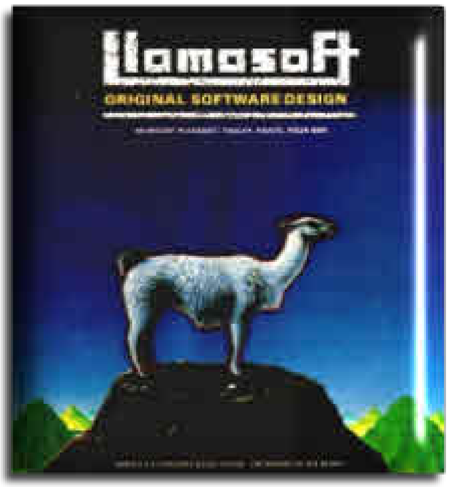
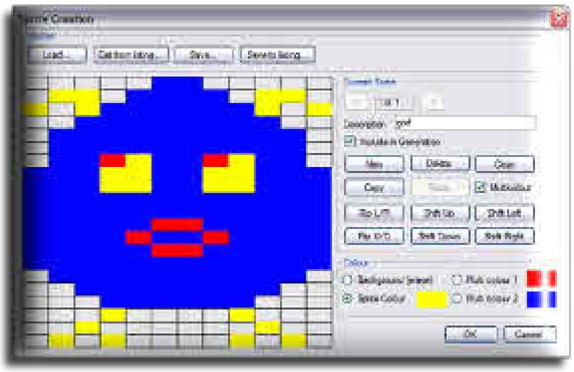
# Commodore Free

## Issue 23 September 2008

**Free to download Commodore magazine**
**Dedicated to Commodore Computers**
**Available as PDF Text SEQ HTML and D64 image**
**www.commodorefree.com**





```
Track 01 Track 02 Track 03   Sector 00
00*      00*      00*
01  ...00 01 ...00 01 ...00
02  ...00 02 ...00 02 ...00
03  ...00 03 ...00 03 ...00
04  ...00 04 ...00 04 ...00
05  ...00 05 ...00 05 ...00
06  ...00 06 ...00 06 ...00
07  ...00 07 ...00 07 ...00
08  ...00 08 ...00 08 ...00
09  ...00 09 ...00 09 ...00
Rtime
15:15 1:00-00 OCTAVE :02 SPEED  :01
Timer 2:00-00 MUSIC  :00 VOLUME :0F
00:00 3:00-00 POSIT  :
```

# Editor

Great news! Sub Hunter is available to pre-order, its interesting for a supposedly dead machine that so much software is both available and readily being developed!

Last month saw an improved Html version of the magazine, this was mainly pushed from Jason Kelk www.commodorecomputerclub.co.uk as the webpage of the magazine is mirrored on this website. Jason contacted me to say the Html had rouge characters odd spaces and would not work in all browsers. I worked with a very patient Jason to resolve the issues finally settling on doing the HTML coding by hand, it wasn`t very difficult and led to a far better version of the magazine. Previously and because of time restraints I had used an application that converted the text into Html by just clicking a button, this was obviously a mistake. I am however concerned that no one contacted me before to complain they couldn`t read the Html version in xyz browser, although it was tested in internet explorer I didn`t have time to test it with anything else. Jason although not fully excited about the Html version declared that it would be ok and stripped out 2k of extra spaces, although Jason is patient and curtious I am sure he felt like just telling me where to get off, it does show what a great person he is to be so controlled, especially as he was busy on his own projects, so thanks Jason

Last month saw the production of a more streamlined version, this gave a few well needed results, I managed some spell checking, an although it may not have been perfect I am sure it was rather better than the pre-checked version, I actually spent 3 hours checking and reading the publication before finalising it all into the various versions. I am no spell checker and would appreciate someone re-reading my work, so if you feel like volunteering speak up. If you want you can do the work without being named, whatever you like.

This issue see a final instalment of the music tutorial from Richard Bayliss in the form of an in-depth look at music production using DMC version 5. Thanks also go to Retro gamer CD and Michael Bevan for allowing Commodore Free the reprint rights the LLAMASOFT - The Dromedary Years article this issue sees part 1 with the conclusion (part 2 ) in next months issue We continue our in the beginning series of tutorials for new users of Commodore machines and finally we have an interview with the creator of Commodore 64 .prg Generator

Ok I need to do some spelling and Grammar checking so that`s me out with the fairies for the next 3 hours see you next month

Regards
Nigel

www.commodorfree.com

# Contents

**NEWS**

**INTERVIEWS**

**TUTORIAL**

**REFERENCE**

**HOW CAN I HELP COMMODORE FREE**..
Ok the best way to help would be write something about.. Commodore (yes for the observant I spelled the com-.. pany correctly this time) _grin seriously though articles.. are always welcome,..

