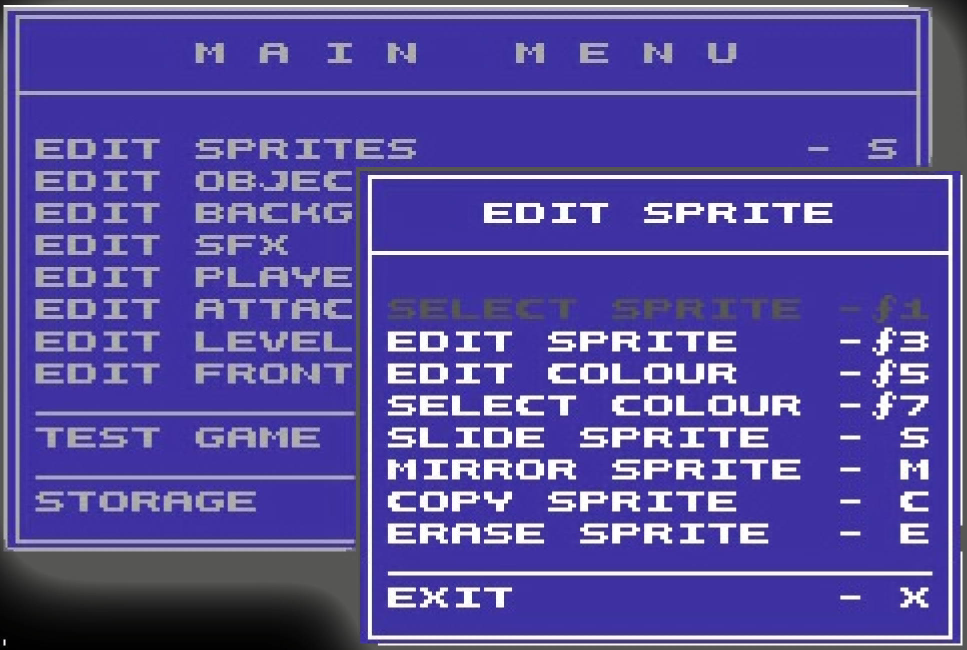
Commodore Free

A free to download Magazine dedicated to Commodore computers.

Issue 96

An Interview with Martin Piper of the SEUCK Redux Project



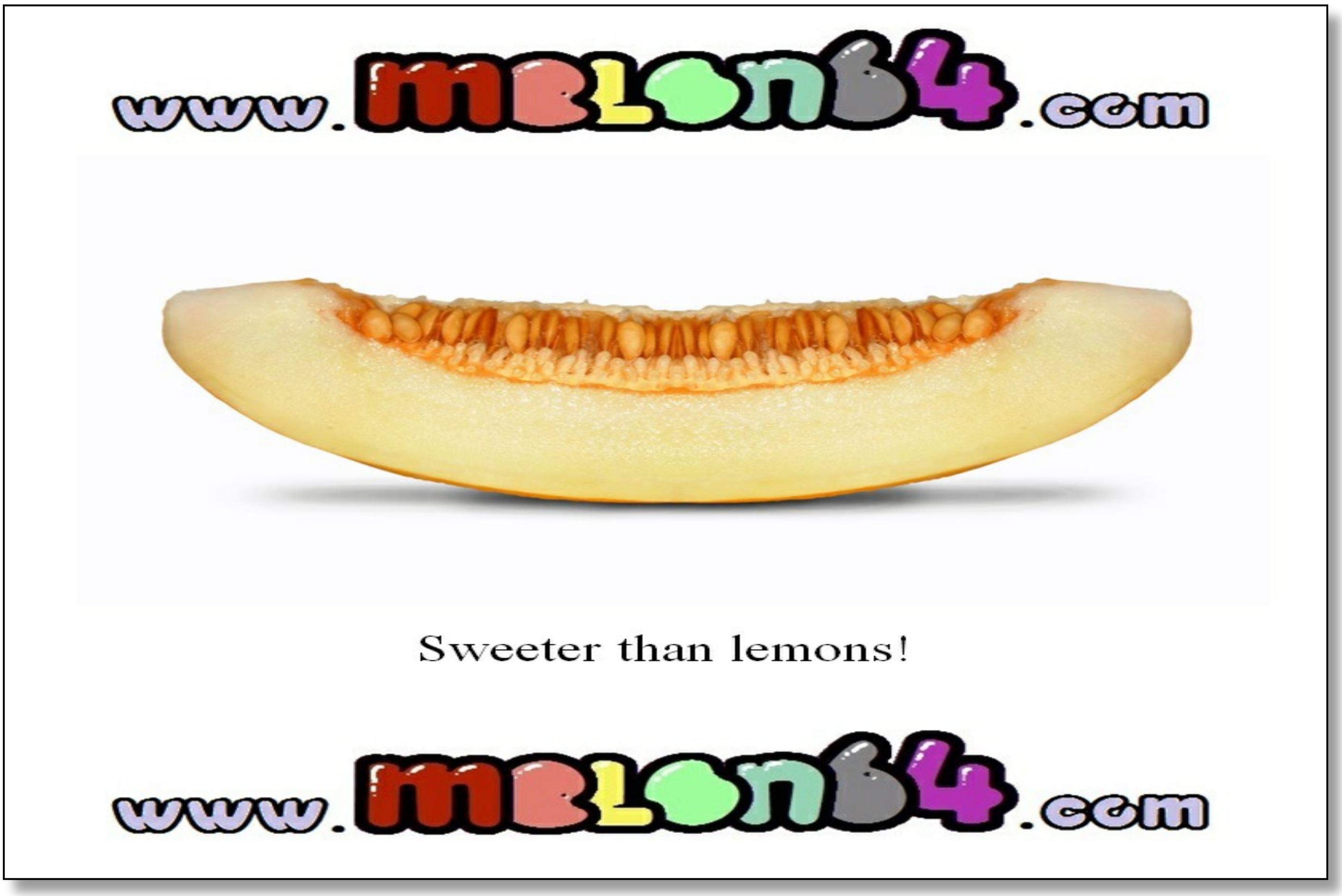
```
        M A I N   M E N U

EDIT  SPRITES              -  S
EDIT  OBJEC
EDIT  BACKG
EDIT  SFX
EDIT  PLAYE
EDIT  ATTAC
EDIT  LEVEL
EDIT  FRONT

TEST  GAME

STORAGE
```

```
        EDIT  SPRITE

SELECT  SPRITE      - ƒ1
EDIT  SPRITE        - ƒ3
EDIT  COLOUR        - ƒ5
SELECT  COLOUR      - ƒ7
SLIDE  SPRITE       -  S
MIRROR  SPRITE      -  M
COPY  SPRITE        -  C
ERASE  SPRITE       -  E

EXIT                -  X
```

C64 Code Hacking

SuperBoot v1.00
for SuperCPU

CONTENTS



Editorial

Another year passes and another starts
Welcome to 2017 !

In the next year I hope to finally get back on schedule with production of Commodore Free, as I seem to have been sadly slacking last year. However, work commitments and real life got in the way and overtook my free time.

I had offers from a programmer and disk magazine creator. I won't go into further details at this time as I have had offers in the past that didn't materialise. This offer is from a very competent Coder and very active member of the Commodore community, so with fingers and toes crossed I hope to bring you more news when I have something solid to write about.

This issue sees an exciting look at Superboot (for the SCPU) from Bert. One thing the SCPU lacks is decent documentation, so any work with the SCPU is or should be greatly received from the Commodore community. Let's face it -- we don't have many packages that fully take advantage of the SCPU`s power and versatility. It would be nice to say that production of the CMD devices in general has seen a turn for the better, but I can't make those statements (more is the pity) and when I do I seem to jinx any further progress.

So this issue we have a few interviews, another section on coding by an active member of the community, and the usual news and updates.

Without further adieu let's roll out another year and welcome back. Oh, and thanks for reading!
Readers' comments seem to have stopped, and I am open to suggestions (good or bad), preferably not rude or abusive!

Regards
Nigel (editor)

www.commodorefree.com

Editor
Nigel Parker

Spell Checking
Peter Badrick
Bert Novilla

TXT, HTML & eBooks
Paul Davis

D64 Disk Image
Al Jackson

PDF Design
Nigel Parker

Contributors
John Fielden
Damian Caynes/Slartibartfast
Joonas Lindberg
Bert Novilla

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

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!

C64 Forever
by CLOANTO

Online Order: www.arcaderetrogaming.com

More Information: www.mcc-home.com

ARG

General News

Retro Asylum Podcast 147

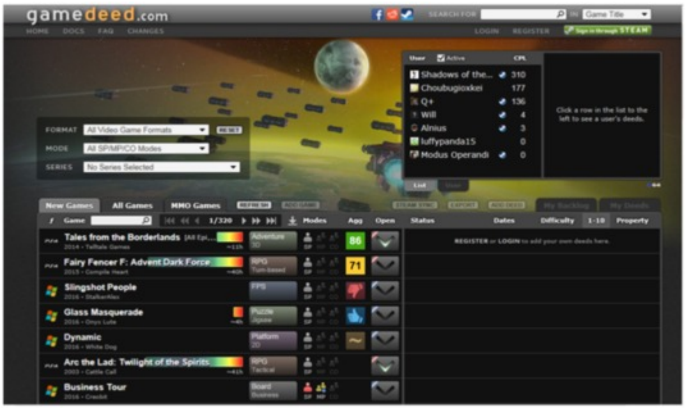
Retro Podcast Asylum is an English podcast for the retro computer fans. In this episode of the podcast are: Dean Swain, Mads Kristensen, & Steve Erickson talk about the current trend of modernizing popular 8-bit systems of the past.

http://retroasylum.com/

Gamedeed.com

Gamedeed.com is a database for all formats of computer games. You can find anything, print screens, articles, personal high scores, and more. The database contains more than 18,000 games

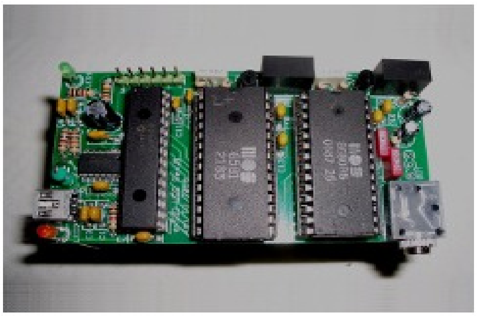
http://gamedeed.com/


gamedeed.com
HOME DOCS FAQ CHANGES
LOGIN REGISTER
FORMAT All Video Game Formats
MODE All SP/MP/CO Modes
SERIES No Series Selected
New Games All Games MMO Games
Shadows of the... 310
Choubugioxkei 177
Q+ 136
Will 4
Alnius 3
luffypanda15 0
Modus Operandi 0
Click a row in the list to the left to see a user's deeds.
Game 1/320 Modes Agg Open Status Dates Difficulty 1-10 Property
Tales from the Borderlands 86
Fairy Fencer F: Advent Dark Force 71
Slingshot People
Glass Masquerade
Dynamic
Arc the Lad: Twilight of the Spirits
Business Tour
REGISTER or LOGIN to add your own deeds here.

exSID USB

Thibaut Varene is working on exSID with this device. It's possible to play SID music from a modern PC, with real SID hardware. Features: 6581 and 8580, 44.1 kHz digis, and nearly cycle-accurate playback.

`http://hacks.slashdirt.org/hw/exsid/`



Retro computer event

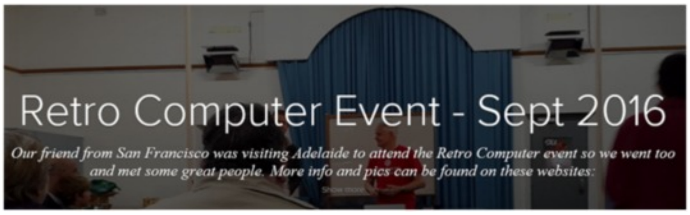
Pictures from the Retro Event The poster said their highlights from the event were meeting Jane M. Jones from Grafix
found here...
https://www.flickr.com/photos/flitterg/sets/72157674915146695/

Other pics (by other people) are here:

https://www.facebook.com/adelaideretrocomputing

http://adelaideretrocomputing.blogspot.com.au/2016/09/september-2016-meeting-report.html

http://amigax1000.blogspot.com.au/2016/09/adelaide-retro-computing-meeting.html



Retro Computer Event - Sept 2016

Our friend from San Francisco was visiting Adelaide to attend the Retro Computer event so we went too and met some great people. More info and pics can be found on these websites:

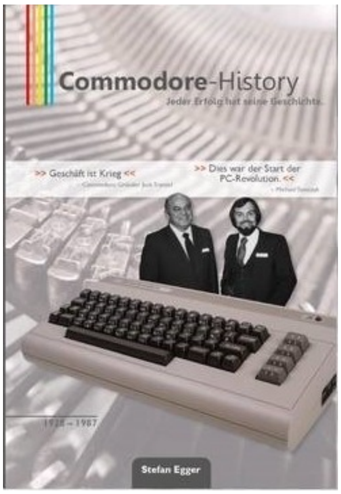
Kilobyte magazine 2

Kilobyte magazine is a pdf magazine in English for 8-bit home computers and game consoles. In this issue: Alan Sugar (Amstrad), diskette cases, NES, TED (FPGA), Jens Schönfeld, Gamescom, Shotgun, Barnsley Badger, Eye of the Gods, Maze of the Mummy and My Life

https://issuu.com/kilobytemagazine/docs/kilobytemagazine2016-2

Commodore History

Stefan Egger is writing a book about the history of Commodore. Stefan has already started an Indiegogo campaign to fund the book. The book will contain all the Commodore and Amiga models and interviews with Michael Tomczyk, Hedley Davis, Bil Herd, Chuck Peddle and Yash Terakura.

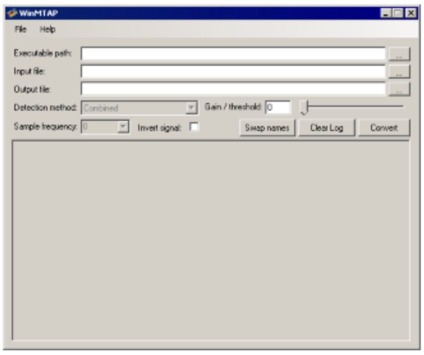

Commodore-History
Jeder Erfolg hat seine Geschichte
>> Geschäft ist Krieg <<
>> Dies war der Start der PC-Revolution <<
Stefan Egger

https://www.indiegogo.com/projects/commodore-history-computer--2#/

WinMTAP 1.0

Gaia has released WinMTAP 1.0 This is a Gui Windows frontend for TAP2WAV. The program makes it easy to convert TAP files for the Commodore 8-bit computers.

http://plus4world.powweb.com/tools/winmtap



Detection method: Combined

Gain / threshold: 0

Sample frequency: 0

Invert signal: □

C2NEmu

Luigi Di Fraia is working on C2NEmu. This device connects the Commodore 64 / VIC 20 Datassette to a PC via a USB cable giving the user the ability to "playback" a TAP file (PC to C64 / VIC20) and "record" TAP files from (C64 / VIC20 to PC).

http://www.lemon64.com/forum/viewtopic.php?t=61717

Homepage
http://www.luigidifraia.com/

Return Issue 26 and 27 released

Return magazine is a German print magazine on the 8-bit computers of A (Atari) to Z (ZX Spectrum). In issue 26 were: Reshoot, Legend of Zelda, D42 Adventure System, Secret Of Mana, Jocopod, Games Magazines from the 80s and 90s (1) Tronimal, Amstrad NC-100 / NC200, Vintage Computing Festival, Scooby Doo Mystery, Bobby Bearing Wonderland Dizzy, Atari Multijoy, Shadow of the Beast II, Pooyan, and Joust.

In issue 27 were the following articles: Lemmings, Secret of Evermore (SNES), Maze of the Mummy (C64), Zeit der Stille (C64), Philips CD-i, Indiana Jones Greatest Adventures (SMD), Rainbow Six: Black Ops (PC), Keyman64, Game magazine from the 80s & 90s (2), AmigaOS (6), Old Bits, Pixelkunst, Crystal Castles, Forgotten Worlds, and Dr.Future - Auf Wiedersehen Monty

https://www.return-magazin.de/

COMMODORE ATARI NINTENDO SEGA AMIGA SONY DOS ARCADE
AUSGABE 27
RETURN
FASZINATION KLASSISCHE COMPUTER UND KONSOLEN
SECRET OF EVERMORE
INDIANA JONES GREATEST ADVENTURES
LEMMINGS
NEUE SPIELE FÜR C64, ATARI & AMIGA
FORGOTTEN WORLDS

JSIDPlay2 - v3.7

A Commodore C64 cycle-exact emulator, optimized for reproducing the SID sound. It is also a SID player for music from a music collection such as HVSC and CGSC. This version fixes some bugs.

https://sourceforge.net/projects/jsidplay2/files/jsiddevice/

Monty Python argument sketch performed by 2 speech synthesisers

Michael Palin is played by the Dectalk Express (Perfect Paul modified to sound more like Stephen Hawking) John Cleese is played by the Intex Talker (Votrax SC-01A) And there you have it. The Monty Python's Argument Sketch performed with two vintage speech synthesizers.

https://www.youtube.com/watch?v=WjMwGWdqHVQ

Commodore software website update

The web page Commodore software has been updated with plenty of new items. Recent entries include: IDE64 v4.1 Release, Catweasel MK4 Short Manual, and Mini X-Pander - Arkanix Labs

http://commodore.software/

COMMODORE . SOFTWARE
Utilities, Apps & More for Your Commodore 64, 128, 16, Plus 4, VIC 20 & PET Computer

History of Personal Computing: Jason Compton & Jason Timmons

A new episode of the Personal Computing podcast is available.

The website says:
Today we'll be further covering the personal digital assistant or PDA, with the Apple Newton clones or "killers," the **Casio Zoomer** and **Amstrad PenPad**. This is the podcast where we take an informal look at personal computing history through the lens of eBay auctions. It's sort of like *Antiques Roadshow*, but all about antique personal computers.

- Replacement C-64 Motherboards
https://icomp.de/shop-icomp/en/shop/product/c64-reloaded.html

http://www.nytimes.com/1993/11/09/science/personal-computers-zoomers-newtons-real-life-so-far-promise-exceeds-usefulness.html

ftp://ftp.oldskool.org/pub/tvdog/tandy1000/faxback/02502.txt

- https://en.wikipedia.org/wiki/PenPad

Homepage of the podcast
http://www.historyofpersonalcomputing.com/

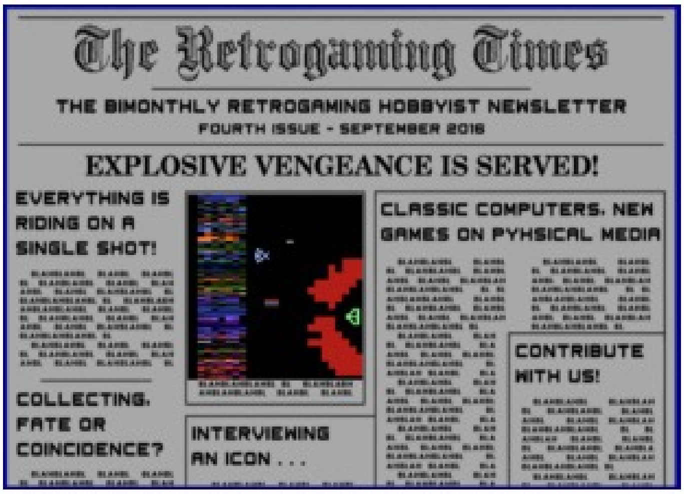
The History of Personal Computing
The Story of the Personal Computer in Audio & Pictures

The Retrogaming Times issue 4

The Retrogaming Times is an English retro computer magazine available as an html webpage. In this edition:

- More C64! - Recent Releases
- Retrogaming Feels - Yars' Revenge
- Apple II Incider - Sundog: Frozen Legacy
- A Pirate's Life For Me - Third Journey
- The Controller Chronicles - NES Controller
- Show Report - California Extreme 2016
- The Paths We Are Lead Down
- TriForce Johnson Interview
- and more...

http://www.classicplastic.net/trt/



The Retrogaming Times

THE BIMONTHLY RETROGAMING HOBBYIST NEWSLETTER

FOURTH ISSUE - SEPTEMBER 2016

EXPLOSIVE VENGEANCE IS SERVED!

EVERYTHING IS RIDING ON A SINGLE SHOT!

CLASSIC COMPUTERS, NEW GAMES ON PYHSICAL MEDIA

CONTRIBUTE WITH US!

COLLECTING, FATE OR COINCIDENCE?

INTERVIEWING AN ICON . . .

Commodore back in Germany

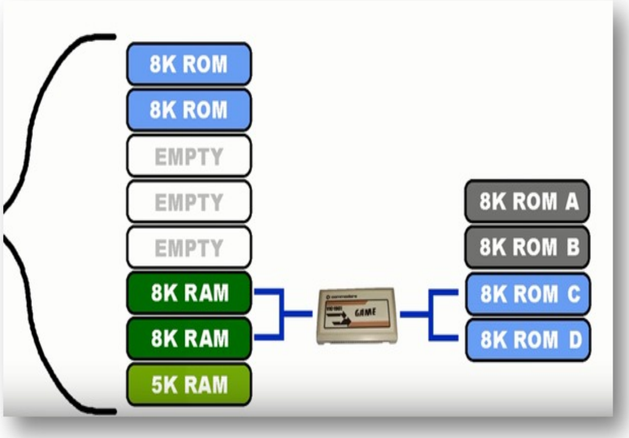
After a pause of several years, fans of classic computers worldwide can look forward to a series of Commodore PCs. The company "Individual Computers GmbH", based in Germany, has licensed the rights of the Commodore trademarks from Polabe Holding N.V.

https://icomp.de/shop-icomp/en/33/items/commodore-back-in-germany.html

How ROM carts work

If you ever wondered how a ROM Cartridge plugged into your computer worked then this is a good place to look. Explained in simple steps covering the VIC, C64, and Atari systems (amongst others). This easy to watch video might help you understand the theory of why cartridges worked and how, and why the C64 had a 16k limitation, and what bank switching was.

https://www.youtube.com/watch?v=NLEMsw1SjDY



Star Trek 25th Anniversary Playthrough: Complete

The complete playthrough of the game Star Trek 25th Anniversary. Join Captain Kirk and company in Star Trek 25th Anniversary. A thrilling adventure game from...youtube.com

https://www.youtube.com/watch?v=IQ6frFkKZYs

CAPTAIN'S LOG

THOUGH THE ENTERPRISE'S PRIMARY MISSION IS PEACEFUL EXPLORATION, THE GALAXY HOLDS MANY SURPRISES. TO BE PREPARED, WE ARE CONDUCTING A MOCK BATTLE WITH THE U.S.S. REPUBLIC.

Steve Hammond interview

The Retro Video Gamer had an interview with Steve Hammond. Steve is a founder of DMA Design. DMA designs has, among other things, developed the classic computer game Lemmings.

http://www.retrovideogamer.co.uk/index.php?option=com_content&view=article&id=496:rvg-interviews-steve-hammond&catid=91&Itemid=160

Eight Bit Magazine 2 (Amstrad, Commodore, Sinclair & more)

Eight Bit Magazine will celebrate early eight bit microcomputers from the 1970's to the early 90's. While most publications cover only the gaming aspect of such computers, we want to cover the hardware and programming side of things as well. After a successful Issue 1 Kickstarter https://www.kickstarter.com/projects/8bitmagazine/eight-bit-magazine we are delighted to offer Even more in Issue 2... You will get a full colour, A5 size, 68 page printed magazine full of great articles written by people with a wealth of knowledge and passion for those computers. Expect programming, technical, and gaming articles for the Commodore 64, Amstrad CPC, Sinclair Spectrum, Atari and more!

https://www.kickstarter.com/projects/8bitmagazine/8-bit-magazine-issue-2-amstrad-commodore-sinclair

THE COLLECTORS GUIDE TO THE:

commodore 64

The cheap 8-bit computer that ended up as the world's number 1 best selling computer. Here's the collectors guide to the Commodore 64

Usborne books download

Remember all the type-in listings you frustrated over, finishing pages of code, then ran the program and found **Syntax Error in line 1 ?** Well, now you can relive that wonderful time by downloading a number of classic Usborne books legally from their website. What I remember mainly from these books was the artwork that accompanied the listings.

The website says:

Usborne 1980s computer books

Many of today's tech professionals were inspired by the Usborne computing books they read as children. The books included program listings for such iconic computers as the ZX Spectrum, the BBC Micro and the Commodore 64, and are still used in some computer clubs toda

Free to download PDF books

https://usborne.com/browse-books/features/computer-and-coding-books/

New Commodore websites online

The Pacific Commodore Expo NW now has a website.

Go to
http://www.portcommodore.com/pacommex

There is a new website for the Southern California Commodore & Amiga Network. Go to

http://www.portcommodore.com/sccan

PaCommEx Pacific Commodore Expo

June 10-11, 2017 at the Living Computer Museum, Seattle, WA

RETRO PROGRAMMER

Quite an interesting YouTube channel with some Commodore related information. It may be an easier way for you to learn things such as memory maps and what those strange poke and peek commands actually do to your machine.

https://www.youtube.com/channel/UCADySP7Hy8TxgfDEe2GZQyw

A few videos but

Commodore 64 Part 1: How The Memory Map Worked
https://www.youtube.com/watch?v=qibJpjJ0sdM

Commodore 64 Part 2: Intro to 6502 Machine Language
https://www.youtube.com/watch?v=ozKcF6Q6DhI

Commodore 64 Part 3: Intro to 6502 Assembly
https://www.youtube.com/watch?v=vMwH9AkXqcE

Vegas Expo videos

A number of videos from the Commodore Vegas Expo v12 have been posted online head to

File transfer C64-to-Linux
https://youtu.be/jNp6tm-rTAI

Chalk Board PowerPad for the Commodore 64
https://youtu.be/OI-fYOtOLpw

CommVEx cake and Bill Seiler reports on his PET 8032 repair
https://youtu.be/sgQYQ0MYHoM

Ricardo Quesada and the UniJoystiCle for the C64
https://youtu.be/LDTGW7ZXje8

CommVEx v12 2016 pre-show and Mazzei mods a VIC-20
https://youtu.be/LqYGLDO2b1k

Drew and the FPGA Arcade Replay, Vampire 500/2000, modded SX-64
https://youtu.be/rBRZtgDvhjQ

CBM engineer Bill Seiler at CommVEx 2016
https://youtu.be/YVFFBTvFRIQ

Actor Todd Bridges and his Commodore connection
https://youtu.be/-rdhlrzO2Yg

Estel and the Font Resource Directory
https://youtu.be/yuorAKDX_Rk

Maurer and the C64 ATX v3
https://youtu.be/1_7wD0LieCA

CommVEx
Commodore Vegas Expo
July 30 & 31, 2016

Update from Robert Bernardo
Fresno Commodore User Group
http://www.dickestel.com/fcug.htm

MEGA65 NEWS

- VHDL codebase almost feature complete.
- Dedicated FPGA board designed.
- Case re-engineered and 3D-modeled, and prototype produced.
- Keyboard and case supply chain clarified.
- GEOS support secured.
- Several fully assembled prototype machines exist.

Mega65 initially planned to launch our fundraising campaign in Q3 2016, but things are not quite there yet.

Quality comes first
The first and most important reason for having to keep you waiting longer is that we want to deliver a product that lives up to our extremely high quality standards. This will be turnkey-ready, bundled with an amazing software package. Connect it to your television set and get started right away. To achieve this more time, more hard work, and further community support are required.

New ideas included
The second reason deliberately allowed for a certain amount of feature creep. This means that when finished, you will receive a much more sophisticated piece of equipment:

- A more powerful FPGA on custom motherboard.
- Improved C64 compatibility (including 6502 illegal opcodes).
- Legacy device support (1581, 1541, and so on).
- GEOS-based configuration menus.
- SD card support.

- “Under Construction” demoscene party - between Xmas 2016 and new years eve - Gernsheim, Germany (http://under-construction.tum-party.net)
- “Retro-Börse” retro trading fair - April 2017 - Mannheim, Germany (http://retro-boerse.de/mannheim)
- “Revision” demoscene Easter party - 14.-17. April 2017 - Saarbrücken, Germany (https://2016.revision-party.net)
- “Evoke” true color demoscene party - August 2017 - Cologne, Germany (https://www.evoke.eu/2016)
- “Maker Faire Bergstraße” creativity festival - 16./17. September 2017 - Bensheim, Germany (https://maker-faire.de)
- “Retro-Börse” retro trading fair - October 2017 - Mannheim, Germany (http://retro-boerse.de/mannheim)
- “Amiga32” Amiga retro celebration - 28. October 2017 - Neuss, Germany (http://amiga32.de)

You want to contribute?
The project is completely based on voluntary work and an extra pair of hands is always welcome. If you can spare some time to contribute to the exciting work we are doing, please do not hesitate to connect to us. What we offer you is a great 8-bit team spirit and the eagerness to make MEGA65 fly. Of course, you will also receive credit and fame on the go, and especially once we go to market successfully!

