

CONTENTS

EDITORIAL

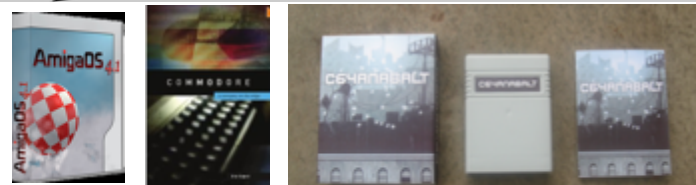
PAGE 3



NEWS

PAGE 5

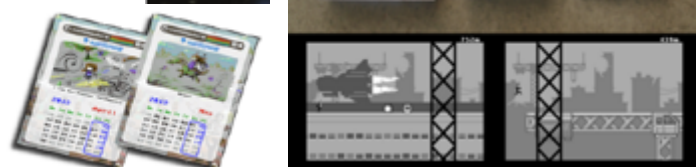
Aminet turned 20
"COMMODORE: A COMPANY ON THE EDGE"
EBOOK CHANGE IMMINENT
AmigaOS 4.1 Update 4 Released



NEWS

PAGE 7

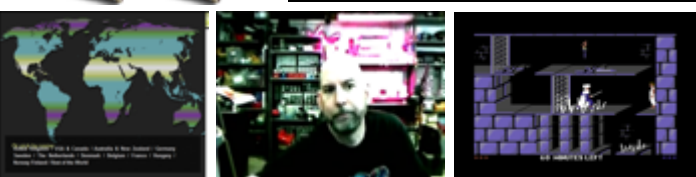
C64anabalt Cartridge Available! (C64)
1541Ultimate I & II Update 2.4
2012 C64-RetroInvaders calendar:
Mermaid graphic artist [2012]



NEWS

PAGE 8

Cracker's Map
BIL HERD SPEAKS TO FRESNO USER GROUP
Prince of Persia - Total Pack for Commodore 64: V1.0



NEWS

PAGE 11

Rear Admiral ThunderDrive



NEWS

PAGE 12

Commodore plus 4 SID card scores an own goal
mind.in.a.box
MEGABIT ROM RETURNS

NEWS

PAGE 13

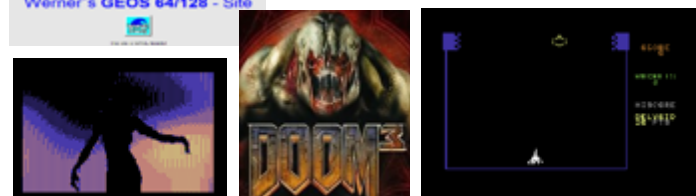
uIEC Manager for GEOS
COMMODORE PROGRAMMING BLOG
UFO DEFENCE



NEWS

PAGE 13

JUST DANCE C64 DEMO
DOOM 3 OPEN SOURCE



COMMODORE 64

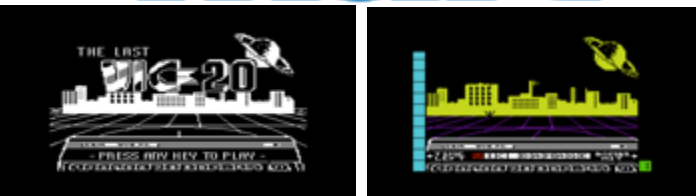
PAGE 15

ASSEMBLER TUTORIAL



COMMODORE FREE REVIEW
THE LAST VIC= 20

PAGE 20



Article on 1541 disk recovery
By Jeremy Smith BSc (Hons)

PAGE 22



RUN/STOP-RESTORE
BY LENARD R. ROACH

PAGE 25

BOOK REVIEW BY COMMODORE FREE

EDITORIAL

Christmas is over and a New Year has arrived!

If I was the kind of person to make New Year's resolutions it would be to produce a better quality of magazine. However I don't do resolutions but do try to produce something of readable quality. On that subject (quality) I would like to thank the plus 4 community for correcting an error from a previous magazine about a Sid card for the Commodore plus 4. The information had been emailed to me and was not accurate; the source was usually reliable so I didn't check the facts. Now I realise I had failed on the delivery of an accurate news item indeed one plus 4 website actually said Commodore Free failed!. This hopefully is corrected; if you read the news section of this issue you will see that a reader has tried to clarify my mistake.

I am and have been planning for some time an issue devoted to the 264 line of machines, this would be the plus4 and the Commodore 16, Commodore 116 etc., if you feel you would

like to write something about the machines then feel free to contact me.

Also this month sees a short 3 issue tutorial on Machine code programming for the Commodore 64, these tutorials were in Micromart magazine, by Shaun Bebbington. The articles have been spruced up cleaned with an oily rag and have been sent to me for inclusion in the magazine of course I do need to plug Micro Mart magazine. So here is the plug then <http://www.micromart.co.uk/>

This issue sees information about fixing disk errors, news items and some game s reviews. And finally we round off with a look at the Run/stop-restore 10th anniversary edition book. Although not strictly a review, I have tried to outline what the book is about in case it wets your appetite to read more.

Regards
Nigel (pretend editor)
www.commodorefree.com



An advertisement for justmice.com. The background is a close-up of a computer keyboard. The text reads: 'www.justmice.com' in a large, orange, stylized font. Below it, in a smaller blue font, is 'Find a mouse for your modern or vintage computer' and '*Plus Commodore components and software'.

The Adventures of

RAMSLAM

**NEW
FOR C64!**

In **Dragonland**



- Password system
- Shoot power-up
 - Tube system
 - Keys
 - Teleporters
 - Trampolines
- Highscore saver
- 15 worlds with 3 levels each



PROTOVISION

More information at www.protovision-online.com

NEWS

Aminet turned 20

Aminet turned 20, and to celebrate we are giving away 3 lifetime memberships!

Upload a program to Aminet to push it over 80,000 files and get a chance to win a lifetime membership from Amiga.org! We will be giving away three to people that help push Aminet over 80,000 files!

After you upload your program to Aminet send us an e-mail with a link and be entered to win.

<http://www.discreetfx.com/contact.html>

Hurry if you want to win because Aminet is very close to 80,000 files already.

Best regards
Amiga.org



"COMMODORE: A COMPANY ON THE EDGE" EBOOK CHANGE IMMINENT

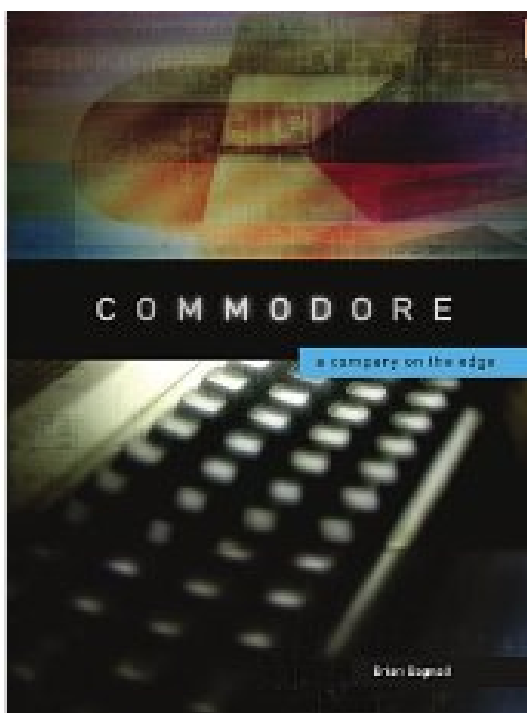
by bbagnall » Mon Jan 16, 2012 11:29 am

Just a heads up that the eBook price for "Commodore: a Company on the Edge" will be changing in about two weeks.

Currently it is listed on Kindle for \$9.99, but a distributor will be taking over duties for distributing the book across all eBook formats in the future (including the Apple book store, Kobo, Sony eReader, etc) and they will be setting the price from now on. They are discussing something in the neighbourhood of \$19.99, although usually eBook sellers like Kindle offer a large discount, so the final price might be something like \$12.99 (but that is just a rough guess).

Anyway, I wanted to warn the community so no one is caught off guard about the change. I hope everyone who has read the book has enjoyed it!

<http://www.commodore.ca/forum/viewtopic.php?f=1&t=3503>



AmigaOS 4.1 Update 4 Released

News from Hyperion

AmigaOS 4.1 Update 4 has now been released. More details and a place to download the update can be found at Hyperion's main web site.

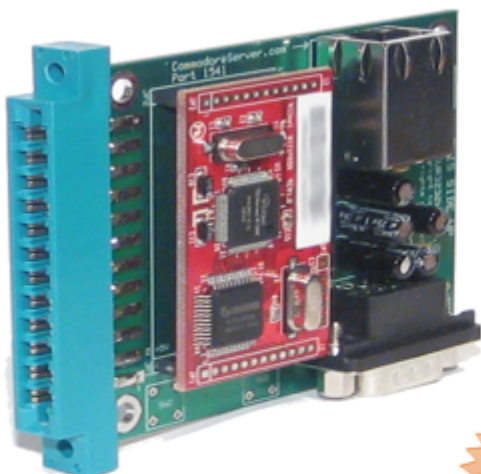
This update introduces a special bonus for AmigaOS 68K emulation fans: official AmigaOS 3.x ROMs and Workbench files. We plan to keep adding more files and helping to improve the emulation experience further in future releases.

Don't forget, for direct customer support please go to Hyperion's support forum.

<http://www.hyperion-entertainment.biz/>



COMMODORE INTERNET MODEMS ARE HERE



starting at
\$66⁹⁹

AVAILABLE IN 2 OPTIONS:

- **INTERNET ONLY**
Connect to Internet Services from a stock Commodore computer.
- **INTERNET +RS-232**
Same as above, but includes a standard RS-232C port to connect to other RS-232 equipment

**USER PORT DEVICE
COMMUNICATES OVER ETHERNET
INSTEAD OF TELEPHONE LINE**

EASY TO PROGRAM

Simply open an RS-232 file and then send data.
For example:

```
10 OPEN 2,2,2,CHR$(10)
20 PRINT #2, "MY DATA"
30 CLOSE 2
```

FIND OUT MORE AT:

<http://www.CommodoreServer.com>



ARE YOU DISK ORGANIZED?

TAKE CONTROL OF YOUR DISK COLLECTION

D64 Disk Management for your Commodore Computer

Never before has there been an Internet-based disk management system and software download center for Commodore computers!
Access, upload and view disks right from your Commodore computer.

UPLOAD YOUR DISKS TO: <http://www.CommodoreServer.com>

NEWS

C64anabalt Cartridge Available! (C64)

C64anabalt is available to buy on cartridge from Retro Gamer shop

The website says

C64anabalt is an official conversion of Adam Atomic & Danny B.'s award winning single-button 2009 indie game Canabalt for the 8-Bit, 64KB RAM, 1Mhz Commodore 64 home computer developed by Paul Koller (Paulko64). This particular version was designed to run from a 16KB cartridge (although there are also tape and disk versions available for free download as well).

