a special command - taking a longer transfer time, of course.

This solution would offer another advantage in that we wouldn't have to care for the slow clock rate of the 64 - the graphics processor itself would run at its preferred clock speed (which could easily be above 150 MHz nowadays), since it would be controlled by an equally fast CPU, which would communicate with the 64/128 via its normal (slow!) interface.

The crux of the graphics chips

Unfortunately, it seems as if all these approaches were doomed to fail, since the times when you could buy a powerful graphics chip at the Radio Shack around the corner are definitely gone. Today's graphics bolides aren't available at low prices, unless you order several thousands - and they're equipped with a PCI or AGP interface for

data exchange. However, this interface is much too complex; too difficult to handle for hardware forges aimed at the C64. Older graphic ICs (like the 10-year-old Trident-4000) might still allow for a quantum leap on the field of C64 graphics, but those are unavailable today, at best you can find them on the second hand market as ISA graphics cards. And the computer around them (e.g. a 386 PC) is usually for free...

The PC as an intelligent graphics sub-system

... and so, the last alternative seems to be (ab)using this PC as a graphics sub-system. Communications could run via the expansion port of the C64/128 and the PC's parallel printer port (LPT): there are more than enough peripherals that connect to the PC using the printer port, showing that it makes for a universal periphery bus; even on the 64/128 field, this is a well known method, just look at the X1541 cable that enables the 1541 disk drive to communicate with the PC's printer port. So why shouldn't we use it as a graphics bus?

At this point, let's refrain from asking if a C64 or 128 can still be considered an actual C64/128 if it had such extensions: the same question could be applied to the SCPU or the SCSI interface. And, finally: wasn't it always part of the Commodore philosophy to make the peripherals more powerful than the computer?
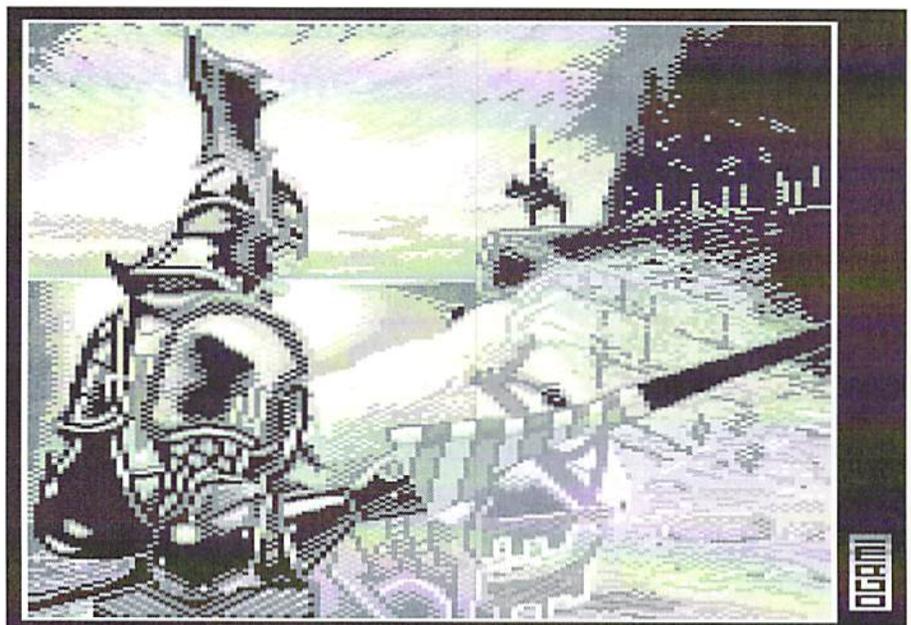
Will a graphics card take C64 graphicians to new frontiers?

Report

Newcomer 2000

by Volker Rust

Picture: Sebastian Bach

In times like ours where new game releases have become rare, it may sound unlikely, but nevertheless it is true: a flood of new games is waiting to wash over all C64 fans. We took a look around at the most active software companies and now we are going to show you the first promising pictures.

CID

A s early as the mid- 90s the C64 magazine wrote a report, illustrated with impressive screenshots, on a project that set out to define new limits: 'Newcomer' by CID (Cinematic Intuitive Dynamix). The game character is deported to a prison island because of a murder committed out of jealousy. But unfortunately the role-playing game, styled after 'Bard's Tale', was only released in a Hungarian version. A little later an English-language preview spread in the scene, announcing a translated version with further added features. The superb intro with numerous animated full screen graphics and a gameplay that apparently offered enormous possibilities raised the hopes and expectations for the complete version but

Moon-struck?

nothing was heard of it for months on end. After four years the expectations for the hit game had nearly completely evaporated. Then the development crew reported back on their Internet homepage, announcing new superlatives (see box).

Considering the size of the game it's no wonder that CID has been working on it for seven years (!) now. They are no exception: the development of modern game projects takes up so much time because programmers do not exclusively develop C64 games anymore. Nevertheless everyone who wants to compete with the highlights of the past needs a massive amount of time for the development phase in
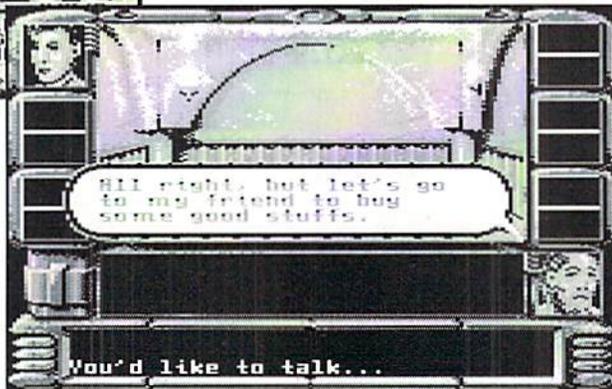
order to create a product that is as perfect as possible.