**WHAT ARTICLES DO YOU NEED**..
Well they vary, contact me if you have an idea but I am.. looking for..

**Tutorials..**
(beginners and Expert),..

**Experiences..**
with Commodore,..

Why I love Commodore machines,.
.
**Interviews.**.
maybe you have access to a power user..

# News

## SUB HUNTER available to pre order

Subhunter is available to pre order from
Psytronic software

http://www.binaryzone.org/retrostore/index.php?main_page=product_info&cPath=82&products_id=430

A scientific experiment to increase the size and number of fish in the sea has gone terribly wrong and now the oceans have been taken over by mutant fish. You must destroy the fish and save the swimmers who have become stranded in their midst. Watch out for agent subs that will do anything to keep this a secret.

**You are the SUB HUNTER!**

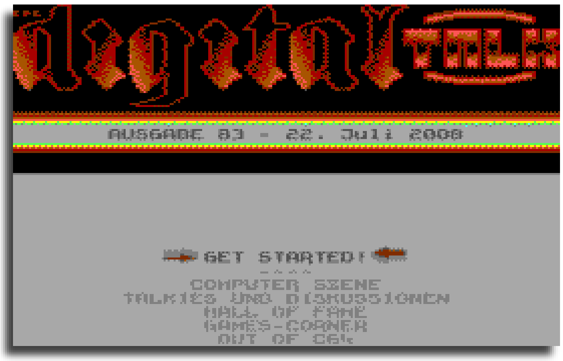
Here's your chance to pre-order the fantastic new C64 game from Richard Bayliss & Frank Gasking - SUB HUNTER! The game features 25 levels of sub-aquatic action, amazing multi-layered parallax scrolling, superb detailed graphics, varied missions including shoot 'em up levels, rescue missions and terrifying boss encounters.

SUB HUNTER will be released on the Commodore 64 in October 2008 (exact release date to be confirmed). The game will be available for the Commodore 64 on tape and will be presented with specially produced full-colour inlay artwork. For a taster of the game you can download a 4 level preview from the Psytronik Software webpage!

** SUB HUNTER GAME CREDITS **
Programming - Richard Bayliss
Graphics + loader screen artwork - Frank Gasking
Music - Richard Bayliss and Thomas Mogensen
Inlay design by Jason Kelk and Jason Kenz Mackenzie
Tape loader - Paul Hughes

## Digital Talk #83

Digital talk electronic magazine number 83 is available to download from

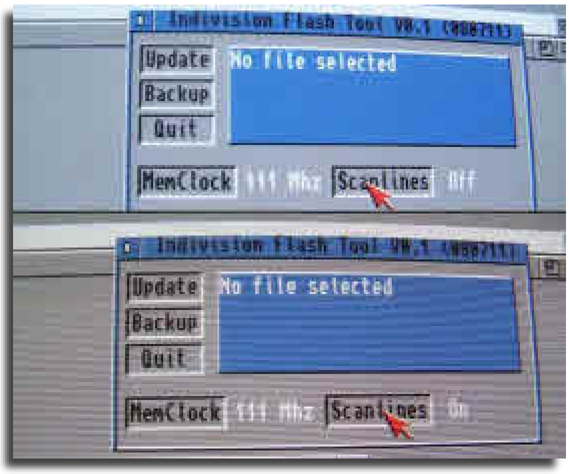

digital TALK
AUSGABE 83 - 22. Juli 2008

GET STARTED!
COMPUTER SZENE
TALKIES UND DISKUSSIONEN
HALL OF FAME
GAMES-CORNER
OUT OF C64

http://noname.c64.org/csdb/release/?id=69160

Individual Computers AGA flicker fixer
--------------------------------------
http://www.jschoenfeld.com/
New product: Indivision AGA

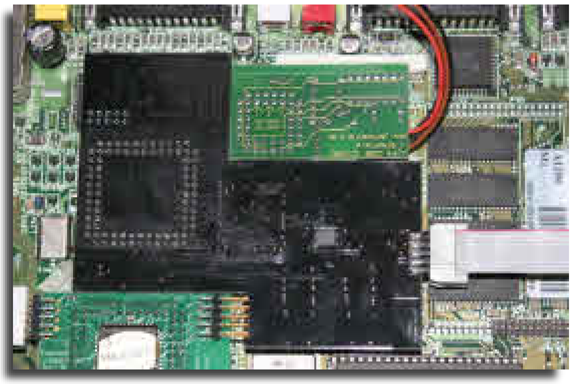
During the 80s, it was common to build computers in a way that there picture could be output onto a domestic a TV set. This was, in order to reduce the overall system cost for the user. TV sets work with lower frequencies compared to today's monitors, therefore the picture flickers. A flickerfixer converts the signal in a way that a modern VGA monitor or TFT-display can be used in place of the TV. The name flickerfixer was mainly created because the problem of a flickering picture is solved. Here at Individual computers we already had a very successful product called Indivision in 2002, indivision was connected to the monitor output of the computer, but the product has been sold out since 2006.