The game was developed as an entry for the RGCD C64 16KB Cartridge Game Development Competition (2011) in which it achieved second place, and the name C64anabalt was suggested by Adam Atomic himself. The physics and procedural algorithms are based on those documented in the original game's open source code.

There are two versions of C64anabalt available; one with a SID chip conversion of Danny B.'s original score by Mikkel Hastrup (Encore), and an alternative build featuring music from the PC indie game ThrustBurst by Andreas Slotte (Ghormak). Unfortunately it wasn't possible to fit them both into a single 16KB ROM, so we've made two versions available to order or download.

Please note that C64anabalt is compatible with NTSC C64's, but lacks the static parallax background cityscape (the background scrolls instead) and it stutters slightly at high running speeds (due to the NTSC machine having less CPU

time available). The game also plays fractionally faster than the PAL version. None of these issues severely affect the play of the game, but it should be noted that the game was coded specifically for PAL machines.

The dove-grey cartridge is packaged in a box designed by Adam Saltsman and comes complete with a printed manual. I will endeavour to ship out games within a week of purchase, but due to these being custom built with separate soundtracks it may take a bit of time (depending on how many orders come in).

<http://www.rgcd.co.uk/2012/01/c64anabalt-cartridge-available-c64.html>

Commodore Free

And I thought Christmas was officially over!



1541Ultimate I & II Update 2.4

A new firmware for 1541 Ultimate (versions I and II) has been released. The update contains fixes which are already included in the custom firmware v2.3 (tlr). In addition, the following functions were added:

- CRT load
- EasyFlash support (read only)
- Flashing and running custom FPGAs (1541U-II only)

According to the change log there has been quite some work "under the hood". The new firmware is available for registered users in the Download section at www.1541ultimate.net.

1541U Firmware Update v2.4A released

After the recently released firmware 2.4 there is now an update to version 2.4A. In the 1541ultimate.net forum users have reported that when executing a cartridge image, the C64 freezes. This bug should no longer occur with firmware 2.4A. Moreover, a complete failure of the 1541U may occur when attempting to install the firmware 2.4, according to the users. Let's hope that this error has been corrected as well. Registered users can download the new firmware from 1541ultimate.net.

Source: www.mingos-commodorepage.com

2012 C64-RetroInvaders calendar: Mermaid graphic artist [2012]

A new year means the need for a new calendar, that need is now fulfilled, (hurray) with this latest commodore related design, Download it , print it out and watch the days flitter away and your life force slowly die.

ErmEnjoy !

<http://noname.c64.org/csdb/release/?id=104058&rss>



NEWS

Cracker's Map

The Recollection Magazine

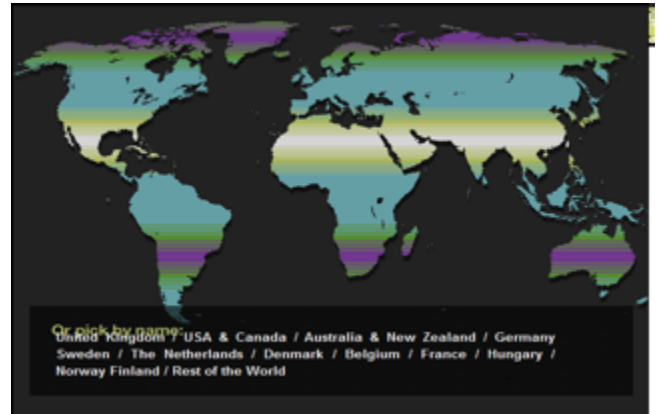
Has a webpage devoted to tracing Hackers all over the world in the Commodore Scene. The site breaks a map of the world down into areas you can select and see the groups and cracking teams that were and still are active in these regions. Here is what the website says about the map

Researched and documented by Jazzcat.

Inspired by the work of Nosah/DCS through the classic paper magazine from the late eighties called Iguana - this segment will focus on the major scene countries' cracking groups and their members, how they started, what they did and why they are remembered. Each group will have primary detail, in some cases directly from the people involved, as well as some interesting trivia dug up from the depths of time. Rather than

being mislead by unreliable information, enjoy the facts about the glory days of pirating.

http://www.atlantis-prophecy.org/reollection/?load=crackers_map



BIL HERD SPEAKS TO FRESNO USER GROUP

Bil Herd speaks to the Fresno Commodore User Group, part 1
<http://blip.tv/file/5814094>

Bil Herd speaks to the Fresno Commodore User Group, part 2
<http://blip.tv/file/5814249>

In a kind of follow-up to the above chats, Bil Herd has posted a new video about the Commodore 116, the \$49 computer he and the other CBM engineers hoped would be good seller for Commodore Business Machines.

To find out what happened to the C116, see Bil's video at
http://www.youtube.com/watch?feature=player_embedded&v=xPD5N43VlSk

Happy New Year!
Robert Bernardo
Fresno Commodore User Group

Commodore Free

Bil herd seems to be going very active and verbal with Commodore, these videos are a very welcome insight in to some Commodore history



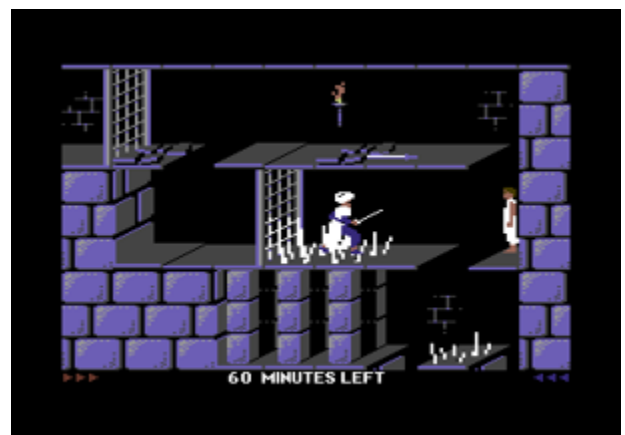
Prince of Persia - Total Pack for Commodore 64: V1.0

75 custom fan-made levels, random level generator and game launcher for C64 version of Prince of Persia

Features

- Automatic cartridge files generation from custom PC fan-made levels
- Built-in random level generator (Aka Random Tournament Mod)
- Random level tweaking (Random decoration, spikes and level mirroring)
- User-friendly level launcher (WinVICE should be installed)

<http://csdb.dk/release/?id=103920>





Commodore C64 Back to the Future!

Connect to TV and Play!

- 2 MByte FLASH
- 16 MByte SDRAM
- S-Video out
- Stereo Audio out
- PS-2 keyboard and mouse
- Micro SD card
- 2 Joystick ports



more than 100
C64 games!



Online Order: www.arcaderetrogaming.com

More Information: www.mcc-home.com



Multiple Classic Computer



C64



AMIGA

Connect to TV
and Play !



The ultimate
Retro Gaming
Device

ATARI 2600



Online Order:

www.arcaderetrogaming.com

More Information:

www.mcc-home.com



NEWS

Rear Admiral ThunderDrive

MyTec Electronics has released the Rear Admiral ThunderDrive, a modern-day replica of the Creative Micro Designs Hard Drive.

<http://69.47.125.137/ThunderDrive/Product/index.html>

or

<http://69.47.125.137/ThunderDrive/index.html>

Here are the details on this new hardware:

- * The first run of approx. 15 ThunderDrive units is for sale.
- * Price - \$599 list (eBay) or \$549 direct (PayPal non-eBay).
- * Payment by Paypal or U. S. Postal Money Order. If paying directly by Paypal, the "Personal/Gift" option is preferred.
- * Contact: workspace(at)wowway.com
- * ThunderDrive case size - (L)7.6" x (W)6.2" x (H)1.5" - very compact.
- * Power requirements - High-efficiency switch-mode power supply included. 12VDC / 5VDC @2A each max. Normal load = approx. 1.45A on 5V and 0.98A on 12V.
- * Hard drive capacity - 4.1GB of actual user space (4.29GB total drive space).
- * Distributed with v1.92 of CMD HD-DOS.
- * DOS is installed from the disk and resides in the ThunderDrive DOS memory area. Kernal routines (or BIOS) are on the ROM. In comparison to the CMD HD, there are very few differences other than init procedures and some added code for clockUP!(tm) handling.
- * An imaged diskette of latest CMD HD utilities is provided, along with references to many on-line repositories for 3rd-party enhancements.
- * No DOS improvements, but the electronics have been greatly improved. The number of discrete components has been reduced on the ThunderDrive board in comparison to the CMD HD, and MyTec has developed clockUP!(tm) doubling during non-IEC intervals, which provides increased performance with large REL files, for example, with BBS software.
- * Maximum hard drive space recognized - 4.1GB. Such a 4.1 gig limit is due to the 8-bit 6502-based controller used with a 6502-based machine, the math being: 256(8-bit) max sectors x 256(8-bit) max tracks = 65536 blocks x 256(bytes per C= block) x 256(8-bit) partitions = 4.29GB HD Space.
- * The ThunderDrive uses a SCA 80-Pin SCSI-3 UW eServer drive mechanism which has been short-stroked and forced into a

SCSI-II mode, yielding nominal space for the ThunderDrive while still performing at 10K RPM speeds. These drives were selected primarily for longevity and form-factor. The drive is equipped with an internal cooling fan.

* The ThunderDrive has been tested to work perfectly with attached, external, SCSI devices - CMD HD, CD-RW, IomegaZip, Syquest, and magneto/optical devices.

* MyTec also provides complete repair service and replacement parts for CMD HD / FD units.



NEWS

Commodore plus 4 SID card scores an own goal

Sid card 1 Commodore free 0

OK I am guilty of this error, I hope this goes some way to explain my error and hope I didn't upset to many people in the process.

Hi Commodore Free,
There are two problems

The first one is that the way it is written may suggest that there is an SID (8580) chip included in the shipped product but this is not the case. Everyone must obtain by himself or herself a SID.

The second is the link being completely wrong. Please use this instead: http://bsz.amigaspirit.hu/nae/index_en.html.

The two cards are quite different. The old one by Solder in its basic form had no ADC/DAC but it was available as an add-on, and it was only capable to run from a plus/4 system clock which is about 9/10th of the c64's therefore its voice differs from the c64.

The one developed by BSZ has integrated DAC but no ADC capability. Furthermore it is capable to synthesize c64 clock for the SID so voice can be the same, it has better sound quality, support for 1351 proportional mouse and selectable audio out routing.



mind.in.a.box

mind.in.a.box have released a new video teaser for their forthcoming album "Revelations". The 7 minutes long video illustrates all songs of the album against a backdrop of images and visual effects.