CIVITAS

Since 1996 the Civitas Group has been working on 'Utopia', a strategy game with lots of details where players have to direct the fate of a whole country. There are numerous opportunities to influence the events, from constructing power plants to defending the country from warlike neighbors. Bills have to be passed and the ecosystem has to be protected. At the same time players should not forget to care for the people of the country. All those who might find the broad spectrum of possible actions too confusing can get an overview from the integrated online help.

The simulated DOS interface at beginning leaves positive impression; hardware that is supported will be recognized. The exclusive sound track by Taxim surely is worth waiting for. Unfortunately, for the time being there are only plans for a German version.

Until now, two playable previews of 'Utopia' have been released. As the coding has reached a rather advanced state in the meantime, we are hoping for a release soon. Various scenarios will be supplied together with the game; nevertheless, the producers already plan additional scenario diskettes so that boredom is


A newcomer makes headlines

Newcomer: Shady deals
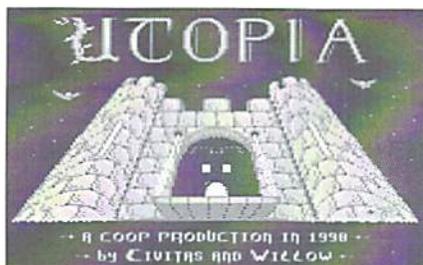
Utopia: Simulated DOS interface

'Heroes & Cowards': controlled solely through menus

The menu-driven mode in 'Utopia'

not an option even after paying the game ad infinitum.

BYTERIDERS

After the release of the extraordinary 'Brubaker' adventure on the Golden Disk in 1992 nothing was heard of the Byteriders for a long time. Then in the GO64!'s interview in the 2/99 issue it turned out by chance that Sebastian Broghammer and Steve Kups had another adventure lying in their disk box that was nearly finished, but not released yet. So the C64 scene could look forward to a new game by the Byteriders. The time could be near now: we have received a version of 'Heroes & Cowards' though it still suffers from some bugs. Passionate adventure game fans will be enthusiastic: the story again is funny, it has some rather difficult but all in all logical mysteries. Small but magnificent graphics by Phil Bergmann (also responsible for pixels in 'Logan' and 'St?mp') and a great soundtrack by Stefan Hartwig make 'Heroes & Cowards' a real treat for all players.

PROTOVISION

We are still waiting for 'Metal Dust', the phenomenal 'shoot'em up' game for the SCPU,

but there is a good reason for that: The mastermind of 'CLiPS', Chester Kollschen, has taken over the responsibility for the project's programming and technically updated the parts that were already coded before. Now the game has been optimized for the SCPU and the computing time gained will be used to make it look even greater. On the other hand the idea to keep the game compatible for the Flash8 acceleration card in the future fell flat. At the moment the graphics maestro Stefan Gutsch (Big User), who amongst others also contributed the title graphic to 'Bombmania' by Classical Games, designs the third level.

SMASH DESIGNS

And even for those C64 fans who have to do without an SCPU a new 'shoot'em up' orgy is waiting. AEG from Smash Designs continued with a new game project directly after he had

finished the great 'Katakis' clone 'Crush'. In the new game, a steel hero fires his way through worlds with the most gentle of scrolling. The working title is 'Turrican 3'. After Chester Kollschen had left the project (see our report in 12/98), AEG took over the programming of the game. In fact, even Manfred Trenz would have been astonished: the game is even faster than his 'Turrican 2' with music during the whole game, which is a feature considered not feasible by most programmers for a long time. We can only hope that AEG will put care in that the final result will also convince through its gameplay. The graphics are actually in part the responsibility of Big User from Protovision. Those who want to get further information about the project have to remain patient until the big Mekka Symposium held on Easter: Smash Designs will probably bring with them some new information ...

STARDUST PRODUCTIONS

For all friends of decent Jump'n'Run games who feel that after playing 'Giana Sisters' and 'Creatures' for 700 times they should set out for new frontiers: Stardust Productions is working on the second 'It's magic' game. It excels its predecessor in all respects although even that was very well done. There is a higher number of adversaries and five worlds with six levels each, the design was improved on, for the first time mysteries were included in the gameplay and music plays during the whole game - what more can you ask for? Programmer Andre Zschiegner says that large parts of the game are finished as far as technical matters are concerned. At the time being the producers are still working on the level design and the final adversaries. For those of you whose fingers already have started itching: Stardust Productions have offered us an exclusive playable preview of 'It's magic 2' for our present magazine disk.
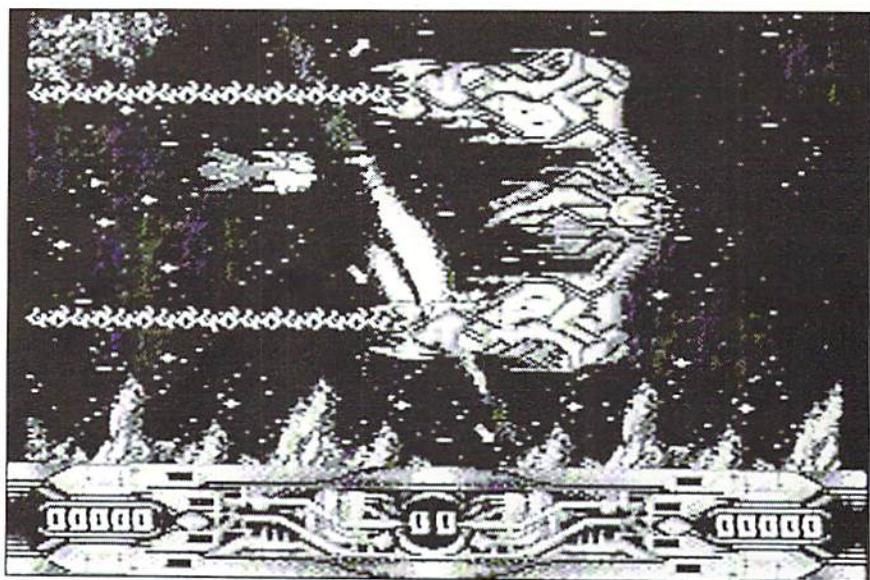
Metal Dust powered by CMD's SuperCPU