Indivision Flash Tool V0.1 (080711)
Update
Backup
Quit
No file selected
MemClock 111 Mhz Scanlines Off

Indivision Flash Tool V0.1 (080711)
Update
Backup
Quit
No file selected
MemClock 111 Mhz Scanlines On

Flickerfixers are still in high demand, one unit of our external flickerfixer Indivision has recently been sold on an internet auction platform for more than 900 US Dollars. That's amount is more than four times the last retail price! This shows are products are popularity, stable and provide value for money. We have already started the development of a new improved flickerfixer the development work began in November 2007, this work is now completed!

The first flickerfixers for the Amiga were first available over 17 years ago, since that time all flickerfixers - including ours - have been using the same basic principles: The picture frequency was doubled, This is not enough for most modern VGA monitors and TFT-displays, because the picture of a computer with a PAL standard was only displayed at slightly less than 50Hz. Most monitors require 60Hz or even more these days.



With Indivision AGA, we're striking a new path. As the product name suggests, this is a flickerfixer very closely tied to the Amiga AGA chipset. This could only be realized by placing the flickerfixer inside of the computer. The product is mounted over the Video-Chip "Lisa". The VGA display is then connected directly to the flickerfixer, further connections are not necessary.

Listing all features of the flickerfixer would go beyond the scope of this news item. You can find the complete technical data in a PDF-overview on the website.

The most outstanding features are:
-24 Bit colour resolution in all screenmodes
-Support for super hires modes
-Highgfx support up to 1024x768 pixels
-All screenmodes can be displayed at 60Hz or more
-no adjustments necessary
-Two options have been added for the retro-gaming enthusiasts that allow a step-by-step approximation of the display from an old TV set: Vertical synchronisation allows reducing the output frequency down to 49.9Hz, which will eliminate tearing-effects. In addition to that, a scanline-emulation can be switched on, This will emulate the dark lines in-between the scanlines of a TV screen.

Development and production of this extremely complex product is very expensive. Our trade partners, who have been serving the Amiga-market for many years, have made considerable financial contributions to the production. To account for this, Indivision AGA is exclusively available only from these resellers until January 1st, 2009: (alphabetical order)

AmigaKit (England)
GGS-Data (Sweden)
Software Hut (USA)
Vesalia (Germany)

Production has already started, and the flickerfixer will presumably be shipped to our trade partners during the first week of September. The suggested retail price is 129-EUR including 19% VAT. This version currently only supports the Amiga 1200 range of computers, although we are already working on a version that will work for other Amiga systems with the AGA chipset.

**VICE VERSION 2 Released**
http://www.viceteam.org/#vice

**Changes in VICE 2.0**
**General**
VICE development has moved to public services: http://sf.net/projects/vice-emu You can read the latest development in the Subversion repository, file bug reports, and similar things. Check it out!

- New Openserver 5.x, Openserver 6.x and Unixware ports.
- New AROS64 port.
- New QNX 4.x port.
- New HPUX (HPPA and IA64) ports.
- New SkyOS port.
- New Atari Mint port.
- New Solaris Express aka OpenSolaris aka Solaris 11 port.

**C64/C128 changes**
- Improved $01 behaviour with regards to unused bit fall-off.
- New experimental NEOS and Amiga mouse support.
- Added a working RTC to the emulation of the IDE64 cartridge.
- New more precise PAL emulation, works only with double size and double scan.

- Added support for the Action Replay 4 and StarDos cartridges.
- Improved REU compatibility
- Fixed CPU timing bug introduced in 1.22 (demo "Borderless").
- Fixed bugs in the CIA emulation (demo "So-phisticated", W. Lorenz testsuite).
- Fixed various bugs in VIC-II emulation: Sprite x-positioning, Latch X/Y, video mode changes, DEN bit handling, raster IRQ, NTSC sprites and sprites in full/debugging border mode (list of fixed demos/games too long to mention here).
- RESTORE key is handled differently now, fixing an NMI bug because the NMI was asserted too long before.

**VIC20 changes**

- Added unconnected c-bus floating byte support.
- New more precise PAL emulation, works only with double size and double scan.

**PET changes**

- Added support for toggling CB2 sound output line.

**PLUS4 changes**

- New more precise PAL emulation, works only with double size and double scan.