Thank you, The MEGA65 Team
http://mega65.org/

MEGA65

RETRO Innovations:

A number of new products have been released on the Retro innovations website. If you haven't visited yet I would advise you to look.

http://store.go4retro.com/

The Retro Hour - Jim Sachs

The Retro Hour is a Podcast created by Dan Wood & Ravi Abbott. The Retro Hour aims to help the world understand the European gaming industry. In this episode: Jim Sachs, who was involved with Defender of The Crown, Saucer Attack, and the CDTV's user interface.

http://www.theretrohour.com/cinemaware-and-amiga-art-with-jim-sachs-the-retro-hour-ep31/

Sprite Castle

Rob "Flack" O'Hara has a podcasts website. You will also find the links to web pages mentioned in the podcast. The most recent games are: Ghostbusters (Activision, 1984) and Complete Computer Fireworks Celebration Kit (Activision, 1985).

http://spritecastle.com/

Bunkerparty 2016

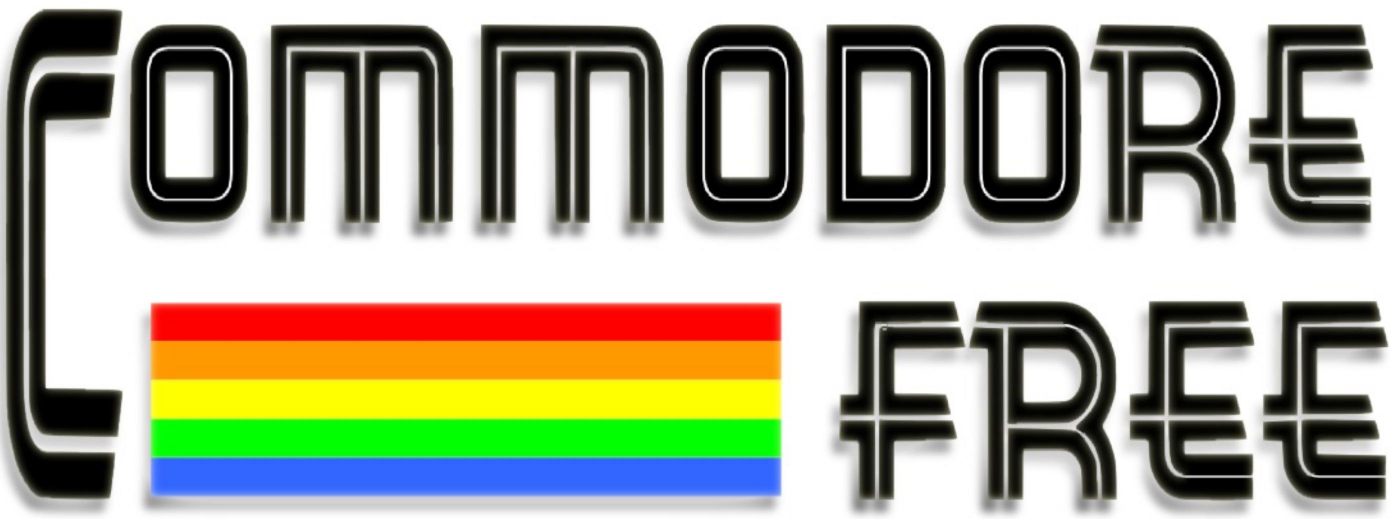
Although this is now old news I put it in for reference. Dienstagstreff organises a Commodore meeting from 9 until 11 September in Bochum - Germany. You can stay for three days, with food and drinks included in the entrance fee. You can also participate in the Mixed-Compo (demos, graphics or games).

http://www.dienstagstreff.de/veranstaltungen/2016/index.php

Karl Hörnell interview

C64.com had an interview with Karl Hörnell, who worked on the following games for the Commodore C64: Fruity, Ronald Rubberduck, Velocipede, Clean-Up, Velocipede II, Fungus, Melonmania, Toad Force, and Clean-Up Service.

http://www.c64.com/



Visit

AltiWorld BBS

Features a totally unique BBS Experience driven by WorldBankBBS Software.

Web Portal:
www.altiworld.com

ALWAYS ONLINE!

Friendly Users
Online BBS Gateway
Integrated IRC Client
Massive Subboard
Full-Screen Editor for PETSCII/ANSI

King Durin wants to see you there today!

ALTiWORLD

bbs.altiworld.com:6400

powered by:
WorldBankBBS

Sysop:
King Durin

ALTiWORLD

bbs.altiworld.com
port 513
sysop: King Durin

powered by: WorldBankBBS

Visit Now!

bbs.altiworld.com

Ports
PETSCII: 6400
Ansi: 513

Commodore 64 News

Hot Style Issue 4

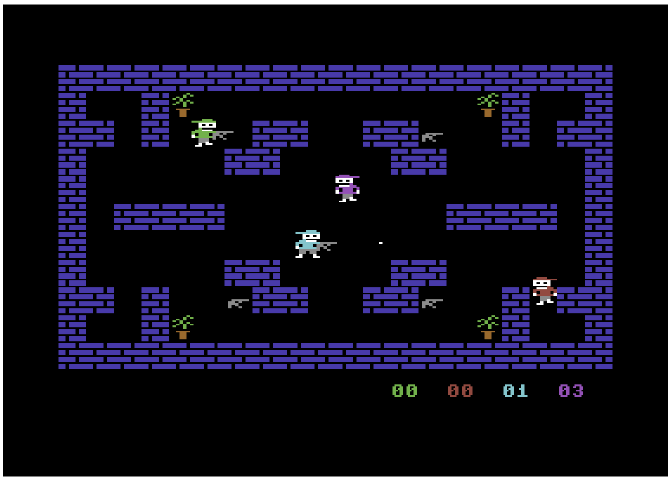
Hot Style is a Polish-language disk magazine for the Commodore C64. In this latest edition are: Decrunch 2016 and Stary Piernik 11. The SIDs in this edition are: Bzykojazz, Cover 2, Janosik, Modula, Noisy Pillars, and Smalltown Boy

http://csdb.dk/getinternalfile.php/151009/Hot_Style_issue_4_TX_DRC.zip
http://csdb.dk/getinternalfile.php/151011/Hot_Style_issue_4_TX_DRC_full.7Z
http://csdb.dk/getinternalfile.php/151010/HS#4_cover_by_Sekator_2016.zip

Shotgun - C64 - Boxed

Shotgun is now available in a deluxe-edition, boxed with a flip cover, 5.25" disk, printed manual, and three extras. The game can be played with four players. You have to collect guns and shoot at each other, but each gun only has 3 bullets. Great for a Commodore night in, as long as you only have three other friends that is !

http://shotgun.drwuro.com/

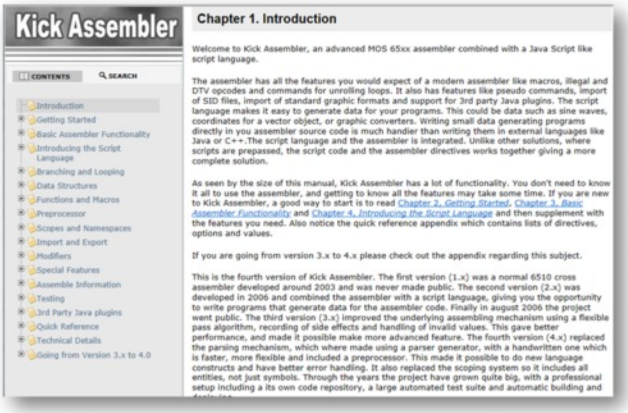


Kick Assembler updates

In the last few months there have been a few updates for Kick Assembler. The most recent are:

New release (v4.4):
- New .errorif directive. (Better to use than asserts during cross branch testing)
- Bugfix: Corrected error with referencing variables in the '.import c64' directives. Thanks to Bastiaan Winde for pointing this out.
- Typo fix in the manual (Thanks again to Bastiaan)
- Misc internal refactoring

http://www.theweb.dk/KickAssembler/Main.html#frontpage



Chapter 1. Introduction

Welcome to Kick Assembler, an advanced MOS 65xx assembler combined with a Java Script like script language.

The assembler has all the features you would expect of a modern assembler like macros, illegal and DTV opcodes and commands for unrolling loops. It also has features like pseudo commands, import of SID files, import of standard graphic formats and support for 3rd party Java plugins. The script language makes it easy to generate data for your programs. This could be data such as sine waves, coordinates for a vector object, or graphic converters. Writing small data generating programs directly in you assembler source code is much handier than writing them in external languages like Java or C++.The script language and the assembler is integrated. Unlike other solutions, where scripts are prepassed, the script code and the assembler directives works together giving a more complete solution.

As seen by the size of this manual, Kick Assembler has a lot of functionality. You don't need to know it all to use the assembler, and getting to know all the features may take some time. If you are new to Kick Assembler, a good way to start is to read Chapter 2, Getting Started, Chapter 3, Basic Assembler Functionality and Chapter 4, Introducing the Script Language and then supplement with the features you need. Also notice the quick reference appendix which contains lists of directives, options and values.

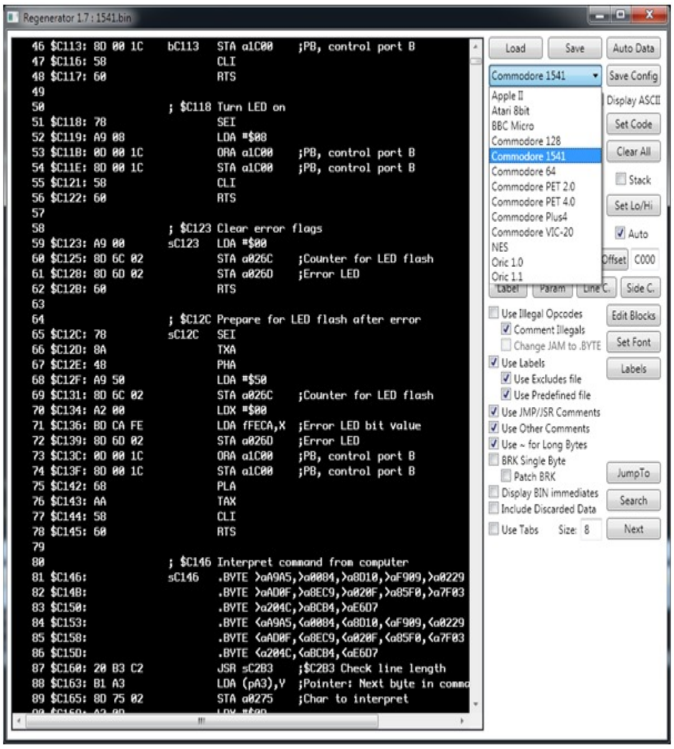
If you are going from version 3.x to 4.x please check out the appendix regarding this subject.

This is the fourth version of Kick Assembler. The first version (1.x) was a normal 6510 cross assembler developed around 2003 and was never made public. The second version (2.x) was developed in 2006 and combined the assembler with a script language, giving you the opportunity to write programs that generate data for the assembler code. Finally in august 2006 the project went public. The third version (3.x) improved the underlying assembling mechanism using a flexible pass algorithm, recording of side effects and handling of invalid values. This gave better performance, and made it possible make more advanced feature. The fourth version (4.x) replaced the parsing mechanism, which where made using a parser generator, with a handwritten one which is faster, more flexible and included a preprocessor. This made it possible to do new language constructs and have better error handling. It also replaced the scoping system so it includes all entities, not just symbols. Through the years the project have grown quite big, with a professional setup including a its own code repository, a large automated test suite and automatic building and

Regenerator 1.7

Regenerator by Tomaz Kac, can disassemble a Commodore 64 machine code program on your Windows PC. Changes in this version are: Multiple-system support, improvements for JMP/JSR, and Immediate Load to $00 - $ff.

http://csdb.dk/getinternalfile.php/150640/Regenerator17.zip



```
$C113: 8D 00 1C    bC113   STA a1C00    ;PB, control port B
$C116: 58                  CLI
$C117: 60                  RTS

                   ; $C118 Turn LED on
$C118: 78                  SEI
$C119: A9 08               LDA #$08
$C11B: 0D 00 1C            ORA a1C00    ;PB, control port B
$C11E: 8D 00 1C            STA a1C00    ;PB, control port B
$C121: 58                  CLI
$C122: 60                  RTS

                   ; $C123 Clear error flags
$C123: A9 00       sC123   LDA #$00
$C125: 8D 6C 02            STA a026C    ;Counter for LED flash
$C128: 8D 6D 02            STA a026D    ;Error LED
$C12B: 60                  RTS

                   ; $C12C Prepare for LED flash after error
$C12C: 78          sC12C   SEI
$C12D: 8A                  TXA
$C12E: 48                  PHA
$C12F: A9 50               LDA #$50
$C131: 8D 6C 02            STA a026C    ;Counter for LED flash
$C134: A2 00               LDX #$00
$C136: BD CA FE            LDA fFECA,X  ;Error LED bit value
$C139: 8D 6D 02            STA a026D    ;Error LED
$C13C: 0D 00 1C            ORA a1C00    ;PB, control port B
$C13F: 8D 00 1C            STA a1C00    ;PB, control port B
$C142: 68                  PLA
$C143: AA                  TAX
$C144: 58                  CLI
$C145: 60                  RTS

                   ; $C146 Interpret command from computer
$C146:             sC146   .BYTE >aA9A5,>a0084,>a8D10,>aF909,>a0229
$C14B:                     .BYTE >aAD0F,>a8EC9,>a020F,>a85F0,>a7F03
$C150:                     .BYTE >a204C,>aBCB4,>aE6D7
$C153:                     .BYTE <aA9A5,<a0084,<a8D10,<aF909,<a0229
$C158:                     .BYTE <aAD0F,<a8EC9,<a020F,<a85F0,<a7F03
$C15D:                     .BYTE <a204C,<aBCB4,<aE6D7
$C160: 20 B3 C2            JSR sC2B3    ;$C2B3 Check line length
$C163: B1 A3               LDA (pA3),Y  ;Pointer: Next byte in comma
$C165: 8D 75 02            STA a0275    ;Char to interpret
```

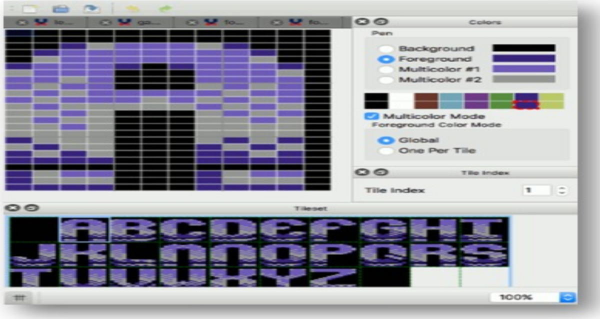
□ Stack
☑ Auto
Offset: C000
□ Use Illegal Opcodes
☑ Comment Illegals
□ Change JAM to .BYTE
☑ Use Labels
☑ Use Excludes file
☑ Use Predefined file
☑ Use JMP/JSR Comments
☑ Use Other Comments
☑ Use ~ for Long Bytes
□ BRK Single Byte
□ Patch BRK
□ Display BIN immediates
□ Include Discarded Data
□ Use Tabs Size: 8

VChar64 v0.1.0

Ricardo Quesada released an update to his character editor for the Commodore 64. The features so far are: Import, export, rotate, invert, clear, shift to left, right, up and down. Changes in this version: Improvements for Export, Save, Recent Files and Multicolour.

http://csdb.dk/release/?id=149507

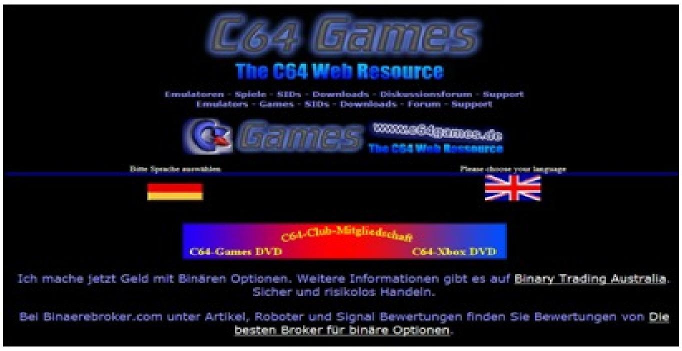
Main website
https://github.com/ricardoquesada/vchar64/



C64games.de

With recent updates to the website, there are now over 6429 games. Recent additions include: Bongo King, Excalibur, HES Games, Kabura, Little Knight Arthur, Madballs, Mystery Master - Felony!, Mystery Master - Murder by the Dozen, Mystic Mansion, and World Games (Cosmi).

http://www.c64games.de/



C64 Games

The C64 Web Resource

Emulatoren - Spiele - SIDs - Downloads - Diskussionsforum - Support
Emulators - Games - SIDs - Downloads - Forum - Support

Games www.c64games.de The C64 Web Ressource

Bitte Sprache auswählen

Please choose your language

C64-Club-Mitgliedschaft

C64-Games DVD

C64-Xbox DVD

Ich mache jetzt Geld mit Binären Optionen. Weitere Informationen gibt es auf Binary Trading Australia. Sicher und risikolos Handeln.

Bei Binaerebroker.com unter Artikel, Roboter und Signal Bewertungen finden Sie Bewertungen von Die besten Broker für binäre Optionen.

SID Known v1.12

Wilfred Bos released an update of SID Known. This version is compatible with the latest HVSC (#65). SID Known is a tool which you can use to identify SID tunes from SID and PRG files.

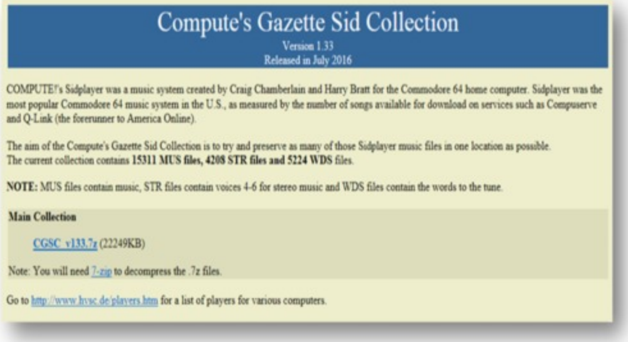
http://csdb.dk/getinternalfile.php/150551/sidknown_v112.zip

http://www.acid64.com/download/sidknown/sidknown_v112.zip

CGSC v1.33

The CGSC (Compute's Gazette Sid Collection) has been recently updated. The COMPUTE!'s Sidplayer is a music system created by Craig Chamberlain and Harry Bratt. The aim of the Compute's Gazette Sid Collection is to preserve as many of those Sidplayer music files in one location as possible. The current collection contains 15311 MUS, 4208 STR and 5224 WDS files.

http://www.c64music.co.uk/



Compute's Gazette Sid Collection

Version 1.33
Released in July 2016

COMPUTE!'s Sidplayer was a music system created by Craig Chamberlain and Harry Bratt for the Commodore 64 home computer. Sidplayer was the most popular Commodore 64 music system in the U.S., as measured by the number of songs available for download on services such as Compuserve and Q-Link (the forerunner to America Online).

The aim of the Compute's Gazette Sid Collection is to try and preserve as many of those Sidplayer music files in one location as possible.
The current collection contains **15311 MUS files, 4208 STR files and 5224 WDS** files.

NOTE: MUS files contain music, STR files contain voices 4-6 for stereo music and WDS files contain the words to the tune.

Main Collection

CGSC_v133.7z (22249KB)

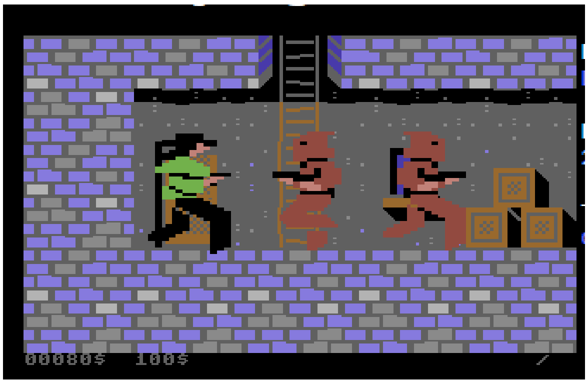
Note: You will need 7-zip to decompress the .7z files.

Go to http://www.hvsc.de/players.htm for a list of players for various computers.

Fort Django

Fort Django is a new game developed by Dr. TerrorZ. You must shoot the bad guys, collect the bags, and find the exit. If you play faster you will get more of a bonus!

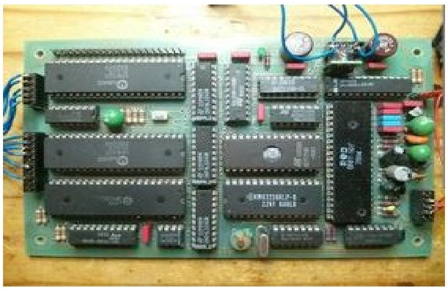
http://csdb.dk/release/?id=149628



C64xs

Freak is building a small Commodore C64 using original C64 parts but with extra added features: Configurable voltages and extra filters for SID and VIC, S-Video output, ESD protection, and multiple kernals and character sets

http://www.forum64.de/index.php?thread/68041-project-log-c64xs-xtra-small/&pageNo=1



C64 - The great refurbishing

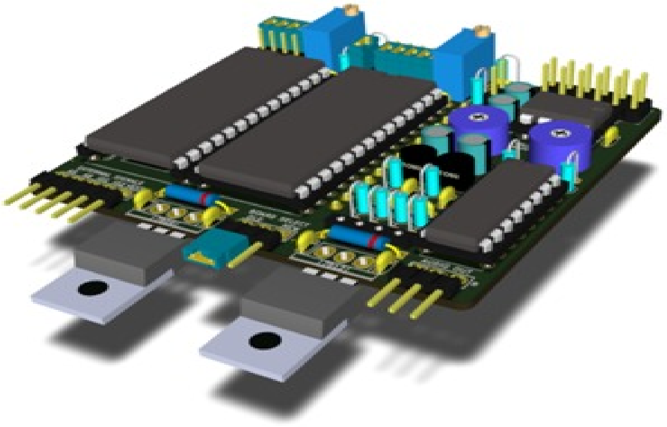
On the Retrohax web page is a refurbish of a Commodore C64. Drygol is refurbishing six Commodore C64s by cleaning and repairing the plastics parts and the keyboard, and then adding a SwinSID NANO and an SD2IEC device.

https://www.retrohax.net/commodore-64-the-great-refurbishing-2/

MixSID

The MixSID is a stereo SID for the Commodore C64. Features include: Stereo channel mixer, stereo pragmatic addressing scheme, flexible IO, audio input, output and boost digital circuits, stereo output, full paddle compatibility, and extensive digital control options.

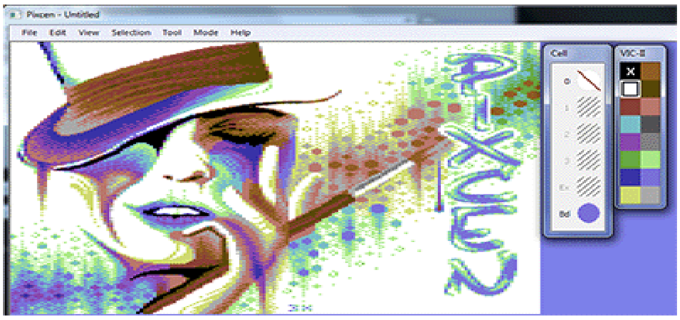
http://henning-bekel.de/2016-07-18-mixisid-1.0-released.html



Pixcen V0.7.0.13

Released by: Censor Design An impressive Windows-based pixel painter application, there are both 32- and 64-bit versions of the application to download.

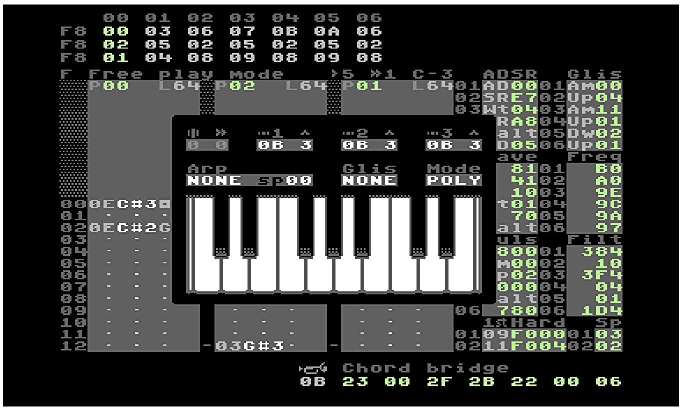
http://censordesign.com/pixcen/versions/



Virtuoso 0.99

Hein has released an update to his music program. The program's features include: Head Editor: sequencer, patterns, tools, and step FLO programs. Jam mode: mono, poly, or stacked. Also included is a Selection of instruments, song, octaves, or song speed and disk menu.⍰

http://csdb.dk/getinternalfile.php/150633/virtuoso-0.99.d64



```
     00 01 02 03 04 05 06
F8   00 03 06 07 0B 0A 06
F8   02 05 02 05 02 05 02
F8   01 04 08 09 08 09 08
F Free play mode    >5 »1 C-3  ADSR  Glis
  P00    L64 P02    L64 P01    L6401AD0001Am00
                                  02SRE702Up04
                                  03Wt0403Am11
                                    RA804Up01
     0 0   0B 3   0B 3   0B 3       alt05Dw02
                                    D0506Up01
     Arp          Glis   Mode       ave   Freq
     NONE sp00    NONE   POLY       8101    B0
                                    4102    A0
                                    1003    9E
000EC#3                             t0104   9C
01 - - -                            7005    9A
020EC#2G                            alt06   97
03 - - -                            uls   Filt
04 - - -                            80001  384
05 - - -                            M0002   10
06 - - -                            p0203  3F4
07 - - -                            00004   04
08 - - -                            alt05   01
09 - - -    - - -    - - -        0678006  1D4
10 - - -    - - -    - - -         1stHard  Sp
11 - - -    - - -    - - -        0109F0000103
12 - - -   -03G#3-   - - -        0211F0040202
                   Chord bridge
                   0B 23 00 2F 2B 22 00 06
```

Polish Garage still using a Commodore 64

A small car repair shop in Poland still uses a Commodore C64c. The machine is used to balance the drive shafts. If it's not broken don't fix it!

http://sploid.gizmodo.com/this-old-ass-commodore-64-is-still-being-used-to-run-an-1787196319

Flashing C64

A web page about how to convert a Commodore C64 into a transparent case with LED flashing lights that can be controlled via remote control.

http://www.breadbox64.com/blog/dashing-flashing-c64/

C64 Endings

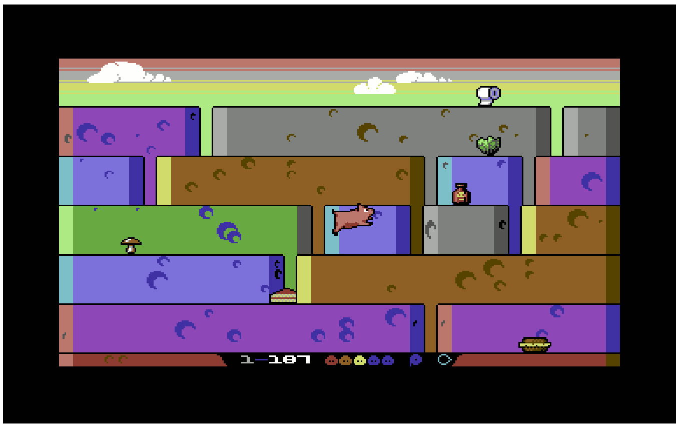
The web page c64endings has a number of new game endings. The latest games are: Jet Set Willy (Software Projects), Impossamole (Gremlin Graphics), Head The Ball (Hewson), Gee Bee Air Rally (Activision), Final Assault (Epyx), E-motion (US Gold) Dark Fusion (Gremlin Graphics), The Curse of the Mad Mummy (Soft gold), Barbarian 2: The Dungeon of Drax (Palace Software), Agent X II: The Mad Prof's Back (Mastertronic), Way of the Exploding Fist (Melbourne House) video Meanies (Mastertronic) Uridium (Hewson) and Taskforce (Players Premier).

https://c64endings.wordpress.com/

QUOD INIT EXIT V1.6

Released by: Retream another update to his pig jumping game In the game you are a little pig, and you need to eat all the food you can find. The game has hires graphics with music from Richard Bayliss.

http://csdb.dk/getinternalfile.php/151112/QUOD_INIT_EXIT.d64



FREEZE64 released

From the C64endings/Hackersoft team comes this home-grown & home-printed, A5, full colour, gloss cover, Commodore 64 gaming, cheat and coding fanzine, which you can order for only £3.50 per issue + P&P. Each issue is full of new cheats and POKES to use with an emulator or real Commodore 64, as well as features on hacking games, coder interviews, game ending reviews, tips, guides and lots, lots more. If you love your Commodore 64 and want to learn more about cheating in games - as well as reading game features and interviews - then this is the fanzine for you!

Latest Christmas edition includes

HUNCHBACK COVER FEATURE: We celebrate Ocean Software's popular and successful platform arcade game Hunchback, sharing some of our POKES and CODES and explaining some of the game code.

- AN EXCLUSIVE INTERVIEW WITH LASSE ÖÖRNI: You may not instantly recognise his name, but you will have definitely have played his recent Commodore 64 games. Lasse is the talent behind the amazing METAL WARRIOR series, as well as the recently released Psytronik action adventure game, HESSIAN. We find out how Lasse started coding in the world of the Commodore 64 and reveal his process of creating his powerful games.
- FREE CHEAT CARD NO. 2: The second of our FREE collectors' Cheat Cards, which are professionally printed in full colour!
- 4K CRAP GAME COMPETITION 'POKES SPECIAL': We've taken the games from the recent 4K Crap Game Competition, and hacked them to pieces to provide you with some fun and useful POKES to play with.
- EXCLUSIVE READER'S XMAS GIFT DOWNLOAD: As if the FREE Cheat Card isn't enough, we also want to give our readers an extra gift as a thank you for supporting FREEZE64.
- HACKERSOFT TUTORIAL: THE LEVEL SKIP – PART 1: How do you fancy learning how to code your own level skip? In this first (of many) Hackersoft tutorials we pick a specific Commodore 64 game and show you how to code in a level skip feature controlled by a single key press. We will take you through the whole process from start to finish.
- GAMES IN THE MAKING: We all love hearing about new Commodore 64 games in the making and so we send out the hounds to sniff out a few more coders who are producing new titles for us all to drool over.
- NEXT LEVEL: In this new, regular feature we reveal an exclusive map featuring all the screens to the first level of Graham Axten's THE BEAR ESSENTIALS.
- GAME ENDING FOCUS: We reveal and discuss the impressive game ending to System 3's LAST NINJA 3.
- AND ALSO... FREEZE64 wouldn't be the same without new and old POKES/codes, sharing more insight into another Hackersoft Pad, and finally more revealing case files from our intrepid Secret Squirrel.