The release dates of "Revelations" are:
20th January at Dreamweb Music (Europe)
24th January at Metropolis Records (Rest of the world)

You can find the video here -->
http://www.youtube.com/watch?v=R_u4JcTmSA

<http://www.mindinabox.com>
<http://www.dreamwebmusic.com>

Commodore free interview is available here
[http://www.mindinabox.com/things/interview.php?id_iv=50& height=600&width=800&jqmRefresh=false](http://www.mindinabox.com/things/interview.php?id_iv=50&height=600&width=800&jqmRefresh=false)
and in issue 40 of Commodore Free magazine
<http://www.commodorefree.com/magazine/vol4/issue40.html>



MEGABIT ROM RETURNS

Nearly 3 years ago Dan "Megabit" Newbury released the Megabit 128 Function ROM, a chip that plugged into the empty motherboard socket of the Commodore 128 and made a slew of C128 and C64 applications immediately available with the press of the F1 key. Now the Megabit 128 Function ROM has returned To see the complete list of applications, specifications of the Function ROM, or even contact Dan for more info or for placing your own apps in the ROM,

see the discussion thread at
<http://www.commodore128.org/index.php?topic=2645.0>

The current pricing for the Function ROM is \$22.50.

Shipping costs is:
USA = \$2.50
Canada = \$5.00
Europe = \$5.00
Australia = \$5.00

PayPal is newbury@planetkc.com

NEWS

uIEC Manager for GEOS

-----Original Message-----

From: Werner Weicht
Sent: 25 December 2011 11:59
To: uiec-users@googlegroups.com
Subject: uIEC Manager for GEOS

Hi all,
Now the final version of my "UIEC_Man" for GEOS, Wheels and MegaPatchV3 is now available on my homepage (<http://wweicht.homepage.t-online.de>) in the download section.

With this program you can change Dxx images on UIEC, SD2IEC and MMC2IEC in GEOS, Wheels and MegaPatch V3. A manual is included. Please read this before you use the program.

Bye
Werner



COMMODORE PROGRAMMING BLOG

I was looking for something on the internet and came across this website, I may have listed it before in these very pages of Commodore Free, well an earlier issue anyhow, I have listed the titles so you get an idea of what to expect they were all created by [digitalerr0r](#)

[Commodore 64 Programming #1: A quick start guide to C-64 assembly programming on Windows](#)

[Commodore 64 Programming #2: Intro to 6502 microprocessor programming](#)

[Commodore 64 Programming #3: 6502 arithmetic's](#)

[Commodore 64 Programming #4: Rendering a sprite](#)

[Commodore 64 Programming #5: Clearing the screen](#)

[Commodore 64 Programming #6: Raster lines](#)

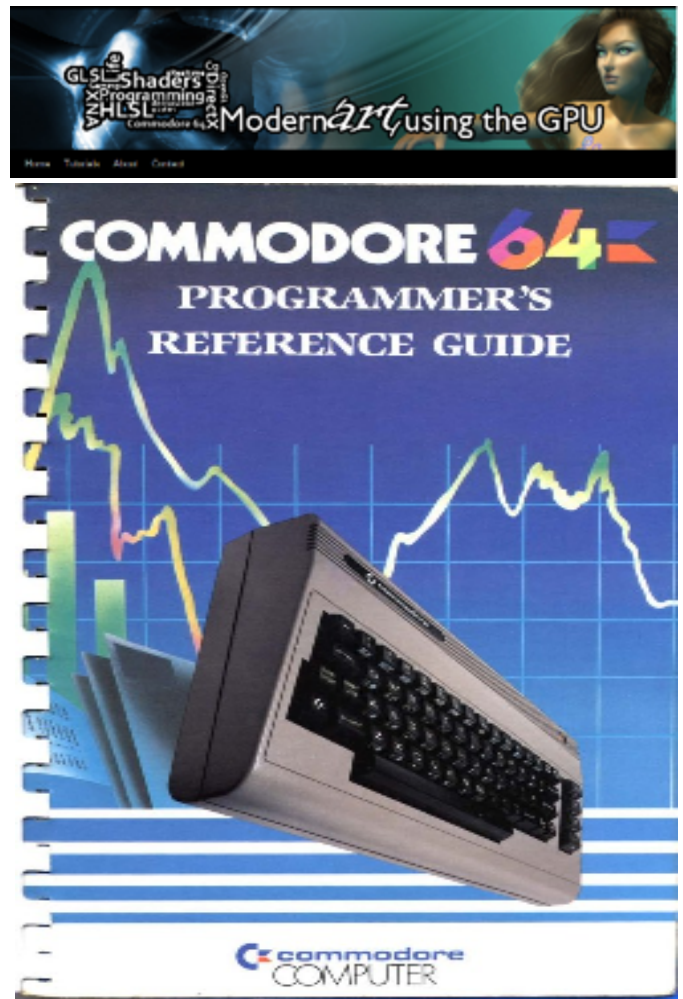
[Commodore 64 Programming #7: Creating and rendering bitmaps](#)

[Commodore 64 Programming #8: Text and custom charsets](#)

[Commodore 64 Programming #9: Interrupts and music](#)

[Commodore 64 Programming #10: Multiple interrupts](#)

The last installment was posted in May 2011
<http://digitalerr0r.wordpress.com/category/commodore-64/>



UFO DEFENCE

Commodore 64 game released Called UFO Defence move left and right to launch a missile and try to kill the aliens at the top of the screen, Reviewed later in this issue

Credits :

Code Nomistake of Delysid

Music Laxyty of Maniacs of Noise, Vibrants

Graphics Nomistake of Delysid Reject of Delysid, Desire

http://noname.c64.org/csdb/getinternalfile.php/100700/UFO_DEFENSE.d64



NEWS

JUST DANCE C64 DEMO

A demo has been released for the C64 called Just dance.

Here is what they say about the demo

Just Dance 64 By algorithm

Demonstrates real-time loading and decompression of compressed audio data on stock hardware.

This is possible without breaking a sweat due to the predefined recompilation routines.

Instead of analysing a byte and extracting bits and performing decisions based on the content, the main core of the routine analyses the entire byte at once. (4 bytes of 8 bit sample data are compressed in a byte) and determines the course of action based on the 'Whole' value of the byte.

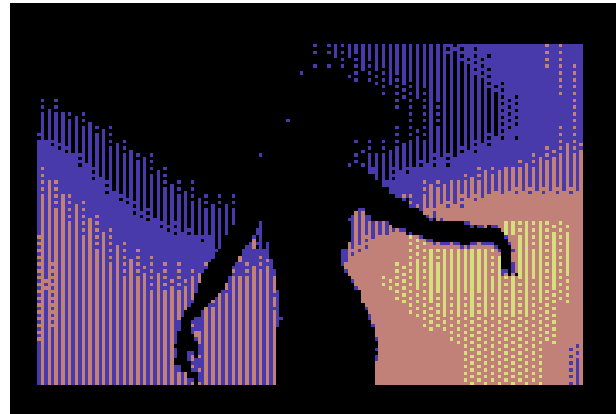
Main core of the decode routine only uses 1 raster line for 4 decompressed bytes! Due to this, it is no problem to load buffer data from disk as well as to combine other routines (the animation on the demo was just meant as a filler to show that the audio routine leaves plenty of processing time left for other actions)

There is over 2 minutes of audio packed to the disk.

The encoder on the PC side uses a variation of the ADPCM routine which results in higher quality audio.

Two optimum step sizes are chosen for each small chunk of the audio sample (instead of increasing/decreasing step value based on previous prediction)

http://noname.c64.org/csdb/getinternalfile.php/100519/JustDance_64.d64



DOOM 3 OPEN SOURCE

The Source code to doom 3 has been released to the public,

Doom 3 GPL source release

This file contains the following sections:

GENERAL NOTES

LICENSE

GENERAL NOTES

Game data and patching:

This source release does not contain any game data, the game data is still covered by the original EULA and must be obeyed as usual.

You must patch the game to the latest version.

LICENSE

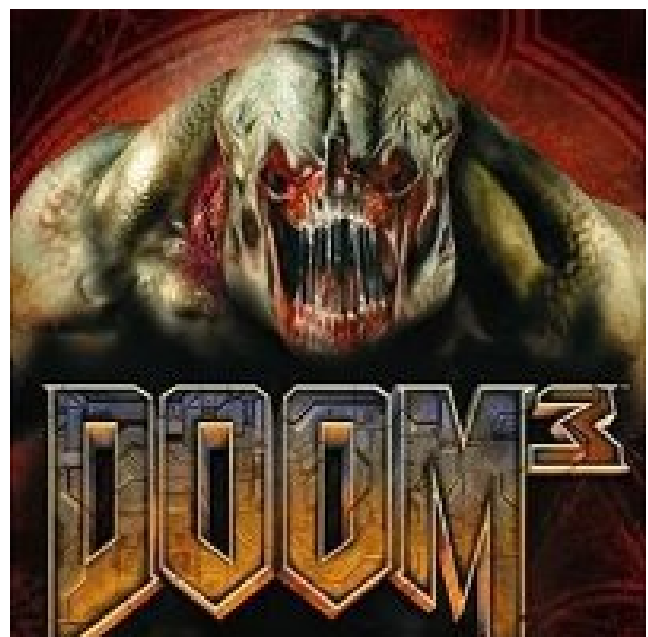
See COPYING.txt for the GNU GENERAL PUBLIC LICENSE

ADDITIONAL TERMS: The Doom 3 GPL Source Code is also subject to certain additional terms. You should have received a copy of these additional terms immediately following the terms and conditions of the GNU GPL which accompanied the Doom 3 Source Code. If not, please request a copy in writing from id Software at id Software LLC, c/o ZeniMax Media Inc., Suite 120, Rockville, Maryland 20850 USA.

EXCLUDED CODE: The code described below and contained in the Doom 3 GPL Source Code release is not part of the Program covered by the GPL and is expressly excluded from its terms. You are solely responsible for obtaining from the copyright holder a license for such code and complying with the applicable license terms.

<https://github.com/TTimo/doom3.gpl>

http://www.gamasutra.com/view/news/38702/Doom_3_Source_Code_Released_To_Public.php



COMMODORE 64 ASSEMBLER TUTORIAL



Author's notes: These tutorials were published in Micro Mart magazine between February and July 2011 in the 'Specialist' section, which also includes Amiga, Apple, Linux and gaming news and views – see www.micromart.co.uk for more information about this publication.

As Micro Mart is weekly, what I decided to do is the assembly in four parts over consecutive weeks, followed by at least a four week break in which time I would write news features. It was never intended for advanced or intermediate-level programmers, but for beginners who might know a bit of BASIC and have always wanted to progress to assembly language. I decided to write a 6502 tutorial after the success of a similar tutorial series that I wrote for the Sinclair ZX Spectrum in 2010, obviously in Z80 assembly. Unsurprisingly, I got a more positive response from the Speccy community than I did for these articles.

The tutorials assume no prior knowledge of machine code. So if you're not a complete beginner, none of this will not be very useful, and if you are a complete beginner, then I'm only giving you a foundation on which to build. Remember: the more you experiment with your code, and the more you read up about the Commodore 64's hardware, the more you will learn. Even better if you enjoy programming, because this will aid your progression as much [or more] than anything else.