**Drive changes**

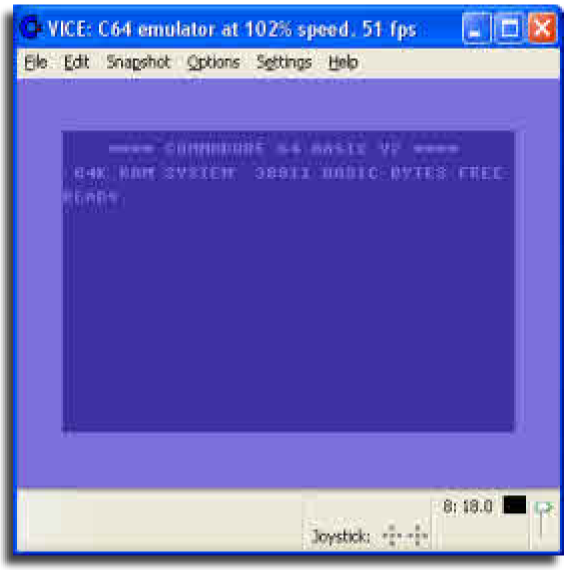
- Fixed block allocation and interleave.

**Unix changes**

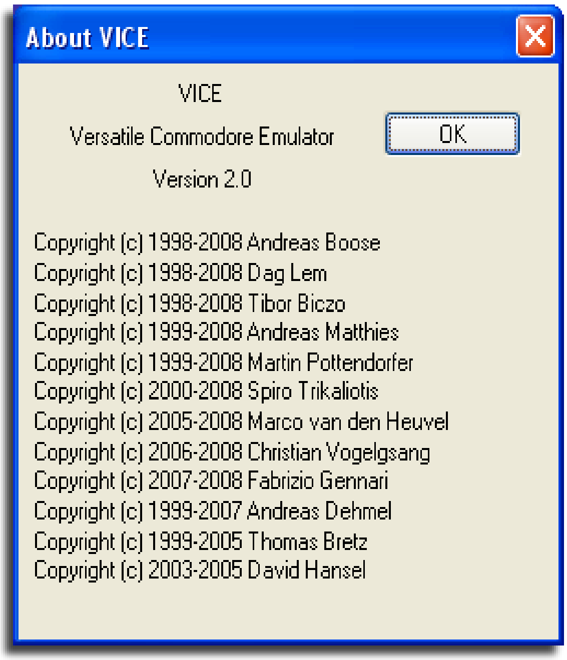
- Broken DGA1/DGA2 code removed
- XRandR fullscreen implemented for all (but CRTC/VDC) videochips.
- XRandR/Fullscreen can now toggle the Statusbar/Menu
- Fixed unix mouse for single sized display.
- Fixed mouse grab and xaw menu popup.
- After a CPU jam user can select to resume emulation without reset.

**MS-Windows changes**

- Fixed the reverse logic of the -confirmexit/+confirmexit and -singlecpu/+singlecpu command-line options.
- Added the ability to copy and paste text from and to the VICE window.


VICE: C64 emulator at 102% speed, 51 fps
File Edit Snapshot Options Settings Help
8: 18.0
Joystick:

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



VICE

Versatile Commodore Emulator

Version 2.0

Copyright (c) 1998-2008 Andreas Boose
Copyright (c) 1998-2008 Dag Lem
Copyright (c) 1998-2008 Tibor Biczo
Copyright (c) 1999-2008 Andreas Matthies
Copyright (c) 1999-2008 Martin Pottendorfer
Copyright (c) 2000-2008 Spiro Trikaliotis
Copyright (c) 2005-2008 Marco van den Heuvel
Copyright (c) 2006-2008 Christian Vogelgsang
Copyright (c) 2007-2008 Fabrizio Gennari
Copyright (c) 1999-2007 Andreas Dehmel
Copyright (c) 1999-2005 Thomas Bretz
Copyright (c) 2003-2005 David Hansel

- Added the ability to paste text into the VICE window by right-clicking on the VICE window.
- Keyboard shortcuts are definable in the configuration file win_shortscuts.vsc
- After a CPU jam user can select to resume emulation without reset.
- Fixed volume slider behaviour when switching full screen mode.
- Key repeat is eliminated OS-wise. This way, the RESTORE key is not used with auto repeat.

**BeOS changes**

- Fixed to also run on Haiku.
- Lots of fixes to the dailogs.
- Fixed the joystick system.
- Added warning dialog when the monitor is accessed without having an open console window.
- Added the ability to copy and paste text from and to the VICE window.

**AmigaOS changes**

- Fixed the reverse logic of the -confirmexit/+confirmexit command-line options.
- Added the ability to copy and paste text from and to the VICE window.
- After a CPU jam user can select to resume emulation without reset.

**OS/2 changes**

- After a CPU jam user can select to resume emulation without reset.