Homepage

http://www.c64endings.freeolamail.com/freeze64/freeze64.htm

Video prievew

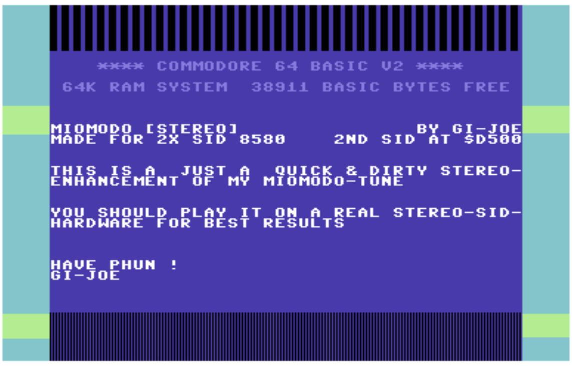
https://www.youtube.com/watch?v=gtZO_0Vjwkw&feature=youtu.be

FREEZE64
THE ULTIMATE GAMING CHEAT FANZINE FOR YOUR COMMODORE 64
HunchBACK
We revisit this classic platformer and inject a little more life into the old dog with our POKES n CODES.
ALSO IN THIS ISSUE:
- AN EXCLUSIVE INTERVIEW WITH LASSE ÖÖRNI
- THE FIRST PART OF OUR LEVEL SKIP TUTORIAL
- MORE NEWS ON GAMES IN THE MAKING
- AN EXCLUSIVE READERS' XMAS DOWNLOAD
- SECRET SQUIRREL OPENS A FEW MORE CASE FILES
- WE REVEAL THE GAME ENDING TO LAST NINJA 3
- POKES, POKES AND MORE POKES!
4K
GAMES COMPO
POKES 'n CODES
SPECIAL!
ISSUE #05
XMAS 2016
£3.50

Miomodo [stereo]

Released by: Mayday! miomodo {stereo} by gi-joemade for 2x sid 8580 2nd sid at $d500 This is just a quick & dirty stereo enhancement of my miomodo-tune. You should play it on a real stereo-SID hardware for best results

http://csdb.dk/getinternalfile.php/151111/miomodo_2SID.sid
http://csdb.dk/getinternalfile.php/151110/miomodo_2SID.d64



```
    **** COMMODORE 64 BASIC V2 ****
 64K RAM SYSTEM  38911 BASIC BYTES FREE

MIOMODO [STEREO]                BY GI-JOE
MADE FOR 2X SID 8580    2ND SID AT $D500

THIS IS A  JUST A  QUICK & DIRTY STEREO-
ENHANCEMENT OF MY MIOMODO-TUNE

YOU SHOULD PLAY IT ON A REAL STEREO-SID-
HARDWARE FOR BEST RESULTS

HAVE PHUN !
GI-JOE
```

SwinSID Ultimate Configurator

Released by: Schema Máté Sebők has created a microcontroller-based hardware emulation of the Commodore 64 SID chip called the SwinSID Ultimate. The design was based on the original SwinSID. His device can emulate the old 6581 or the newer 8580 SID chips, and can run in NTSC or PAL mode. To go with the design is a tiny menu-driven tool that allows you to configure the various options of the SwinSID Ultimate.

Review Blog here

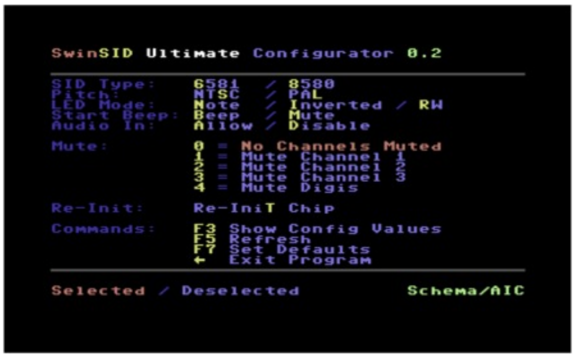
https://ilesj.wordpress.com/2016/04/24/swinsid-ultimate/

Facebook page

https://www.facebook.com/swinsidultimate

Config software

http://csdb.dk/getinternalfile.php/151059/swinsidu-config.prg

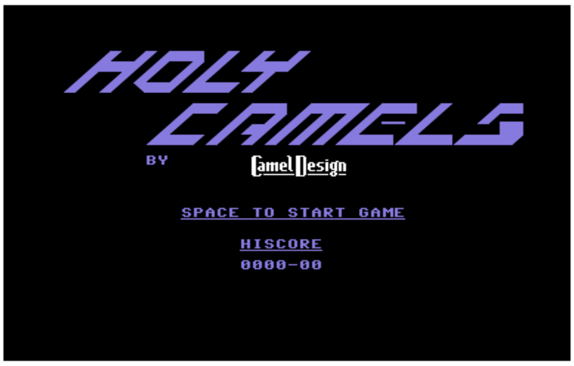


```
SwinSID Ultimate Configurator 0.2

SID Type:     6581  / 8580
Pitch:        NTSC  / PAL
LED Mode:     Note  / Inverted / RW
Start Beep:   Beep  / Mute
Audio In:     Allow / Disable

Mute:         0 = No Channels Muted
              1 = Mute Channel 1
              2 = Mute Channel 2
              3 = Mute Channel 3
              4 = Mute Digis

Re-Init:      Re-IniT Chip

Commands:     F3 Show Config Values
              F5 Refresh
              F7 Set Defaults
              ←  Exit Program

Selected / Deselected          Schema/AIC
```

Holy Camels

Released by: Camel Designs, Megastyle Seems to be a game where you have to press the right key to stop the camel from being killed. The key or letter will appear on the left and scroll quickly to the right. Hit the right key to save the camel. Hmm... I think I never managed to hit the right key yet!

http://csdb.dk/getinternalfile.php/151067/HOLYCAMELS.PRG



Lotek64 issue 53

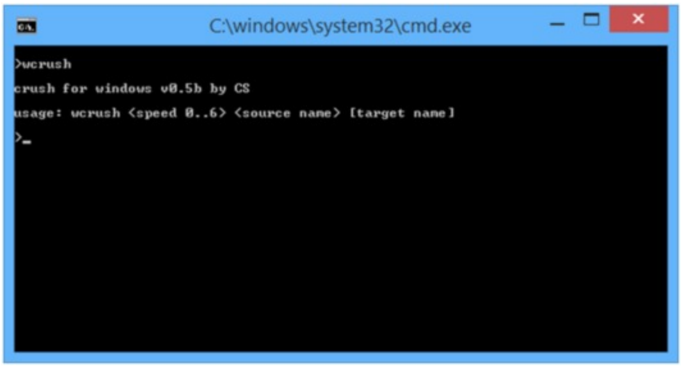
A new edition of the German-language PDF magazine is available. The articles in this edition include: Lo*bert, Editorial, News, Retro-Publishing: poly.play, Mirco Geldermann/Magic Disk 64, Diskdrive – C65, PLLUG, Amiga-Prototyp Velvet, USB-Adapter Unicorn – Atari ST, TV-Gameshow, DK Bongos, Resume, YouTube-Channels, Star-Wars, Quod Init Exit, Nerd-Roadmovie, Coco-Notes, U-Bahn-Kunst, Commodore Meeting Wien 2016, Chipmusik, GameBoy-Music, SIDologie, and Videogame Heroes #19: Q*bert.

http://www.lotek64.com/hp/index.php

Wcrush V0.7b

Crush for Windows was developed as a replacement for Taboo's crush.exe levelpacker, which unfortunately doesn't work on 64-bit Windows without ugly tricks like using DosBox, etc. The included binaries work on Windows 98SE and Windows 8.1 64 bit, so they should (maybe) work on all Windows versions that came in between (hopefully).

https://sourceforge.net/projects/wcrush/



```
>wcrush
crush for windows v0.5b by CS
usage: wcrush <speed 0..6> <source name> [target name]
>_
```

World Conquest

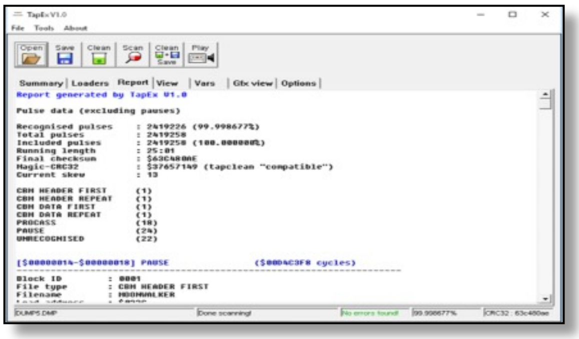
Released by: Hokuto ForceWORLD CONQUEST is a computer strategy war game based on the popular board game RISK. In this game two enemy factions make war for global dominance (you can play against another human player or the computer). Similarly to RISK, both factions control Regions of the World using Armies, gain Reinforcements from owned Regions, and attempt to conquer bordering Regions using area movement.

http://csdb.dk/getinternalfile.php/151005/WORLD CONQUEST.D64
http://csdb.dk/getinternalfile.php/151006/World Conquest tape.rar

Tapex

A program for analyzing, verifying, and cleaning Commodore 64 cassette files (TAP V0 TAP V1 and DC2N RAW files (16, 24 & 32 bit). Changes in this version: Support for Martech block loader, Visiload, cult, Bleepload, Audiogenic, Hitload, Firebird loader, ODEload, Burner, Micro Load, Alternative Software loaders, CHR loaders, power load, Anirog Tequila Sunrise loader, and Flash Load.

http://csdb.dk/release/?id=149992



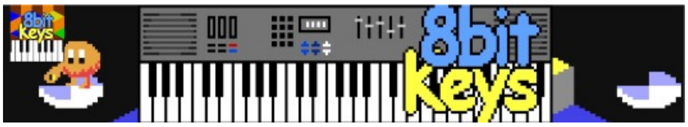
```
Report generated by TapEx V1.0

Pulse data (excluding pauses)

Recognised pulses     : 2419226 (99.998677%)
Total pulses          : 2419258
Included pulses       : 2419258 (100.000000%)
Running length        : 25:01
Final checksum        : $63C480AE
Magic-CRC32           : $37657149 (tapclean "compatible")
Current skew          : 13

CBM HEADER FIRST      (1)
CBM HEADER REPEAT     (1)
CBM DATA FIRST        (1)
CBM DATA REPEAT       (1)
PROCASS               (18)
PAUSE                 (24)
UNRECOGNISED          (22)

[$00000014-$00000018] PAUSE              ($000AC3F8 cycles)
-------------------------------------------------------------
Block ID          : 0001
File type         : CBM HEADER FIRST
Filename          : MOONWALKER
```

DUMP5.DMP | Done scanning! | No errors found! | 99.998677% | CRC32 : 63c480ae

Hmmm...
I also found this

It covers the Commodore FM Sound Expander in some detail, but has a number of videos on obsolete 80s music hardware. Anyone remember the Miracle keyboard? The website says you couldn't edit your own music, but that's not right as I have software that lets you compose, although limited in length. I won one in a music competition for use with the Amiga. I remember some issues as it has a tutorial, and maybe I was the only person to go though it, but about 2/3rd though it would crash. They sent me various disks but it always crashed at the same place -- in the end I gave up complaining.

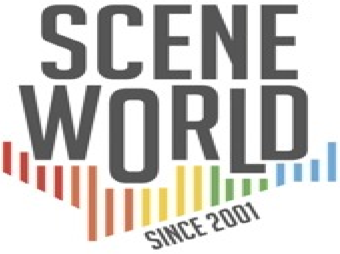
https://www.youtube.com/channel/UCcTt3O4_IW5gnA0c58eXshg



Scene World Podcast Issue 26

Scene World Podcast is all about the Commodore scene. In this latest edition: Paul Solo Mine A Video Game Con, Jeroen Tel, True Cinema Retro, Tarnkappe.info, Return Magazine Project SIDFX, 3D Realms' Rad Rad Rodgers, and Rodgers Kickstarter.

http://sceneworld.org/blog/2016/10/13/podcast-episode-26-a-video-game-cons-paul-solomine/



Light Hammer v2.0

If you can solder and have access to a 3D 0printer, you can now build your own light hammer. Visit Jan's website for more information

His website says

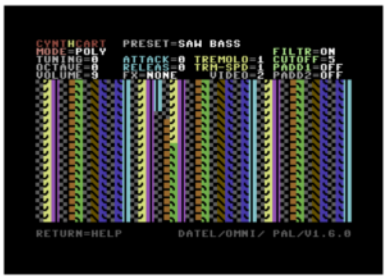
To prevent confusion, a Light Hammer is just like the lightpen and lightgun, a CRT-based input device for your computer... well, retro computer that is. Because nowadays in the age of touch screens, who needs a lightpen? And, in the age of LCDs, not many people own a CRT anymore. Therefore, lightguns aren't what they used to be. The good old lightgun is just a lightpen with a trigger and an extra lense. In order for the lightpen/gun to operate you'll need a CRT monitor. If you have such a monitor and a retro computer like the C64, then this might be of interest to you. The Light Hammer opens up a whole new world of possibilities.

http://home.kpn.nl/bderogee1980/projects/lighthammer/lighthammer.html

Cynthcart v2.0

Cynthcart is a cartridge for the Commodore 64/128 that turns the computer into a standalone analog synthesizer. Cynthcart's key-command user interface allows it to be played and controlled without a monitor, making it especially suitable for live performances. The original Commodore 64 piano keyboard overlay can be used (illustrated above), and Datel, Passport, Sequential, and Kerberos MIDI interface cartridges are now supported.

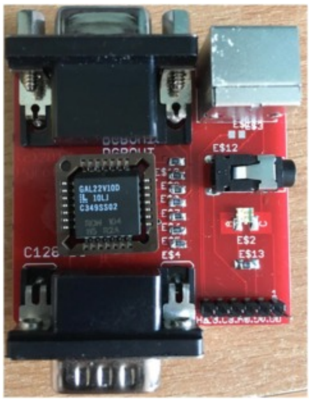
New software is available. The changes in this version include : 30 pre-set sound patches, arpeggiator, tremolo, analog filter, MIDI support, on-screen help, color visualization display, SID hex editor, and PAL / NTSC pitch tables with automatic selection.



```
CYNTHCART    PRESET=SAW BASS
MODE=POLY                                FILTR=ON
TUNING=0     ATTACK=0  TREMOLO=1   CUTOFF=5
OCTAVE=0     RELEAS=0  TRM-SPD=1   PADD1=OFF
VOLUME=9     FX=NONE     VIDEO=2   PADD2=OFF

RETURN=HELP         DATEL/OMNI/ PAL/V1.6.0
```

http://www.qotile.net/cynthcart.html

80-column CGA to RGB converter



Pyrofer is working on a converter for the 80-column CGA video signal into RGB. So, you can connect your Commodore C128 to an RGB monitor. The refresh rate 15 kHz (but no scan doubler). Read more on these links:

http://c-128.freeforums.net/thread/336/final-attempt-80col-cga-converter

http://www.amibay.com/showthread.php?86858-C128-CGA-to-RGB-adapter

The Internet for Commodore Users

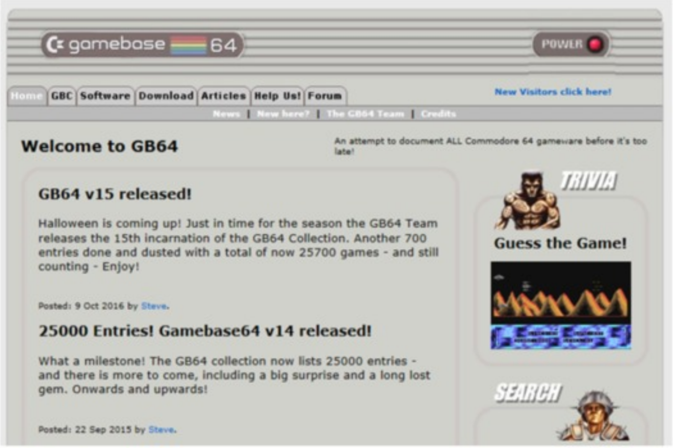
Back in the day, I bought the spiral-bound book, "The Internet for Commodore Users" by Gaelyne Gasson. It was the third, and little did I know at the time, it was the final edition. Every once in awhile, C= users ask about getting their machines on-line, and I refer them to TIFCU if it answers their questions. (TIFCU was written before C= wi-fi modems and other newer means of connecting.) All these years I had thought that the book would never be seen again. I was wrong! In September I met with Gaelyne in Adelaide, South Australia, and she informed me that C= users can still access the book by downloading the .PDF. I was gladly surprised. If you want your own copy of TIFCU, read the description at: http://market.vcsweb.net/about-tifcu-the-internet-for-commodore-users/ and click on the button at the bottom for purchasing information. If you just want to directly purchase TIFCU, go to: http://market.vcsweb.net/product/tifcu-the-internet-for-commodore-c64-128-users/

Robert Bernardo - Fresno Commodore User Group
http://www.dickestel.com/fcug.htm

Gamebase GB64 - v15

GB64 is quoted as "An attempt to document ALL Commodore 64 gameware before it's too late! "and have released a new version. The total number of games is now over 25,700 entries, 700 more than the previous version. There are many updated files

http://www.gb64.com/



Welcome to GB64

An attempt to document ALL Commodore 64 gameware before it's too late!

GB64 v15 released!

Halloween is coming up! Just in time for the season the GB64 Team releases the 15th incarnation of the GB64 Collection. Another 700 entries done and dusted with a total of now 25700 games - and still counting - Enjoy!

Posted: 9 Oct 2016 by Steve.

25000 Entries! Gamebase64 v14 released!

What a milestone! The GB64 collection now lists 25000 entries - and there is more to come, including a big surprise and a long lost gem. Onwards and upwards!

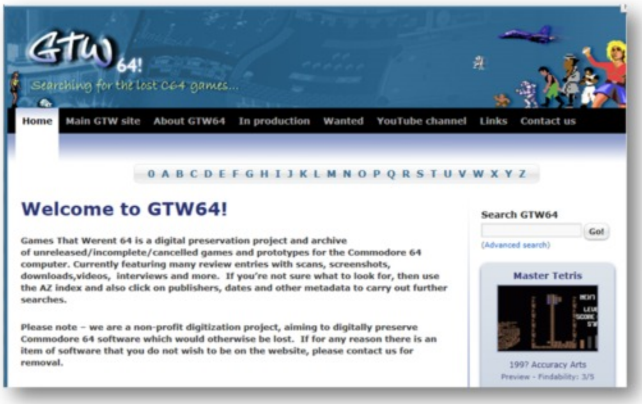
Posted: 22 Sep 2015 by Steve.

GAMES THAT WEREN'T

Games That Werent 64 is a digital preservation project and archive of unreleased/incomplete/cancelled games and prototypes for the Commodore 64 computer. Currently featuring many review entries with scans, screenshots, downloads,videos, interviews, and more. If you're not sure what to look for, then use the AZ index and also click on publishers, dates, and other metadata to carry out further searches.

Recently added are : Abyss Zone, Aliens, Blaster Twins, Bomb Mania 2, Bongo King, Commando Libya - part 2, Final Lap, Freebooter, Heroquest V1, Karate, Mythos, Pet Person, Race with the Devil, Scarabaeus 2 and Space Invadaz. Update: Armalyte V1, Astaroth: Angel Of Death, Attack On Centralis, Baja Buggies, Beavers, C64GS cartridge titles, Chicago, Custodian, Cyber Tank, Dork's Dilemma, Dynablaster, Edward Randy, Enterprise, Escape From Colditz, Flashpoint, Forbidden Planet, Gargantuan, Graeme Souness Vector Soccer, Habitat, Hydrogenese, London Exchange, Maelstrom, Mag Max V1, Mario Bros, Paddle Mania, Piper Software games, Purple Saturn Day, Quadrillion, Quondam, Restrictor, Roadstar Xri, Robin Hood, Tangram, The City of Ehdollah, Tour De Force, Tycoon Tex, Up Up And Away, and Zorro In Wonderland.

http://www.gamesthatwerent.com/gtw64



Welcome to GTW64!

Games That Werent 64 is a digital preservation project and archive of unreleased/incomplete/cancelled games and prototypes for the Commodore 64 computer. Currently featuring many review entries with scans, screenshots, downloads,videos, interviews and more. If you're not sure what to look for, then use the AZ index and also click on publishers, dates and other metadata to carry out further searches.

Please note – we are a non-profit digitization project, aiming to digitally preserve Commodore 64 software which would otherwise be lost. If for any reason there is an item of software that you do not wish to be on the website, please contact us for removal.

Master Tetris
199? Accuracy Arts
Preview - Findability: 3/5

2016 Reset C64 4KB Game Coding Comp

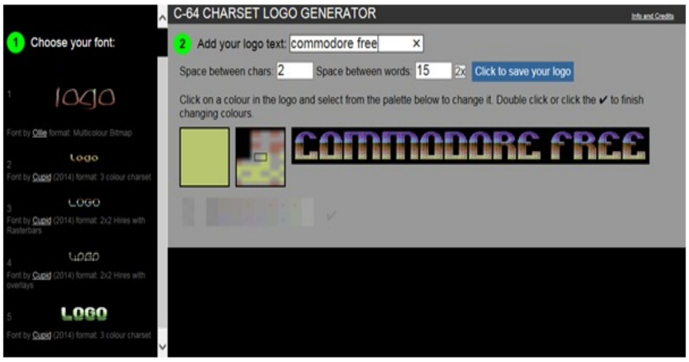
The current entries of the 2016 Reset C64 4KB Game Coding Comp are: M4573RB4G3L5 (Cout Games), Colon 10 (Cout Games), Dog (Vanje Utne), Granny's Teeth (Richard Bayliss), Goblin (Vanja Utne), Valentine's Day Shopping Simulator (Karmic), Watschinator (Endurion), Antarctica (Antarctica), Picross 4k (Oziphantom), Zombie Massacre (Wanax), Lumberjack (Majikeyric), Bonkey Kong (Graham Axten), Super Ski (Andreas Gustafsson) and ? (Paul Koller).

https://www.facebook.com/ResetC64Magazine/

C64 Charset Logo Generator

An online logo generator with the character sets of the Commodore C64. Jump on the website, select a logo font, and type in your message. Idea and coding by Chris Heilmann. Charset ripping, credits, and clean-up by Dejan Petronijevic and Daniel Kottmair.

https://codepo8.github.io/logo-o-matic/#goto-orc



C-64 CHARSET LOGO GENERATOR

1 Choose your font:

1 Font by Ollie format: Multicolour Bitmap

2 Font by Cupid (2014) format: 3 colour charset

3 Font by Cupid (2014) format: 2x2 Hires with Rasterbars

4 Font by Cupid (2014) format: 2x2 Hires with overlays

5 Font by Cupid (2014) format: 3 colour charset

2 Add your logo text: commodore free

Space between chars: 2 Space between words: 15 2x Click to save your logo

Click on a colour in the logo and select from the palette below to change it. Double click or click the ✔ to finish changing colours.

Silicom Tunnel - C64

Programming: Roberto Ricioppo (Using Sideways SEUCK)
Graphics: Roberto Ricioppo
Music: Richard Bayliss
Genre: SEUCK - Horizontal Scrolling
Tape Loader: Thunderload 7 - by Martin Piper and Richard Bayliss

Fly across the Silicom Tunnel, fighting against the invaders that approach your ship. You do (after all) want to escape the tunnel from the evil alien empire. Warning: aliens will shoot back, so you had better watch out :)

http://tnd64.unikat.sk/f s.html

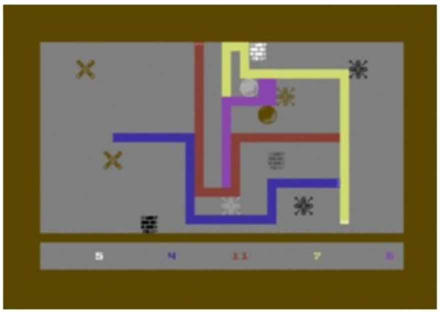
Snafu'64

(For 1 to 4 Players with the Protovision 4 Player Interface).

SNAFU '64 is a classic snake trap game where you must direct your snake so that it always has room to keep moving freely. At the same time, try to block your opponents so that they are forced into a collision.

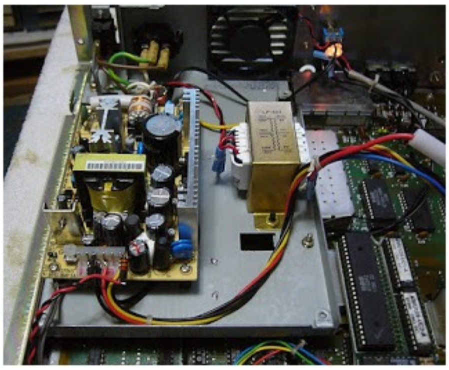
http://csdb.dk/release/?id=150866

http://csdb.dk/getinternalfile.php/152192/snafu64.prg



Custom Built Commodore PSU

Over the years, C= repair technician Ray Carlsen has built "very custom", special-order power supplies for Commodores and Amigas (in addition to his regular, universal power supplies). Check out the photos of his very custom power supplies at: http://blog.retro-link.com/2016/11/custom-power-supplies-from-ray-carlsen.html



Robert Bernardo
Fresno Commodore User Group
http://www.dickestel.com/fcug.htm

Commodore SX-64 repair

Iz8dwf has created a video about repairing a Commodore SX-64. **This is what he says:**

Bad Kernel and Chargen ROMs. This time there was no need to use any particular test instrument to completely repair this computer. Of course having a good set of known good spare chips helps a lot.

Also, whenever one has an SX-64 on the bench, it's worth checking a PCB mistake that affects all (at least all I have seen) these machines: one of the 9V AC user port pins is connected to GND, on the original C=64 board, no 9V AC pin is connected to GND, so I break the PCB track to eliminate this connection. If left connected, it could ruin some user port tools that expect no connection between 9V AC and GND and/or the SX-64 power supply (I never tried, but why taking the risk anyway?)

Replacement ROM with my own EEPROM board design here:
https://www.youtube.com/watch?v=kyXOuIfQgjw
https://www.youtube.com/watch?v=zBl4MbWcAF4

www.thefuturewas8bit.co.uk
Not just a place to buy SD2IEC's or a C64p
The future really was 8 bit!

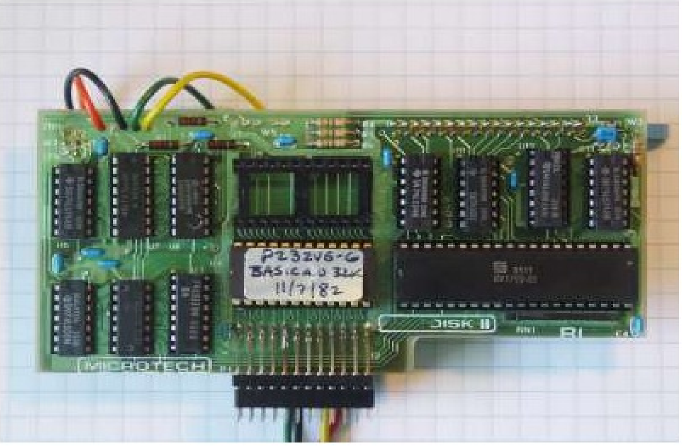
Commodore PET News

PEDISK II

The CGRS MICROTECH **PEDISK II** was an early floppy disk system for the Commodore PET/CBM computer. The PEDISK did not use the IEEE bus like the Commodore disk drives, but instead used a standard floppy disk controller chip connected to the inside of the PET/CBM machine. It could support both 5.25" and 8" standard floppy mechanisms as configured from the factory. The firmware on the board was able to load CGRS's proprietary DOS from disk. Up to 3 drives (all of the same type) were supported.

This is a project to re-create/clone the PEDISK-II board

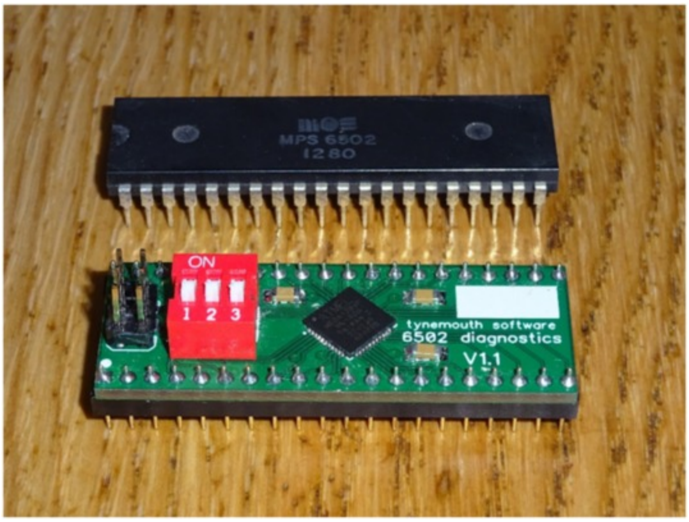
http://www.6502.org/users/sjgray/projects/pedisk2/index.html



PET Diagnostic

Tynemouth software has developed a new system that can carry out a diagnosis on a PET / CBM computer. The diagnostic system is placed in the socket of the CPU 6502 and then performs its various tests.

http://blog.tynemouthsoftware.co.uk/2016/07/pet-diagnostics.html?m=1



Commodore PLUS 4/16 News

Botticelli Bilderdisk number 41

Erich released another Botticelli Bilderdisk for the Commodore C16 and the Plus/4. This is a double-sided disk filled with pictures in the Botticelli format. Magica viewer is used to display the pictures that are a mix of handmade and digitized pictures.

http://plus4world.powweb.com/software/Botticelli_Bilderdisk_41

Crackers 'Demo 5 - Plus / 4

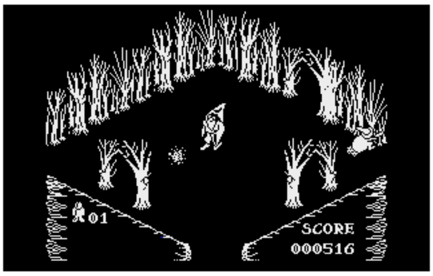
You can now use Crackers' View Demo 5 on the Commodore Plus / 4. This is an old-school mega demo, created by Ati, Andy, Bubis, Chronos, Csabo, Degauss, gerliczer, Harry Kichy, Lavina, Luca, Mad, MMS, Moldi, Murphy, Ninja, Nukem, PPCS, siz, Skoro , Spektro, 5tarbuck, TMR, and Unreal.

http://plus4world.powweb.com/software/Crackers_Demo_5

Pentagram - Plus / 4

In this game you must help Sabreman find the lost Pentagram, an artifact with magical powers. The game is based on the Atari 800 source code Mariusz. A version has also appeared for the C64. The Plus / 4 game also has support for a SID card.

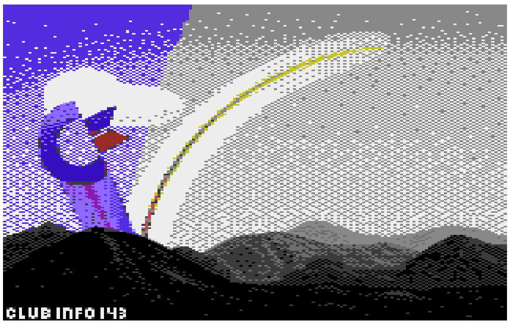
http://plus4world.powweb.com/software/Pentagram



Club Info 143

This is a German-language d64 disk magazine for the Commodore C16 and the Plus/4 featuring the following articles: Forum, Scene, Tips & Tricks, C16 Diskmonitor, Pac-Pac, Pentagram, Frutty Man, Verwuerfelt, Rebound, Computer Spass, Hardware, Spionagelehrling, Turm von Hanoi, Monster Hunter, Borrowed Time, Dateimanager, and other systems.