I retain the copyright for these articles, which are used in Commodore FREE with permission. If you would like to contact me about them, then you may do so through the Micro Mart forums and a link is provided. Without any further ado, here is the first four weeks, which covers some of the very basics. The original images that I provided with the articles are included for illustration purposes only.

The Commodore 64 - Part 1

As with the last series of tutorials (which was an introduction to Z80 assembly language for the Sinclair ZX Spectrum, published in Micro Mart), this will walk you through the beginnings of assembly language for a popular 8-bit home computer from the 1980s. This time though, it'll be the Commodore 64, which is a 6502-based micro computer. I may go at a pedestrian pace in the first four parts, which is partly due to my frustrations with machine code tutorials from years past, however (though it won't be totally necessary) I'll be assuming that you have some programming knowledge of a high-level language such as BASIC. And there is a thread on the Micro Mart forums in which you may ask any questions that

you have, all of which I will endeavour to answer. This will also be the place to post your code and contact me.

Before we look at some example code, I'll remind you that programming is a way of thinking and problem solving; remembering the syntax of a particular language isn't always useful (though it certainly helps when you hop from one language to another), and in this case, assembly can be rather cryptic. As I've pointed out in the past, pencil and paper can be one of the most useful tools any programmer can have, as working out the logic and structure of your program first, or particular routine within it, will in most cases help eliminate the frustrations of programming it later.

First of all you will require a good C64 emulator, Notepad or something that will allow you to write plain text files and an assembler for your Windows PC, Mac or Linux system. I'll be using is The Dreams Assembler as well as WinVICE [both for Windows]. Links to the downloads are available at tinyurl.com/C64-Coding.

Let's now take a look at some basic information about our target platform. Announced at the Consumer Electronics Show held in Las Vegas, USA way back in January 1982, the Commodore 64 was the successor to the CBM/PET series and VIC-20. Boasting a whole 64K of RAM and a high-resolution screen of 320 by 200 pixels with 16 colours, 8 hardware sprites and three-channels of mono-audio, CBM pushed this 8-bit as the first fully-featured home machine for under \$600USD. Whilst this may not have been so affordable back then in the UK, it was the cheapest machine in its' class in North America, and eventually became the most successful 8-bit computer of all time.

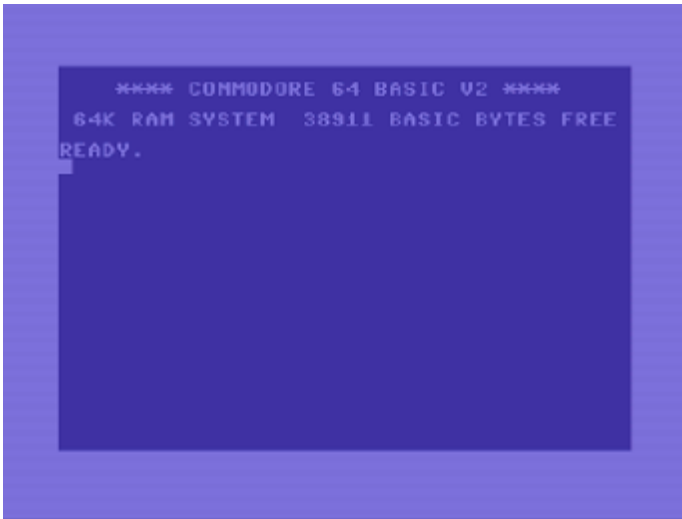
It contained a modified 6502 processor, the 6510, which ran at approximately 1Mhz. The differences were minimal from a 6502, but it allowed 'bank-switching', meaning that all 64K of RAM could be accessed by the programmer. The machine could use cartridges, cassette tape or diskette media, had two joystick ports as standard and a user port, which was usually for connecting Centronics-compatible printers. Various upgrades appeared throughout its' commercial life-span, including memory expansions, accelerators, hard drives and even RAM drives.

Bread bin to slim

The Commodore 64 went through many revisions during the 1980s. The original machines used recycled VIC-20 cases (known as the bread bin, due to its' shape), and the technology was also bundled into a PET case - the Educator 64 - and in 'luggable' form, the SX-64, which was also the first colour portable computer, sporting a built-in disk drive and a 5" colour screen.

In the mid-80s, Commodore cost-reduced the computer into a cream slim-line casing, known as the C64c, and later took away the keyboard for the C64GS (Games System), which was bundled with a rather flimsy joystick, using cartridges only.

This was the beginning of the end of the commercial life for the once-mighty 8-bit, until a brief resurgence in 2004, when the Toy:Lobster company launched the C64DTV, a battery-powered device that plugged directly into your TV, containing 30 classic games plus hidden features, such as the ability to turn it into a 'real' C64 by soldering in a PS/2 keyboard onto the PCB, and adding an IEC serial port to connect a compatible Commodore disk drive.



First steps – Part 2

When you're thinking about writing a program or routine, it's always worth jotting down some objectives first, as there is often more than one way to achieve what you want, and of course more than one language in which to do it. On the Commodore 64 you could, with a compiler and some persistence, use the machine's somewhat limited BASIC to do most things that the technology is capable of. However, using assembly has certain benefits, most obviously speed and direct control over everything that the computer does.

If you haven't done so already, go to tinyurl.com/C64-Coding and download The Dreams Assembler, unless you have a C64/6502 assembler that you are comfortable with.

For our first example, let's say that we want a simple routine that will change the border and screen colour to black and clear it. Many of you may remember how to do this in BASIC, and it can be achieved in one line. First, you will need to POKE the relevant memory locations for the inner screen and border with the value that you want (zero to 15 representing the colour). Then, you will need to PRINT either a shifted CLR/HOME within quotation marks, or find the PETSCII value

with the CHR\$ command. Open your text editor and let's see how to do this in assembly. Comments are included (after the semi-colon) for clarity.

; This will tell our assembler where to put this routine in memory:

***=49152**

; Here is our appropriately-named label:

BEGIN

lda #\$00 ; Equivalent of LET A=0 in BASIC
sta \$D020 ; Stores A into memory location \$D020 (like POKE 53280,A)

sta \$D021 ; As above, but into \$D021

lda #\$93 ; Now, let's change the value of the accumulator (A) to \$93

jsr \$ffd2 ; Jump to 'Char out' ROM routine

rts ; Return from sub-routine (in this case, back to BASIC)

Save your file with a .asm extension (i.e., save it as test1.asm), exit your text editor and drag and drop the file onto your assembler (if you are not using The Dreams Assembler, please refer to the instructions for assembling your source code). Assuming that there are no errors, you should see a file called 'a.out' created in the same directory as the assembler. Now open your emulator (VICE is recommended), and either drag and drop a.out into your C64 emulator, or load it in according to the instructions of your particular emulator. You can rename this to whatever you like, and also change the file extension to '.prg' if your emulator is being fussy about loading the file. You should see a 'READY.' prompt after a short virtual load. To call the routine, type:

SYS 49152

and press enter. You will notice that the whole screen is turned black and cleared, and you should see the 'READY.' prompt again. This is hardly very exciting, but it's a start. Try changing the values in of the accumulator in the routine (LDA #\$xx) and see what happens. Note that you can not go above 255 or below zero, and that all the numbers within the code is in hexadecimal, with the exception of the memory pointer.

Hex to decimal

Computers do everything in binary, a number system to the power of two. This can be difficult to read as we're used to dealing in decimal. Our program is located at 49152, or \$C000 in hexadecimal. In binary, this would be 1100000000000000, so calling this using SYS could cause a crash if you miss or add a naught accidentally.

Hexadecimal works to the power of 16, which fits binary better than decimal (16 = 10 hex = 10000 binary). As we don't have a single numeric digit to represent 10 through to 15, we substitute these for the first six letters of the alphabet, therefore A in hex is 10 and F is 15, and hex 10 is 16. We use the Dollar (\$) sign in our code to tell the assembler that the number is in hex. To convert any 16-bit hex number, use the following formula: (N3x4096)+(N2x256)+(N1x16)+N0, so \$FEDA is (15x4096)+(14x256)+(13x16)+10, or 65242.


```

Administrator (C:\Windows\system32\cmd.exe - x86 - Test1.com)
File Edit Search View Options Help
C:\Windows\system32\cmd.exe - x86 - Test1.com
: This will tell our assembler where to put
: this small routine in the C64's memory:
*=$9152
; Here's our label:
BEGIN
  ldx #$00          ; Equivalent of LET A=0 in BASIC
  stx $0000        ; Stores A into memory location $0000 (53280)
  txa              ; As above, but into $0001
  ldx #$93        ; Now, let's change the value of the accumulator (A)
  jsr $FFD2       ; Jump to 'Char out' routine
  rts              ; Return from sub-routine
END
FileHelp Line:1 Col:1

```

Character out – Part 3

As you may already know, the Commodore 64's screen has two areas, the outer screen (border) is generally a solid colour, without some well-timed programming or trickery, at least. The inner screen area (simply referred to as the 'screen') is what we're interested in. It has a pixel matrix of 320 x 200 pixels in bit-mapped high resolution mode, or 40 x 25 characters cells. The screen RAM is mapped in memory at location 1,024 (which is hex 0400) taking up 1,000 bytes of RAM. The colour RAM for each corresponding 8 x 8 cell at is at 55296 (hex D800) - in other words, if you have a character stored at location 1025, the foreground colour for that would be 55297; we'll worry about this in later instalments. For now, we'll look at writing characters to the screen.

From last week's example, we called the ROM routine at 65490 (hex FFD2) to write the contents of the accumulator (A) to the screen. What this routine does is output the character to a device, in this case it is device zero unless we want to say otherwise, and device zero is the screen. As the accumulator can only hold a one byte value, we can only write one character to the screen at a time. Consider the following routine.

; Let's set up a constant:

CHAROUT=65490

; We'll put our program into memory location 49152:

***=\$c000**

BEGIN ; Here's where we'll loop back to

ldx #\$48 ; LET A=72 (decimal)

jsr CHAROUT

ldx #\$45 ; You can work out the decimal equivalent

jsr CHAROUT

ldx #\$4c

jsr CHAROUT

ldx #\$4c

jsr CHAROUT

ldx #\$4f

jsr CHAROUT

inc \$d020 ; This will increase the contents of the memory location 53280 by one

jmp BEGIN ; This is an unconditional jump (like GOTO) back to BEGIN

Compile the routine as usual, drop it into your emulator and call it by SYS 49152 and press enter. The first thing that you will notice is that a word will flood the screen, and the border will continuously change colour. To stop the program, use RUN/STOP and RESTORE (Caps lock and Page Up on WinVICE).

The screen is flooded with text very quickly, certainly in comparison to the BASIC equivalent, but it could be quicker still as this example is very inefficient and poorly coded. For a start, we have a duplicate line of code if you can spot it.