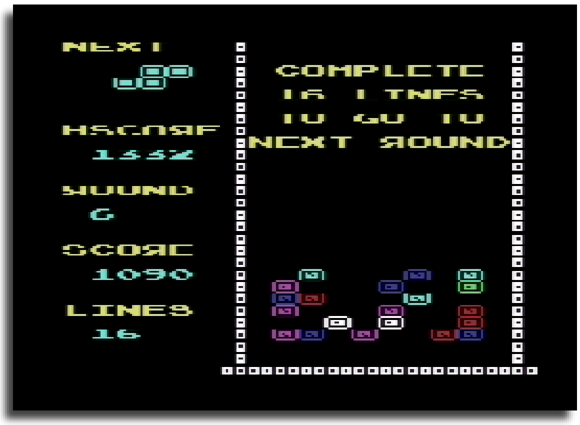
**Miscellaneous changes**

- Added support for more 3rd party basic extenders to petcat.
- Extended BMP saving to support 24bit.
- Fixed the joystick code of all emulators to handle state changes on both joysticks at the same time.
- New monitor commands and features.
- New memmap feature which allows tracking of memory accesses, activated by the configure option enable memmap.
- Fixed bug in ffmpeg/FFV1 driver. Should be really lossless now.

### VIC20 TETRIS

Another version of the popular puzzler TETRIS has been released for the VIC 20 and can be downloaded from here

http://one.xthost.info/nbla000/tetris/



### Bil Heard phone interview

On Thu, July 24, 2008 2:47 pm, Robert Bernardo wrote:

CBM engineer Bil Herd suggested the idea of a video conference into this year's CommVEx, instead of the usual pre-recorded, yearly CBM Animals video. He has rounded up Andy Finkel, CBM engineer who has worked on the VIC-20, C64, and Amiga as the focus of this conference.

Bil said, "I was able to get hold of Andy Finkel who was head of the games group the entire time I was there and went on to do LOTs of Amiga stuff, he literally knows where some of the bodies were buried on the Amiga sw."

For those who haven't kept up, the video interview became a telephone interview. That 1 hour, 40 minute telephone interview is now on-line! It is broken up into 4 parts of about 6-8 megs each (22-30 minutes for each part). Go to http://bios.tx0.org/ in order to download the .mp3's. On the interview you will hear Bil, Andy, myself, Cameron Kaiser, and Keith Henrickson.

My thanks to Bil and Andy for giving a most enlightening look into Commodore Business Machines, and thanks to all who came up with questions for the two engineers.

Good stuff!
Robert Bernardo
Fresno Commodore User Group
http://videocam.net.au/fcug

### SCACOM
### Aktuell issue 7

German magazine SCACOM Aktuell issue 7 is available for Download and to read Online at www.scacom-aktuell.de.vu.

Also a special Gold Quest 4 magazine is included and 2 new background images. You can read 48 pages of content and additionally 22 pages about "1 year of SCACOM".

SCACOM Aktuell Team
www.scacom.de.vu

### REU PROGRAMMING REFERENCE

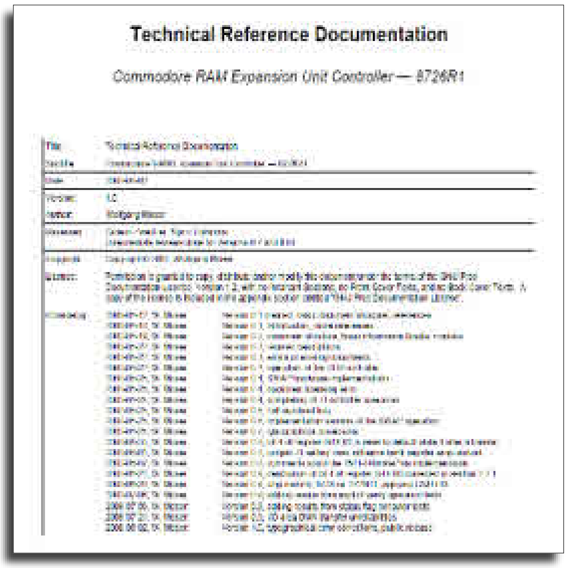
Wolfgang Moser
in the past three months I have written a technical reference guide about the Commodore REU's controller chip, the CSG8726R1. The main intention of the document was to help emulation authors improving their REU compatibility. Actually I recieved feedback from Gideon Zweijtzer and Spiro Trikaliotis who were able to improve the REU compatibility of the 1541-Ultimate 1.1 and VICE 2.0 from early pre-release versions of the documentation. Bo Zimmerman was so nice to put an archive up to the "Funet" archive (now hosted at zimmers.net), it consists of the mastered document in OpenOffice format, the published format as PDF and also a readme file. The whole document is covered by the GNU Free Documentation License:

Look it up at:
http://zimmers.net/anonftp/pub/cbm/documents/chipdata/CSG8726TechRefDoc-1.0.zip

**These files are contained in the download:**
- CSG8726R1-TechnicalReference-1.0.pdf
- CSG8726R1-TechnicalReference-044.odt

The article consists of a technical description and discussion of the hardware register interfaces of the Commodore RAM Expansion Unit (REU). This technical documentation tries to describe the hardware implementation aspects of the 8726R1 controller chip of the Commodore REU as they show up to a programmer.