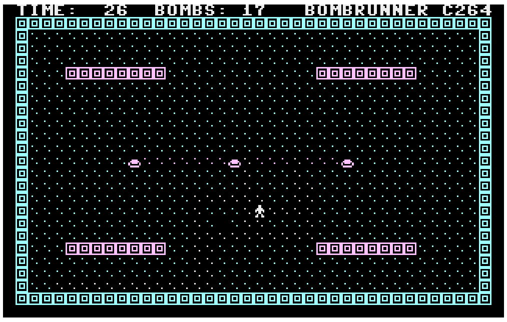
http://plus4world.powweb.com/software/Club_Info_143



Bomb Runner

Mark Quante (264er_Fan) has released a new game written in BASIC. The game is compatible with both NTSC and PAL equipment. The goal of the game is to collect all the bombs before time runs out.

http://plus4world.powweb.com/software/Bombrunner



TIME: 26 BOMBS: 17 BOMBRUNNER C264

Commodore Plus/4 Game Music Collection 1984-2016

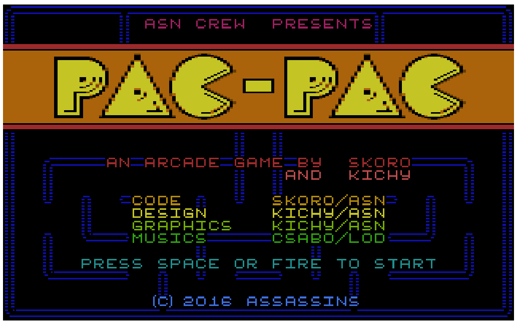
Skoro has created a YouTube collection of classic Commodore 16 /Plus4 music. Amongst the included works are classics like : Auf Wiedersehen Monty, Daley Thompson's Star Events, Hollywood Poker, Icicle Works, Kazik And The Ghosts, Leaper, Skyhawk, Terranova, Treasure Island, Winter Events, Adventures In Time, Get It, Majesty Of Sprites, Pac -Pac, Sgt. Helmet Zero, and Xplode Man.

https://www.youtube.com/watch?v=Y-6DYhcOBPo

Pac-Pac

The Assassins crew have released another new game for the Commodore Plus / 4. The game is coded by Skoro , with graphics from KiCHY and music by Csabo. In the game you must help Pac Pac 64 clear the different screens by eating dots. Various bonuses appear but no power pills in the corners like the original . The game has a smooth game play and beautiful colors.

http://plus4world.powweb.com/software/Pac-Pac



ASN CREW PRESENTS

PAC-PAC

AN ARCADE GAME BY SKORO AND KICHY

CODE SKORO/ASN
DESIGN KICHY/ASN
GRAPHICS KICHY/ASN
MUSICS CSABO/LOD

PRESS SPACE OR FIRE TO START

(C) 2016 ASSASSINS

YAPE v1.0.11

Another new release of the popular plus4/16 emulator, with usability improvements around the built-in monitor and the tape format handling. Let's line up all what's been changed:

- memorize window size when not using GDI rendering
- new selectable tape signal detection method (added 'edge triggered') for WAV samples
- loading MTAP's with non-native frequency such as ones recorded with NTSC bit set (report courtesy of Luca)
- load C64/VIC20 tapes recorded in MTAP format with the KERNAL loader
- even line hue phase offset in CRT emulation mode
- tape code cleanup
- sound engine related improvements
- speedometer improved
- preset screen size bug
- monitor cleanup
- TAP save overwrite warning (bug report courtesy of Luca)

http://yape.homeserver.hu/

```
COMMODORE BASIC V3.5 60671 BYTES FREE
READY.
```

Welcome to Yape.

0006
commodore
VIDEO MONITOR
MODEL 1701
commodore VIC 20

Commodore Vic 20 News

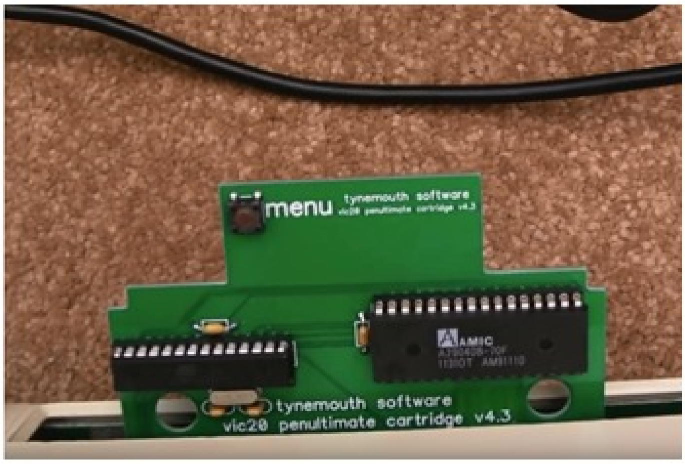
VIC20 Penultimate Cartridge

GadgetUK164 created a video about the new Penultimate Cartridge for the Commodore VIC20
YouTube has this information:
A quick look at the new Penultimate Cartridge from Tynemouth Software!

Tynemouth Software:
https://www.thefuturewas8bit.com/index.php/shop/commodore.html

thefuturewas8bit:
http://blog.tynemouthsoftware.co.uk/2016/07/vic20-penultimate-cartridge-preorder.html

https://www.youtube.com/watch?v=ByNCK-GhJQM


menu
tynemouth software
vic20 penultimate cartridge v4.3
tynemouth software
vic20 penultimate cartridge v4.3

VIC-20 Reloaded

Steve J. Gray is working on a new 8-bit project.
The project is to create a new VIC-20cr (Cost Reduced) motherboard which could become a starting point for modified VIC motherboards. For example, we could use a single 32K ram chip and make a VIC-32. It could combine multiple ROMs into a single ROM to save board space, or we could even add extra colour RAM to expand its graphics capability. A SID chip could be added, or a different video chip, or the addition of a second joystick port.

http://www.6502.org/users/sjgray/projects/vic20r/index.html

Vicious Jamz

THE VICIOUS TWENTY presents VICIOUS JAMZ by R'ZO (RYAN LISTON) A 17 track + 5 bonus track audio CD of VICmusic from R'zo. Free download contains tracks written using BASIC, Fisichella, and Vic Tracker. The Download includes printable CD booklet and .pdf booklet.

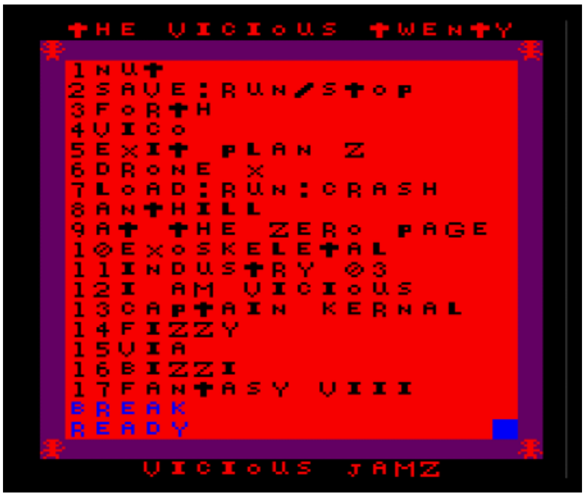
Listen to and down load
https://r-zo.bandcamp.com/album/vicious-jamz
Sample tracks available to listen free on Soundcloud
-Nut https://soundcloud.com/rzo-4/nut

-SAVE:RUN/STOP
https://soundcloud.com/rzo-4/saverunstop

-exoskeletal
https://soundcloud.com/rzo-4/exoskeletal

Full package download for those who cannot access bandcamp
https://1drv.ms/u/s!As2kXhqHVPDzimtlt_AA5_Zn6zTw

Denial website link
http://gator3293.hostgator.com/~sleeping/ipw-web/bulletin/bb/viewtopic.php?f=15&t=8217



```
THE VICIOUS TWENTY
1NUT
2SAVE:RUN/STOP
3FORTH
4VICO
5EXIT PLAN Z
6DRONE X
7LOAD:RUN:CRASH
8ANTHILL
9AT THE ZERO PAGE
10EXOSKELETAL
11INDUSTRY 03
12I AM VICIOUS
13CAPTAIN KERNAL
14FIZZY
15VIA
16BIZZI
17FANTASY VIII
BREAK
READY
VICIOUS JAMZ
```

It Came from the Grave

Jeffrey Daniels has released another game. Requirements: a VIC 20 Keyboard (WASD), joystick optional
Description: Escape a graveyard full of ghouls.
Download:
http://www.sleepingelephant.com/denial/games/itcame2.prg

Dropout

Dropout is a traditional block puzzle game for unexpanded VIC 20 PAL version / NTSC version

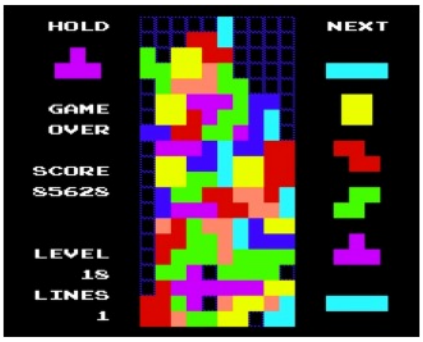
Dropout is tuned for fast play and implements many block-game features that were not yet common in the golden 8-bit era:

- SRS (Super Rotation System: each piece has 4 orientations, except square)
- Hold function (swaps current piece with the last piece stored in HOLD box)
- Wall kicks (smoother play near walls or locked structures)
- Lock delay (pieces do not lock instantly)
- Infinite spin (movement and rotation reset the lock delay, easier to move)
- DAS (fast key repeat with delay)
- Shows next 6 pieces
- Ghost piece (white outline showing where the piece is dropping to)
- Pre-rotation (Rotate controls are read during the 100 ms line clear delay)
- T-Spin recognition (T-Spin clearing 2 lines is worth more than 4 lines)
- Most L, J, S, Z twists are possible, not all thoroughly tested though
- Scoring system similar to many recent block games, except no combos
- TGM-style randomizer: previous 4 pieces are less likely to appear again

Four game modes:
[F1] Standard Long Play - start slow, level up to insane speeds
[F3] 40 Lines - clear 40 lines as fast as possible, under a minute is great
[F5] 2-Minute Challenge - score as many points as you can in 2 minutes
[F7] Master Mode - start at level 15, if you make it to level 24 you're good

PAL/NTSC differences:
- PAL version does a graphics hack to display 8x12 pixel square blocks
- NTSC version uses native "flat" pixel ratio and 8x8 blocks (and 60 Hz timer)

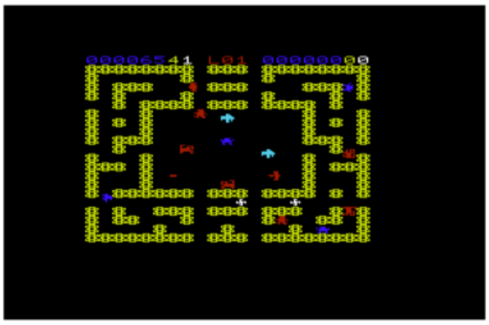
http://www.cncd.fi/aeeben/download/dropout.zip



VICrossroads

Steve Harter created the (C64 original), Vic Porter (VIC-20 porting) Requirements: VIC-20 with 8k RAM expansion
This is a VIC-20 port of the C64 mazey shoot'em up

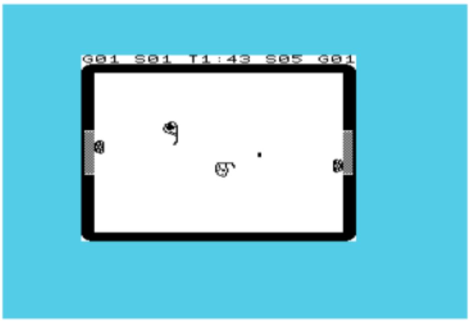
Download:
http://sleepingelephant.com/ipw-web/bulletin/bb/download/file.php?id=975



Hat Trick

Origional Author: Capcom (C64 port), Vic Porter (VIC-20 port) Requirements: VIC-20 with 16k RAM expansion, supports 24k expansion for extra graphics
Description: VIC-20 port of the C64 ice hockey game

Download:
http://sleepingelephant.com/ipw-web/bulletin/bb/download/file.php?id=1018

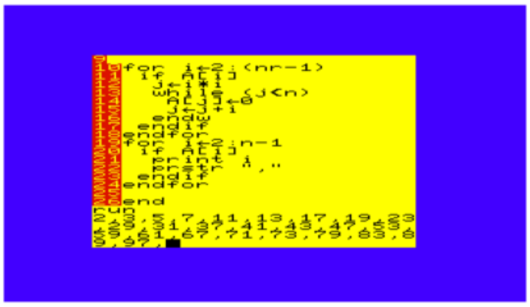


VICScript v0.1

Author: Bobbi
Requirements: VIC-20 with 32K expansion, plus disk drive / SD2IEC
Description: Novel interpreted scripting language
NOTE: This is an initial release. Please let me know of any bugs you encounter (or any suggestions for the next release).

VICScript is a simple scripting language for the Commodore VIC-20 with 32K expansion and disk drive. VICScript includes a minimal line editor which may be used for editing program text (or editing short text files). Alternatively, files may be prepared in an external editor and saved as plain PETSCII files. VICScript is primarily intended to be a fun, simple language for twiddling with hardware. The only native types are 16 bit-integers and arrays of 16-bit integers. A full range of C-style operations for logical and bitwise manipulation of values is provided. Constants may be expressed in decimal or hex, and values may also be displayed in either base. VICScript also aims to provide a basic set of structured programming primitives, such as named subroutines, multi-line if / else / endif conditionals, and a flexible while loop. The flavour of the language may be described as a cross between C and BASIC.

Discuss here:
http://sleepingelephant.com/ipw-web/bulletin/bb/viewtopic.php?f=2&p=90950#p90950

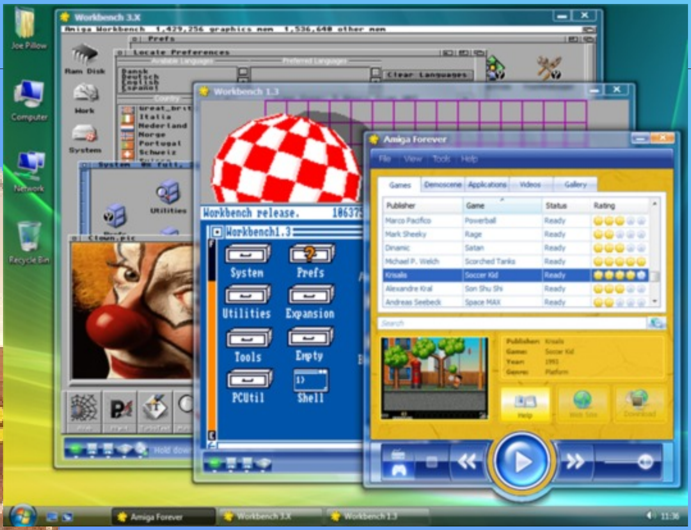


AMIGA FOREVER AND COMMODORE 64 FOREVER

Amiga Forever
http://www.amigaforever.com
http://www.facebook.com/AmigaForever

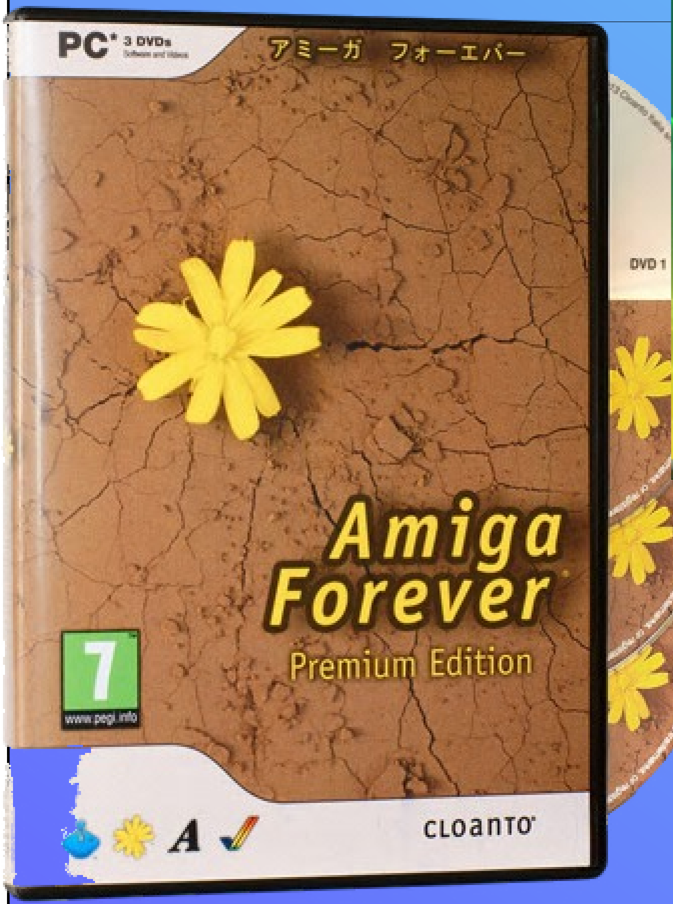
C64 Forever
http://www.c64forever.com
http://www.facebook.com/C64Forever

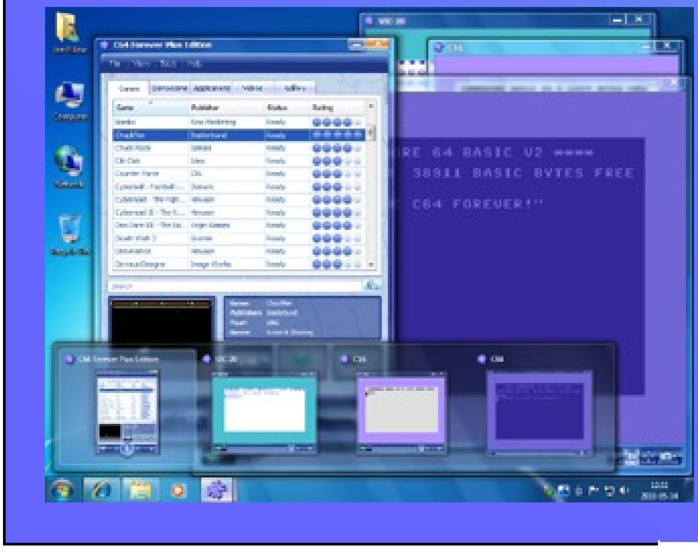
RetroPlatform Project
http://www.retroplatform.com



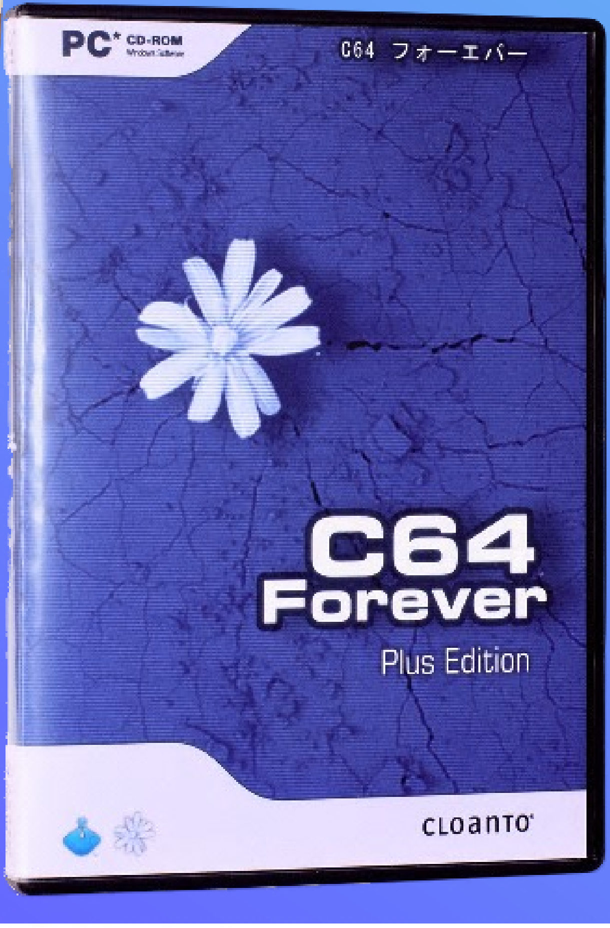
Publisher	Game	Status	Rating
Marco Pacifico	Powerball	Ready	●●●○○
Mark Sheeky	Rage	Ready	●●○○○
Dinamic	Satan	Ready	●●●○○
Michael P. Welch	Scorched Tanks	Ready	●●●●●
Krisalis	Soccer Kid	Ready	●●●●○
Alexandre Kral	Son Shu Shi	Ready	●●●○○
Andreas Seebeck	Space MAX	Ready	●●○○○

Publisher: Krisalis
Game: Soccer Kid
Year: 1993
Genre: Platform





```
RE 64 BASIC V2 ****
 38911 BASIC BYTES FREE
 C64 FOREVER!"
```



Commodore Amiga News

Icaros Desktop 2.1.3

Paolo Besser has released an update for the Icaros Desktop. Enhancements or updates to: Mapparium, OWB, Simple Mail, portable, Final Burn Alpha emulator meteMP3 player, and ColorCLI scripts

http://www.icarosdesktop.org/

Vampire 500 v2

Igor Majstorovic has created a video about the vampire 500 v2. The video lasts 40-minutes, He also talks about the plans for the Vampire 1200. English subtitles are available.

https://www.youtube.com/watch?v=tNjuulAWntc

Komoda & Amiga Plus Issue 5

Komoda & Amiga Plus is a PDF English and Polish magazine for the Commodore user. In this edition: Blocky Skies, Back to the Golden Age, Boxx 2 Cinnamon Writer, Dungeon Master, mp3info, Fresh News, Following the footsteps of Ghosts'n Goblins, Resurrection: Shadow of the Beast, Pixel Heaven 2016, Talking Heads: to cheat or not to cheat ?, Johnny presents, Doom, Werewolves of London, Night of the Valkyrie, Game Over II, The Lands of Zador, SID Adventures: Linus, Platman, Qwak. Vortex Crystals. Soulless and Fast bitmap plotting.

http://ka-plus.pl/

Amiga Future Issue 122

This is an English- and German-language Amiga magazine. In this edition: Foreword, News, Playfield, Chaos Guns, Crystal Kingdom Dizzy, Vampire: Curse Of The Night, Total Pattern, Blocky Skies, Lila, Center Court 2, WinUAE, OS4 Depot, Sum, USA 1011 MP3 Player, Furia 33 , Trevor's Soapbox, FPGA, Classic Reflections:. Eyetech Group (1), Demoscene, Blitz Basic (1), Richard Lowenstein, Matthias Heinrichs, and Christoph Gutjahr

AMIGA FUTURE
122
FURIA 33
TURBOKARTE FÜR AMIGA 600
BLOCKY SKIES
CHAOS GUNS PREVIEW • VAMPIRE: CURSE OF THE NIGHT
TOTAL PATTERN • LILA • CENTER COURT 2 • WINUAE 3.3

http://www.amigafuture.de/

The Retro Hour - Amiga Format Magazine

The Retro Hour is a professional podcast created by Dan Wood & Ravi Abbott. The Retro Hour aims to help the world understand the European gaming industry. In this episode: Amiga Format Magazine

http://www.theretrohour.com/amiga-format-magazine-the-inside-story-the-retro-hour-ep36/

AMIcast Issue 16

AMIcast is an English podcast about the world of Amiga. In this episode: An interview with Aaron Smith (Online-Shop "Amiga On The Lake").

http://www.amigapodcast.com/

Amigos podcast

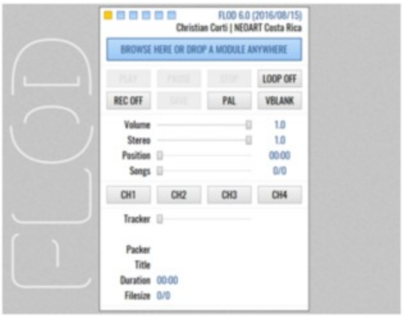
Amigos recently was updated new items include: Alien Breed, Ruff 'n Ready, Amigo Scour, The Hound of Shadow, CD32 TV, New Star Electronics, Prince of Persia, Water bombs at dawn, They sold a million, No Turrican is an island, Connecting the Amiga to a BBS, Populous, and The Amiga is good for your mental health.

http://www.amigospodcast.com/

Flod

Flod is a JavaScript / AS3 module written by Christian Cort, its able to play on the Amiga music in a modern browser.

Christian writes
This site requires a modern browser compatible with ES6. The latest versions of Chrome, FireFox, and Opera are supported (Edge will be supported in the future). Internet Explorer does not support ES6 nor the Web Audio API, so it will never work.


FLOD 6.0 (2016/08/15)
Christian Corti | NEOART Costa Rica

Volume 1.0
Stereo 1.0
Position 00:00
Songs 0/0
Tracker
Packer
Title
Duration 00:00
Filesize 0/0

http://www.neoartcr.com/

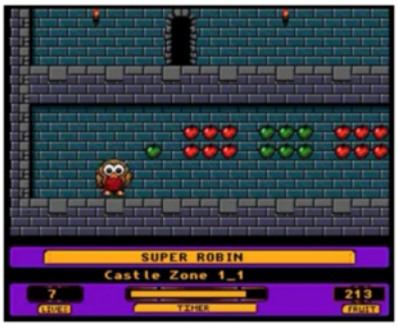
Super Robin

Developed using the Reality Game Creator, this game has 31 levels. You have to try to collect all the fruit and find the exit, avoid the enemies, jump into teleport doors, and push switches.

https://www.youtube.com/watch?v=cYvz7t7VRRQ

By amigapd
http://www.amigapd.com/

More info
https://gaming.youtube.com/game/UCuwiFnGoJ-Y3PBtc2ouJKpw

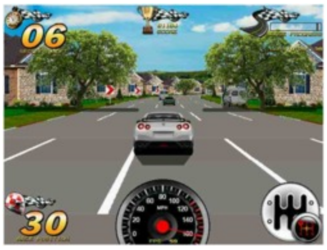


Amiga Racer v3.62

Michael St. Neitzel is working on the development of the game Amiga Racer.
www.amigatec.com/download/amigaracer.lha
http://distrita.com/amiga-racernews/

Minimal Setup
Please note: With the minimal system requirements some game play limitations may apply.
CPU: PowerPC (PPC) CPU with at least 800 MHz GFX: Graphics Card that is able to display 640x480x8 INP: Keyboard SFX: Soundcard OS4: AmigaOS 4.1 (Update 6)

Optimal Setup
CPU: PowerPC (PPC) CPU with 1 GHz or greater
GFX: ATI Graphics Card with FullHD Support
INP: Keyboard + Joystick
SFX: Soundcard

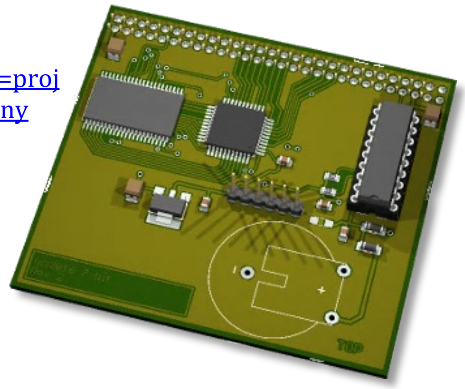


http://www.os4depot.net/index.php?function=showfile&file=game/driving/amigaracer.lha

New Amiga hardware developments

Retro 7-bit have created some new hardware for the Amiga. The IDEA ZII - IDE controller (Zorro II) for the Amiga 2000 Wicher 2000 - Turbo card for the Amiga 2000, Wicherek 14/28 - turbo card for the Amiga 500 and Wicher 5k / 1M - CHIP RAM for the   Amiga 500.

http://retro.7-bit.pl/?lang=en&go=projekty&name=nastepny



Prisma Megamix Music Card Available On AmigaKit.com

Amigakit Announcement
AmigaKit.com is pleased to announce stock and availability of A-EON Technology's music card. The Prisma Megamix Prisma is a new Zorro card expansion for big box Amiga's and also the Amiga 1200 Towers with an available compatible clockport socket (desktop A1200 version coming soon). Based on VS1063 chip, it can play back many music formats in full 16-bit 48Khz audio and additionally mix with the Amiga's native Paula sound. When it is decoding and playing back an MPEGA audio file or various other formats, it frees up the Amiga to do other things. An MHI driver is supplied with the card for AmigaAmp and other various music playing software.

Specifications
* Clockport and Zorro II interfaces (if operated in Zorro II slot, clockport can be used for compatible add-on cards)
* Compatible with Amiga A1200 T, A2000, A3000, A4000, A4000T.
* Based on VS1063 chip
* Playback of wide range of formats: MP2, MP3, WMA, OGG, LC-AAC, HE-AAC, FLAC, ALAC, IMA ADPCM, G.711 u-law, G.711 a-law, G.722, and WAV PCM formats
* Pass through port for Paula mixing. Paula mixing is automatically activated by hardware, no software interaction needed
* CDDA header
* MHI support for use with MHI compatible software such as AmigaAmp
* can be expanded by optional digital module (S/PDIF)

Direct Product Links
AmigaKit UK Product Page
http://amigakit.leamancomputing.com/catalog/GBP.php?url=product_info.php?products_id=1218

AmigaKit USA Product Page
http://amigakit.us/product_info.php?products_id=1218

AmigaKit Europe Product Page
http://amigakit.leamancomputing.com/catalog/GBP.php?url=product_info.php?products_id=1218

AmigaKit Canada Product Page
http://amigakit.leamancomputing.com/catalog/CAD.php?url=product_info.php?products_id=1218

Retro 7-bit

Develop new hardware for the Amiga. On the site: The IDEA ZII - IDE controller (Zorro II) for the Amiga 2000 Wicher 2000 - Turbo card for the Amiga 2000, Wicherek 14/28 - turbo card for the Amiga 500, and Wicher 5k / 1M - CHIP RAM for the Amiga 500.

http://retro.7-bit.pl/?lang=en&go=projekty&name=nastepny

Amiga DevCon 2016

(held on October 7 to 9) Amiga DevCon 2016 (October 6 and 7). Both events are a 100% volunteer effort as is the AmiWest show itself.

Some information about DevCon from the http://wiki.amigaos.net/wiki/Main_Page

Some of the articles covered were:

- The anatomy of an Amiga disk device driver. We will be dissecting the p5020sata.device which is a SATA device driver used in the upcoming AmigaOne X5000. This isn't just any custom made driver. It is melding between Linux libata and the traditional AmigaOS trackdisk device drivers.
- Hans de Ruiter will be presenting his tutorials on Warp3D Nova. For anyone that wants to program for Warp3D Nova be sure to obtain/borrow/steal a card for the DevCon. See the Warp3D Nova press release for assistance on choosing a card.
- Personal Projects. You are encouraged to bring your own personal projects to the DevCon where we, as a group, may be able to help you out.

Although the events have now passed, at least you have a heads up for next year!