A more logical way to write it is to separate the data from the main routine, reading each byte with a conditional loop before calling CHAROUT. Even then there is a better way to output characters. Let's have a look at a more refined example:

CHAROUT=65490 ; Constant

***=\$c000 ; Store routine at 49152**

BEGIN ; Label

ldx #\$00 ; LET X=0

txa ; LET A=X

LOOP ; Marker

lda DATA,x ; Read byte at memory location DATA+X and store in A

cmp #\$00 ; Compare A to zero

beq END ; If true (i.e., A=0), branch to END label

jsr CHAROUT ; Call CHAROUT routine

inx ; Increase X by one (X=X+1)

jmp LOOP ; Jump to LOOP marker

; Here is our label for the conditional loop above:

END

rts ; Return from sub-routine (to BASIC in this instance)

; Here is our text data (stored in bytes):

DATA

.text "hello",0

Save and assemble this new version and try it. Apart from the fact that the program is much tidier and logical, it's also more readable. Using the ROM routine at hex FFD2 still isn't the best way to write to the screen; We know where in RAM it is mapped so we are able to store bytes directly there using a similar method as above, and we'll look at that next week. For now, mess around with the latter piece of code and see if you can improve it.

```

C:\Program Files\WinVICE\WinVICE.exe - C:\Program Files\WinVICE\WinVICE.ini
File Edit Search View Options Help
C:\Program Files\WinVICE\WinVICE.ini
1 CHAROUT=65490 ; Constant
2 *=$c000 ; Store routine at 49152
3 BEGIN ; Label
4 ldx #$00 ; LET X=0
5 txa ; LET A=X
6 LOOP ; Marker
7 lda DATA,x ; Read byte at memory location DATA+X and store in A
8 cmp #$00 ; Compare A to zero
9 beq END ; If true (i.e., A=0), branch to END label
10 jsr CHAROUT ; Call CHAROUT routine
11 inx ; Increase X by one (X=X+1)
12 jmp LOOP ; Jump to LOOP marker
13 ; Here is our label for the conditional loop above:
14 END
15 rts ; Return from sub-routine (and to BASIC in this instance)
16 ; Here is our text data (will be stored in bytes):
17 DATA
18 .text "hello",0

```

Bites and nibbles – Part 4

To quickly recap, I gave you some basic information about the Commodore 64's screen and colour RAM (sometimes referred to as colour nybbles – note the spelling), and a quick example of writing text to the screen. Hopefully, you've been experimenting with the code and have made it display all kinds

of text, such as 'I AM ACE!'.

You may remember that we were calling the ROM routine at hexadecimal FFD2 to output characters to the screen one at a time; machine code (which is what your assembly routine becomes) is much quicker than BASIC, so you won't see each character being written as it happens. But even taking this into account, there is a quicker way still to display your message. Try the following example:

```

; Constant:
SCRRAM=$0400
; Here's where our program will reside in memory (12*4096)
*=$c000
START      ; Marker
  ldx #$00  ; X=0
  txa      ; Transfer X register to accumulator (A=X)
  sta $d020 ; Store A at memory location
  sta $d021
LOOP       ; Loop marker
  lda STRING,x ; Read byte at location STRING+X into A
  cmp #$00    ; Compare contents of A to zero (terminator)
  beq END    ; If equal, branch to END
  sta SCRRAM,x ; Store A at SCRRAM+X
  inc       ; Increase X by one (X=X+1)
  jmp LOOP   ; Unconditional jump to LOOP
END        ; END marker
  rts      ; ReTurn from Subroutine (back to BASIC)
STRING    ; Data marker
; Test data - 40 characters long. Zero used as terminator:
.text "0123456789012345678901234567890123456789",0

```

What is happening is that the routing is using a conditional loop which reads in each byte from the segment of memory called STRING to the accumulator. We compare its' contents to our terminating value (zero), and if true it branch to our END marker; if not the accumulator is written directly to the screen RAM position plus X. X is then increased to the next character position on the screen.

This not only saves some processor time, it will also allow you to write characters to the whole screen area, as CHAROUT (hex FFD2) will force a line-feed when it writes to the last character of the bottom line, as it does in BASIC. But we've only written text to the top line here. Filling the screen RAM won't work by just increasing the number of characters in the STRING block because the counter we're using (the X register) will only hold values between zero to 255. The screen RAM is 1,000 bytes, so trying to write more than 256 characters with the above example won't happen as it will never reach the terminator.

To write to the whole screen, try the following example:

```

SCRRAM=$0400
COLRAM=$d800 ; Constants
*=$c000      ; Resides at 49152
  ldx #$fa    ; X=250 (decimal)
  lda #$01
  sta $d021   ; Store A at 53281
  lda #$56    ; A=86 (decimal)

```

```

LOOP      ; Loop marker
  sta SCRRAM,x ; Store A at SCRRAM+X (first 1/4 of screen RAM)
  sta COLRAM,x ; Store A at COLRAM+X (first 1/4 of colour RAM)
  sta SCRRAM+250,x; etc...
  sta COLRAM+250,x
  sta SCRRAM+500,x
  sta COLRAM+500,x
  sta SCRRAM+750,x
  sta COLRAM+750,x
  dex      ; Decrease X by one (LET X=X-1)
  cpx #$ff ; Compare X to 255
  beq END  ; If equal (true), branch to END
  jmp LOOP ; Jump to LOOP
END       ; END label
rts      ; ReTurn from Subroutine (back to BASIC)

```

What you will get with this routine is a screen full of shift+V on your C64 (a chunky X). It counts down with the X register, remembering that when we decrease it by one and X is zero, it wraps around to 255 (hex FF), so 0-1=255 rather than -1 (this is actually a two's compliment representation of -1, which you may look up on the Internet – but if you don't understand this then don't worry about it for now). We're also writing to the colour RAM at the same time using the same logic. See if you can apply this to the first routine to get it to write to more than 256 characters of text. If you succeed, post your code at www.tinyurl.com/C64-Coding, where you may also ask any questions and contact me.

Colours

The Commodore 64 is able to display 16 colours, 0 through to 15 inclusive, as follows: black, white, red, cyan, purple, dark green, dark blue, yellow, orange, brown, light red (pink), dark grey, medium grey, light green, light blue and light grey. By default, the colours repeat, so colour 16 is black, 17 is white etc... This is why we got away with writing a value higher than 15 to the colour nybbles in our second example.

I bet you've always wanted to know how to write to the screen RAM directly

COMMODORE FREE

Thanks to Shaun and Micromart

AMIGA FOREVER AND COMMODORE 64 FOREVER

Amiga Forever

<http://www.amigaforever.com>

<http://www.facebook.com/AmigaForever>

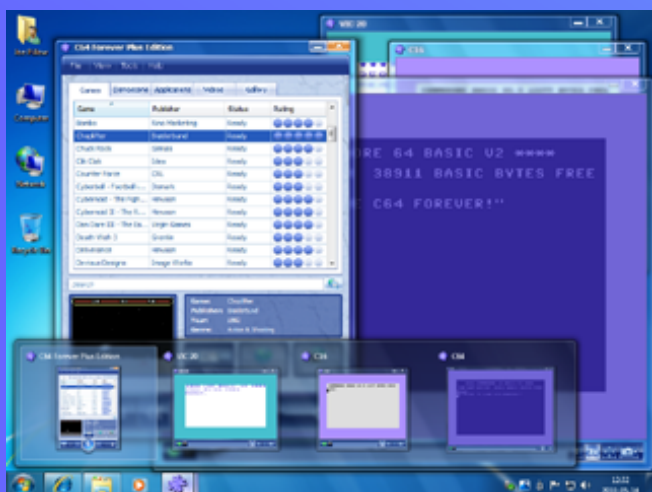
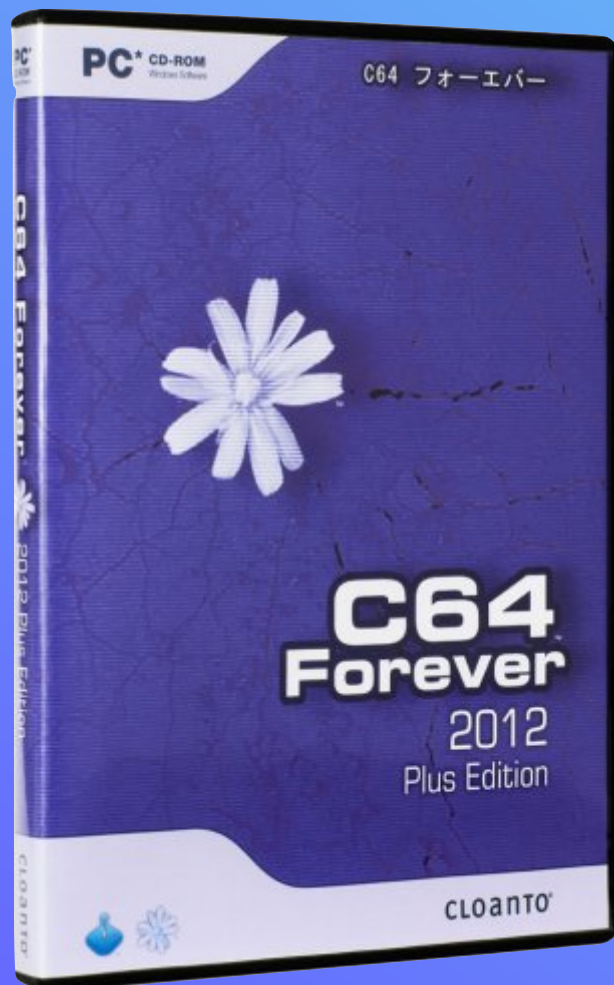
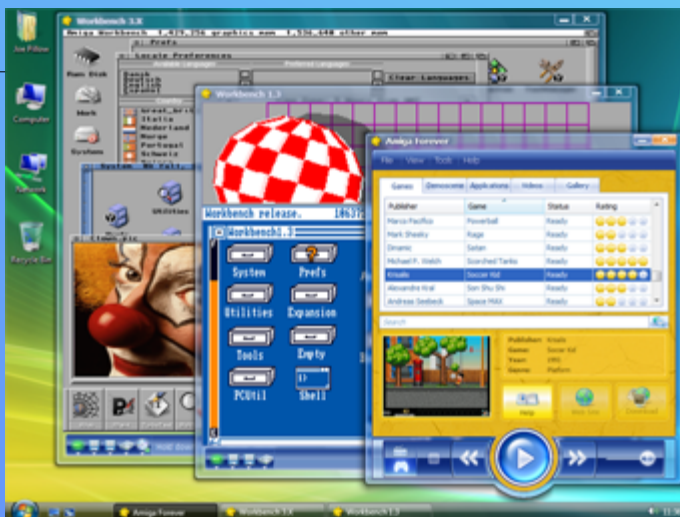
C64 Forever

<http://www.c64forever.com>

<http://www.facebook.com/C64Forever>

RetroPlatform Project

<http://www.retroplatform.com>



COMMODORE FREE REVIEW

THE LAST VIC= 20

Another game for the Vic 20

<http://www.sleepingelephant.com/ipw-web/bulletin/bb/viewtopic.php?p=62548#62548>

orion70 writes

Hello guys, new VIC software is not exactly flourishing this year, so here's a little game in hi-res .