Sincerly yours,
Womo



Technical Reference Documentation

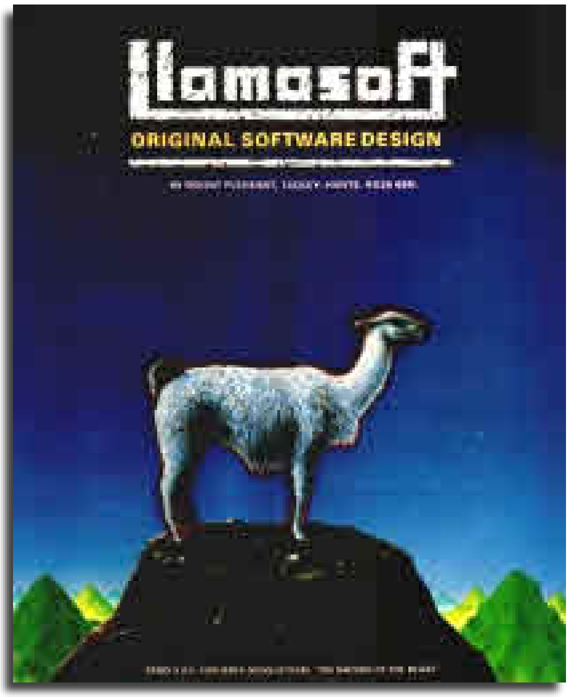
*Commodore RAM Expansion Unit Controller — 8726R1*

[illegible]

# LLAMASOFT - The Dromedary Years (Part One) Article Written By Michael Bevan

In 1980 the then 18 year old Jeff Minter was still in unknown quantity in the fledgling world of home computer gaming. Within ten years Minter had founded his own company 'Llamasoft', unleashed a catalogue of over twenty classic 8-bit games and the term 'Minter-esque' had entered gaming lexicon to describe his highly individual, uniquely psychedelic and indeed exceptionally hairy style of game design.

The seeds of Llamasoft were sowed after Minter left secondary school to go to a sixth-form college in Basingstoke. Still unclear what he intended to study, and with his best O-level results having been in English, physics and mathematics, the hand of fate intervened in the shape of Minter's head form teacher, who suggested an extra-curricular Computer Studies course, in addition to A-levels in his strongest three subjects. Minter, whose vague knowledge of computing had previously amounted to little more than what he had read in Issac Asimov science fiction novels like 'I-Robot' and had hardly even clapped eyes on a computer terminal before, signed up. Although at first frustrated with the content and laboriously slow teaching methods of the Computer Studies course, an encounter with his brother's newly acquired home Pong machine, along with Breakout and Space Invaders at a local travelling fair stimulated the young Minter's imagination. Knowing there was suddenly potential in the computing field to interest him he did not as yet however quite make the connection between the very limited Commodore PET machine he was using at college and this brave new world of video gaming opening up around him. Jeff recalls:



'Nobody had ever really heard of programming video games back then. You had arcade games, which people knew about, Space Invaders and Asteroids, that kind of thing, and in the home things like Pong and the VCS, but the idea that someone could just sit down and create their own games was completely alien'.

Things changed when Minter one day walked into his college computer room to see something interesting. A fellow student was sat at the Commodore machine moving a crude blob around the screen by pressing the keypad. Asking how he had achieved this, the reply would change Minter's life forever. The student simply said he had 'typed it in'. Grabbing a bunch of programming manuals, Minter quickly learned PET BASIC, followed by Assembly language and began spending every lunch-time in the computer room hunched at the screen, attempting, somewhat crudely, to replicate the arcade games he had seen.

'At college we had one Commodore Pet between the whole college and I think only about four geeks in the whole place who spent their time in the computer room making these weird little games. We never thought for an instance that there would ever be a market for this stuff, it was just something we did.'

Hooked as he was, Minter was still just a hobbyist and when the time came to leave college the 'weird little games' had to be put aside when the hairy one enrolled on a full-time university course. Things did not go completely as planned however as he soon found himself struggling with his course, and hankering for the creativity he had pursued back during all those college lunch hours.

'I left college and went to university and I was very bored at university because I was on a maths and physics course, and I wasn't interested in maths and physics, I was interested in playing with computers. So in due course I got kicked out. And after I got kicked out I was supposed to be going back to a polytechnic and I spent three months there and then I got quite seriously ill. While I was ill I thought.. sod it.. I'm gonna start just making games because it just so happens that in the intervening time Uncle Clive had just brought out the ZX80 and the ZX81 and Commodore had brought over the Vic20 and so this arcane stuff which we'd been doing in college, just the four of us, suddenly opened up and I wondered whether, given that I couldn't really do anything else because I was ill, I could actually write some games and see if I could sell them.'