Amiwest

http://www.amiwest.net/

Amiga devcon

http://wiki.amigaos.net/wiki/AmiWest_2016_DevCon

Personal Paint 7.3b for AmigaOS 3

A-EON Technology Ltd announced a new version of Personal Paint for AmigaOS 3 / 68k. The new version is a free upgrade for existing 7.3 customers. New customers can obtain PPaint from the main website: www.ppaint.com

The new 7.3b version is a maintenance release. It is available to download for registered customers as an LHA archive from the A-EON secure website or AMIStore App Store.

Features:

● Works with the Classic AmigaOS and WinUAE / Amiga Forever emulation
● True colour animation frame thumbs in the StoryBoard window
● High resolution gadget imagery
● Thin borders style GUI option for a more modern look
● 9 new built in brush shapes
● 100 new Custom Brushes
● Pressure sensitive drawing
● Dithered transparency
● New Airbrush Linemode
● Extended paint functionality with image processing tool

Required Hardware:

● Amiga A500, A500+, A1500, A2000, A3000, A4000, A4000T
● 1MB Memory
● Kickstart 3 or later
● Hard disk drive with 10MB or more free space

Recommended Hardware:

● AGA capable Amiga computer: A1200 or A4000
● 2MB Graphics (Chip) Memory or RTG graphics card
● 68020 CPU or better

DCFS and LNFS Explained

Hyperons web blog had this entry:
http://blog.hyperion-entertainment.biz/

The AmigaOS Fast File System (FFS) was created back in 1988 and, believe it or not, is still in use today by some die hard enthusiasts who insist it is still pretty good. FFS has several modes which has enabled it to survive far past its expiry date. Two of those modes have never been described before. That is about to change. Back in 1992, Randell Jesup added what is known as the Directory Caching File System (DCFS) mode to FFS. This was meant to speed up directory listings on floppy disks. A rarely used and mysterious mode with very little documentation.

Fast forward to 2001, when Olaf Barthel created an FFS re-implementation and added the Long Name File System (LNFS) mode. Until this point users were stuck with 30 characters due to the original implementation.

Curious? Read all the gory details on the most complete and official AmigaOS Documentation Wiki.
http://wiki.amigaos.net/wiki/DCFS_and_LNFS_Low_Level_Data_Structures

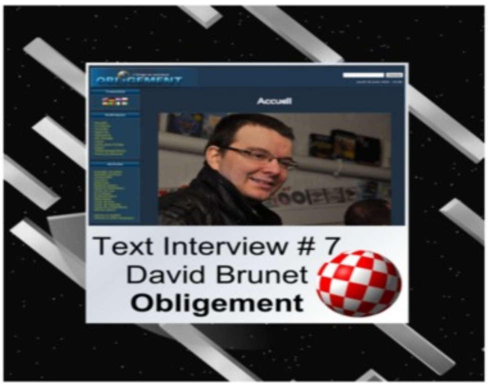
Format - DF0

Formatting disk...

AMIcast-Text Interview 7 - David Brunet - Obligement

AMIcasts guest is the guy behind the Obligement portal - David Brunet. Obligement is one of the most popular places in Amiga world. And it doesn't matter that the website is in French. :) Translation tools are very integrated. Let's talk.

http://www.amigapodcast.com/2016/08/amicast-text-interview-7-david-brunet.html



Internet Archive posted 10,000 browser-playable Amiga titles – go, play

https://techcrunch.com/2016/08/08/amiga/

The website says:

We can't vouch for the quality of all of the Amiga titles that were recently posted up on Archive.org, but there sure as heck are a lot of them – 10,000+, by the site's count, including favorites as *Where in the World is Carmen San Diego, King's Quest* and *Double Dragon*, along with what looks to be a fair amount of redundancy. I'm not really sure what the difference is between *Deluxe Pac Man v1.1* and *Deluxe Pac Man v1.7a*, but I suspect it's fairly minor, even for completists.

https://archive.org/details/softwarelibrary_amiga

Amibian

Amibian is a raspberry Pi3 image with the Uae4arm emulator installed. Amibian transforms your Raspberry Pi3 in an Amiga. Features: Easy to install Midnight Commander for working with files, USB and Ethernet support.

What you need

- A Raspberry Pi3*
- A fresh copy of Amibian
- Win32 Disk Imager**
- Amiga Kickstart roms
- Games, Programs, Workbench

* It runs also on over-clocked Rpi zero, Rpi 1, and Rpi 2
** This app is needed to flash SD
http://gunkrist79.wixsite.com/amibian

Retro Commodore

Retro Commodore has a web page with a number of high-quality scans available. The most recent scans are: Joystick, SUPER Pascal COBOL64 / COBOL128 Software System, Data Becker FORTH für Commodore 64, Scala Multimedia 200/300 full box, computer, Lancelot Manual, Commodore 1701 Monitor User's Guide (German), SOFT Today, Inter Word 2.0 (Danish), VIC Computing Christmas 1981 Commodore 1081 Manual + Schematics, TwinExpress User's Guide, Supra Drive Operator's Manual and Word Sync Addendum, angiotensin II Cover + Instructions, VIC Forth Handbook (Swedish) Benutzer Handbuch zu Digital Image (Germantown), Ami-back Tools manual addendum, MiG-29 Fulcrum (HitSquad) PagerSetter User's Guide + Extras, Network PC Instructions, SOFT type-ins, Pool of Radiance Manuals, Amiga ROM Kernel reference manual: Exec, Introduction to the Amiga 2000 , BASIC programs for the Commodore 64 (Dutch), Maxtron Computer Cassette Unit, and Railroad Tycoon.

http://www.retro-commodore.eu/

Vampire 2 Review

Dan Wood has a YouTube review of the Vampire 2 Turbo Card for A600 as well as demonstrating how to install the hardware. He shows how fast the A600 is with this new Turbo Card installed.

https://www.youtube.com/watch?v=8S3B8a8N83k

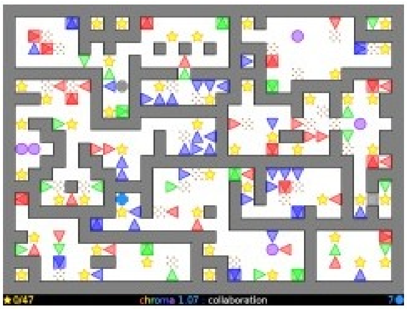
WHDLoad

WHDLoad permits the Amiga user to read diskette-based games from your hard drive. Modified in this recent update are: Assassin Special Edition, Hard'n'Heavy, The 100 Most Remembered C64 Game Tunes, John Madden Football, Flimbos Quest, Manhunter, Fire Force, Beyond Dark Castle, F-16 Combat Pilot, Strip Poker Strip Poker 2 / Deluxe Strip Poker, Line of Fire, Rock Challenge, and Final Fight.

http://www.whdload.de/

Chroma - AROS

Chroma is an abstract puzzle game. A variety of colourful shapes are arranged in a series of increasingly complex patterns, forming fiendish traps that must be disarmed and mysterious puzzles that must be manipulated in order to give up their subtle secrets. Initially so straightforward that anyone can pick it up and begin to play, yet gradually becoming difficult enough to tax even the brightest of minds. Have you got what it takes to solve Chroma?



http://archives.aros-exec.org/?function=showfile&file=game/puzzle/chroma.i386-aros.zip

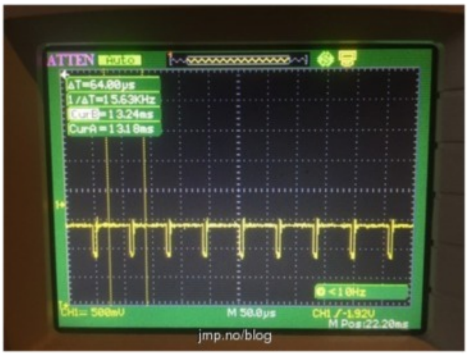
Amiga and the 5th audio channel

Stian Søreng has developed a system to add a fifth audio channel to the Amiga computer. He uses the composite video channel to produce an extra audio channel.

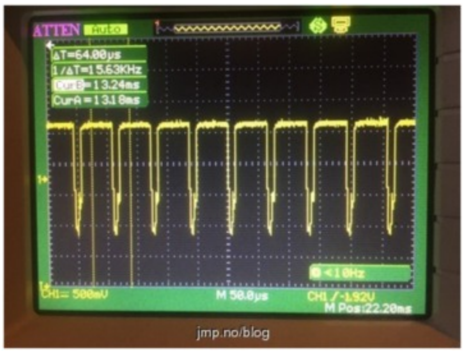
His blog states.

Ok, so it's not a real, real audio channel, but I'm talking about something capable of outputting audio: the Amiga composite video port. It's the same socket as the other two audio ports, it outputs terrible video quality and is not in use on most Amigas. So why not put it to work? The concept is simple: a bright color outputs a high voltage, while a dark color outputs a low. By changing the screen colors at just the right speed, I'm able to generate audio frequencies and use them to play music.

BLACK



WHITE

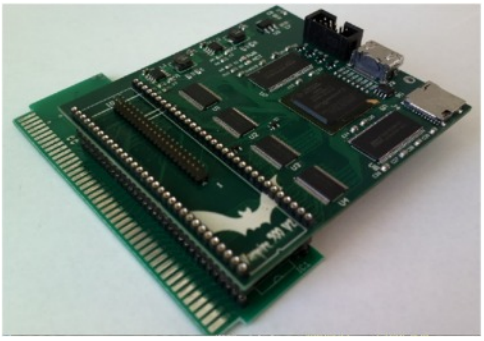


http://jmp.no/blog/a-5th-audio-channel-on-the-amiga

Vampire 500 v2

Simo Koivukoski created a video showing an Amiga 2000 with the Vampire 500 v2 accelerator. One of the first tests is with RiVA 0.50 and Picasso II (CyberGraphX screen mode 320x240 / gray). The second test is MP3 playing with Amplifier and Delfina Lite DSP.

https://plus.google.com/+SimoKoivukoski/posts/ZQ4aKvobvEn



AmigaRemix

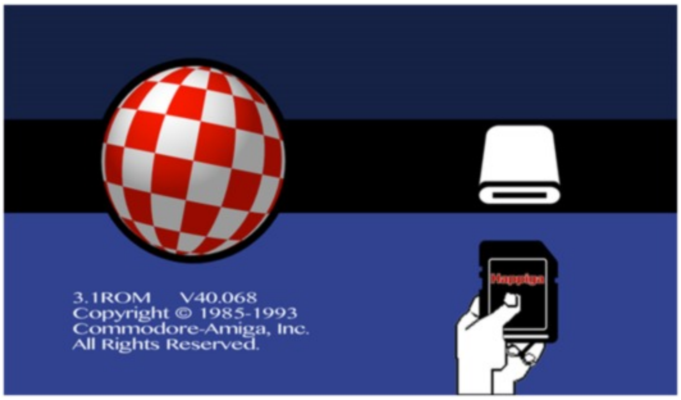
Amiga remix is a webpage with remixed Amiga music The following Amiga music was added to the web page: Assassin Special Edition - Title, Lotus Turbo Challenge 2 - Title, Agony - Loading Mountain (Breathless RMX), Turrican 2 - The Greath Bath, Pinball Dreams - Beatbox Theme (daXX Remix), Human Target (daXX Remix), Monday (Maf remake) and Another World Ending (Another Rebelcitizen Ending Remix).

http://www.amigaremix.com/

Happiga

Happiga is a emulation system that can be used on (amongst other things) the Raspberry Pi. It uses the UAE4ARM emulator to emulate an Amiga 4000 AGA including Picasso96. You can use a real keyboard and joystick via the Keyrah V2. Changes in this version: An update for Raspbian, support for 128 MB Z3 RAM, a new menu and other small improvements.

http://www.happi-game-center.com/fr-happiga.html



3.1ROM V40.068
Copyright © 1985-1993
Commodore-Amiga, Inc.
All Rights Reserved.

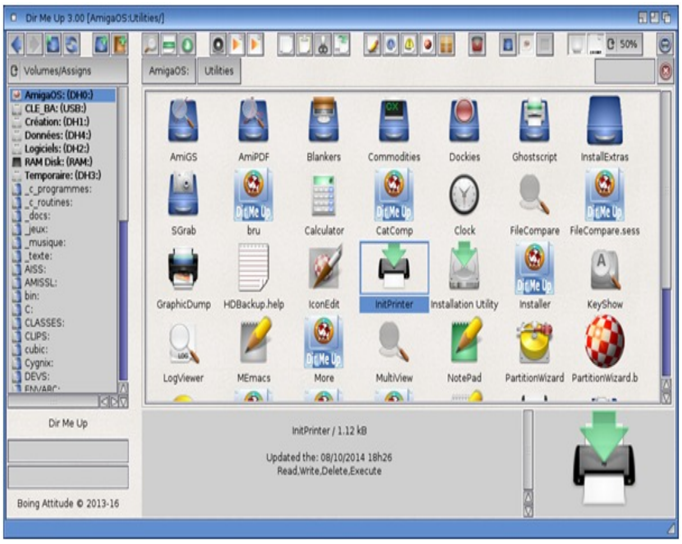
Dir Me Up 3.0 - Amiga

Dir Me Up is a program to explore and manipulate files, links, and drawers of your Amiga.

You browse quickly and easily the content from your hard drives, USB sticks or CDs with advanced features: browsing history, current path being accessible via buttons, multiple sorts, filters, preview (image, text)... Handle one or several files: you can easily run, view, cut, copy, paste, rename, comment, manage attributes and much more. Of course, you can choose to click icons, use a contextual menu or press the keyboard shortcuts. Connect Dir Me Up with the best existing software on Amiga OS 4: video, music, internet are all pleasures you can enjoy by a double-click.

Changes in this recent version: Shell-, ARexx- or Python scripts. Configurable Toolbar buttons, progress bar, improvements for dialog screens and menus, graphical view of partitions, and much more

http://glames.online.fr/products/dirmeup.html



BoingsWorld Issue 78

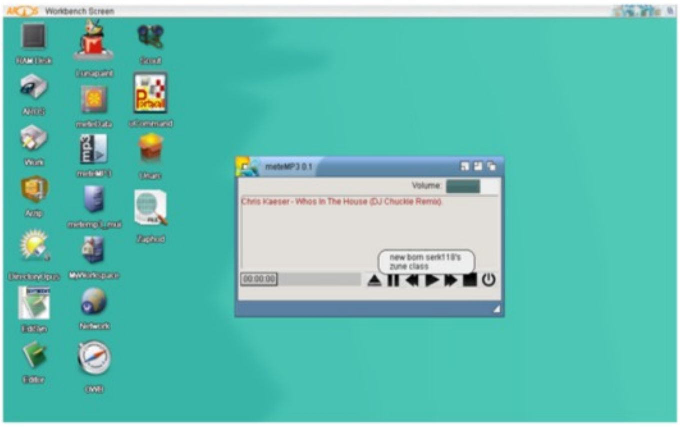
A new episode of BoingsWorld (German language podcast) is now available with the following articles: Warp3D Nova, FPGA Graka MNT VA2000, X5000, Solid Gold Collector's Edition, Center Court 2, Video Center Court 2 AGA, Project: Lila, Amiga I miss you, AmigaG-TI, Computermuseum München, Amiga Wiki, and Kerkos

http://boingsworld.de/

meteMP3

MeteMP3 is an Mp3 player for Aros. A new version is now available with recent changes: Drag & Drop, custom memory allocation / de-allocation, command line play via cli/shell, and improvements for the playlist.

http://archives.aros-exec.org/?function=showfile&file=audio/play/metemp3.i386-aros.zip



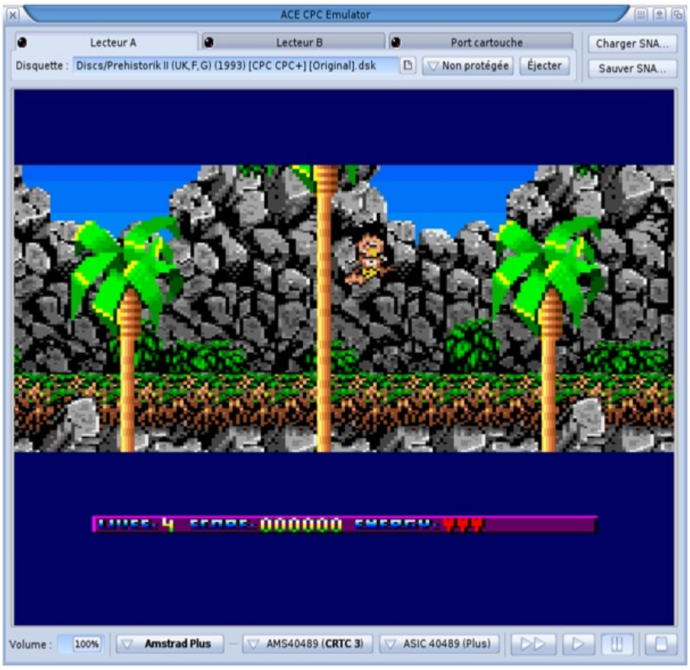
ACE v1.12 - MorphOS

ACE is an Amstrad CPC emulator developed by Philippe Rimauro. The emulator is written in C and its GUI is based on MUI 4. The changes in this version are: Update from gcc4 to gcc5. Improvements for CRTC 0/1, VSync, Kempston mouse, AMX mouse, Tape2HD and the disc drives. .

ACE is not based on any existing emulator and was coded from scratch, using a real CPC as a reference in order to provide the best Amstrad CPC emulator ever. The current version of ACE does not offer all the expected features yet. Nevertheless, most of the Amstrad CPC and Amstrad Plus games and demos are

working perfectly; In addition to the high quality emulation, ACE features full screen and overlay display, editor and analyzer windows for all emulated CPU and chipsets, advanced breakpoints management, a very powerful set of AREXX commands for cross-development, a plugin interface... and much more...

http://aminet.net/package/misc/emu/ACE-MorphOS



Ignition Beta 4 - AmigaOS 4.x

This AmigaOS 4.x version is made by Achim Pankalla, which is based on the 68K version from Axel Dörfler. Ignition is a modern and powerful spreadsheet application with a small but relational all-purpose database. You can realize your address database and manage your apartment house or whatever you like. You can calculate with times and dates as if they were regular numbers. The whole system can be controlled via ARexx. You have several possibilities to visualize your data. The different diagram and object types are implemented via an add-on architecture. It's currently in beta status, meaning it's pretty much feature complete, but might still have a few annoying bugs.

Beta 4 changes are

- An improved and reworked version of the German manual thanks to Helmut Haake.

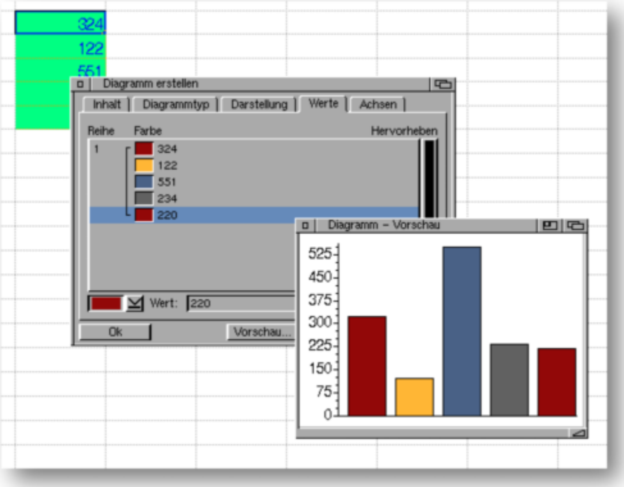
- The import of TurboCalc files should now work much more seamless, and now supports multiple pages, among others.

- CSV import now supports all ASCII characters.

- A Dutch language catalog thanks to Johan Banis.

- A number of crucial bug fixes

http://ignition-amiga.sourceforge.net/



Amiga Future November/Dezember 2016 • € 6,50 • CD-Edition € 9,50 • www.amigafuture.de
123
AMIGA
FUTURE
RESHOOT
GAMESCOM 2016
MESSEBERICHT
ALARCITY • GIANA SISTERS SE
TALES OF GORLUTH 2
INFECTION • SYNERGY • FRYIN PAN
TALES OF GORLUTH

Amiga Future issue 123

A new version of the German/English Amiga Future magazine has been released with the following contents:

Previews:
Playfield, Alarcity, Tower 57.

Reviews:
Synergy, Giana Sisters S.E., Reshoot, Tales of Gorluth 2, OS4 Depot Spiele, Infection, Amibian, FryingPan.

Special:
Trevors Soapbox, Classic Reflection: Eyetech Group Part 2, Demoscene.

Workshops:
Cooking for beginners, Blitz Basic Part 2.

Showreports
Gamescom 2016, Classic Computing 2016, Commodore Expo Las Vegas 2016, Evoke 2016.

Others:
Interview Amiga Waves, Editorial, Content, News, Imprint, Content CoverCD, Letters to the editor.

Unfortunately, the Developing Demo Effects Workshop we had planned for this issue will not now be included, but we hope to offer this workshop in a future issue, at a later date.

CoverCD contents:

Extras
LuettjeBookholler, aheart-C64games

Public Domain ...

AmigaOS4
2PicPuzzle_os4, aiss, amicraftnova, aos-imagefx-icons, ffmpeg_os4, jabberwocky, libbz2, MCE-OS4, odyssey, ReportPlus-OS4, reportplus, wet_update, WormWars-OS4,

AROS
frontier-elite2-gl.aros, webptools051_aros,

Classic
A-Shell, Adom, Adom_68020, Adom_68040, ADOM_Guide, AnnoFortnightly, APool-GL, Bally, bkp, BootEm, Bootleg_21, boot_star, Calimero_3.6, CopPack10, cranker, DataPack, DefPack, dmg2img_1.6.5, HollywoodSP, HWP_MUIRoyale, HWP_RapaGUI, Identify, impvcfsm, Lepani, lhasa, libdwg-0.6, MaxPak, MCE, MegaBoot, MUIbase-3.3, MultiBoot, NOMAD_Boot, OmegaPacker, PDCurses, pgs33a_33, Preferences, ReportPlus, Savage, Sensor, ShellUpdate, Smart, soliton_decks, Uedit4k2, unz550xA, wet_update, WormWars, zint_2.4.2, zip-bin, zip-src,

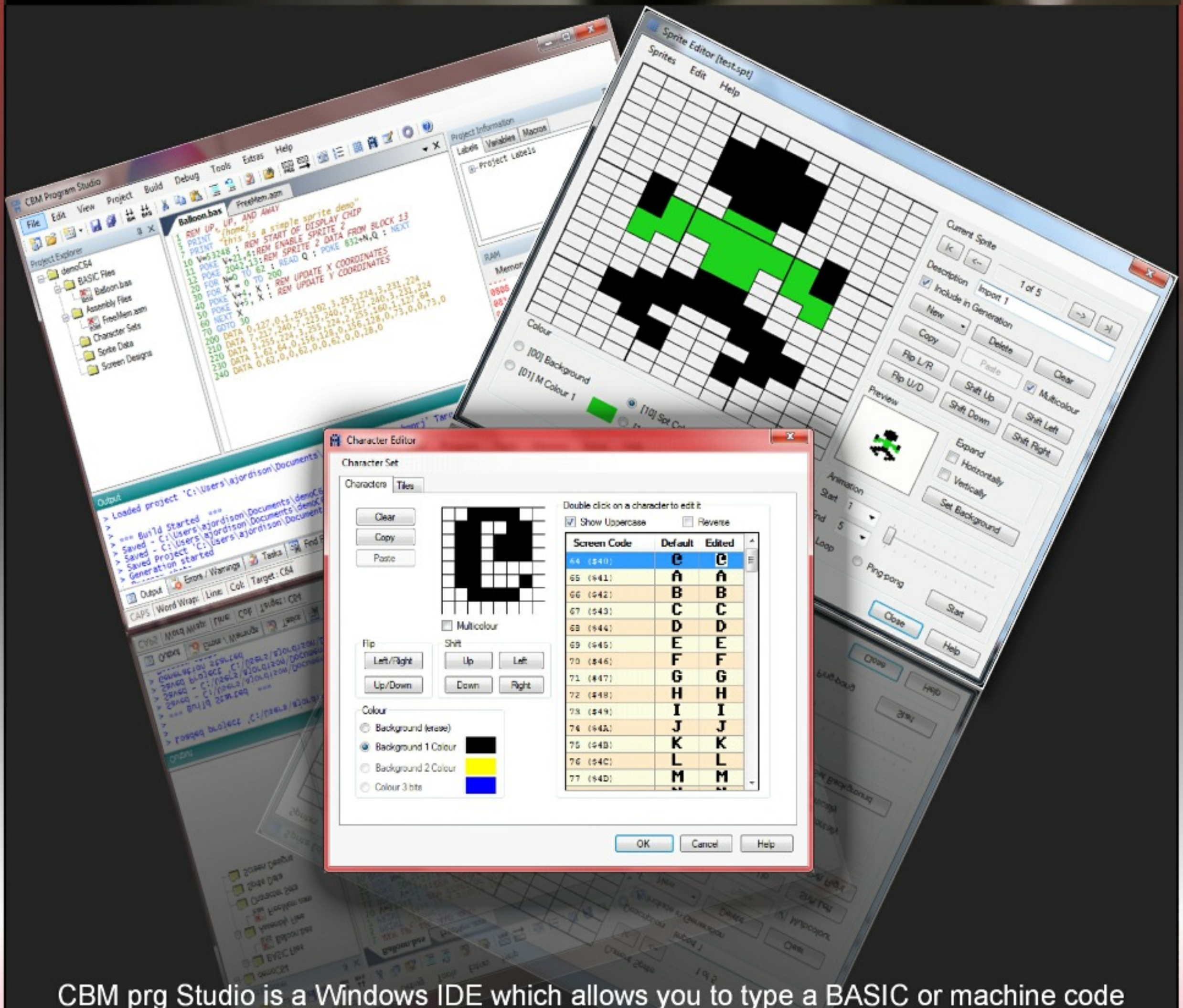
MorphOS
2PicPuzzle_mos, gnuplot-MOS, lazymines_mos, MCE-MOS, ReportPlusMOS, WormWarsMOS, yahzee_mos,

Multimedia
BoingsWorld Podcast Episode 78 and 79, Retrokompott 19 (GameCon)

http://www.amigafuture.de/kb.php?mode=article&k=4972

CBM .prg Studio

A handy Commodore 8-bit program development utility for Windows



```
1 REM UP, UP, AND AWAY
5 PRINT "{home}"
7 PRINT "this is a simple sprite demo"
10 V=53248 : REM START OF DISPLAY CHIP
11 POKE V+21,4:REM ENABLE SPRITE 2
12 POKE 2042,13:REM SPRITE 2 DATA FROM BLOCK 13
20 FOR N=0 TO 62 : READ Q : POKE 832+N,Q : NEXT
30 FOR X = 0 TO 200
40 POKE V+4, X : REM UPDATE X COORDINATES
50 POKE V+5, X : REM UPDATE Y COORDINATES
60 NEXT X
70 GOTO 30
200 DATA 0,127,0,1,255,192,3,255,224,3,231,224
210 DATA 7,217,240,7,223,240,7,217,240,3,231,224
220 DATA 3,255,224,3,255,224,2,255,160,1,127,64
230 DATA 1,62,64,0,156,128,0,156,128,0,73,0,0,73,0
240 DATA 0,62,0,0,62,0,0,62,0,0,28,0
```

Screen Code	Default	Edited
64 ($40)	@	@
65 ($41)	A	A
66 ($42)	B	B
67 ($43)	C	C
68 ($44)	D	D
69 ($45)	E	E
70 ($46)	F	F
71 ($47)	G	G
72 ($48)	H	H
73 ($49)	I	I
74 ($4A)	J	J
75 ($4B)	K	K
76 ($4C)	L	L
77 ($4D)	M	M

```
> Loaded project 'C:\Users\ajordison\Documents\demoC64'
> *** Build Started ***
> Saved - C:\Users\ajordison\Documents\demoC64
> Saved - C:\Users\ajordison\Documents\demoC64
> Saved Project 'C:\Users\ajordison\Documents
> Generation started
```

CBM prg Studio is a Windows IDE which allows you to type a BASIC or machine code program and convert it to a '.prg' file, which you can run on an emulator or the real hardware. It also includes character, sprite and screen editors and a fully featured 6510/65816 debugger.

The following machines can be developed for:

- Commodore 64 (and SuperCPU)
- Commodore 128
- Commodore VIC 20
- Commodore 16
- Commodore Plus/4
- Commodore PET BASIC 2 machines, e.g. PET 2001
- Commodore PET BASIC 4 machines, e.g. PET 4000/9000

http://www.ajordison.co.uk

Happy Programming

(primarily for Commodore 264 BASIC 8-bit computers)
(i.e. / e.g. Commodore 16/+4, Commodore 128 -in 128mode)
By **John Fielden**

In this issue I have managed to find time and energy to create something new, and that is the point of the article. No matter what is going on in life, I'm always happy programming! And of course, sharing.

I'm even happier if these published articles help people in any way, whether on programming matters or those difficult to talk about, socially taboo issues -- like the Spectrum, Amstrad, etc. (Ha-ha-ha!).

You may remember that previous articles have mentioned bullying as one difficult subject. And it just so happens that is relevant again. In this issues article, the programming consists of an elite table. These are prevalent in all walks of life. The most memorable is a book I read by a Psychic Master who claims to have been helped by The Messiah Christ Jesus to put 'The Gifts' in order. I was initially surprised to see Healing at the bottom, and Wisdom about seventh. Though Discernment and Understanding were are at the top of that list. The author of said book sadly died on 29 February 2016, and there does not seem to be anything much of his works that has survived. And I must assume that is how he wants it. So, haven't named him. R.I.P. (A lot of people testified back to the presenter on the radio that they'd been helped by him. Me included.)

Now, this article would be dull, if it consisted merely of the first program "Top 10" which is to become the basis for our experiment. Whether the c16 (etc) is capable of Re-dimensioning Arrays. Which was not an early consideration by makers of languages because people were, are expected to know the size they want and have it all decided on paper before typing up their program. But times change of course.

In this article, we will explore ways to RE-DIMension the array matrix without the existence of the actual command. Not being about sorting array methods, these are not discussed. With regard to the fact computers tend to use tables to keep some kind of hierarchy up to date. Full provision has been added to allow control to be passed solely to the user to facilitate sorting out their own lists. -Which may in itself be a unique advantage novelty!

Data invariably needs saving for another day with small size tables used in the examples, and are easy to expand. As there are invariably right ways and wrong ways.

We'll try:- first; change the program. (In this case. the variable 'T' on line 30. But this changes the amount of T-data that the computer expects to load into memory. So, an error will occur and in the standard program that makes no account for this. The program will crash. (Though, by my reckoning, it should come back with DS error #50 in the User Manual.