It's magic 2

OUT OF ORDER SOFTWORKS

Out of Order Softworks, too, started a new adventure project immediately after finishing 'Leisure Suit Leo 2', and they have set their aims very high. In this game, Roy Sheldon is a journalist dogged by bad luck and left stranded by a car crash in a small town in the UK, where he takes up private investigations to solve a mysterious murder case. There will, of course, be the usual complicated Larry-Laffer-like encounters with the female sex and peeping is definitely allowed.

This time the producers went without using the D4-Adventure system and engaged two programmers at the same time for the main programming: Dizzy Devil from Out of Order and Carlos from Cascade. The control of Roy Sheldon was developed from the Lucasfilm adventures' "SCUM" system and can also be controlled with the mouse. The inventory displays small high-resolution pictures instead of object names. In addition to the usual adventure gameplay, players will be able to have long, interactive conversations with complex animated characters. Both German and English versionsof the game are planned.

'Legend of Kyril': Status of project: Unknown

Rumor has it ...

Concerning some other new game projects we have no exact information about how far the programming is done. For example Digital Excess, known for uncountable 'Game On' and 'MagicDisk' productions like the fantastic 'Hi Ska Do', have recently started programming for the C64 again. At the Symmek 1999, the programmer Thomas Koncina showed us the first pictures from the successor of

'Bomberman' clone 'Detonator', and especially the gigantic explosion effect left a very favorable impression.

We are still waiting for the long announced 'Legend of Kyril' game by Taurus Software, too. Latest rumors claim that the programmer's military service prevented a quick completion of the project. Other game projects we were excitedly waiting for, such as the 'Zone of Darkness' adventure and the space shooter 'Rigor' seem to have been completely put on ice by the various teams.

When exactly will they be released?

This article is consciously trying to avoid giving exact release dates for the games. The reason is easy to see: even the programmers can only answer in rough estimations. The only answer coming with any truth or guarantee concerning

modern game projects is 'as fast as possible'. As programmers, graphics designers and musicians are normally paid next to nothing and only very few people are willing to help develop extensive game projects, it is hard to fore see the exact date of completion and of release. Moreover, the motivation of programmers is further lowered by the extremely low sales figures of the games. Even though there are still a few thousand players playing with the C64, very few games manage to sell more than fifty copies. That's not a very (financially) rewarding amount of money in comparison with the amount of work such a production takes.

But nevertheless we can hope that 2000 is a good year for games. Of course we will keep you abreast of the latest news concerning the projects mentioned above as well as any other new games. One final remark remains to be made: Don't copy games, buy them!

We would like to thank Jörg Dröge (Nafcom/ P.O.L./The Stock) for his help and support.

Roy Sheldon's last voyage?

First new buddies: Roy meets British beef
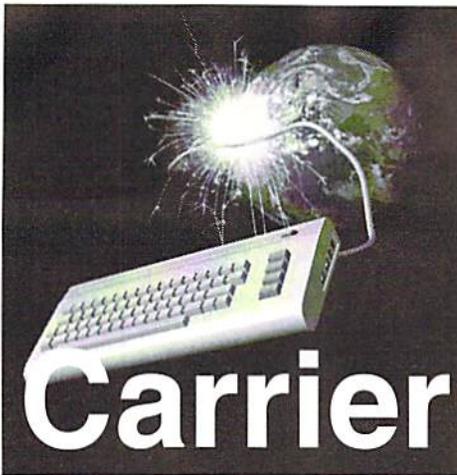
CarrierDetect

Picture: Sebastian Bach

Crystal Ball Gazing: The Internet and our Commodores

by Gaelyne R. Gasson

As the theme of this issue has to do with the future, I've been asked to expound on what the future holds for us in the telecommunications area. Hopefully there are a few programmers reading this, as I'd love to share some ideas about what I think of in terms of the ideal Commodore Internet software suite... Come along and dream with me...

The sky is the limit!

The fun thing about looking forward is that we can strip off any limitations that we (sometimes) add to projects. This is good, because some of the best Commodore products have come about because "impossible" wasn't part of the vocabulary of the designer.

Web Browsing with a Twist

Development of Maurice Randall's Wave is discussed in another article, so I won't talk about it too much here. Instead, let's take a look at the possibilities that having a web browser run on the Commodore system can bring.

Something that I haven't seen mentioned is the ability to add features to our own web browser that benefit us as a community. Since the web browsing software is on our computer, anything it can do, we can make

use of on web pages developed for the Commodore community. One of these might be Commodore graphics viewers built into the browser... not just the boring GIFs and JPGs from other computer platforms, but fun things like Doodle, Koala and even GeoPaint files.

Having our own Webcam is certainly in the realm of possibilities, especially since most of the hard work for it has already been done by Nate Dannenberg. He's created an interface for the Connectix Quickcam (http://www2.southwind.net/~natedac/quickscan.html). Many of the online webcams are simply still images that are captured every few minutes and saved to the web directory, and Nate's device could certainly be used for this purpose.

Listening to music as we browse. Other platform browsers are able to play sid, midi and wav files, and there's no reason why we couldn't have this ability as well. There are a few sites on the web that have SID music files set to play when someone browses the page, but it'd be so much cooler if we were listening to the SIDs from our own Commodore system - especially with the stereo SIDs.