Thanks to his parents recent gift, a brand new Vic20, the fruits of this decision would include a conversion of the arcade/puzzle game Deflex (originally dreamt up on the that old college PET), a 'City Bomber' clone, a 'Breakout' clone, and Rox, an odd Asteroids-meets-Missile Command hybrid, all written in Commodore BASIC. Although crude looking now, Minter was becoming increasingly adept at programming and was producing titles that actually bettered some of the games doing the commercial rounds in that era. Rox, particularly was inspired by a desire to improve on a woeful third-party Asteroids clone for the ZX81 he had played and indeed, misguidedly, purchased.

'Rox, was a very simple game, but was something a bit special to me. I'd made those other few games on the Vic, and they were fine, but were all simple and all things that had been made before. Rox was something else, a little game that I made up myself, redefining the VIC characters to create a lunar surface upon which a little lander descended, and which was then beset by falling meteors. I made it look nicer than any game I'd made before, designing a custom character set so that even the text appeared in a futuristic style, and adding a proper hi-score table and everything'.

Thankfully recovering from illness, Minter set about the task of actually trying to market some of the games he had created. Previously a visit to a Sinclair hobbyist 'Micro Fair' in London had resulted in an expression of interest in some of his ZX81 games from specialist publisher DKTronics. Keen to capitalise on this interest Minter had expended much effort to produce a ZX81 version of Atari's Centipede which was unique on the machine in being able to display modified hi-res graphics instead of the Sinclair machine's standard character set. Unfortunately, when DKTronics released both Centipede and Minter's ZX81 hi-res ROM program they conveniently 'forgot' to pay him any royalties instead only handing over a derisory one-off sum. Frustrated and upset by such underhand business tactics, Minter vowed to start a proper business of his own, one which would cut out the middle-men and put him back in control of the distribution of his creations on the new Commodore machine. The genesis of the famous company name would evolve from a seemingly random event during a programming session on his beloved Vic20.

'One day I was home alone one afternoon, lying on the floor idly playing with the graphic design tool, and thinking about animals. I pushed the cursor round and drew the outline of a llama. I sat and swigged my tea, looking at the llama, and then, on a whim, directed the cursor to some empty space below the image of the beast, and wrote there, in tiny letters, the word Llamasoft!'

And thus the legendary Llamasoft was born. Minter decided it would be wise to bring in a partner to help him launch his new company, and teamed up with Richard Jones, one of the PET 'geeks' he had known at college. Jones was to run the business side of the company allowing Minter to concentrate fully on developing and creating new games. Initially distribution was to be by mail order adverts in the computer press and at computer shows rather than through high street vendors. Although he had a few titles including Rox, and a newer but fairly simple game called Headbanger's Heaven already completed for the Vic, it was obvious that better, more advanced titles needed to be created if Llamasoft were to make an impression with the games-playing public. As a diversion before starting work on designing these more demanding launch titles, Minter amused himself by revisiting his old 'City Bomber' game in a slightly misguided attempt to get his brand new company noticed in the outside world.

'City Bomber' was in the early computing community a generic title for a very basic arcade-style game in which a player controlled a plane descending over a blocky cityscape. The sole player input, a 'Bomb' button, caused a single bomb to drop from the plane and destroy a portion of the buildings below. The object of the game was to destroy the buildings so the plane could land without crashing into them on its descent. Minter took his old code for his own version, added a graphic for a waving Argentinean flag, programmed a crude version of Rule Britannia for the game's soundtrack, and called the game 'Bomb Buenos Aires!' This was 1982, the year of the Falklands conflict, and Minter had intended the game as a bit of a daft joke to parody the fervent and over-the-top resurgence of patriotism the country was experiencing. It was a plan that back-fired spectacularly and Llamasoft ended up having to apologise to the Press Complaints Commission after adverts for the game in computing magazines had drawn the ire of publications such as The Daily Telegraph.

Minter, sensibly, returned to creating original and non-controversial games. Inspired after seeing Eugene Jarvis' classic Defender for the first time to create a rudimentary clone, the resulting Defenda premiered at the Commodore Show in London later that year, with Minter producing several hundred copies of the game by hand in the hope of selling them to interested punters. As Jeff recalls:

'Defenda, as I called it, was a bit shit. However, back then, everything was a bit shit, and there was the possibility that Defenda might be marginally less shit than some of the other games out there. And games in machine code for the Vic were rare.'

Indeed given the call for fast, assembly language Vic games, Minter managed to shift most of his stock of Defenda at the show. The best thing to come out of the event for Llamasoft though was the interest of a representative of a US company called Human Engineered Software. HES was looking to get into the games distribution business in the States and was looking for product. Defenda fitted the bill extremely well. If Minter was able to provide HES with the game on ROM for the US market there was a great possibility of a not insignificant financial benefit for Llamasoft from the deal. Although now wary of third-party distributors, Minter was keen to follow through. Unfortunately the relationship with business-partner Jones had gone into decline. As the so called business 'guru' of the company Jones was proposing that profits should be split 70/30 in his favour. Minter, justifiably was not impressed by this logic, being the sole creator of the company's product. An inevitable split followed, with Jones leaving Llamasoft to form Interceptor Micros. A tense rivalry between these two companies would continue throughout the rest of the decade until Interceptors' demise in 1990.