You could however, make a new file with the required number of list spaces. first; change the load-in part of the program to read the old value of T as a constant. Whilst keeping the save-to disk T-data to the new top number for the table.

This may have to have a new file name. If it is on the same disk for instance. The file name to load in being different to the one saving out. You may wish to change the handling of this to the user -within the program!

secondly; load in the data of the file which is to be 'expanded'.

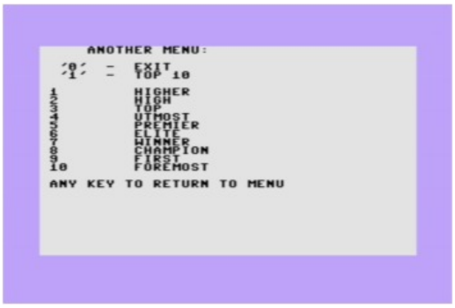
thirdly; type in the extra data for the newly improved table.

fourth; save this work to the new table

next; change the load in to the new table number, and new name; on the main program.

finally; save the main program. All up to date. You might feel it better to save all versions of the main program to make future expansions quicker. But here we are leaping ahead. We haven't started with our Top10 yet.

```
COMMODORE BASIC V3.5 60671
BYTES FREE

READY.

10 REM *RE-DIM ARRAY*
20 REM **WORK ROUND*
30 T=10
40 DIM T$(T)
50 REM
60 REM *DON'T FORGET THE HOUSE-
WORK-*
70 REM *ONCE THE ARRAY IS
CHANGED!*
80 REM

100 REM -THE TOP 10-
110 SCNCLR
120 PRINT"    ANOTHER MENU:"
130 PRINT
140 PRINT"  '0'  -  EXIT"
150 PRINT"  '1'  -  TOP 10"
160 PRINT

200 GETKEYA$
210 IF A$="0" THEN END
220 IF A$="1" THEN 300
290 GOTO 200

300 REM *TABLE*
310 FOR J=1 TO 10
320 READ D$
330 PRINTJ,D$
340 NEXT J
350 PRINT
360 PRINT" ANY KEY TO RETURN TO
MENU"
370 GETKEY A$
380 GOTO110

10000  DATA
HIGHER,HIGH,TOP,UTMOST
10010 DATA PREMIER,ELITE,WINNER
10020  DATA
CHAMPION,FIRST,FOREMOST

READY.
```



```
    ANOTHER MENU:
  '0'  -  EXIT
  '1'  -  TOP 10
1         HIGHER
2         HIGH
3         TOP
4         UTMOST
5         PREMIER
6         ELITE
7         WINNER
8         CHAMPION
9         FIRST
10        FOREMOST
ANY KEY TO RETURN TO MENU
```

First
we change line 310, The '10' to a T. Otherwise when we change the program to increase the amount of places allowed on the table, we will still only get 10 in the PRINTed list. (Instead of the 11 and 12, I have opted for in the following example/s)

Second.
Change line 50 to:-

```
50 TT$="TABLE"   :REM *DISK FILE
FOR ELITE TABLE*
```

This makes naming the file easier, using the variable at line 8230 (for load), and line 9510 (to save).

Third.
We will come to the rest of the changes in due course.

In the following code that should be added later. Lines 9200 to 9340 are the subroutine ready to load in the Table. For example when after shutting the computer down for the night, the table can be recalled to memory the next day. Whilst lines 9500 to 9620 are the ones that ran that previous night when "You", the user asked your beloved computer to store them more permanently. -ie. to disk. Knowing that despite needing sleep, you couldn't wait to return to such an interesting program the next day!

```
?SYNTAX ERROR
READY.

8200 REM **********************
8210 REM ***LOAD ELITE TABLE***
8220 REM **********************
8230 OPEN1,8,2,TT$+",S,R"
8240
IFDS=62THENCLOSE1:GOTO12000
*(trap for errors!)
8250 PRINT"O'LOADING TABLE"
'<CLEAR HOME>
8280 FOR U=1 TO T
8290 INPUT#1,T$(U)
8300 PRINT U, T$(U)
8320 NEXT U
8330 PRINT"LOADED ET."
8340 CLOSE1
8350 GOTO 400

BREAK
READY.

9499 REM ******************
9500 REM *SAVE TO DISK*
9505 REM ******************
9510 PRINT"O' SAVING:   ";TT$
'<CLEAR HOME>
9520 OPEN1,8,2,TT$+",S,W"
9530 FOR Q=1 TO T
9540 IFER<>0 OR EL<>0 OR ST<>0  OR
DS<>0 THEN TRAP12000
9550 PRINT#1,T$(Q)
9560 REM *SPACE ADVISABLE SEE
1560*
9570 PRINT#1
9580 NEXT Q
9590 PRINT#1
9600 CLOSE1
9650 GOTO 400
9700 STOP   ***

BREAK
READY.

12000 REM *ERROR TEST TABLE*
12010 PRINT"Ó           ERROR TEST"
<CLEAR HOME>
12020 rem IF ER=14 THEN 11000
12030 rem IF ER=18 THEN 11000
12040 rem IF ER=30 THEN 7900
12200 PRINT ER,ERR$(ER), "ER"
12210 PRINT EL,EL$(EL), "EL"
12220 PRINT ST,ST$(ST), "ST"
12230 PRINT DS,DS$(DS), "DS"
12500 STOP


READY.
```

We've covered the TRAP/RESUME & RESUME NEXT command in prior issues. All the codes are in the relevant manuals. It's better to declare the TRAP [LINE] early in the program.

That leaves us to fill in the rest of the base code So that the program makes sense. In other words, so that the subroutines can be called as necessary. And the Table can be changed and edited as promised etc.

Time for a cuppa, and a Think!

Brainstorm to Solution:

I want to have a highlight bar, instead of or an alternative for the standard: from x to y. that I normally plump for as it's easier to code. Provision to expand (or reduce) the (pre-saved) table size; may prove problematic when all of it won't fit onto one page. This will have to be anticipated in the code, and tested at the design stage.

Also consider that things may have expanded so much, we don't know the size of the data on disk.

Or the possibility that the user may want to reduce the DIMensioned provision to actually save memory/ load/save time. Perhaps speed is required over size, suppose departments are separating and don't need to be on

```
STRING TOO LONG ERROR
READY.
```

Add the following code: <SHIFT> + <RETURN>
USE THE ' Symbols, for information only!   <SHIFT>+<RETURN>

```
DELETE 60-290
DELETE 350-380

30 T=10 : T=11 :DIMT(12)            'for various tests!
40 DIM T$(T)
50 TT$="TABLE"  :REM *DISK FILE FOR ELITE TABLE*
60 P=15                             ' Page list
70 B1$="’"                          ' revs on
80 B2$="ÒĐ"                         ' revs off and black
90 CR$="‘"                          ' 1 cursor down arrow
100 CD$="‘‘‘"                       ' 3 down
110 S$="Ý     "                     ' 1 left and 5 spaces (adjust to own requirements.
120 PL$="““‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘"   ' 2 HOME, 23 /24 DOWN ARROW
130 B4$="’ž"                        ' revs on and colour green (key 6, with right ctrl on pc)
140 COLOR0,2 : COLOR1,1        ' the ink is black, the page is white.  Together we learn
                                              - we're very different!
150 TRAP 12000                     ' !!!  prevents BREAK __ERROR IN xxxx  Messages.

330 T$(J)=D$
```

one large database. It may be that sequential files (denoted with the 'S' in the OPEN statement) aren't the best way forward. That leaves Relative or Binary files to consider. ...As there is much on relative files (e.g. in 'Commodore 1551 disk drive users guide' etc.) this article will only cover 'seq.' files).

It seems best to first finish the started program. With the "From x, To y?" change method. And a simple count for the maximum a screen can hold, and wrap around on the maximum. The version with a selector bar will require more thought. And the idea for making the table bigger and save-able on one program may not be possible with SEQuential files.

```
400 REM *MAIN*
410 SCNCLR
420 PRINT"            ’ ÂTABLEÄ Ò" 'spells "TABLE" in revs on & flash on mode.
430 PRINT
440 PRINT"CURRENT FILE NAME: ";TT$
450 PRINT
460 D=1
470 FOR J=1 TO T
480 IF J => P THEN STOP
500 REM
510 REM    IF J=D THEN PRINTB1$;
550 PRINTJ;S$;T$(J)
560 REM    PRINTB2$
570 NEXTJ
600 REM
610 PRINT
620 PRINT"KEYS: 0/Q=EXIT, 1/L=LOAD TABLE,"
630 PRINT"2/S=SAVE TABLE, 3/E=EDIT, 4/X=SWAP"
640 PRINT" BAR: <SPACE> = EDIT, <RETURN> = SWAP"
650 PRINT" USE <UP> / <DOWN> ARROWS.
660 REM  PRINT
670 PRINT"5=ENLARGE, 6=REDUCE COUNT, 7=EDIT FILE"
680 GOSUB 1400
700 REM
710 GETKEYA$
720 IF A$="0" OR A$="Q" THEN 7900
730 IF A$="1" OR A$="L" THEN STOP
740 IF A$="2" OR A$="S" THEN STOP
750 IF A$="3" OR A$="E" THEN 900
760 IF A$="4" OR A$="X" THEN 1200
770 IF A$="5" OR A$="N" THEN STOP
780 IF A$="6" OR A$="R" THEN STOP
790 IF A$=" " THEN C=D :PRINTD;"  ";:GOTO940
800 IF A$=CHR$(13) THEN 2000

810 IF A$="Ñ" THEN U%=1 :GOSUB 1500
820 IF A$="“" THEN U%=2 :GOSUB 1500
830 IF RT>0 THEN 2020
890 GOTO 710
900 REM *** '3' OR 'E' ***
910 PRINT PL$
920 INPUT "NUMBER TO CHANGE";C
930 IF C<1 OR C>T THEN 400
940 PRINT
945 PRINTPL$;: INPUT" CHANGE TO: ";C$
950 T$(C)=LEFT$(C$,25)
960 C=0 :C$=""
990 GOTO 400

1200 REM *'4' OR 'X'
1210 PRINT
1220 INPUT" FROM 'X' ";X
1230 IF X<1 OR X>T THEN 400
1240 INPUT" SWAP TO 'Y' ";Y
1250 IF Y<1 OR Y>T OR Y=X THEN 400
1260 T$(0)=T$(X)
1270 T$(X)=T$(Y)
1280 T$(Y)=T$(0)
1290 T$(0)="" : REM *RESET 4 NEXT TIME*
1300 GOTO 400
1400 REM *BAR*
1410 PRINT"““";CD$;CR$;
1420 PRINTB1$;D;S$;T$(D);B2$
1430 RETURN

1500 REM *MOVE BAR*
1510 PRINT"““";CD$;
1520 FORJ=1 TO D
1530 PRINTCR$;
1540 NEXT J
1545 REM  IF RT=1 THEN RT=2 :GOTO 1560
1550 PRINT"
        "
1560 PRINT"Ñ";D;S$;T$(D)             'Arrow up once
1565 IF C1=D THENPRINT"Ñ";B4$;C1;S$;T$(D);B2$ 'Arrow up once
1570 IF U%=2 THEN 1650
1580 D=D-1 :IF D=0 THEN D=T
1590 REM   PRINT"““";CD$; 'clear home + clear home (*)
1600 REM  FORJ=1 TO D
1610 REM  PRINTCR$;
1620 REM  NEXT J
1630 REM  PRINTB1$;D;S$;T$(D);B2$
1640 GOTO 1680
1650 REM *** D O W N ********
1660 IF C1=D THENPRINT"Ñ";B4$;C1;S$;T$(D);B2$ 'Arrow Up once
1661 REM   IF C1=D THEN PRINT"Ñ";C1;S$;T$(D)                 'ignore
1670 D=D+1 :IF D>T THEN D=1
1680 PRINT"““";CD$;                    'as 1590;
1690 FORJ=1 TO D
1700 PRINTCR$;
1710 NEXT J
1720 PRINTB1$;D;S$;T$(D);B2$
1790 RETURN
```

* 2 clear home' keys are safer than one. when user input is required. Due 'esc T' which creates it's own window, being possible by hook or crook. 2 inverted S' on the BASIC screen restores the window/screen to it's normal size.

```
?BAD SUBSCRIPT ERROR
READY.

2000 REM *<RETURN>*
2010 RT=1  :C1=D
2030 PRINT"Ñ";B4$;C1;S$;T$(D);B2$ 'up arrow
2040 GETKEY A$
2050 IF A$="Ñ" THEN U%=1 :GOSUB1500      'up
2060 IF A$="“" THEN U%=2 :GOSUB 1500
2070 IF A$ = CHR$(13) THEN 2100
2090 GOTO2040
2100 REM *WAIT FOR SECOND RETURN*
2110 C2=D
2120 REM  C3=C1 : C1=C2 : C2=C3
2130 T$(0)=T$(C1)
2140 T$(C1)=T$(C2)
2150 T$(C2)=T$(0)
2160 T$(0)="" : REM *RESET 4 NEXT TIME*
2170 RT=0 :C1=0 :C2=0 :C3=0
2200 GOTO400

7900 REM
7910 SCNCLR
7920 PRINT"   EXIT, ARE YOU SURE Y/N ?"
7930 GETKEYA$
7940 IF A$="Y" THEN 8000
7950 IF A$="N" THEN 400
7960 GOTO 7930
8000 END

READY.
```

Now the program can be - <SHIFT> +<RETURN>

Have a play around with that. And when we return to the programming, it will be time to look at actually Saving & Loading Files.

```
...
  EXIT, ARE YOU SURE Y/N ?

READY.
```

Here things start to get really exciting. Input, or copy/paste the code from 8200 to 12090. Now, if we could be certain to always follow the rules, we could get on with this quite happily. But, as you will know, things don't always run smoothly. So, it is time for an exercise in Debugging! We are actually going to get things wrong ON PURPOSE to see what might go wrong!

So, Set up on a completely new disk or file. RUN the program. And instead of pressing 'S' to save, PRESS 'L' instead!

This should produce the following error:-

```
?ILLEGAL QUANTITY ERROR IN 12020
READY.
```

Type "HELP" to see what the fuss is about. And we see that ERR$(ER) appear to be the culprits. But actually they are merely the messengers. And as the saying goes "Don't shoot the messenger, if the message is bad!" -until you find a new one???

Type "PRINT ER" or "?ER"

The number 14 occurrs: Pg. 163 of The commodore 16 manual, informs us that -

"A number used as the argument of a function or statement is outside the allowable range."

In other words because we haven't saved the Table yet, the number being requested to call it back is too high. As we start the FOR/NEXT loop at 1. But as yet have nothing saved.

Now, if the program is for commercial use. We don't want it crashing on some unsuspecting user. So. lets find a way around the crash. Take out the REM statements in lines: 12020, 12030 & 12040

Next we look at how the computer handles a mis-match of files. Change line 30, so that it causes variable 'T' to equal 11. RUN the program, Edit and Change the list all you want, Save it, then stop the program. Change line 30 again to 10. And yes, RUN the program.

Most will have found The c16 (etc.) handles this with no qualm. Although the last item in the list will be lost. But, when we try the other way round. And here is the last chance to SAVE everything You've done. If You haven't already. And even if you have' do it again to ensure everything is up to the last second. (I've got down to version 'd' by now).

Change line 30 to read 'T' as 10. (You might need to use The DIM statement as we're getting into big numbers).

RUN the program, save the file.

Stop again and make T in line 30 equal 11.

Now RUN, and try to Load in the program you have just saved.

It just freezes, and will stay like that forever, if you let it, looking for the last/next file on the list. Which as we know, does not exist. And so, a way to prevent this from ever happening again, is needed! (Question is, How?)

One way is to put a number at the top of the file on saving. At the number T is currently at. So, that if this does not match the last number in the FOR/NEXT loop in our Load-in subroutine. We can have the program churn up a notice to the user."FILE MISMATCH ERROR" or words to the effect. This works well for private use, but what if there is a cause to use a different program that has diferent variables? Some will have already noticed that the variable FOR... loop used in the load routine is different to that of the save routine to store what again changes to T$(T)., T$(U) or T$(Q) etc.

Add the following:

```
8260 INPUT#1,TB
8270 IF TB<>T THEN 11000

9525 PRINT#1,T
```

And while we're at it. Here's the code for the programs response to these errors:

```
LIST 11000-11900

11000 PRINT"‘   FILE CANNOT BE
FOUND!"
11100 STOP
```

Test the program as before. You may need to delete the saved 'TABLE' file. (each time) for these experiments to work. And if things are going to plan, we should have been sent to the one size fits all encatchment area of the program. Notice with the use of both '<' and '>' in line 8270. that now even smaller files won't be accepted. This is easy to change should you wish to.

This however is cumbersome, as we're limited to what the program says T should be. So either we have to make provision for length of file as well as it's name. In which case, before long it will become necessary to have a proper file menu. With names and sizes of everything on disk/folder! (not covered here)

Well, I've got this far. But in the meantime; there has been a minor issue that's been persistently on my mind, that I feel could be done better. It's simple to correct so I haven't been in a rush. But now my head needs a rest from the tough/er stuff. So, lets take this opportunity to get it sorted out.

Follow the 'rollercoaster'!:

The scribble of Instructions looks scruffy at the bottom of the table screen. It's time to clear this up and give it it's own screen. Here we go:

```
LIST620-670
```

go over the numbers with the cursor to make 620 into 3030, so that by the time you've gone down the list 630 is 3040,, and 670 becomes 3080.

At a free line type;

```
DELETE 620-670

3000 REM *INFORMATION SCREEN*
3010 SCNCLR
3020 ?" 'qq'   'I'NFORMATION SCREEN!"
```

We can make this a proper menu. You might wish to give each instruction its own line. Thus may end up with something like this:

```
3000 REM *INFORMATION SCREEN*
3010 SCNCLR
3020  PRINT"‘    'I'NFORMATION
SCREEN!"
3030 PRINT"‘ TABLE KEYS: "
3035 PRINT"‘ 0/Q = EXIT "
3037 PRINT"‘ 1/L = LOAD TABLE,"
3040 PRINT" 2/S = SAVE TABLE, "
3045 PRINT"‘ 3/E = EDIT "
3047 PRINT" 4/X = SWAP"
3050 PRINT"‘ BAR:  "
3055 PRINT"  <SPACE> = EDIT "
3057 PRINT"  <RETURN> = SWAP"
3060 PRINT"   -USE <UP> / <DOWN>
ARROWS.
3070 PRINT
3080 PRINT" 5 = ENLARGE "
3085 PRINT" 6 = REDUCE COUNT "
3087 PRINT" 7 = EDIT FILE"
3090 PRINT
3100 PRINT" PRESS A KEY "
3110 GETKEYA$
3120 RETURN
3200 STOP
```

change line to:

```
11100  GOSUB3020 :GOTO400
```

add:

```
11010 PRINT" PRESS A KEY"
11020 GETKEY A$
```

and:

```
840  IF  A$="I"  THEN  GOSUB  3000
:GOTO 400      'chaos here. without the
Goto400
```

The Program should now work smoothly enough. Though, with the SELECTion BAR, the screen will flash in order to change the selection. This is simply because it was easier to clear the screen and re-write the list. Instead of spacing out and re-writing the appropriate selections. (though curiosity might make me look at this later. I'm tempted to keep this one to myself, so that if you do choose to build it yourself, you may see the 'fun' I had with the original BAR. Many corrections were involved! (A clue: The hard bit is having the computer remember the lines that need to be spaced out. Possibly easier in GRAPHICs mode, with the CHAR, RDOT(N). Pages 121, 110 & 146 in The C=16 User Manual respectively. -And covered in these volumes somewhere. But if that's too much upheaval. Try looking up the POKEs and PEEKs for the screen memory map! Either way would have been a more standard approach.

It was interesting to make The Selection BAR compatible for JoyStick use! First notice that the joystick ports use back up keys. (Not even with a {SHIFT} or {C=} pressed, nor are they obscure, and don't even make much directional sense, all in all.) lines 770 & 780 will therefore conflict with JoyPort #1.(D65R, left, right,up, down. T to fire!)

The {RETURN} key Locks out the other features (when in 'green' /selection mode. There's a couple of ways around this. 1 is to add a new subroutine to test if the keypress is a valid one. And if it is use the selection as the number to edit/change/swap or delete. The second is to merely cancel the selection bar and return focus to the elite table whilst carrying out the latest command.

Elsewhere, It's still a bit abrupt to say "File cannot be found", and I don't think an appology is quite the way to go. I want whatever can be brought, be it 5 records in a file or 500. To take whatever is there without crashing or error reporting. -Or at least to know what is there, and the work required beforehand!

One simple way used is to change:

```
8270 IF TB<>T THEN T=TB :?"FILE
MIS-MATCH"
```

Be Aware of the problem that the computer has already made a provision in memory for such a table with these variables to be stored.

```
xx T=10
xxT1=40
```

But that it was ordered to provide a maximum of 10 in the list. No less, and certainly No more than 10. So it logically assumes that something is wrong and declares it. Bear in mind it doesn't ask questions it isn't programmed to ask. In this case. "Do I add to this. Your highness?" -So, we need to code this in -particularly the "Your highness" bit!

```
...
code
...
xx Open to load-in = T   ;{FT=T} -{with
AV as name of file}
...
xx OPEN to save to disk = T1   :{FT=T1}
-{drop AV for a different name}
...
```

Another thing noticed 'playing around' while building the program, Is that it's too easy to over-write a file. There is no provision to check with the user here yet. Although, beyond the scope of this article, as it's getting quite long. It's a worthwhile feature. Have a subroutine check for the file.

I would welcome reader solutions, on counting the number of records on a file by OPENing it to have a look at the length)

I can confirm that I've tried everything I can think of (just about!) and COPY, SCRATCH etc. does not work with Yape even on standard .PRG, (which is most likely a good thing!) but leads me to wonder what else isn't possible on Yape that perhaps is with the real machine. (This is absolutely no detriment to YAPE -which I absolutely Love! And am ever endebted to for getting me back into programming. More so, as for the life in me, I could not get VICE working!)

So, though I do think it's possible somehow, and in my experiments got near, I am admitting defeat here. -for the time being! Address solutions to me as I want to be able to check that some-one hasn't misconstrued what I'm saying and send in something for something perhaps near but different.

```
?BAD SUBSCRIPT ERROR
READY.
```

The ReDimensioning bit (or as near as).

The only solution I have thus far, could be done as a subroutine. But it will have wider ranging use and versatility; easier to modify for other programs etc. as a stand-alone program.

I aim to have the user decide (within reasonable memory capacity: I need to do tests on this. For 16k., 64k, and 128k) to know what the maximum size file is.

The user will have to tinker with the program itself to match the variables of their import that they wish to save into the bigger list, then add to it. The program could go one better by asking the user where they wish to start. (i.e. may have two to join etc.)

Then the program will go about making the new, bigger file, putting the old one/s to it where requested and saving it.

DRAWBACK, as I said before, with Yape (my version anyway) commands like SCRATCH etc. aren't possible. Therefore I will have to leave it to the user to rename files according to their use. That is to say the name "TABLE1" will have to be changed by DOS on the PC. Once the old, smaller file has been renamed (e.g. TABLEX1).

?TYPE MISMATCH ERROR
READY.

```
5000 REM *RE-DIM ARRAY*
5010 SCNCLR
5020 PRINT"     *RE-DIM ARRAY*"
5025 REM -----------
5035 REM -----------
5110 TT$="TABLE1"
5120 SZ=1200
5200 REM *DETAIL*
5210 INPUT"‘ SIZE OF FILE ";SZ
5220 PRINT" SIZE: ";SZ ;"  SURE Y/N ?"
5230 GETKEY A$
5240 IF A$<>"Y" THEN 5210
5290 DIM F$(SZ)
5500 OPEN5,8,2,TT$+",S,W"
5510 PRINT" CREATING AND SAVING EMPTY FILE"
5520 FORJ=0 TO SZ
5530 F$(SZ)=" "
5540 PRINT#5,F$(SZ)
5680 NEXT J
5690 CLOSE5
5900 REM *ADD FOR CHECKING: ERRORS, EXISTS ETC.*
```

READY.

Note: It may also be necessary to use a variable and allow the user to decide the channel. In the meantime I have used 5.

?SYNTAX ERROR
READY.

Epilogue: Promoting good code

In the end, there was a way to put both "From x TO y?", and The Selection Bar in. But the work required was still double.

Deleting the early code paved the way to allow for the extra variables required for the re-thought out program.

At time of being sent of for editorial approval. I only have one Joystick. And can't remember from some 30 years ago what the backup keys for JoyPort#2 are.

I never got around to preventing the screen flash when re-sorting the list. Doing so may require a complete overhaul to the way the list is printed. Even thinking about it feels like such a CHAR -or chore!

Despite that lack of PEEK inspiration. I feel the objectives that this article set out with are covered satisfactorily. And that the novice user will gain much from the explorative way the piece is written.

The list itself can be added to/ To about 20 before it requires clever commands to halt the list, and account for multiple pages etc. Remember to account for this with the appropriate DIM statement on the provided line.

You may have noticed the table can be used for numerous things. And as Christmas approaches, you may be considering your new years resolutions.

The next Happy Programming article covers some obscure commands not yet covered in the Happy Prog'ing slot. And as the early idea was to cover all commands in The c16 User Manual it is time to deal with the last few. So colourful is one of the demo's and seems an appropriate title. Because it looks to me like these are the last to be covered, and so to cover any more in the 264 series of computers would require moving up to see the c128.

<u>Here's the standard noticees, -and begging bit.</u>

Happy Programming is for every-one. If You've enjoyed or benefitted from any of the articles in Commodorefree magazine. Please donate to the magazine as websites cost money to keep going.

The opinions expressed here are not necessarily that of Commodorefree, it's representatives, staff, readers etc.

end note: Programmed on Yape software (emulator for pc) and written in Word. Whilst everything is thoroughly checked and deemed correct at time of printing. No computer problem of any hardware kind should arise due to use/copying of programs. No liability by commodorefree, or it's representatives, staff etc., including John Fielden may be assumed. In the unlikely event that anything does occur it is most likely to be co-incidence. Often old machines break down simply because they are old. If using a pc consider the operating system, (which it usually is!) -- In any case: I am skint!

Now, at last I can bid you Good day, and wish You ALL -

HAPPY PROGRAMMING!
Most Kind Regards,
John Fielden

INTERVIEW WITH Commodore 64 Brain Steve Morrow

https://www.youtube.com/user/SeetheTruth4Yourself

Visit my forum: http://www.programmermind.com

Sometimes a picture is worth a thousand words. Some problems you have been stuck on can seem simple after watching a short video, so when I found this YouTube channel with videos for Atari Basic/Assembly, Commodore 64 Basic/Assembly, Dark GDK (c++), Atari/ with Commodore games, various Tutorials, and Grand Theft Auto IV Hacking I thought that some other readers may find content useful.

COMMODORE FREE
Please introduce yourself to our readers.

STEVE MORROW
Hello, everyone. My name is Steve Morrow. I was born in Ohio and have been living in Phoenix, Arizona since 1998. My passion is programming in multiple languages from Basic, assembly language, C#, C++, Java, and so on.

CF
What started your interest in Computers?

SM
My interest in computers began back in 1984 when I first discovered an Apple IIe computer at school. I knew nothing of computers at that time. Much later however in high school, I purchased an Atari 65xe in 1989, learned programming, and about a year later finally owned a Commodore 64.

CF
What is your day job?

SM
I currently work for PayPal in Customer Service.

CF
OH! Ok I guess that needs no explanation, although to be fair, PayPal have always been helpful to Commodore Free. Do you have a favourite machine, and if so, why?

SM
I cannot find a better machine than the Commodore 64. I am getting one repaired as this interview is taking place. The Commodore 64 has changed my life for the better. I have learned to reconnect with my roots and spent many hours decompiling printouts while working through solutions.

CF
How long has the YouTube channel been running, and how many visitors on average does each video receive?

SM
I started the channel back in 2009 and currently have 1,620 subscribers. I don't know the average amount of people that join as of yet, but I have faithful followers I communicate with constantly through the channel, email, and Skype.

CF
So what prompted you to start the channel, and what equipment do you use?

SM
The vision for this channel began when I graduated from college and wanted to post my experiences live for recruiters to see (in hopes of getting into the Information Technology industry). I had no idea I was going to start a channel since YouTube was quite new then. Eventually I landed a position as a Web Developer. It turned out later that I used my YouTube as a tutorial guide and built a website around my experiences with programming on a larger scale.

I currently use Camtasia Studio for my editing, Audioblocks for the music, VICE C64 for the live demonstrations, HP Webcam Control Center for the live video shots, a Logitech HD 1080p desktop camera, Excel, Powerpoint, and sometimes just my regular hardware for examples.

CF
What feedback have you had from the various communities and conversely like Commodore Free, what criticisms and negative feedback have you had?

SM
In the beginning I obtained great advice from a disk magazine publishing company called LoadStar (formerly SoftDisk magazine) . They taught me to write programs that people can comprehend and easily disassemble. Lately I have had feedback through websites like Lemon64 regarding my machine language tutorials. Through this "instructive criticism" I have learned to hone in on my strengths and build from there. I have often found people to be my greatest resources. I have finally come to realize that I still have a lot to learn and grow from.

CF
Can you expand on some of the content our readers should be most interested in with the Atari and Commodore BASIC and machine code, and what are you hoping to cover with these?

SM
The Atari home computer still has a special place in my heart. I owe credit to learning both BASIC and machine language on that system. I spent hours in libraries and typing until my hands cramped.
However the biggest journey has been with the Commodore 64. I learned to respect this machine and overcame learning curves to finally have a submission published by LoadStar magazine in 1995. It was a graphic/text adventure called Quest for the Mad Bomber.

CF
I notice that for the C64 you're partway through programming a game. What level of watcher would be able to follow this and what speed do you think you will be introducing the vide-

os? Presumably, you intend to slowly build up on the last, as lots of readers complain that learning from books they understand the first few chapters, then it seems to skip off and find it impossible to follow.

SM
Well currently I was working with a team developing a Breakout clone, but I ran into a lot of restrictions, change of career, etc that derailed me for a short time. I tend to focus my YouTube videos on anywhere from the "beginner" to the "intermediate" viewer. For the machine language tutorials, it is recommended to have a solid background in BASIC since they tend to be more complex (since the logic is speaking to the 6510 processor itself). My goals are to start simple, such as I am doing with the Commodore 64 Very Basic tutorials, break down the logic, and add a little progression to keep the pace of interest.

I think Books can be complicated to learn from since everyone has a different experience and story. Although I did consume a lot of books in the past, the most important thing I learned to do was reverse engineering code, and investing more time in the Commodore 64. The best advice I can give is to experiment, fail, and try again until you succeed.