Display other types of files. When he's not developing Wheels, or the Wave, Maurice Randall also develops GEOS software for working with PostScript files. Without wishing to add too much to his work load, a way to view PostScript files from the Wave would be a neat addition.

Design our own web sites. Some lucky members of the Commodore community are already using the Wave to view their web pages before uploading them to their Internet provider. The next step would be to be able to upload the web page to the web directory right from the program used to view it (or create it). This is something that's built into many "other platform" web design software

and is a great time saving feature. As well as software to develop web sites, there's a ton of other smaller software to help web masters, such as color pickers, HTML validators and programs to tidy up our code.

Web server software. If we can develop a web browser and edit our own web pages on the Commodore, it stands to reason we should be able to have a web server as well. Especially as a server takes less resources than a browser does. Ideally, this would require a 24 hour connectionto the Internet, or the server would only be available when it's placed online (similar to how some BBS's are only available during evening hours). Having our own web server would let us develop web based application an not be limited by other operating systems. In my "day job", I develop web sites and applications for the web, and one of the tools I use is a "web preprocessor" which allows me to make use of the computer operating system to do things such as count, compare and make decisions based on "if" statements. It's very similar to programming in BASIC. It's fun... but it'd be even more fun to develop Commodore web based software. For instance, with our own server we could have fill out forms that asks questions of the visitor and creates a Commodore file for download based on the answers given.

Email

Naturally, we'll need a Email client (that's the program that fetches your mail and displays it). Some "other platform" Email clients have a nasty habit of sending Email in HTML format (the same format as is used for displaying web pages). Hopefully our Commodore Email client will handle this by either saving the mail to display in the web browser, or will display it without the HTML coding as Pine currently does. Two other features I'd like to see are address books, not just for Email addresses though. Many times we receive Email with web URLs in them, and it'd be a nice addition if the address book also stored the URLs so if I need to find Joe Shmoe's web site and he's sent me mail at any time all I need to do is access the address book and select his name. It could then give me the choice of either sending him Email or starting the web browser so I could browse to his home page.

FTP

FTP stands for "File Transfer Protocol", and if we can browse the web, browsing FTP sites is also a possibility, more-so because graphics aren't required. It can be built into

the web browser, but it can also be stand alone as well. Hopefully it would store FTP sites we've visited in a history file so we could easily return to places we've been to before. One of the UNIX shell FTP programs I use is called "ncftp". It can automatically connect to an FTP site and send the word "anonymous" for the login name and my Email address for the password. This is a feature I'd love to see a CBM FTP client have.

Chat Clients

Having our own ICQ client would be nice, as it would allow us to chat with other users who also use ICQ. We could also have our own "AIM" (AOL Buddy) chat client, and IRC software is certainly possible, since one is included with Daniel Dallmann's SLIP demo. Programmers interested in these types of software might be able to get a head start by taking a look at the Open Source software developed for the Linux operating system, since most of the information required would be included. As these are completely text-type protocols, it should be interesting to see the types of user interfaces developed for them.

Telnet Clients

A telnet client is almost exactly like a term program, except that instead of dialing via the phone line, the software connects to other systems using the Internet. A telnet client would need to have many of the same features as a term program does, such as x, y and zmodem protocols, as well as a buffer so you could buffer text and transfer files.

Call for Diversity

A friend of mine has been insistent on not "redesigning the wheel", and wishing to combine his tcp/ip programming with existing online software such as a terminal program. I've done all I can to get him and a few term program programmers together, but at the same time I think we could be losing a lot if we only stick to one way of doing things online. As a few have noted, the Wave (and Wheels) requires extra hardware that not everyone is able or willing to add to their systems. We need software for both ends of the spectrum and in the middle as well. Look at how many word processors we have for our computer... each have their own particular features and benefits and I think it'd be a shame if we only had one or two choices. A quick look at a list for Email clients for the Windows operating system lists 27 different

programs that each have their own features and 'flavour'. The same is true for other operating systems such as Linux. I honestly think it's this diversity that makes it all worthwhile. If you're a programmer thinking about developing an Internet application, don't assume that just because a program already exists that you shouldn't try to develop one of your own. Freedom of choice is a wonderful thing.

A note to programmers listening in... one thing that should be kept standard is the tcp/ip stack - one stack should fit all applications, GEOS, Basic, Assembler, the lot. This is how other operating systems have managed to have such diverse software (with rare exception), and is probably one part of the wheel that doesn't need reinventing.

Don't despair if some of the software under current development is beyond your grasp in respect to the hardware required. Many of the utilities CAN be scaled down later.

Impossible? Heck no. Much of the UNIX shell software in use on daily basis have very small requirements, so there's absolutely no reason why similar software couldn't be made for our Commodores. Nothing is impossible!

Gaelyne R. Gasson is the author of The Internet for Commodore C64/128 Users. You can find her online at http://videocam.net.au/gaelyne/
(ad)

Uncle Wanja's Mailbox

After an extended delay, the other editors forced me into working through my pile of letters... But the letters I found were so good I almost have to thank them for it, as I had nearly forgotten what fun a letters column can be. Anyway, right now I'm drooling again, waiting for your new letters to arrive. One thing I'd particularly like to hear about, is what the Commodore World readers think of our magazine!

The first letter this issue comes from a guy who practically lives in my neighbourhood, and it won't be long before I meet him at my university. He's planning something that surely doesn't only interest me, but everybody with a foible for nostalgia...