It's basically a silly "game-and-watch" style arcade, with a 5-position cannon controlled through user-defined keys. You simply have to destroy 19 enemy bombs each level, before they hit your city or time runs out.

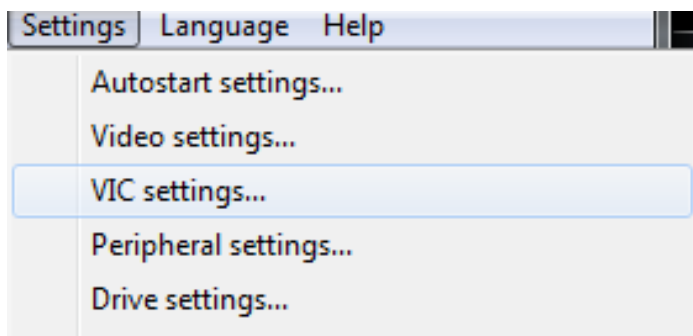
Only a *diversion*, nothing deep, but hey, it runs with a mere 8K expansion, which is less than my average

COMMODORE FREE

This game created by orion70 is a

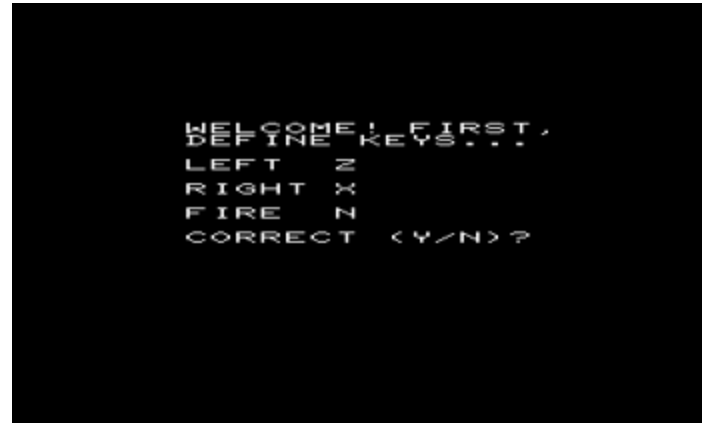
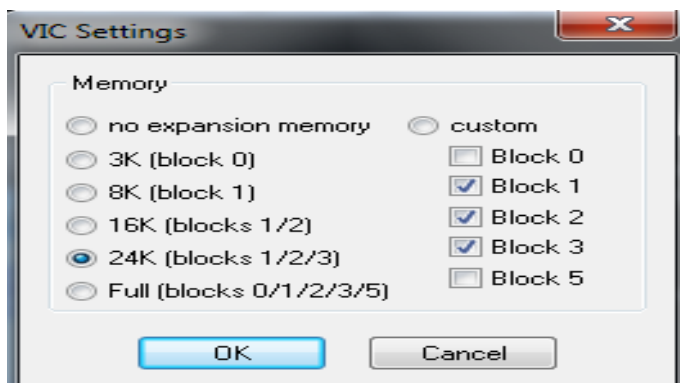
conversion of the Commodore C16 game entitled "Earth Defender", created by R. Adling. The game uses the hi-res 160x192 graphics from Mike's Minigrafik extension.

The aim of the game is to defend your home planet. You must destroy all the bombs with your canon. The game can be played on a real VIC-20 with 8KByte extra RAM or with an emulator. To load the game in WinVICE for example you need to go to the settings menu



Then select VIC SETTINGS

And tick the box 24k (block1/2/3) otherwise it doesn't seem to work correctly, I tried 16k but this didn't work for me either, examples were not recognising the keys when prompted to define keys (see below)



Loading the game, that incidentally is available as both a monochrome or a colour version,(except the colour version allows a joystick to be used as well) we are greeted with a an option for Joystick or Keys, (except the mono version but you guessed that didn't you) selecting keys we are prompted to define them. then the splash screen is displayed. The screen has the words "press any key to play" and as these are hard to ignore and doing so take us to our play field



With the game now loaded and us (the user) sat on the play field we have to defend ourselves from attack, its at this point the game may feel familiar to some readers. What happens is that very small missiles appear looking not to dissimilar to a full stops; and these start to move down the screen, in various angular lines of angularity, ermm couldn't think of a word to describe this..... but they do not leave a trail behind them. It's therefore difficult to see these attacks, mainly as the bombs are too small and I suppose you would say this ; adds to the difficulty of the game. Your task is to blow up these "missiles" before they can do damage to your city.

You move the selected keys to go left and right controlling the gun firing system. When you think you are in a line with the missile let a bomb go, if you hit a bomb a crosshair (it looks like a small cross hence the name) will lie where the missile was and you gain a point marked on the right of the screen, get to

10 and you go to the next level where it's more of the same but faster.

If you miss and the bomb hits the city then it will destroy a section and score a hit this is shown by the VIC damage when destroyed the game will end, on the far left is show the Time .

The game is very playable, although it will not be to everyone's liking, but the graphics look really nice on the VIC especially in the colour version of the game. The sound is basically white noise and small explosions but they do build up the tension in the game,

So all that is left is to score the game.

Although the game moves fast I found it frustrating to play however its a game I do like (in concept)

Graphics 7/10
Sound 6/10
Playability 5/10

Last ability 5/10

You will play it a few times then get fed up but you will go back to it! At a later time

Nice effort especially with the graphics



Article on 1541 disk recovery

By Jeremy Smith BSc (Hons)

Notes on recovering old C64 disks sent in by Commodore Free reader Jeremy Smith

What You Need

- First, you need a PC with a parallel port.
- Secondly, you need an XA1541 active cable (not X1541). These can be bought on eBay from <http://www.ebay.co.uk/sch/vasilisy/m.html>



- Thirdly, you need a C64 1541 drive, or compatible (e.g., Oceanic, Exceleator).
- A screwdriver
- A glasses cleaning cloth
- A bottle of Isopropyl Alcohol (e.g., a CD cleaning bottle)

Getting Started

Plug the cable into the PC's parallel port, then plug the other end into either of the din sockets at back of the drive. PC's ports are very sturdy, but if you need to take care, turn off the machine before connecting. The 1541 probably won't mind hot swapping the serial cable at the other end.

Then you need to Get this software:

opencbm-0.4.2-i386.zip

from this website

<http://www.trikaliotis.net/opencbm#download>

Unzip the files into the path C:/Program Files/Opencbm
If you can, run cmd from Start -> Run -> type cmd [Enter]

CD "C:/Program Files/Opencbm/exe"

Run instcbm, this installs the driver.
Next, put a disk in the drive and turn it on.

To test things, type in a CMD window:
cbmctrl dir 8

If it comes up with a list of files, good everything's working if not check the connections and that you have installed the files correctly!

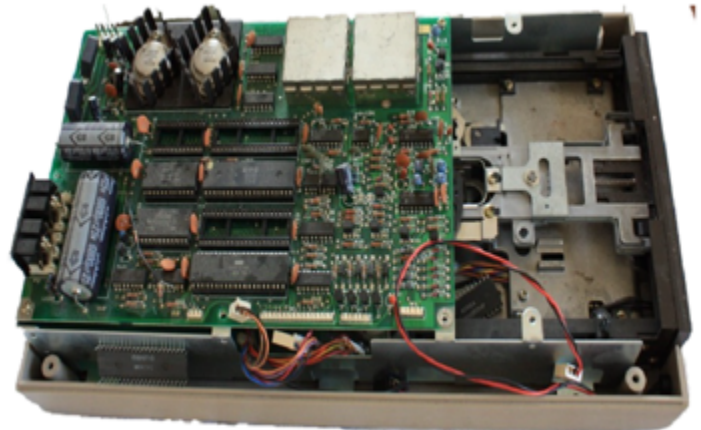
Taking the Lid off the 1541

Turn off the drive and unplug the cable and power.

Next, unscrew the 4 screws the underside of the drive, then lift the beige lid off. Put it somewhere safe.

There's a black thingy on a spring which lifts up, don't force it, and under that is a white head with a black line on it. This is what you'll need to clean every so often.

Be very careful to put the drive somewhere safe, so it doesn't get water or drinks spilt on it. That's the problem with taking the lid off. However, if you want, you can put the lid back on after cleaning the head, it just takes a bit longer.



Put the power and drive cables back in the drive, and turn it back on.

Backing Up Your Disks

Now there's a tool called 'd64copy' which is called so (from the command prompt in C:/Program Files/Opencbm):

```
d64copy 8 "my1stdisk.d64"
```

If you run this, it should go through all the sectors like this:

```
C:/Program Files/Opencbm\exe>d64copy 8 "my1stdisk.d64"
1: *****
2: *****
3: *****
4: *****
5: *****
6: *****
7: *****
8: *****
9: *****
10: *****
```

It should get to 35. If there's any read errors, it will warn you.

This:

683 blocks copied.

Should be 683 blocks. If it says 680, 3 errors have occurred.

Cleaning the Head After Dusty Disks

Here's where it gets interesting...

Some disks are old and because of this, the ferrite is coming off.

This tends to clog up the drive head, hence the alcohol cleaner.

Dab a little corner of the cloth in the alcohol and lift up the black thingy and clean the head with the corner of the cloth.

Don't worry about breaking it, but don't wet the bit of felt at the underside of the black thingy.

So what happens with some disks is they report loads of errors, just the sector list flying up the screen. Don't worry, they're not permanently unreadable. You have to run d64copy several times.

Getting Errors Down to Zero

The trick is to keep running d64copy until the errors get down to 0 - or close.

If you have a cluster of bad sectors (say, 11, 13, 14) then run d64copy thus:

```
d64copy.exe 8 -s 11 -e 11 "my1stdisk.d64"
```

And thus:

```
d64copy.exe 8 -s 13 -e 13 "my1stdisk.d64"
```

And:

```
d64copy.exe 8 -s 14 -e 14 "my1stdisk.d64"
```

This is much quicker than re-doing all 35 sectors. Just do it again and again and hope the sector clears up.

Hints and Tips

- Residue gets brushed off by the head against the disk, so re-reading the sector again and again removes errors.
- If 1 side a problem, read side 2 several times as the head passes over same point

- If one sector works, preserve it, so specify -s and -e and it writes always to the same file so -s 1 -e 10 to -s 11 -e 35 is the same as -s 1 -e 35

- Recommended is, when you put a disk in, do like thus:
cbmctrl dir 8 >001.txt

Where 001 is disk #1. Mark the disk with tape and write '001' on it in marker.

- If you can't get rid of all errors, you can try cbmcopy.
cbmcopy -r 8 filename

If you're lucky, run this on all the files on the disk, and hopefully the bad sectors won't be in those files. The bad sectors could be in 'unused' sectors!