CF
You also said you teach BASIC programming. Do you feel it's important to start with BASIC and then move to machine code, or would you advise just moving to machine code and skipping BASIC for the would-be programmer? In Commodore Free this is a common question I get asked from people, although there are two answers and enough on both sides. I usually recommend getting some grips of BASIC before moving on as the code you learn will not be lost. Loops are common as are memory locations, etc.

SM
It can depend on your experience. For any beginner, I am highly recommending (from experience) to start with BASIC. Although I have many people requesting "machine language tutorials" that don't enjoy BASIC, I encourage learning a higher level language since it all builds upon as you say "the fundamentals" of grasping higher concepts when learning to write in assembly language. As I've learned the machine is like a robot, teach it to do small tasks, and then build upon them after you are satisfied with the results. For the advanced user, I am definitely recommending to stick with machine language tutorials and do some reverse engineering. As long as you have the "basics" under control, machine language is only a step away. Also, do a lot of "Googling". As my college professors used to say, "Google is your best friend."

CF
What's planned next for the channel?

SM
I've been promoting an assembly language project in production. The mission is to organize a team of devoted programmers, artists, musicians, and story writers to create the next greatest simulation or game. I am currently recruiting through Lemon 64, Twitter, through my channel, and other websites. The goal is to keep a structured approach to programming using Powerpoint as a guide and Skype to communicate with the team.

CF
You also plug your forum here http://www.programmermind.com
Can you explain some of the content on this site... does it mirror the YouTube videos, for example?

SM
Yes. I am doing my best to have it mirror my YouTube channel. Like any website it can take a lot of work to maintain. However I am confident that I can enhance it as I continue to listen to my users and accept change.

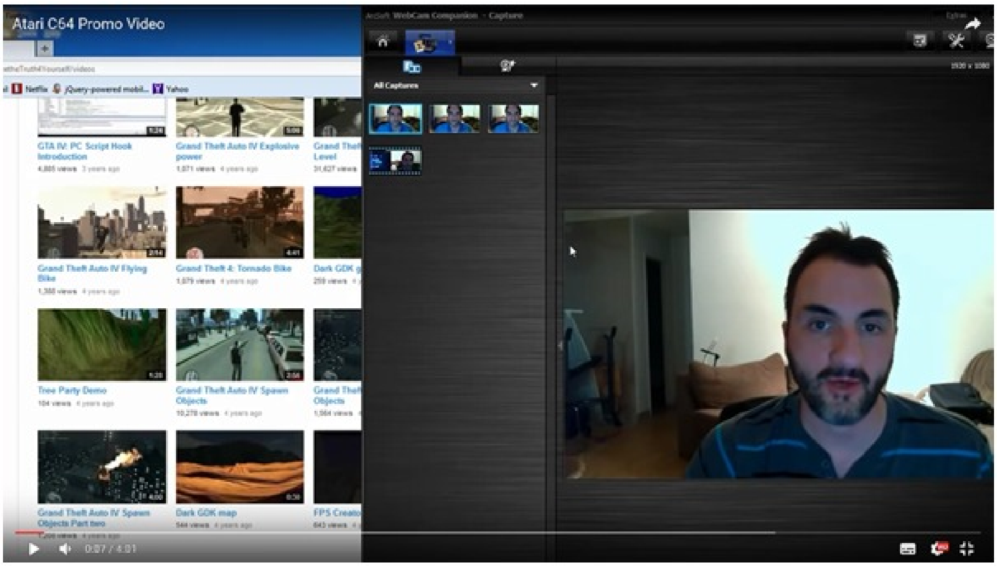
CF
Do you have any comments to end with, or maybe a question I didn't ask that you feel is important?

SM
I would like to take the time to thank the readers for peering into my small universe. It is my hope that you can keep your passion, whether it is with programming, games, projects, and keep a positive attitude while moving forward.

I also mentioned that I am having my Commodore 64 repaired and my vision is to start uploading programs using the 1541 Ultimate II tool on my Commodore 64 to my desktop via an SD card that will operate in VICE C64. These programs are buried on floppy disks that I cannot wait to share with the world.

Finally I present a closing question. Where can I connect with more people who have a soft spot for the Commodore 64? It has been a journey and I look forward to the future as this machine is brought back into the forefront. It has been a pleasure sharing with you all today and I am grateful to have found Commodore Free. Thank you for your time.

THE Commodore basic tutorial can be followed here (currently in 8 parts)
https://www.youtube.com/watch?v=FSRgKy82VkI



SMILA Interview
By Damian Caynes/Slartibartfast

Damian Caynes
Who were your main inspirations when you started out as a c64 graphic artist?

SMILA
I worked on my first gfx in 1986 so there weren't really any "famous" game artists around at the time.
The arcades inspired me more than any artist did.

DC
What games inspired you to get into game development?

S
Manic Miner / the Ultima games as well as arcade shooters like Terracresta, I loved the amazing designs on the alien ships and the amazing cabinets.

DC
You have a long history with the c64, working for Acclaim and doing titles for Ocean and US Gold. Would we know any of the games you worked on?

S
I've worked on well over 100 games but I guess the biggest games I've worked on are - Forsaken, NBA Jam, Shadowman, Alien Breed, Worms.

DC
What was your favourite game that you have released?

S
On the c64 it's probably Soulless as it was a return to doing 8-bit games after a few decades, working with heavy restrictions is a real blast. Other platforms I would probably say Shadowman, as we had a great team and we really did have a lot of fun creating it.

DC
Which game was the most fun to design?

S
I think I would say MY LIFE, I had a clear idea of how I wanted it to turn out and got the design done really quickly, achim (the programer) and myself managed to get the game done really quickly.

DC
You design games as well as do the graphics, but you're not a coder, surely you must think like a coder to design c64 games?

S
I've been in the games industry for 30 years so I guess all parts of development have just rubbed off on me.

DC
You have a distinctive graphical style, how did this evolve?

S
Years of working in the industry I guess, I've worked with some amazingly talented people over the years and you always learn little tips and tricks to help you improve.

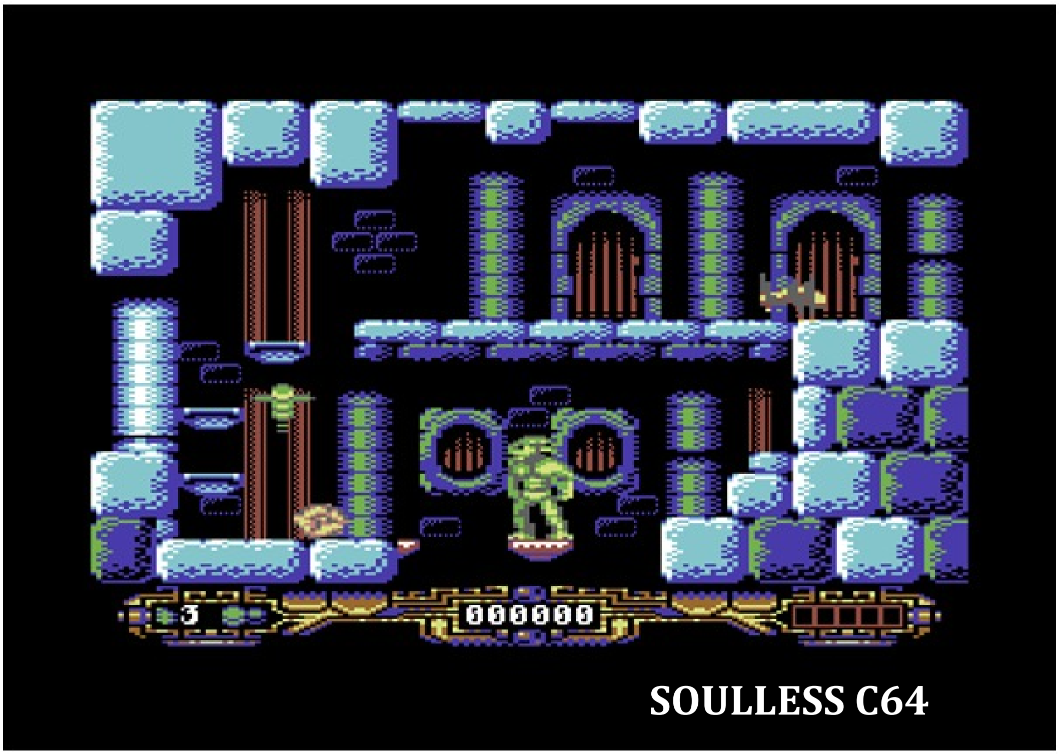
DC
You're now working on a sequel to Soulless. Can you tell us what the main features are in the sequel?

S
With the first soulless I based it on two of my favourite c64 games, Draconus and Impossible Mission, this time I've looked at another one of my faves "Sacred Armour of Antiriad". It's a multi-load over 4 different looking levels with the ability to upgrade your weapon and find a suit which allows you to fly. It will also have a few boss battles.

DC
You also do cover artwork for some games. Do you do any graphic design work outside of the c64?

S
My main job these days is a concept artist on TV and film, I also create album and book covers, I still keep my finger in games a little though, recently doing work for team17.

DC
Excellent, we certainly can't get enough of your original game concepts, and can't wait for the next one! Thanks for taking the time to talk to us!



SOULLESS C64

Interview with Slartibardfast
By Joonas Lindberg

I recently interviewed C64 scener and long time friend of mine, Damian Caynes, also known as Slartibartfast in the scene. He has been a member of various groups and has made some really impressive looking demos, and he is also planning to make some games for the C64.

I talked with him to find out about his history and to discuss his C64 plans for the future. Read the interview below

http://csdb.dk/scener/?id=3910

Joonas Lindberg
When did you first use a C64? What is your first memory?

Damian Caynes,
My mother bought my first c64 when she scored $10k for a damaged wrist, we took it home with a datasette and our first games were Brian Bloodaxe and Bombjack. We waited an hour for a game to load initially cos we heard tapes were so slow! But soon we also got a 1541 and an Action Replay V cart, and I never looked back

JL

So what games inspired you to get into design and development?

DC

So many different games, but mainly the games of Andrew Braybrook, Jeff Minter and anything I could get my dirty mitts on through the Action Replay V ML monitor

JL

Which coders inspired you to become a coder?

DC

Well as I said, the work of Andrew Braybrook and Jeff Minter, as they had dev diaries in ZZap!64 that revealed the coding techniques behind their games. But also the work of Martin Walker, Jon Wells, The Rowland Brothers, and Antony Crowther.

JL

Who are your favourite SID musicians?

DC

Oh it's really hard to answer that question, as there's so many talented scene musicians. But, in short, I'd say the work of Rob Hubbard, Matt Gray, Steve Rowlands, Chris Huelsbeck, Jeoren Tel, Ben Daglish, and so many more that I can't remember at the moment.

JL

What was your first piece of code?

DC

Well, my first piece of code was probably just incrementing the border colour in a loop, but the first piece of raster code I wrote was a group of sinus colour bars animating on screen, I was really proud of it and it was the start of years of intro and demo coding.

JL

Have you ever coded an intro or a demo?

DC

Well actually, the only things I released in the scene were demos, but in classic Australian style they were little more than really stylish linked intros, but I did get a lot of praise for my parts.

JL

As a member of the scene, have you ever done a crack?

DC

No, even though I was in a cracking group in the 90's scene, I didn't release any cracks or even put an intro on a crack. But this week I joined a two man cracking crew with my old mate Fungus, and PROS will rule as a modern cracking crew.

JL

Have you ever had a game commercially published?

DC

Well, that's actually a funny and slightly embarrassing story. When I was 17 I was engrossed with making SEUCK games, this was a year before I started coding ML stuff. My brother and I no-

'? METHINKS IT'S LURVELY!!!! |

UTURE 1S1ON ES1GNS

CODED BY SLART1 FOR THE 1KV DEMO COMP!!!

ARE GOIN' DOWN HERE, SO READ 'EM TO BE

ticed that a local c64 game publishing company (ECP) was releasing an awful lot of SEUCK games, and on closer examination we discovered that most of them were made by the owner of the company. So we polished of our quite average shooter, The Final Attack, and submitted it to the company. A week later the owner got in touch and said he was publishing the game, and offered an advance on sales. Needless to say we were very surprised at this, but gladly took the money and bought an Amiga 500 with RAM expansion and paid for a $1000 phone bill. Our parents informed the local newspaper and TV station, who praised us as "young entrepreneurs" and "computer whiz kids". It was very embarrassing for a kid in Year 11 of high school, and I got no credit for it because the cool kids knew it was made in SEUCK. About two months later ECP got back to us and said there had hardly been any sales of the game, and they wanted their money back! Well, I was only 17, with no source of income, and they couldn't sue me because I was too young. So I ignored their pleas, and about six months later the company went bust and was bought out by Electronic Arts.

JL

Have you ever coded a game for the C64?

DC

I have actually coded several games, but have never completed and released a project. Back in the 90's I made a simple light bikes game and Tetris clone, and submitted them to Beam International in Melbourne, to show that I could code 6510 and would be able to handle 65816 on the SNES. Unfortunately they didn't even know what to do with the disks, as the c64 was on the out and out at the time. More recently, in 2005, I half finished an awesome PETSCII game, TRONSCII, a 2 player multi scrolling playfield TRON clone, but I got stuck and that was like almost 10 years ago It was really cool tho, two multidirectional scrolling grids, colour cycling light bike trails... I couldn't finish the project and it lay dormant for a decade, but recently I decided to submit it to Games That Weren't, and it will be featured in the December update.

Jl

What game projects are you planning at the moment?

DC

Well I'm planning three different game projects at the moment, of varying complexity. Firstly there's Magic Bag, a Match-3 RPG game, where you match magic gems to improve your stats, cast spells, or attack. Then I'm planning to make a multidirectional shooter using the SEUCK Redux engine. Finally my long term project will be a hybrid of Habitat and the SCUMM engine, developed to show off a friend's c64 modem, called The City. As well as these projects, I'll be working on simple arcade conversions with my group mate Fungus.

JL

It's been a great pleasure interviewing you. Thank you for your information about your upcoming projects and thanks for taking your time with this interview.

DC

Thanks! I just have to let the readers to know to follow my C64 Code Hacking blog, and continue to read the tutorial series in this magazine, and remember, code is life!

http://c64.inspired-digital.net/

C64 Code Hacking
By Damian Caynes/Slartibartfast

Hey, I'm Slartibartfast, formerly of Aussie demo crews Futurevision Designs, Battery and Lithium(old school) and Onslaught, Chrome and Hokuto Force (contemporary), recently having joined PROS to code intros for the cracks my man Fungus does.

I've been coding for the c64 almost since I first got one in the 90's, took a break from the "death" of the scene in the mid-90's until returning in the early 2000's to join Onslaught and Chrome in around 2005. I released several short demos in that time, started work on a game, and burnt out around 2007 and left the scene again. That brings us to today, where I decided if I was going to make a game now, I'd have to polish up my skills. This started as an idea to rejoin Hokuto Force, but then the CSDB intro competition came along and I met Fungus again for the first time in years, and we formed a cracking crew, PROS.

I figured I could disassemble my old demo parts, which were little more than linked intros, and quickly teach myself to demo again. And with the help of VICE monitor and Regenerator Disassembler, it worked!

So follow on and read all about the process in creating my first intro in a decade, along with source code snippets and eventually the source to the entire intro.

First Steps... Setting up a development environment!

The most important tools in a modern c64 coders arsenal are their IDE (Integrated Development Environment, a code editor) and their choice of cross assemblers.

While there are many IDEs and cross assemblers available with varying levels of complexity and feature sets, we're going to keep it simple by using the

Popelganda Relaunch64 IDE
http://www.popelganda.de/relaunch64.html

and all sources will be compatible with the
TASS64 https://sourceforge.net/projects/tass64/

(or 64TASS) cross assembler, which is a popular choice as it is also backwards compatible with Turbo Assembler sources, which was the coder's choice of assemblers on the c64 in the good old days.

Simply download Relaunch64 for your platform of choice, we will be using Windows, and download TASS64 and extract it to a folder on your hard drive, preferably inside the Relaunch64 folder you just created.

Now, unlike other IDEs that require a complicated make file to compile your source, setting up your cross compiler in Relaunch64 couldn't be easier. Simply go to Options/Preferences, click on New Script, and add a title for your script, in this case "TASS64". Now open your TASS64 folder, and simply drag the executable file into the Relaunch64 script window. Relaunch64 will automatically create a command line for your cross assembler. Now just click save script, and your development environment is now completely setup!

The basics of c64 ASM coding

I won't go into the absolute basics of c64 ASM coding in this article; the topic is too diverse and has already been covered in other articles on the wide web, but before we get into the code for our first raster routine I will give you some links to some excellent tutorials available on the web.

Commodore 64 Programming #1: A quick start guide to C-64 assembly programming on Windows
https://digitalerr0r.wordpress.com/2011/03/19/commodore-64-programming-a-quick-start-guide-to-c-64-assembly-programming-on-windows/

C64 Assembler Tutorial
http://www.c64.ch/programming/

The Hex Files, learn demo coding!
http://www.oldschool-gaming.com/articles_index.php?view=c64_hex_files

64bites, a great tutorial series if you're willing to pay for it
http://64bites.com/

Fairlight programming docs
http://www.fairlight.to/docs/

Introduction to c64 demo coding
https://odd.blog/2008/01/11/introduction-to-c64-demo-coding/

Richard Bayliss' Assemble It!
http://tnd64.unikat.sk/assemble_it.html

All You Need to Know About Your C64
http://codebase64.org/doku.php

Codebase64, learn demo coding tricks!
http://codebase64.org/doku.php

These links will teach you the essentials of c64 code hacking, of special interest is All You Need to Know About Your C64, and Richard Bayliss' excellent Assemble It! series. Note that the sources you find on the web may be for other assemblers, but it shouldn't be too hard to convert between different cross assembler sources once you know what you're doing.

Your First Raster Interrupt Routine

The first thing I would need is a basic raster routine, for playing music and creating colour splits (my PETSCII logo uses a black background colour surrounded by reversed blue characters, so there needs to be a split to change the background colour to blue to match).

I looked at my early demos, and they depended on Kernal registers and routines, and used the stack to store the registers between interrupts. This would not do for a modern intro, so instead I looked to **PETSCII Ate My TinySID** http://csdb.dk/release/?id=18202 as it also had the PETSCII proportional font scroller I needed.

Here's the source of the basic routine, with one raster interrupt, playing music. TASS64 format is used.

```
* = $080D
; Clear the screen
LDX #$00
– LDA #$20
STA $0400,x
STA $0500,x
STA $0600,x
STA $0700,x
LDA #$01
STA $D900,x
STA $DA00,x
STA $DB00,x
STA $DAE8,x
INX
BNE –
SEI
LDA $DC0D
LDA $DD0D
JSR $1000 ; init SID tune
LDA #$7F
STA $DC0D ;CIA1: CIA Interrupt Control Register
STA $DD0D ;CIA2: CIA Interrupt Control Register
LDA #$1B
STA $D011 ;VIC Control Register 1
LDA #$01
STA $D01A ;VIC Interrupt Mask Register (IMR)
LDA #$06
STA $D020 ;Border Color
LDA #$00
STA $D021 ;Background Color 0
LDA #$00
STA $D012 ;Raster Position
LDA #<start ; set the raster interrupt to the start label location
STA $FFFE ;IRQ
LDA #>start
STA $FFFF ;IRQ
LDX #<nmi
LDY #>nmi
STX $FFFA ;NMI
STY $FFFB ;NMI
LDA #$35
LDX #$FF
STA $01
TXS
INC $D019 ;VIC Interrupt Request Register (IRR)
LDA $DC0D ;CIA1: CIA Interrupt Control Register
LDA $DD0D ;CIA2: CIA Interrupt Control Register
CLI
loop JMP loop
start STA nexta+1
STX nextx+1
STY nexty+1
LDA #$06
STA $D020
STA $D021
JSR $1003
LDA #$1B
STA $D011 ;VIC Control Register 1
LDX #<start ; send the interrupt back to the start label location
LDY #>start
LDA #$00
STX $FFFE ;IRQ
STY $FFFF ;IRQ
STA $D012 ;Raster Position
INC $D019 ;VIC Interrupt Request Register (IRR)
nexta LDA #$00
nextx LDX #$00
nexty LDY #$00
nmi RTI
*=$1000
.binary “music.prg”,2
```

You'll notice it uses what is known as self-modifying code, instead of using the stack to push and pull the registers between interrupts, this saves raster cycles over using the stack for when you're doing cycle intensive demo work. So, for instance, A is stored in nexta+1 which will load the value after the interrupt is finished. This is necessary so your interrupts don't mess up the normal goings on of the c64.

Rasters: Your First Colour Split

This routine is largely the same as our bare bones routine, except an extra interrupt routine has been added. $FFFE and $FFFF are set to the location of the next label, and $d012 is set to $32(the first line of the screen) + (12*8) (12 lines). Then after that routine has been executed $FFFE and $FFFF are reset to the location of the start label, easy as that!

```
 *=$080d
 LDX #$00
- LDA #$20
 STA $0400,x
 STA $0500,x
 STA $0600,x
 STA $0700,x
 LDA #$01
 STA $D900,x
 STA $DA00,x
 STA $DB00,x
 STA $DBE0,x
 INX
 BNE -
 SEI
 LDA $DC0D
 LDA $DD0D
 JSR $1000
 LDA #$7F
 STA $DC0D ;CIA1: CIA Interrupt Control Register
 STA $DD0D ;CIA2: CIA Interrupt Control Register
 LDA #$1B
 STA $D011 ;VIC Control Register 1
 LDA #$01
 STA $D01A ;VIC Interrupt Mask Register (IMR)
 LDA #$06
 STA $D020 ;Border Color
 LDA #$00
 STA $D021 ;Background Color 0
 LDA #$00
 STA $D012 ;Raster Position
 LDA #&lt;start
 STA $FFFE ;IRQ
 LDA #&gt;start
 STA $FFFF ;IRQ
 LDX #&lt;nmi
 LDY #&gt;nmi
 STX $FFFA ;NMI
 STY $FFFB ;NMI
 LDA #$35
 LDX #$FF
 STA $01
TXS
 INC $D019 ;VIC Interrupt Request Register (IRR)
 LDA $DC0D ;CIA1: CIA Interrupt Control Register
 LDA $DD0D ;CIA2: CIA Interrupt Control Register
 CLI
- LDA #$00
 BEQ -
 JMP -

 start STA nexta+1
 STX nextx+1
 STY nexty+1
 LDA #$06
 STA $D020
 STA $D021
 JSR $1003
 LDA #$1B
 STA $D011 ;VIC Control Register 1
 LDX #&lt;next
 LDY #&gt;next
 LDA #$32+(12*8)
 STX $FFFE ;IRQ
 STY $FFFF ;IRQ
 STA $D012 ;Raster Position
 INC $D019 ;VIC Interrupt Request Register (IRR)
 nexta LDA #$00
 nextx LDX #$00
 nexty LDY #$00
 nmi RTI
 next STA nexta1+1
 STX nextx1+1
 STY nexty1+1
 LDX #$00
- INX
 CPX #$3D
 BNE -
 LDA #$00
 STA $D021
 STA $D020
 LDX #&lt;start
 LDY #&gt;start
 LDA #$00
 STX $FFFE
 STY $FFFF
 STA $D012
 INC $D019
 nexta1 LDA #$00
 nextx1 LDX #$00
 nexty1 LDY #$00
 RTI
 *=$1000
.binary "music.prg",2
```

Now as a challenge, consider the other VIC registers you could split, such as $d016 to have a scrolling area of the screen. We'll cover basic scrollies in the next article, so take some time to absorb this article and stay tuned for the next instalment!

An Interview with Martin Piper of the SEUCK Redux Project

By Damian Caynes/Slartibartfast

As you may know from personal experience SEUCK (The Shoot-em-up Construction Kit) is an awesome tool for game construction. In fact, some people use it's graphic and sfx editors for other game projects. But unfortunately it can be quite limiting and many games for the platform can suffer from being quite similar.

Enter SEUCK Redux,

http://codebase64.org/doku.php?id=base:scrolling_and_seuck_file_player

written by Martin Piper in 2008; an entirely new SEUCK engine, written from scratch, with

full source code,
https://github.com/martinpiper/C64Public

and allowing you to use a re-mastered version of the SEUCK engine with new scroll code and sprite multiplexor, or alternatively the new Sideways SEUCK engine. Several games have already been released with enhanced features of SEUCK Redux, including

Super Tau Zeta 2 Redux,
http://tnd64.unikat.sk/friends/Super_Tau_Zeta_2_redux.zip

The Adventures of Alain,
http://tnd64.unikat.sk/friends/Adventures_of_Alain.zip

Another Day, Another Zombie, Final
http://tnd64.unikat.sk/friends/Another_Day_Another_Zombie_FINAL.zip

and the technically brilliant

Nuclear Strike Force
http://tnd64.unikat.sk/games/Nuclear_Strike_Force.zip

All with new title screens and enhancements such as power-ups.

The engine can also be used to create other forms of c64 games; as it has a multi-directionally scrolling engine, a powerful new sprite multiplexor, as well as the SEUCK routines for attack waves and sound effects.

I chatted with Martin recently about the SEUCK Redux project, and about his C64 coding career, read the interview below.

When did you first use a c64? What is your first memory?

Martin Piper
I think my first C64 was around 1987 (so 14 years old) when I upgraded from a VIC20. I think the first game I bought with the computer was Wizball. I remember the John Lewis store in Bath (UK) selling a bundle at the time.

So what games inspired you to get into design and development?

MP
Coin-op arcade games were a great inspiration, like After Burner, because their hardware was way beyond anything a mere home computer had. They gave me something to try to emulate on the trusty old C64 and other previous machines. These ambitious targets

kept on making me strive to improve my coding.

Which coders inspired you to become a coder?

MP

I suppose the coders that inspired me back in the early 1980's were those who wrote type-in listings printed in magazines and the numerous series of short books with listings of simple BASIC games.

But when I got the C64 the world of great games using machine code was just beginning to open up before me. Andrew Braybrook's diary of a game for Morpheus really started me off with wanting to learn much more about machine code. Then Martin Walker's Citadel diary really helped with honing my knowledge about design and attention to detail.

When did you start programming and on what platform?

MP

I actually started coding back in 1981 (8 years old) when my dad bought a ZX81 and quickly gave up trying to use it. So it quickly ended up in my bedroom where I could use my childhood perseverance and focus to learn BASIC. I learnt a little bit about machine code back then, but not enough to be that useful. Then came the VIC20, which I think had a busted VIC chip (or memory timing) as the graphics were very glitchy. Then the C64.

What was your first piece of code on the c64?

MP

Some BASIC demos, where I used compiled BASIC to try to get some speed, also some BASIC text adventures. My first real program which is when I had that real Eureka! moment was when I got a 6502 assembly reference guide and using pen and paper wrote my first bit assembly. I used the opcode lookup table in the book to convert to hex, which I then entered as BASIC pokes and started it with a SYS statement. The code was something like:

```
 sei
        inc $d020
        inc $d021
        jmp $c000
```

I wanted to see how much faster this machine code was compared to doing something similar in BASIC. I nearly fell off my chair with surprise when I saw how tiny the narrow strips of colour were compared to BASIC. I then twigged that using this style of code I could use the raster register much more quickly than ever before. My next bit of assembly code was like this:

```
  sei
  ldx #0
l1
  lda #100
l2
  cmp $d012
  bne l2
  inx
  txa
  and #7
  sta $d016

  lda #150
l3
  cmp $d012
  bne l3
  lda #0
  sta $d016
  jmp l1
```

All the opcodes and branch offsets were calculated by hand and laboriously poked into memory. But in the end I had a split screen scroll with the other part of the screen static.

Ideas kind of very quickly exploded from there as I wrote my own assembler in BASIC at that point and really started on writing larger assembly projects. The game Tusari happened shortly afterwards and was my first real finished 100% assembly game.

Have you ever coded an intro or a demo?

MP

Yes several, that are not very good. :) Somewhere in the archive of all my old work disks (http://www.gamesthatwerent.com/2015/02/martin-piper-assets/) there are various unfinished demo ideas for wireframe 3D plotters, or scrollers etc. There is even somewhere in there a graphical demo creator application that allows graphics, music and colours bars to be setup with customised text and screen positioning.

Who are your favourite demo scene coders?

MP

Back in the day I have to say Triangle inspired me mostly because they did bigger demos with great design. These days Booze Design and Censor Design have to be right up there at the top of my list. These are groups, not just coders, because code needs the whole package or music/graphics/design to be really great.

OK, OK, if you really twist my arm to choose a specific coder then it would be The Human Code Machine, he seems like a nice chap and I respect his work.

SEUCK Redux is a revelation, did you plan to integrate SEUCK into your engine from the beginning?

MP

It wasn't planned as such. I was working on resurrecting my old work disk projects and enjoying tidying them up and exploring the old code. But it struck me how decades of knowledge in the console games dev industry that followed the C64 times kept on making me see massive improvements in my old teenage code. It was at this point I saw the enthusiasm that some people, Richard Bayliss mostly, had for SEUCK games. Back in the day I liked SEUCK but I saw that it had rather glitchy code. So I started poking around looking for the graphics file format, reverse engineering it by tweaking things in the editor and finding the changed bytes in memory. I thought to myself that maybe it would be useful to try writing some code to play these SEUCK files but using entirely new code with modern algorithms and ideas instead.

Which lead to this post on CSDB: http://csdb.dk/forums/?roomid=11&topicid=60233&showallposts=1

At this point I had just found the graphics, blocks and map data for the levels in memory and got the scroller working.

The c64 demo scene wasn't initially receptive to your project, did you find that discouraging?

MP
After writing that first post and before replying to the thread again I think I had already managed to figure out most of the animation code and enemy attack wave triggering and had it tied into the multiplexor and scroller code. So the few naysayers really didn't dissuade me, actually on the contrary their lack of vision encouraged me because it highlighted how the C64 scene really needed a good quality entirely open source game engine available for others to learn from.

I think now the large number of redux games with really good extra bespoke code that expand the game way beyond what SEUCK could do is a great vindication of the idea of sharing and reusing code when possible. It shows a generic open source game engine for the C64 is really feasible.