Dear GO64 team,

Recently, some friends and I have initiated a project. It's about the production of a CD and putting up a web page with similar contents, namely a complete history of video games. There are already several pages with information, images and technical facts on more then 800 game machines, and a game database containing almost 24000 entries. Today, almost 60 persons from all of Germany are taking part in the project, writing game reviews, or articles about people like Sid Meier

After searching the net, your page seemed like the best place to turn to, as we're still searching for game experts on the C64. Although I hardly missed a game on the 64, I don't have the time for it, now that I'm the head organizer. So we're looking for game connoisseurs who could get enthusiastic about the project, and write reviews to C64/C128 games (C16 also) for inclusion in the database. It would be very nice if someone in your team matched this description, or if you'd present the project on your web page and maybe win us some participants. I'd like to emphasize that it's a non-commercial project, that we're not making money of it.

Of course, it would also be helpful if you knew of a club or somewhere I could turn to. By the way, the project's entitled "Spieltrieb" (play instinct) and can be reached on the net under 194.64.245.10, and will later be reachable at the domain "www.spieltrieb.net". I'll include a link to you on the Spieltrieb pages. PS: the pages may not look that great, as they're still under construction, and don't have their final design yet - but we're improving.
Ciao, and thanks a lot, Klaus Brandhorst (klaus.brandhorst@uni-bielefeld.de)

Hello Klaus!
Sounds like an interesting project (thanks for linking to us!) and if I find the time you can be sure I'll write some article about the good old C64 to include on your web page. Also, I'm sure that GO64!/Commodore World was just the right place to turn to - there must be lots of C64 users who still have a ZX81, CPC, Amiga, Acorn or an Atari, and rest assured that there are plenty of game experts in our readership! Furthermore, I'd like to point you to http://www.c64.com, where you'll find lots of information on C64 games. I wish you good luck with your project! PS: Any readers who'd like to support this project, can turn to me, or send us an email.

The next letter is a bit old, but I'll bring it up anyway, as I suppose most readers haven't heard about the event it tells about:

Greetings Wanja!
Whoa man, whoa! I'm still totally out of myself. Even though it was some time ago (in October 1999 -Ed), I must say it really was a surprise! I'm talking about your radio appearance at FRITZ! How cool! For all those who don't know Fritz: it's a radio station located in Potsdam in Brandenburg. On Monday to Friday evenings, they have a show called "Blue Moon". "Blue Moon" is a sort of forum aired between 22.00h - 1.00h, that focuses on a subject the callers can discuss. And recently, they were talking about the C64, opening the show with a talk with Enno Coners. Interesting indeed! They also played some musics, from the studio as well as by callers through the phone. It was a real surprise, and man am I proud of my good old 64! What a fantastic idea! Thanks a

lot! That's it for today! One for all! All for the 64!
- Toni Neumann -

Hi Toni!
I cursed myself for not living in Berlin, when I heard about Enno's appearance on the radio... I'd really have liked to hear that, too! Unfortunately, the radio station didn't send us the copy of the show we asked them for; otherwise I could have listened to it (therefore, if any of you readers has taped the show, we'd be glad if you could send us a copy!) The alternative radio station at the university of Davids/California (KDVS, on 90.3 FM, and on the Internet at http://www.kdvs.org) has a very special offer of a similar kind: "Hard Hat Mack" regularly features the "SID6581 Radioshow", playing exclusively C64 musics from the years 1982 through 2000!! If you have the Real audio plugin installed in your Internet browser, you can listen to the show live on the net. According to the current programming (also found on the web page), it's on Tuesdays from 8 pm - 9 pm local time - that's 5 am in Germany, just right for breakfast! While we are at it: the next letter comes directly from the states, from a former subscriber to Commodore World:

Hello GO64!,
I received, today, my first issue of your magazine. I was impressed by the format as I slid the magazine out of the mailer (and not by just the cover photo). Having read all the articles (I am a quick reader!) I will say I think I will enjoy future issues. I will "spin" the disk later. One matter bothers me, however: The absence of US advertisers (except CMD). Do you plan to entice, at least, the few who were in Commodore World? I would presume CW's list of subscribers was large. Those readers will want to know "who's around" in the US, and where they can get "stuff". Commodore users WORRY about those things. Well, I anticipate more interesting articles. Good luck!

- George T. Gallagher -

Hi George!
Glad to hear that you like our magazine. In

Are there any requests, suggestions, complaints, or comments about the GO64!/ Commodore World Magazine, or do you just feel the urge to chat a bit with our own "uncle Wanja"? Just send us those letters, before Wanja's mailbox starves!

CSW Verlag
Key-Word: Wanja's Mailbox
Goethestraße 22
D-71364 Winnenden
Germany

The faster way is by e-mail:

w_gayk@bielefeld.netsurf.de

or

brix@plush.de

Wanja's homepage: **www.plush.de/brix**

fact, we're trying to get as many writers of the Commodore World into the new team as we can, in order to make sure the former readers of the CW have as much fun with the new magazine as with the old. Therefore, we really appreciated that Gaelyne Gasson was the first to join us, and also kept on providing her "Carrier Detect" column to our mag. In the meantime, Jason Compton ("Just for Starters") and Bruce Bruce Thomas ("Graphic Interpretation") are also working with the team, and we're looking ahead to working with them. And yes, of course we'd like to include advertisers from North America, the UK, Europe or from Australia in the magazine. All those who are interested, please write to the publisher's address for further information, or send an email to go64@go64.de. Another possibility is or "Breadbox Oases" column, where sources for c64-related hard and software are showcased from time to time. And that's where you - all our readers around the world - come in: it's up to you to look around and send us short articles about the breadbox oases you can find! Please include a photo, if possible. For anything concerning this matter, you can also turn to us via email, or by writing to the publisher's address. Remember, the more help we get from you readers, the better your and our GO64!/ Commodore World is going to be! - Now to something completely different: the following letter is a little bit weird, but still not too hard to interpret. Either this reader has a really deranged attitude towards women, or he's got an abnormal preference for those tree-wetting creatures. It's up to you to decide which of the two hits the root of the matter, or whether I'm totally wrong:

Howdy letterbox uncle,

With this letter I'd like to inform you and the other younger readers and makers of the GO64! about how it all began in the beginning: God created man. And he saw it was good. Very good indeed. So good god might have been absolutely content. And see, it was great! And in the image of God he created man. And since it was a man, God must be a man, too. And no man is perfect. Instead of leaving it as it was, after, maybe, inventing football, sweat-absorbing socks and cleavage for mammals, God was being a little too enthusiastic about creating. The Lord God said, "It is not good for the man to be alone." Without any doubt, a thesis man has to agree to. Likely, our ancestor felt similarly, and was looking forward to the coming togetherness. Surely, he'd already thought up a name for the companion, and was pondering if he should first teach him to "sit" or "go". Being thus naive should cost him a rib and his peace. In exchange, he got a woman. Until our day, it couldn't be finally cleared whether this was a fair tradeoff or not. God, it seems, had a bad conscience, as man still got his dog later.

-Oliver Schmidt-

Oliver, I have no idea if the church would include this creation story in their books - but could it be your girlfriend said something against the C64, while your dog barks approvingly when you load "Create with Garfield" to build a picture showing Odie giving greedy Garfield a kick in the Butt?? Maybe he's even enthusiastic about a hamster blown up by a microwave in Maniac Mansion, while the "dumb girls" complain about animal misuse? Am I totally wrong? No matter what - there's an urge for action: herewith, I, the approved Prof. Dr. letterbox uncle Wanja, prescribe you three hours of Samantha Fox Strip Poker, to help you get back to healthier thoughts... If that shouldn't help, just play Infocom's adventure "Hitchhiker's Guide through the Galaxy", and forget the woman and the dog until you've finished it (my personal favorite in crisis coverage)... Then, when you've stopped laughing, write another letter. With Douglas Adams' humour, this could take a while.

In the meantime, let's have a look at the last letter for today (which reached me shortly after Christmas):

Dear Wolfram, Volker, Sven, Nikolaus, Helmut, Wanja, Arndt and Staff,

Such an aggregate of talented people in one place !! I am delighted and enlightened by your great work on computer science in general, and on the Commodore equipment in particular. You light up my life. I believe that the latest issue you sent represents my twenty-fourth magazine associated with Commodore World, and I am supposed to be paid up through magazine thirty-one, through July of year 2000 if you send one each month. But I can see that your cost is much greater than CW's, so please keep me on your mailing list and please let grampa know when you want more money. I don't want to lose track of you, and if there is anything to reincarnation, I want all of us in the same family, headed up by my favorite composer, J.S. Bach. A long time ago I lived in Germany, in Landstuhl near Kaiserslautern, and at one time I knew the German language. It has all slipped away, so forgive me for not being able to handle German. Your English is remarkably fine. Thank you specially for making your educational work available to those of us limited to English. One pressing question: Where is WINNENDEN ?? I can't find it in my three supposedly comprehensive atlas books. If it is not on maps, it is probably a wonderful, kind, compassionate, excellent place. My heart is filled with love for you. Have a fine Christmas and New Year, and God go with you always, in everything you do.

- Love Always, Grampa George
(Ptahseti@aol.com) -

Hello Grampa! First, let me say a big "Thanks" for your motivating words! The reason why you can't find Winnenden in your atlas is the fact that it's a rather small place in the vicinity of Stuttgart. An excellent place? Maybe not... or maybe it is, as it's where the GO64!/Commodore World is sent out into the world. I'm planning on an article about us, so the readers of the English release get to know us a little better. It often seems to me that some people have an absolutely hyperbolical image of us in mind; since at heart, we're just a wild bunch of 64 enthusiasts, as you all are. Surely, GO64! wouldn't exist if we weren't. But that's a different article. I think I speak on behalf of the other editors when I wish you and all other readers much fun with the coming issues of GO64!/Commodore World.

See you next issue!

Yours,
-Uncle Wanja-

CMD WAREHOUSE BARGAINS
1-800-638-3263

CMD's Warehouse Bargains include many hard-to-find used and (where noted) new items. In most cases we have only one or two of each item in stock, so be sure to place your order before your selection has been taken! A minimum order of $20.00 applies to all Warehouse Sale orders. Shipping charges are not included in the prices shown. (*See our main advertisement elsewhere in this issue for shipping information*).

TELEPHONE ORDERS ONLY ON THESE ITEMS

Mailed in orders for non-reserved out-of-stock items will result in a merchandise credit. No cash refunds. All items listed here are sold "as-is", and all sales are final. Prices and specifications subject to change without notice. Not responsible for typographical errors.

BOOKS
32 BASIC Programs for the C64	$8.00
Anatomy Of The 1541	$8.00
Basic Computer Games	$6.00
C128 Computer Aided Design (new)	$8.00
C64 Programmer's Reference Guide	$25.00
Commodore 64 Favorite Programs Explained	$10.00
Commodore 64 For Kids from 8 to 80	$8.00
Commodore 64 Games Book	$5.00
Commodore 64 Programs for the Home	$8.00
Compute's First Book of C=64 Sound &Graphics	$10.00
Computes Second Book of Commodore 64	$10.00
Computes Second Book of Machine Language	$10.00
"Guide to Care, Feeding and Training of C-64"	$6.00
How To Use The Commodore 64 Computer	$10.00
Inside Commodore DOS	$10.00
Osborne 1 User's Reference Guide	$15.00
Superbase The Book	$10.00
The Elementary Commodore 64	$10.00
The Official Book Of King's Quest-Daventry	$10.00
Using & Programming the C64	$8.00
Using Newsroom	$10.00
Your Commodore 64-A Guide to the C 64	$10.00