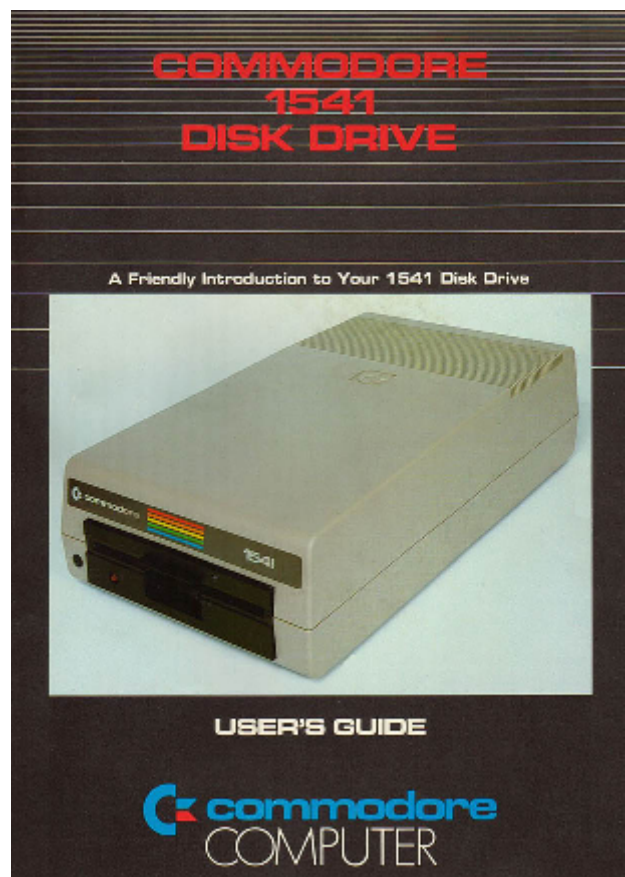
Disclaimer

I've tried to be careful to ensure you don't have any problems, but any damage:

- A parallel port damaged
- A 1541 drive damaged
- A 1541 drive head damaged
- A disk damaged

Is your own responsibility and liability.

And Commodore Free doesn't have any responsibilities or liabilities for printing the article





<http://www.cmdweb.de/>

Here you find information about the [SuperCPU](#), [FD disk drive](#), [HD series](#), [JiffyDOS](#) and [RAMLink](#). To order, pick your closest [distributor](#).

Furthermore, you can visit our [About CMD](#) section or the [Download area](#).

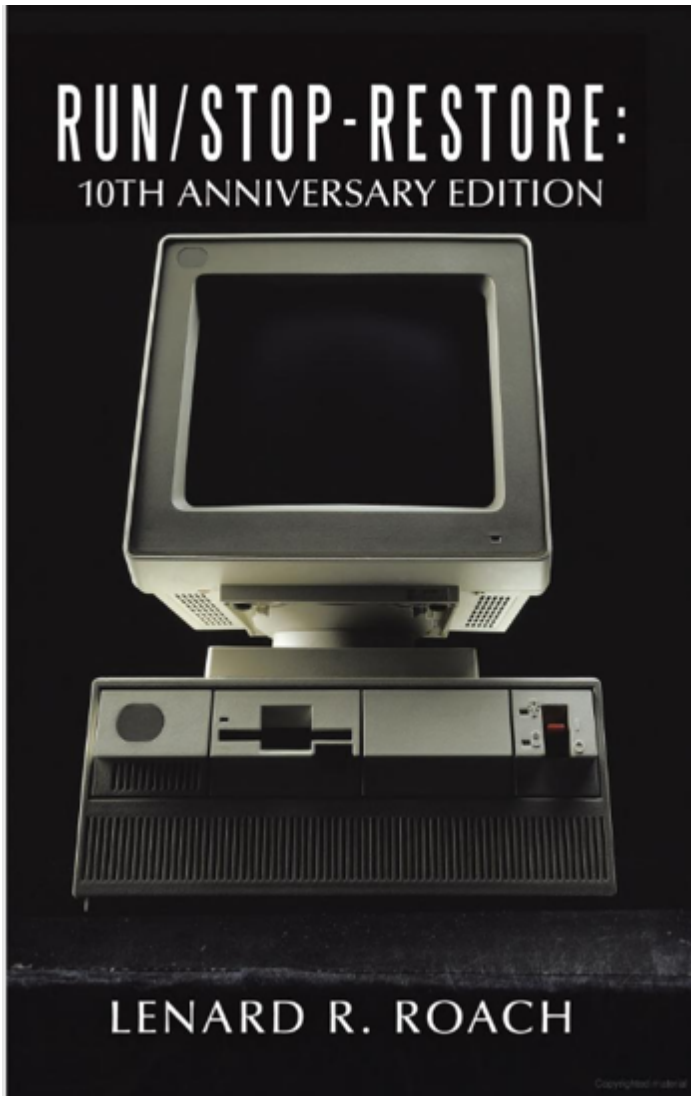


www.commodorecomputerclub.co.uk/

Welcome to the home of the **Commodore Computer Club UK**. We have set up to support all popular Commodore 8-bit computers, from the Commodore (CBM) PET range, through to the C128, so if you have a C64, VIC-20, C16, Plus/4 or even a C64DTV, and would like to know more about the club, please read the [club FAQs](#) or visit [our forums](#) if you have any further questions.



RUN/STOP-RESTORE BY LENARD R. ROACH BOOK REVIEW BY COMMODORE FREE



<http://www.abebooks.co.uk/9781456719241/RunStop-Restore-10th-Anniversary-Edition-Roach-1456719246/plp>

<http://www.uread.com/book/runstoprestore-lenard-r-roach/9781456719241>

And other online retailers in various formats

Product Details

- Paperback: 132 pages
- Publisher: AuthorHouse (March 10, 2011)
- Language: English
- ISBN-10: 1456719246
- ISBN-13: 978-1456719241
- Product Dimensions: 6 x 9 x 0.3 inches
- Shipping Weight: 7 ounces

HISTORY

A book that took 10 years to make! A book about a bygone era of computing that never really rolled over and played dead, more like dug a tunnel and went underground. Here is a modern collection of ancient writings about a computer thought of as extinct-- the Commodore! Relive or discover for the first time what it was like to use and work with the best selling single board computer in history through the eyes of one who still admires its complex simplicity.

COMMODORE FREE

The question on this review is "where to start!" The beginning is an obvious start and after reading the book a few times I wrote the following about it:

This book isn't going to be to everyone's liking, that's for sure; if you are a power Commodore user and expecting some sort of programming book looking at memory addresses and locations then sadly this isn't the book for you. If you are new to Commodore and can't or don't remember the early days and really do wonder "what the internet was like on a 300 baud modem" then I suppose this may be a read for you. However if you do remember 300 bps modems you may wish to quickly forget all about them, although to be fair the Bulletin board services were predominantly text based; and as such seemed faster than loading some webpages on a broadband connection of the current day.

HOUSE TOUR

The book starts out with a tour of the author's house and its various locations, there usages and placements relative to other rooms in the house. I found this is an unusual start to a book of any description. This is also an unusual start to a book about commodore and its history. This book is what I would personally call an "Experience book". By that comment I mean A look back at Commodore history from one users perspective; and

Sample chapters of the book can be viewed here

http://books.google.co.uk/books/about/Run_Stop_Restore.html?id=inSoymyQC9EC&redir_esc=y

And the book can be purchased from These outlets

<http://www.amazon.com/Run-Stop-Restore-Anniversary-Lenard-Roach/dp/1456719246>

http://www.waterstones.com/waterstonesweb/products/lenard-r-roach/run_stop-restore/8514784/

http://ebookstore.sony.com/ebook/lenard-r-roach/runstop-restore/_/R-40000000000000394012

<http://www.tower.com/run-stop-restore-10th-anniversary-edition-lenard-r-roach-hardcover/wapi/118016339>

his computing progression through the years using a Commodore computer, his joys at getting applications to work and frustrations as to why something wouldn't. "SYNTAX ERROR?"

The book has been rewritten for the 10th anniversary and as such various amendments have been added, some are minor grammatical changes some are complete rewrites to the text. The author looks at using Commodore machines to enter cyberspace at the breakneck speed of 300 baud and how he connected to bulletin board servers using the humble phone line, no actual details of how to connect just experiences of the events. The author writes about "his" personal first Bulletin board service that he ran as a sysop, how the equipment was acquired, one night from a shifty stranger (you need to read the story) and how it was all taken down into his basement; setup and configured and how once working it soaked up so much electricity that the authorities came knocking on his door thinking he was a "pot" grower (again you need to read the story of the Pulpit, the book also describes why the name was picked and what the BBS would stand for. Then after a prank caller came pestering people on his BBS the author decided that the service must come to an end and pulled the virtual plug on the service.

We learn from the writer that baseball cards and paper clips when inserted by the writer's sons into various Commodore disk drives do not really aid to the drives function in any way! And we learn that cats of any breed are the same, in all countries; they all try to sabotage machines and walk all over the keys, brushing up against piles of disks and sending them crashing to the floor.

The whole of the book was written using Geowrite and the GEOS operating system for the Commodore 64, spell checking provided by GEOSpell, then the book was converted to word format by Wrong is write. So it does show you what can be done with a Commodore machine. Heck with enough dedication the Commodore can create a whole book ready for publishing, still no mean feat by any standards.

The Author writes about his early days with Goes and the constant error messages about GEOS desktop being missing, how he overcame this problem like others have done with a simple copying of 1 file (the GEOS desktop). The user writes about the many machines he saw being scrapped because of the Year 2000 bug and how he continued to use his Commodore even after the dooms day came and went. Of course even Commodore machines didn't escape totally unscathed from the Year 2000 bug, GEOS itself and applications needed some patching for the date problem.

We learn about the writer's love of programming and how some of his applications were published like "Checkit out" published in run magazine, and how the author felt about the whole process. We also hear about how copyright has to some effect killed one of the writer's applications and how he no longer feels he can change /amend or reinvent the application for fear of repercussions from the copyright holders he sold the software to.

The author looks at some of the Commodore games specifically written for girls and finds out that not to many games for girls

existed. We also learn of the author's hatred of the various screen savers that appear on other machines and how he has always used the "off" screen saver IE turn the monitor off as it saves the screen and also saves some power.

Amusing and witty, although I wouldn't say it's a book that "must" read, from the links you can see on google books for example you can preview various chapters

I asked bill a couple of questions related to the book

Q. How was the book written what hardware and software were used

It was written on a Commodore 128 using GEOS 2.1,

Q .was it still re worked on Commodore

The original text was done on the Commodore, but the publisher couldn't work with GEOS text so I used Wrong Is Write and Big Blue Reader to convert the text out of Commodore and into Word which is something they could work with.