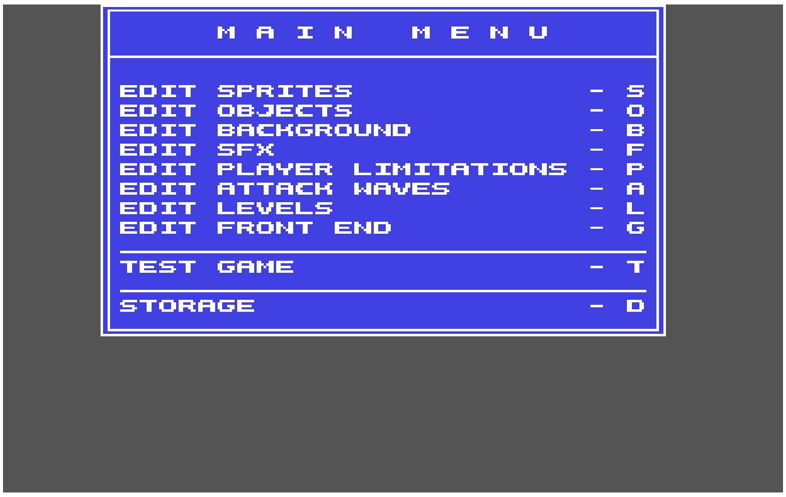
What is your favourite SEUCK Redux game?

MP
Richard Bayliss and Alf Yngve are prodigious when it comes to producing games. It's great that Richard really took onboard the idea of reusing the redux code. I think for me the favourite has to be "Legion of the Damned"

because it contains the special spell effects that make it obvious that it's obviously a lot more than just a horizontal scrolling SEUCK game.

Closely followed by "Nuclear Strike Force" because it really pushes the multiplexor quite hard, something the original SEUCK could not do.

Your engine can also be used to make multidirectional scrolling games, how hard would it be to create such a game with your engine?



```
M A I N   M E N U

EDIT SPRITES               - S
EDIT OBJECTS               - O
EDIT BACKGROUND            - B
EDIT SFX                   - F
EDIT PLAYER LIMITATIONS    - P
EDIT ATTACK WAVES          - A
EDIT LEVELS                - L
EDIT FRONT END             - G

TEST GAME                  - T

STORAGE                    - D
```

MP
The game engine is quite generic. So enabling the 8-way scrolling code (using Scroller_MultiDirection = 1 inScrollEntry.a) shows how to scroll the screen with 4x4 map blocks, full colour scroll and multiplexed sprites moving with the background. This code can be expanded to include animations from SEUCK or other sources.

Have you considered creating a level editor for the Redux multi-directional scrolling engine?

MP
I tend to use CharPad and SpritePad these days and just import the data. I used to make map editors out of a mixture of assembly and BASIC that were bespoke for each game idea.

Tell me about the other game projects you have planned for the future?

MP

Look through my old work disks and you'll probably see repeating themes and attempts at certain types of games.

For example racing games (http://www.gamesthatwerent.com/gtw64/racing-game/) and paradroid/citadel style games (http://www.gamesthatwerent.com/gtw64/citadel-2/).

I've spent most of my time recently honing various things to do with animation (http://csdb.dk/release/?id=117482&show=notes#notes),

data compression, music/sfx players, tape/disk loaders, Behaviour Driven Development (Cucumber driven testing for 6502 https://github.com/martinpiper/BDD6502),

source level remote debugging (https://github.com/martinpiper/ACME/tree/master/VICEPDBMonitor)

and some AI algorithms for the C64. These side projects are basically working towards something to do with the repeated themes in my old work disks. I feel this itch has not yet been scratched. But I've learned now never to promise anything specific until it is practically done, so I'm staying quiet Whatever happens, I know the code will eventually be made open source for everyone to enjoy. Just like the Berzerk redux port was open sourced.

Thanks for taking the time to talk to us, and thanks for your coding efforts with SEUCK Redux, we look forward to your future releases!

MP
You're welcome, and thank you for the interview invitation. I've enjoyed the trip down memory lane.

SuperBoot v1.00

for SuperCPU

Revised Edition

SuperCPU Initialization and Payload Launch Vehicle

Create large, multi-bank SuperCPU programs in read-only memory. Boot automatically.

Written by
Bert Novilla (satpro)
11/20/2016

Note: Emulation is currently the only avenue for activities of this sort. Thankfully, emulation in VICE is excellent. But, just for the record... *Actual hardware would be more excellent.*

Auto-boot a SuperCPU?

In this article, I will show how to create the program that starts and controls a SuperCPU. This program is a block of code commonly referred to as a boot or BIOS. What is a boot? For our purposes, a boot is "*hardware initialization code in read-only memory that passes control to a secondary, payload program in RAM.*" The payload can be anything. During computer initialization, the "boot" sets things up and then hands off to the payload. In this article, we will build a SuperCPU boot from scratch.

Our boot program, which we will call SuperBoot, requires neither C64 Kernal nor BASIC in order to function. Rather, it assumes neither is present. In a way, SuperBoot is a payload-assist service. It initializes computer hardware and helps launch a secondary program, aka the payload. SuperBoot starts at computer power-on, when the CPU is in Emulation mode and retrieving the RESET vector. Following this, SuperBoot implements a set of distinct steps, leading to an eventual handoff to some other control program. I use a version of this template to launch my experimental 16-bit operating system, Winc64. Without SuperBoot, I would have to load Winc64 every time I wanted to use it.

Why SuperBoot?

SuperBoot is very small and blindingly fast. It starts in VICE similar to a cartridge, automatically, except that the secondary program it delivers, the payload, will be running at a full 20 MHz in fast static RAM. This is akin to using a CMD or VICE image, except their payloads, C64 Kernal and BASIC, run in Emulation mode and thereafter, must load everything from disk or cartridge. If a program does switch to Native mode, hidden SuperCPU interrupt handlers still switch to Emulation mode and back *every time an interrupt occurs*. To make matters worse, SuperCPU cartridges operate at 1 MHz, and the only cartridges that will work contain 8K.

With all of this in mind, I decided to write SuperBoot.

Design Choices.

SuperBoot is minimalist. Working in 65816 Native mode, it initializes a *virtual* 65816, SuperCPU, and C64. It blanks the screen and border to black and passes control to a secondary, payload program at location $1000 in Bank $00. The rest is up to the payload program.

By keeping it small and simple, we eliminate many headaches. Actually, a full-featured BIOS could do much more. The CPU is in Native mode and C64 set for all RAM, so both 65816 and SuperCPU behave by the book *in full Native mode*, and all Emulation mode complications and quirks vanish. The payload SuperBoot delivers will not have access to a C64 Kernal or BASIC. It will be in an environment we refer to as "bare metal" and this means *there is no safety net*. One could add BASIC or Kernal as additional payload programs, but *to SuperBoot and SuperCPU*, they do not yet exist.

Ready. Reset. Go.

At power-on, a set of instructions executes, first in RAM Bank $00, then in ROM Bank $F8. This occurs *millions of cycles before* a C64 Kernal accepts control and shows the blinking "READY" prompt. Chances are, many readers have never seen or read about the first instructions that execute in a SuperCPU. This is immediately after power-on, in read-only memory located in Bank $F8. Before that there is RESET, the point in time when a 65816 is *reset electrically* and begins executing instructions. RESET is the starting point for everything that happens next. As it turns out, we can specify *what happens next.*

Native mode, SuperCPU, and its registers.

A 65816 CPU performs operations on 16-bit data and directly accesses a 24-bit address space. SuperBoot places the payload in fast static RAM (banks $00-$01), which is memory that runs at the full 20 MHz. A surprisingly small number of hardware registers will control a SuperCPU effectively.

To aid in learning how a SuperCPU works, I studied the original binary image from CMD as well as source code for the VICE emulator. I also obtained unpublished VICE boot ROM source, and it was invaluable. While not plentiful, there was enough information online to gather a reasonably accurate picture of how a SuperCPU works. I also documented dozens of other locations, which are not important here.

CMD 2.04

The CMD 2.04 image (128K) is elaborate and contains an enormous amount of code tasked with proprietary device communication, among many, many other things. A zillion C128 machine checks litter the startup code. The 2.04 binary includes the famous "CMD Space Rocket" splash animation, JiffyDOS, RAMLink, and a useless second bank, which is devoted to the C128. The C64

Kernal is heavily patched for use in the SuperCPU

VICE 0.05-0.06
In contrast, the VICE 0.05 image (64K) consists of minimal code devoted primarily to controlling a SuperCPU and dealing with interrupts in 8-bit Emulation mode. VICE does not worry about a C128, RAMLink, or JiffyDOS, but just like CMD, performs a memory test and confines itself to Emulation mode.

SuperBoot v1.00 boots a SuperCPU 76% faster than VICE, and 99.9% faster than CMD. The secret is... Native mode.

Power-on and Bootmap mode.
At power-on, a 65816 is in 8-bit Emulation mode. Instruction execution commences at the 16-bit address pointed to by locations $FFFC-$FFFD, in Bank $00 -- the normal RESET vector everyone knows. Normally this address is in ROM, but the SuperCPU contains RAM in Bank $00. There is no ROM. To overcome this issue, the SuperCPU contains an operational environment known as *Bootmap mode*.

In Bootmap mode, the CPU "sees" the upper 32K bytes of *read-only* memory (ROM) Bank $F8 instead of random-access memory (RAM) in Bank $00, excluding the 4K I/O block at $D000. With clever address mapping, Bootmap mode fools a SuperCPU into seeing ROM where RAM exists.

Note: VICE requires ROM images to contain 64K, 128K, 256K, or 512K bytes. 64K segments are numbered 0 - 7, and SuperCPU reserves banks $F8-$FF for read-only memory. In this article, read-only memory in Bank $F8 is referenced as "ROM Bank $F8".
The power-on indirect vector landing point, which can be as simple as a long jump back to ROM Bank $F8, is located within the second 32K block of memory in ROM Bank $F8, excluding the $D000 I/O block. While Bootmap mode is active, a 65816 thinks it is working in actual Bank $00 memory. *This is how a SuperCPU finds the non-existent RESET vector at $FFFC.* Just after RESET via $FFFC, SuperBoot lands at $FFE0 -- right into the loving arms of a long jump back to ROM address $F8:0000.

With CMD and VICE images, a SuperCPU is a 16-bit machine forced into compatibility with the limited 8-bit environment of a C64, Kernal, and BASIC. An operating system must allow programs to run in Native mode, yet both the Kernal and BASIC run assuming a 6510. Without modification, the stock C64 Kernal will not function properly in Native mode, running at 20 MHz. There are several additional SuperCPU hardware issues with an Emulation mode kernal, as noted elsewhere in this document. SuperBoot's approach is to activate Native mode at 20 MHz and stay there, and to assume a C64 Kernal does not exist. If desired, one could modify and integrate a C64 Kernal into the payload mix, as CMD and VICE do, but a it is *not* required for SuperCPU operation.

Later on, Bootmap mode is disabled and normal mapping resumes as initialization continues. The final task in ROM Bank $F8 is to pass control to the payload program. Arbitrarily, I chose location $1000 in Bank $00 as the hand-off address, but it could be *any address* in Banks $00-$01, with a few exceptions. At any rate, a new control program will take over from there. SuperBoot is finished.

"Inside SuperBoot"

At power-on, a SuperCPU is in Bootmap mode, a 65816 is in Emulation mode, and a C64 is in reset. After a long jump from $FFE0 in Bank $00, hardware initialization *begins* in ROM Bank $F8, but does not remain there for very long. Read-only memory is slow, relatively speaking -- barely half as fast as static random-access memory -- so only important tasks are performed by instructions in ROM Bank $F8. Everything else takes place later at full-speed in fast static RAM Bank $00.

Below is an example showing the last 192 bytes of ROM Bank $F8. This block is largely unused, yet vital to the boot process. Native mode interrupt vectors at $FFE4-$FFEF behave just as they do with a C64, but are in a different place. Notice the long jump at $FFE0. This is a four-byte springboard to ROM Bank $F8, the place where hardware initialization begins. *($FFE0 is SuperBoot's "$FCE2" equivalent)* SuperBoot is not concerned with any of the interrupt vectors, and leaves interrupts disabled. The payload will determine and configure interrupts.

In the table below, the *mostly empty* 192-byte block shown is visible in Bank $00 only while Bootmap mode is active, so SuperBoot copies this block from ROM Bank $F8 to RAM Bank $00 in order to be visible at all other times.

The 160-character copyright message block space, located at $FF40 and extending to $FFDF, is cosmetic and optional. VIC Bank 3 bitmap graphics will not affect the copyright message because it is out of range.

Source Code Format.
In all code samples, there are no symbolic names, labels, or macros. The source text is a pseudo-monitor listing, designed to be easily understood.

Vector Table, Springboard, and Copyright Message at $F8:FF40
Mapped to $00:FF40 when Bootmap mode is active

```
*= $ff40            ;your message goes here (max. 160 chars)

*= $ffe0            ;"Reset"     5C 00 00 F8

     jml $f80000    ;springboard: long jump to ROM at $f8:0000

*= $fffc         ;RESET  E0 FF
.word $ffe0       ;RESET points to "Reset" ($ffe0)
```

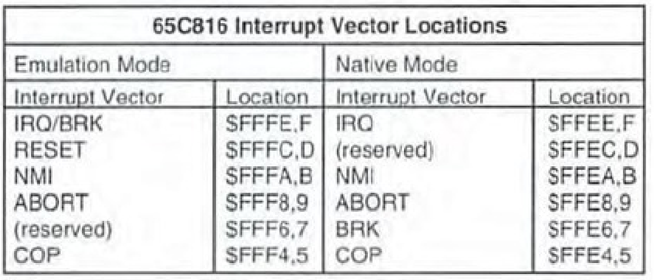
65816 Startup

First, SuperBoot initializes the 65816 CPU. Some values, such as Stack Pointer and Direct Page, are arbitrary and default. The payload will probably change the values anyway. The Data Bank register is set to $00, which allows SuperBoot to use normal addressing in RAM Bank $00 -- *while executing instructions in ROM Bank $F8*. SuperBoot will remain in Native mode until handoff, but ultimately, the payload determines what happens after SuperBoot has finished. In reality, a robust boot program would anticipate a rogue-entry situation, e.g. a warm start. For simplicity, SuperBoot assumes the only entry point is through hardware reset.

For this first code fragment, the CPU operates with a 16-bit accumulator and 8-bit index registers. As stated earlier, interrupts will remain disabled until handoff.

65816 Initialization at $F8:0000 (Start)

```
*= $f80000        ;$F80000 in bank rom $f8
;SuperBoot lands here after a long jump from $00ffe0...
;*** 65816 is in Emulation mode, SuperCPU is in Bootmap mode ***

;----- 65816 CPU NATIVE MODE INITIALIZATION -----
78    sei      ;disable interrupts
18    clc
FB    xce      ;enter Native mode
C2 20    rep #$20 ;set 16-bit AM, 8-bit XY
A9 FF 01 lda #$01ff
1B    tcs      ;Stack Pointer (SP) = $01ff
A9 00 00 lda #$0000
5B    tcd      ;Direct Page (DP) = $0000
AA    tax      ;copy 0 into 8-bit X
DA    phx      ;do a "push-pull" of a 0 byte
AB    plb      ;Data Bank (DB) = $00, R/W data in Bank $00
```



65C816 Interrupt Vector Locations

Emulation Mode		Native Mode	
Interrupt Vector	Location	Interrupt Vector	Location
IRQ/BRK	$FFFE,F	IRQ	$FFEE,F
RESET	$FFFC,D	(reserved)	$FFEC,D
NMI	$FFFA,B	NMI	$FFEA,B
ABORT	$FFF8,9	ABORT	$FFE8,9
(reserved)	$FFF6,7	BRK	$FFE6,7
COP	$FFF4,5	COP	$FFE4,5

Disable C64 hardware.

Now, it is time to visit a few key hardware registers in the C64 itself. Because read-only memory is slow memory, SuperBoot limits time-consuming activities while in ROM. For now, it will shut down C64 I/O, disable simulated C64 ROM memory, and blank the screen to a nice, clean black.

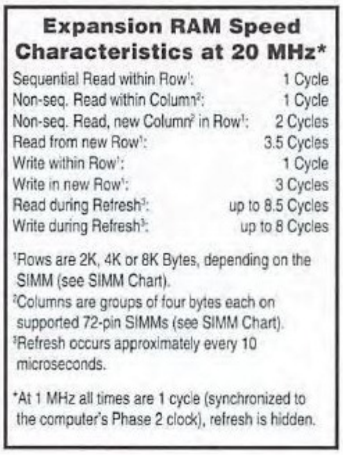
Accessing C64 hardware.

For a SuperCPU, accessing C64 hardware is always expensive, even while operating in static RAM. In order to be 100% effective, C64 I/O accesses should be spaced about 20 CPU cycles apart. Anything less, on average, could result in a slight delay while waiting for the next 1 MHz I/O access cycle. In reality, one will probably never have to worry about this, as there is a one-byte hardware buffer to help mitigate I/O mirroring wait states.

The SuperCPU runs at 20 MHz; the C64 hardware runs at 1 MHz. You either wait, or do something else for 19 cycles while C64 I/O accesses take place. It really is this simple. The payload program should limit its activity to static RAM Banks $00-$01 as much as practicable. Static RAM *guarantees* 20 MHz. All other memory is slower, to varying degree. Period.

64K RAM, Disable C64 I/O Hardware
MMU, VIC, CIA, and SID

```
A2 E5    ldx #$e5       ;MMU:  disable simulated ROM, enable I/O
86 01    stx $01        ;c64:  $a000/$e000 = RAM, $d000 = I/O
A2 2F    ldx #$2f       ;set data direction
86 00    stx $00

A2 00    ldx #$00       ;border (clean, black boot screen)
8E 20 D0 stx $d020      ;VIC border register
11 D0    stx $d011      ;blank the screen output to border color

A2 7F    ldx #%01111111 ;#$7f   disable all cia interrupts
8E 0D DC stx $dc0d      ;CIA #1 Interrupt Control Register
8E 0D DD stx $dd0d      ;CIA #2 Interrupt Control Register

AD 0E DC lda $dc0e      ;CIA #1 Control Register A
29 FE FE and #$fefe     ;16-bit mask
8D 0E DC sta $dc0e      ;cia #1 16-bit stop timers A & B

AD 0E DD lda $dd0e      ;CIA #2 Control Register A
29 FE FE and #$fefe     ;16-bit mask
8D 0E DD sta $dd0e      ;cia #2 16-bit stop timers A & B
```



Expansion RAM Speed Characteristics at 20 MHz*

Sequential Read within Row[1]:	1 Cycle
Non-seq. Read within Column[2]:	1 Cycle
Non-seq. Read, new Column[2] in Row[1]:	2 Cycles
Read from new Row[1]:	3.5 Cycles
Write within Row[1]:	1 Cycle
Write in new Row[1]:	3 Cycles
Read during Refresh[3]:	up to 8.5 Cycles
Write during Refresh[3]:	up to 8 Cycles

[1] Rows are 2K, 4K or 8K Bytes, depending on the SIMM (see SIMM Chart).
[2] Columns are groups of four bytes each on supported 72-pin SIMMs (see SIMM Chart).
[3] Refresh occurs approximately every 10 microseconds.

*At 1 MHz all times are 1 cycle (synchronized to the computer's Phase 2 clock), refresh is hidden.

SuperCPU hardware

From here, SuperBoot moves on to SuperCPU hardware. To unlock write access, it performs a store (poke) to location $D07E. This activates SuperCPU hardware registers and permits write access. Storing to location $D07F de-activates the registers and resumes read-only status.

In Bank $00, the $D200-$D3FF block holds 512 bytes covered by hardware write-protection, accessed as described above. For example, CMD stores the amount of free RAM in this block. SuperBoot does not use any of this memory.

The Free RAM Test and Checksum.

One of the ways SuperBoot saves time is by eliminating the RAM test. This may have been necessary with real hardware, but we are *emulating* a 16MB SuperCPU. We know our RAM is good, and we know how much there is. There is no need for this extremely costly, long loop. For the same reason, SuperBoot does not need a checksum calc.

Note: Bonus! We gain 128K, because CMD (and VICE) reserve RAM Banks $F6-F7 for “system use” and mark those banks as off-limits.

20 MHz Turbo is selected, and memory optimization is turned off in preparation for the payload copy. Bootmap mode is disabled, and hardware registers are de-activated.

Initialize SuperCPU Hardware

```
;----- INITIALIZE SUPERCPU HARDWARE -----

8E 7E D0 stx $d07e      ;activate hardware registers
;--
A2 84    ldx #%10000100 ;#$84 payload will specify a vic bank later
8E B3 D0 stx $d0b3      ;= full speed (turns off all mirroring)

8E 7B D0 stx $d07b      ;software speed select = 20 mhz
8E 73 D0 stx $d073      ;system disable 1 mhz
A2 04    ldx #$04
8E 78 D0 stx $d078      ;set SIMM type (16 mb)
8E B6 D0 stx $d0b6      ;disable bootmap ($f8 mapped into $00 @ $8000-$ffff)
;--
8E 7F D0 stx $d07f      ;de-activate hardware registers
```

That is it. All that remains is to copy the payload program to RAM Bank $00.

The Payload Copy.

The possibility exists for some program to require use of all 512K ROM. For efficiency, a payload should place as much code and data as possible into RAM Banks $00 and $01. For that, SuperBoot copies memory from ROM Bank $F8 to RAM Bank $00, in one or more payload blocks. In our example here, SuperBoot delivers data to two different places in RAM Bank $00 using instruction MVN.

CPU registers A, X, and Y are set to 16 bits for the copy. For ease here, source and destination addresses are the same, but do not need to be. Remember to preserve the Data Bank register after executing the MVN instruction. From ROM to RAM, transfer rate is approximately 1 byte/us. (1MB/sec.). This is about one byte per C64 cycle.

Deliver the Payload Program to RAM Bank $00

```
C2 30    rep #$30          ;go16axy
8B       phb               ;save data bank because it is not retained

A9 FF 07    lda #$0800-1   ;size-1  copy the 2048-byte payload
A2 00 10    ldx #$1000     ;src addr
A0 00 10    ldy #$1000     ;dst addr
54 00 F8    mvn $f8, $00   ;copy 2048 bytes from $f8:1000 to $00:1000

A9 BF 00    lda #$00c0-1   ;size-1  copy the 192-byte vector table
A2 40 FF    ldx #$ff40     ;src addr
A0 40 FF ldy #$ff40        ;dst addr
54 00 F8    mvn $f8, $00   ;copy 192 bytes from $f8:ff40 to $00:ff40

AB       plb               ;restore data bank
```

Wrapping up.

Earlier, SuperBoot disabled memory optimization. Memory optimization targets a VIC bank and only mirrors data from the specified 16K bank.

With that, the initialization routine is complete! The payload program at $00:1000 takes over and SuperBoot exits. At this point, a typical payload program might load its graphics or display a splash screen. SuperBoot however, is finished.

The End

```
;-----  done, exit to payload program coldstart  -----

5C 00 10 jml $001000 ;payload coldstart
```

Some final thoughts.
That is all there is to it. 124 bytes, 2325 1 MHz cycles -- with a 2K payload. Not bad, eh? The computer is now humming along at a full 20 MHz, in the fastest type of RAM available. For this article, SuperBoot was limited intentionally. One idea is to expand this template with more features and options. For example, you could load settings from the REU, the contents of which can be saved in VICE until the next session a la RAMLink. For this article we have assumed the payload will manage its own operating environment, and purposely limited our initialization.

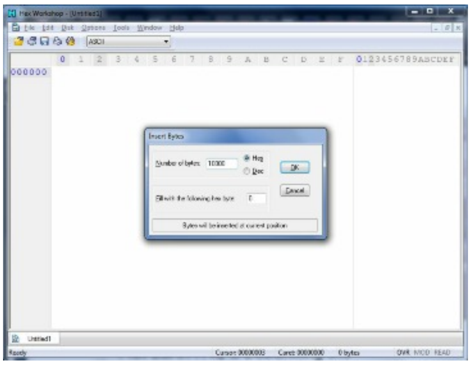
Enjoy.

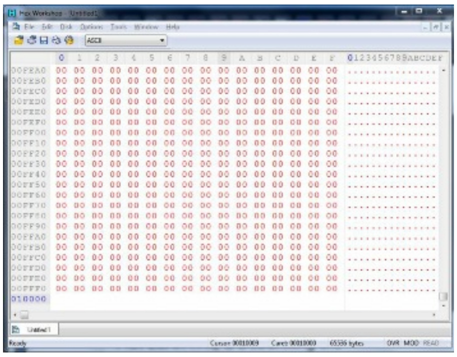
For further information, please email me at bert@winc64.com.

How to Build the ROM Image

In order to build an image of SuperBoot, you will need a hex editor. For version 1.00, we will build a 64K image, but you may also try 128K, 256K, or 512K. Programs should download code and data to static RAM Banks $00-$01 prior to time-intensive operations involving the data.

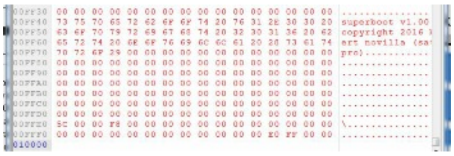
1. Create a new 64K image, filled with 00 bytes. The image size is 65536 bytes.


Hex Workshop - [Untitled1]
File Edit Disk Options Tools Window Help
ASCII
Insert Bytes
Number of bytes: 10000
● Hex
○ Dec
OK
Fill with the following hex byte: 0
Cancel
Bytes will be inserted at current position
Untitled1
Ready
Cursor: 00000003
Caret: 00000000
0 bytes
OVR MOD READ



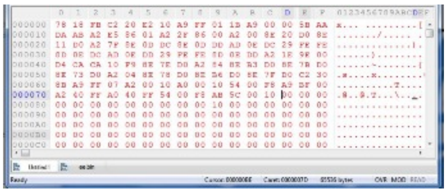
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
00FEA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FEB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FEC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FED0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FEE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FEF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
010000
```

2. Insert: E0 FF at $FFFC. Insert: 5C 00 00 F8 at $FFE0. Copyright text at $FF40 is optional.



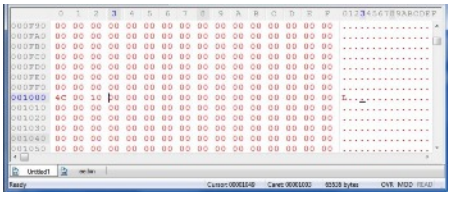
```
00FF30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF40  73 75 70 65 72 62 6F 6F 74 20 76 31 2E 30 30 20  superboot v1.00
00FF50  63 6F 70 79 72 69 67 68 74 20 32 30 31 36 20 62  copyright 2016 b
00FF60  65 72 74 20 6E 6F 76 69 6C 6C 61 20 28 73 61 74  ert novilla (sat
00FF70  70 72 6F 29 00 00 00 00 00 00 00 00 00 00 00 00  pro)............
00FF80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FF90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00FFE0  5C 00 00 F8 00 00 00 00 00 00 00 00 00 00 00 00  \...............
00FFF0  00 00 00 00 00 00 00 00 00 00 00 00 E0 FF 00 00  ................
010000
```

3. Insert SuperBoot binary at $0000.



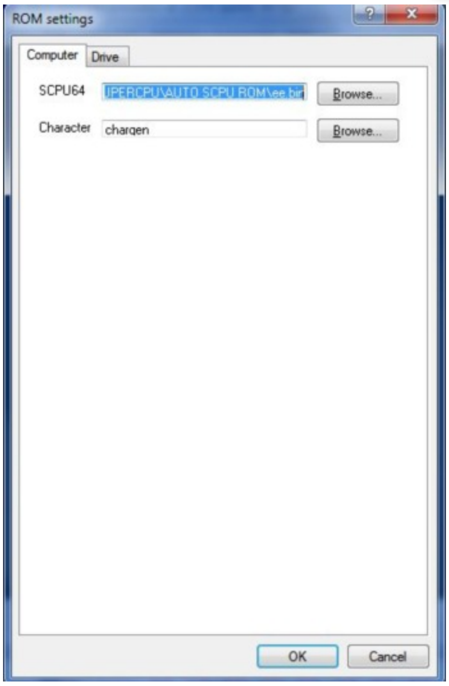
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
000000  78 18 FB C2 20 E2 10 A9 FF 01 1B A9 00 00 5B AA  x..............[
000010  DA AB A2 E5 86 01 A2 2F 86 00 A2 00 8E 20 D0 8E  ......./........
000020  11 D0 A2 7F 8E 0D DC 8E 0D DD AD 0E DC 29 FE FE  .............).
000030  8D 0E DC AD 0F DD 29 FE FE 8D 0F DD A2 1E 9E 00  ......)........
000040  D4 CA CA 10 F9 8E 7E D0 A2 84 8E B3 D0 8E 7B D0  ......~.......{.
000050  8E 73 D0 A2 04 8E 78 D0 8E B6 D0 8E 7F D0 C2 30  .s....x........0
000060  8D A9 FF 07 A2 00 10 A0 00 10 54 00 F8 A9 DF 00  ..........T.....
000070  A2 40 FF A0 40 FF 54 00 F8 AB 5C 00 10 00 00 00  .@..@.T...\.....
000080  00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00  ................
000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

4. Insert payload binary at $1000.



```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
000F90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000FA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000FB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000FC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000FD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000FE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000FF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001000  4C 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00  L...............
001010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

5. Click **Settings/ROM Settings** and load the image into VICE.



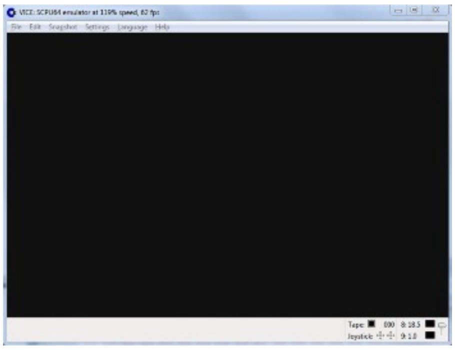
ROM settings

Computer | Drive

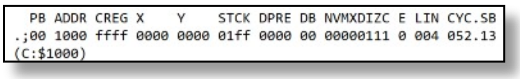
SCPU64: UPERCPU\AUTO SCPU ROM\ee.bin

Character: chargen

6. Click **File/Reset/Hard** and perform a Hard Reset (Ctrl+Alt+R). You should see something like this.


VICE: SCPU64 emulator at 119% speed, 62 fps
File Edit Snapshot Settings Language Help

A register display should look like this. Use Alt+M to enter the monitor, then the <R> command.



```
  PB ADDR CREG X    Y    STCK DPRE DB NVMXDIZC E LIN CYC.SB
.;00 1000 ffff 0000 0000 01ff 0000 00 00000111 0 004 052.13
(C:$1000)
```

That's it! We are finished.

Commodore Free Magazine

www.commodorefree.com

Editor
Nigel Parker

Spell Checking
Peter Badrick
Bert Novilla

Text , HTML & Ebook Conversion
Paul Davis

D64 Disk Image
Al Jackson

ISSUU formatting
Alessandro Di Nepi

PDF Design /Editor /webhost /text collector
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

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.