BUSINESS
64 Powerpack-NEW	$10.00
Better Working File & Report (new)	$15.00
Business System- Sales Analysis Management	$10.00
Datamanager 128	$15.00
Datamanager 128	$20.00
Datamanager 2 (new)	$15.00
Easy Finance 1	$5.00
EPYX Microsoft Multiplan Spreadsheet	$10.00
Financial Cookbook-NEW	$8.00
General Ledger for the C=64-NEW	$15.00
Paperback Filer 64	$10.00
Paperback Planner 64	$10.00
Payables for the C=64-NEW	$15.00
Planner's Choice -Personal Planning System	$8.00
Power Assembler-(new)	$10.00
Practicalc 64	$5.00
Practicalc-(new)	$10.00
Practfile 64-(new)	$10.00
Receivables for the C-64-NEW	$15.00
RUN Productivity Pak II	$8.00
Silent Butler (Check Writing system)	$15.00
Superbase Starter-NEW	$10.00
Swiftcalc 128	$20.00
Swiftcalc 64	$10.00
The Consultant (Database Management)	$15.00
The Home Accountant	$15.00
The Home Banker	$10.00
The Home Manager	$10.00
The Manager	$10.00
The Word Machine/Name Machine (new)	$5.00
Valuecalc	$8.00

CARTRIDGES
Alf in The Color Caves	$10.00
Alpha Build	$10.00
Big Bird's Special Delivery	$10.00
Bubble Burst	$10.00
Dragonsden	$8.00
Facemaker-(new)	$10.00
Heskit	$10.00
Kids on Keys-(new)	$10.00
Kindercomp	$10.00
Lemans	$8.00
Linking Logic	$10.00
Mach 5 Cartridge	$15.00
Mastertype (new)	$10.00
MS Pac-Man	$8.00
Pinball Spectacular	$8.00
Radar Rat Race	$8.00
Sea Speller	$10.00

Speed Bingo/Math	$10.00
The Final Cartridge	$15.00
Up & Add 'em	$10.00

CARTRIDGES (No Docs)
Blueprint	$8.00
Clowns	$8.00
Ducks Ahoy	$8.00
Kindercomp	$8.00
Lemans	$8.00
Linking Logic	$8.00
Magic Desk I	$8.00
Maze Master	$8.00
Memory Manor	$8.00
Popeye	$8.00
Speed/Bingo Math	$8.00
Up & Add 'em	$8.00
Vic 20 -Hesware-Gridrunner	$5.00
Vic 20- Avenger	$5.00
Vic 20- Jupiter Lander	$5.00
Vic 20-Omega Race	$5.00

DESIGN
Awardware	$8.00
Delta Drawing-(new)	$6.00
Jingle Disk-NEW	$6.00
Printer's Devil for Print Shop	$8.00
The Christmas Kit-NEW	$10.00
The Newsroom	$10.00
The Newsroom (new)	$15.00
The Print Shop	$15.00
The Print Shop Companion	$10.00

DISK UTILITY
Commodore 64 Software Bonus Pak	$5.00
Disk Utilities	$5.00

EDUCATIONAL/CHILDREN
Computer Science	$5.00
Early Learning Friends	$10.00
Easy Lesson and Easy Quiz	$5.00
English I	$5.00
ENGLISH II	$5.00
First Men on the Moon -Math-(new)	$6.00
Grade A Math	$5.00
In Search of the Most Amazing Thing-(new)	$6.00
Jungle Book Reading-(new)	$6.00
Koalagrams Spelling -use with Koala Pad-(new)	$6.00
Logic Builders	$6.00
Math Busters	$8.00
MECC Expeditions	$8.00
MECC Odell Lake	$8.00
Paint-A-Rhyme -for use with Koala Pad-(new)	$6.00
Peter Rabbit Reading-(new)	$6.00
Snooper Troops Case #1 (new)	$6.00
Snooper Troops Case #2	$5.00
Snooper Troops Case #2 (new)	$6.00
Snoopy's Skywriter Scrambler	$10.00
Story Machine	$8.00
SUNBURST-Memroy Building Blocks	$10.00
Tales of Mystery	$8.00
The Argos Expedition	$8.00
Up for Grabs-(new)	$6.00
Weather Tamers	$8.00
Where in Europe is Carmen San Diego (new)	$20.00
Where in the USA is Carmen Sandiego (new)	$20.00
Where in the World is Carmen San Diego (new)	$25.00
Wizard of Words	$10.00

GAMES
20 Solid Gold C64/128 Entertainment Hits	$10.00
20 Solid Gold Hits -5 Disks	$10.00
Aliens	$4.00
Amnesia	$6.00
Archon	$10.00