Q. how was the text laid out for the format

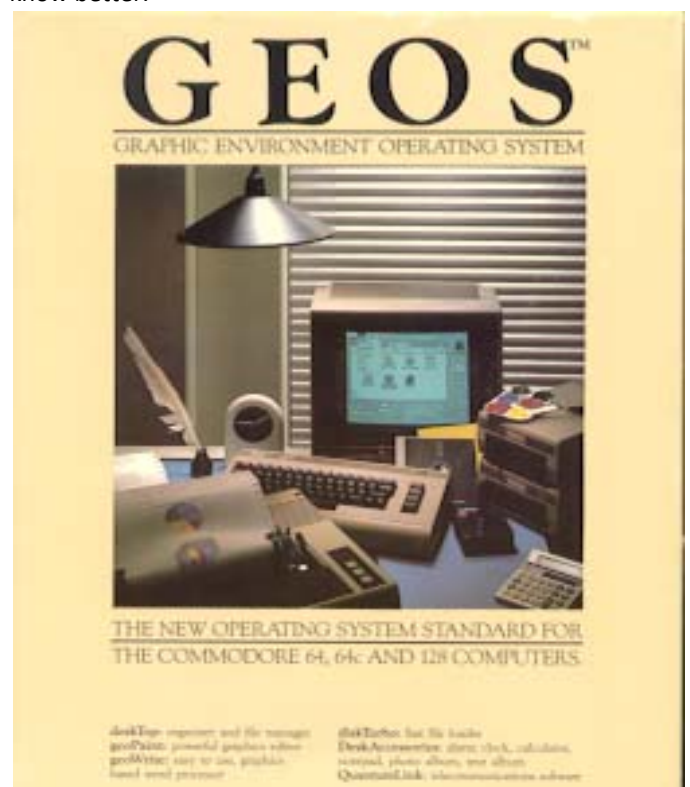
The text was laid out to run 5 by 7, this was I could get the book done in Hardbound for free.

Q. How was it then sent for printing

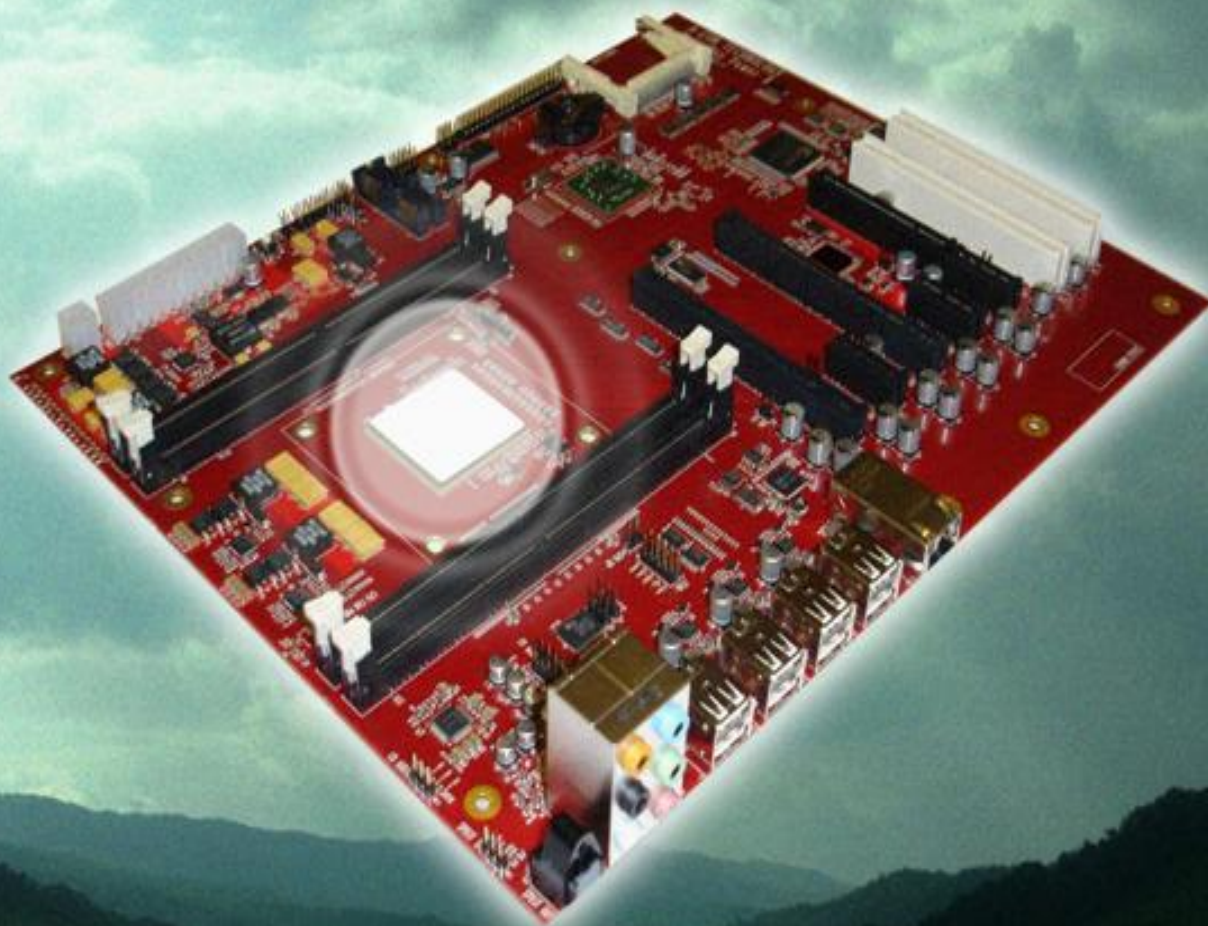
It was sent to the publisher through the email as an attachment, formatted for the most part the way you have it there

Q. who created the cover etc

The cover was stock item found in Authorhouse's archives. As a noob to the publishing gig, I thought I had to live with what they sent. I was given too much freedom to work and didn't know what to do with it. When I publish my second book, I will know better.



AmigaONE



X10000
I BELIEVE

**On the Way to Fun:
An Emotion-Based Approach to
Successful Game Design**

Roberto Dillon (Author)

Product details

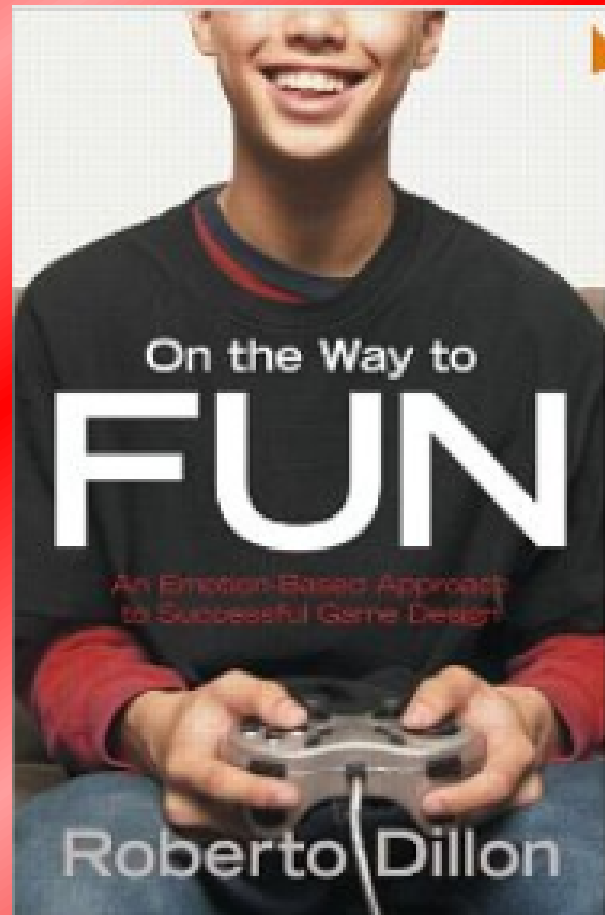
Paperback: 200 pages

**Publisher: A K Peters/CRC Press
(8 Mar 2010)**

Language English

ISBN-10: 1568815824

ISBN-13: 978-1568815824



**Golden Age of Video Games:
The Birth of a Multi billion
Dollar Industry [Paperback]**

Roberto Dillon (Author)

Product details

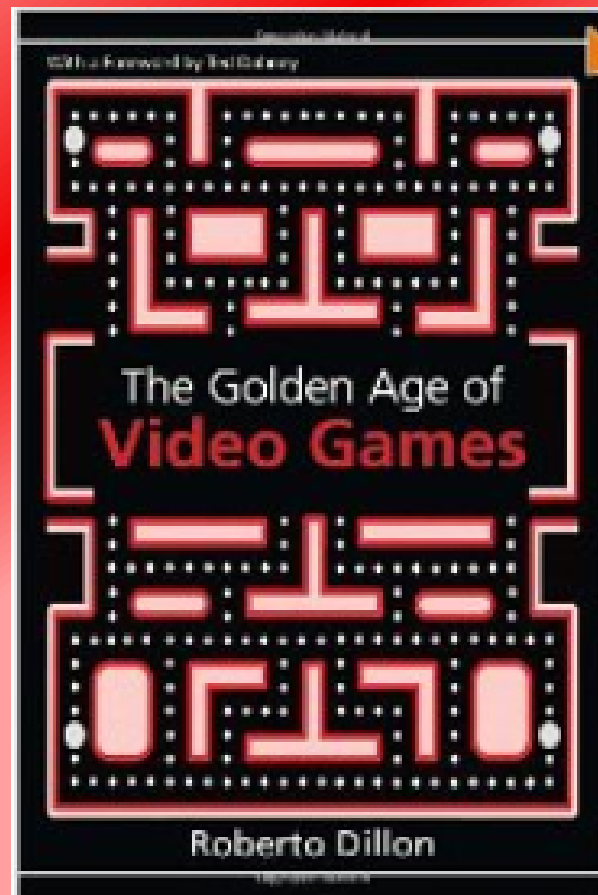
Paperback: 209 pages

**Publisher: A K Peters/CRC
Press; 1 edition (28 April 2011)**

Language English

ISBN-10: 1439873232

ISBN-13: 978-1439873236



Issue 58

2012

Editor
Nigel Parker

Spell Checking
Peter Badrick

Text & HTML Conversion
Paul Davis

D64 Disk Image
Al Jackson

PDF Design
Nigel Parker

Website
www.commodorefree.com

Email Address
commodorefree@commodorefree.com

Submissions

Articles are always wanted for the magazine. Contact us for details .We can't pay you for your efforts but you are safe in the knowledge that you have passed on details that will interest other Commodore enthusiasts.

Notices

All materials in this magazine are the property of Commodore Free unless otherwise stated. All copyrights, trademarks, trade names, internet domain names or other similar rights are acknowledged. No part of this magazine may be reproduced without permission.

The appearance of an advert in the magazine does not necessarily mean that the goods/services advertised are associated with or endorsed by Commodore Free Magazine.

Copyright

Copyright © 2011 Commodore Free Magazine
All Rights Reserved.