Ardok-The Barbarian	$6.00
Batman	$10.00
Batman-The Caped Crusader	$10.00
Beach-Head	$6.00
Below The Root (NEW)	$10.00
Beyond Zork-NEW	$6.00
Blood Money-NEW	$6.00
Blue Angels-Formation Flight Simulation	$10.00
Bop N Wrestle-NEW	$6.00
Bureaucracy	$10.00
CavemanUgh-lympics	$4.00
Championship Baseball	$6.00
Championship Golf (new)	$6.00
Championship Wrestling	$6.00
Chomp (new)	$4.00
Classic Concentration	$6.00
Commodore 64C Test Pilot-NEW	$6.00
Conflict in Vietnam-NEW	$6.00
Countdown to Shutdown	$4.00
Crossfire	$4.00
Cyrus Chess	$6.00
Deadline	$4.00
Demon's Winter	$6.00
Dig Dug	$6.00
F15 Strike Eagle	$10.00
Feud	$6.00
Fist - The Legend Continues	$6.00
Flight Simulator II	$15.00
GBA Championship Basketball-2 on 2-NEW	$6.00
GFL Championship Football	$6.00
Gold Record Race-(new)	$6.00
Grand Prix Circuit-(new)	$10.00
Guerilla (new)	$6.00
Hacker	$4.00
Hardball (new)	$6.00
Heartland (new)	$6.00
High Roller-NEW	$6.00
Howard the Duck-Adventure on Volcano Island	$4.00
Invaders of the Lost Tomb-(new)	$6.00
Jack Nicklaus Greatest 18 Holes of Golf-NEW	$6.00
Jaws	$6.00
Jeopardy-Junior Edition	$6.00
John Elway's Quarterback	$6.00
Jordan vs Bird	$10.00
Karate Chop (new)	$6.00
Kings of the Beach (new)	$6.00
Knight ORC	$6.00
Kung Fu II-Sticks of Death (new)	$6.00
Last Ninja 2-NEW	$6.00
Legionnaire	$5.00
Lords of Conquest-(new)	$6.00
Lords of Conquest-NEW	$6.00
Mean Streets	$10.00
Megaplay Volume 1	$4.00
Minit Man	$6.00
Murder by the Dozen	$8.00
Navy Seal	$6.00
Navy Seal (new)	$8.00
Operation Whirlwind	$6.00
Pathworks	$6.00
Power Drift	$4.00
Predator	$6.00
Presidential Campaign	$6.00
Project Space Station	$4.00
Q-Bopper	$6.00
Qix (new)	$6.00
Questron II	$4.00
Raid Over Moscow	$8.00
Raid Over Moscow	$6.00
Rambo III	$4.00
Rick Dangerous	$4.00
Robbers of the Lost Tomb	$4.00
Sanxion	$6.00
Sargon II-(new)	$6.00
Sargon III-(new)	$8.00
Serve and Volley (new)	$6.00
Shinobi	$6.00
Silent Service	$4.00
Sky Fox	$6.00
Snow strike-NEW	$6.00
Spiderman	$4.00
Sporting News Baseball	$6.00
Star Trooper-(new)	$6.00
Stationfall	$6.00
Summer GamesII	$4.00
Superman	$4.00
Superstar Ice Hockey	$6.00
Suspended -A Cryogenic Nightmare	$5.00
Sword of Fargoal	$4.00
Tag Team Wrestling	$4.00
TETRIS	$6.00
The Last Ninja	$4.00

The Lords of Midnight	$6.00
The President is Missing (new)	$5.00
The World's Greatest Baseball Game	$4.00
Thud Ridge-NEW	$6.00
Times of Lore	$4.00
Titan-NEW	$6.00
Total Eclipse (new)	$5.00
Ultimate Wizard	$6.00
War in Middle Earth-NEW	$6.00
Wheel of Fortune	$8.00
Wild Streets-NEW	$6.00
Word Flyer	$6.00
Z-Pilot	$4.00
Zenji	$6.00
Zork I	$6.00
Zork II	$10.00
Zork III-NEW	$8.00

GEOS
Deskpack 1-NEW	$10.00
Fontpack 1-NEW	$10.00
GEOBASIC (new)	$10.00

HARDWARE
1351 Mouse w/Manual-no disk	$20.00
1702 Monitor -Color 40 Column-refurbished	$99.00
1802 Monitor-Color Monitor 40 Column-refurb	$139.00
Aprotek 1200 Baud Mini-Modem (new)	$30.00
Assorted Joysticks	$5.00
C128D Keyboards (NEW)	$69.95
C128D Keyboards-REFURBISHED	$59.95
C64 Keyboard (NEW)	$49.95
C64 Keyboard (refurbished)	$39.95
CPU64 Version 1 Base Model	$139.00
Datasette	$5.00
EPYX Fast Load Cartridge w/Manual	$10.00
Homewriter 10 Printer w/manual	$45.00
Koala Pad Only no docs	$20.00
KXP 1080I Printer	$75.00
Laser 190E Printer	$30.00
Magic Mouse	$10.00
Magic Voice Speech Module	$50.00
Multiplexer	$50.00
Okimate 10 Color Printer w/manual,disk,ribbons	$45.00
STAR Gemini 10X w/Manual	$25.00
Super Expander 64	$8.00
The Voice Messenger -Speech 64	$20.00
Used SlimLine cases for Commodore 64	$10.00
VICMODEM for VIC20	$5.00

MISCELLANEOUS
10 Assorted Commodore Magazines 1985-86	$10.00
64 Doctor	$5.00
Home Video Producer	$10.00
Instant Music-NEW	$6.00
LOADSTAR Disks#46,65,66,67,68	$10.00
Synthesound 64	$5.00
Vic 20 Cassette-Intro to Basic Part 2	$10.00

MISCELLANEOUS SOFTWARE
Aerobics	$6.00
Bobsterm Pro (new)	$10.00
Commodore	$5.00
Commodore Technology	$5.00
Music Construction Kit	$5.00
SAT Score Improvement System	$5.00
Writer/File Pak 1541	$8.00

PROGRAMMING
Better Working Power Assembler	$10.00
GEOBASIC (new)	$10.00
Hesware - Graphics Basic	$10.00
Introduction to BASIC/Part I	$8.00
Oxford Basic	$10.00
Oxford Pascal	$10.00
Simon's Basic	$10.00
Start Programming w/Gortek and the Microchips	$5.00

WORD PROCESSING
Easy Script	$5.00
Mastertypes Writer	$8.00
Mirage Concepts-Word Processor -Professional	$10.00
Outrageous Pages-NEW	$20.00
Super-Text Professional	$5.00
Supertext Word Processor	$10.00
The Printed Word	$8.00
The Write Stuff 128-NEW	$20.00
Word Writer 128	$20.00
Word Writer 6-NEW	$15.00
Wordpro 3 Plus/64-NEW	$15.00
Wordpro w/turbo Load & Save-NEW	$15.00
Writers Choice	$8.00