In the US there was also a small problem for the proposed distribution of Defenda - Atari. Word was out that they were starting to get fed up with third party unlicensed versions of arcade games that they owned the rights to and were possibly about to take the matter to the courts. In this environment there was no question of the game being released in its current form. In the UK the game had been re-named Andes Attack in an effort to no-so-blatantly rip of Defender. Later, after a visit to see The Empire Strikes Back, Minter hit on the idea of changing the game's graphics and tweaking the game-play in an attempt at recreating Empire's Hoth battle-scenes, replacing the film's AT-AT walkers with the next best thing that came to mind.. camels. The resulting Attack of the Mutant Camels or AMC is one of Llamasoft's most memorable releases of the Vic 8-bit era.

The biggest success of Minter's fledgling career was just around the corner however. At the time Minter was frequenting a lot of trade events and shows touting his growing Llamasoft catalogue and thus spending a lot of time around the UK capital.

'After each day's show was over we'd sometimes head off into London to eat or to go to the arcades in the West End, and at that time the excellent movie 'Blade Runner' was just out in cinemas and the walls of the Tube had many posters up advertising it. I hadn't yet seen the movie myself but the posters were striking. The name of the film and the typeface of the font looked interesting... and for some reason I was thinking about the next game I wanted to do, which I wanted to be another unexpanded Vic game but wasn't sure what it would be like at that point. And while staring at one of the Blade Runner posters waiting for the Tube, the name 'Grid Runner' popped into my head'.

Minter had been toying with the idea of updating the Centipede concept for a while, and decided to start work on a new game design implementing his ideas. Updating the game's background to a futuristic-looking grid setting and creating new enemies such as the infamous 'grid zapper' while also increasing the frenetic pace of the original, Minter named his modest new creation Grid Runner, and mailed off a copy to HES in the US awaiting a verdict on what he thought was a good, if not particularly commercially viable game. What happened next could never have occurred to him in his wildest imagination...

Sources: 'The History of LLamasoft' (www.llamasoft.co.uk), Google Tech Talk - 19.03.07

Continued Next Month

Many thanks to Retro Gamer CD www.rgcd.co.uk and to Michael Bevan for agreeing to the reprint of this article

TND
the new dimension

# Music Scene C=64 DMC Tutorial

## Getting Started with DMC 5

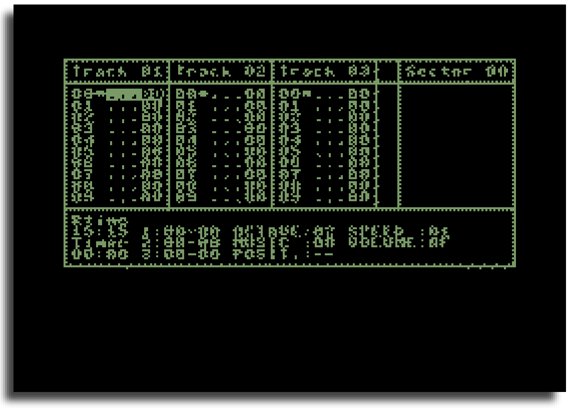
**by Richard Bayliss Special thanks to Rio / Rattenrudel for updating**

**DOWNLOAD DMC Music Editors**
**(Pack contains DMC V2.1 (GRAFFITY), DMC V4.0 (GRAFFITY) + Docs (By Richard), DMC V5.0, DMC V5.0+ (CREAMD/C64.SK), DMC V5 PACKER, DMC V5.0 SCANNER and DMC V7.0 (GRAFFITY+UNREAL)**

**www.redesign.sk/tnd64/DMC%20Music%20Editors%5BTND%5D.zip**

**Getting Started with DMC 5**

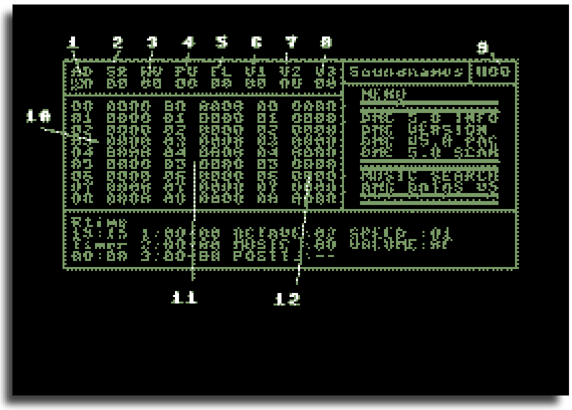
**Introduction**
This tutorial goes through the use of DMC 5.
Compared to DMC 4, DMC 5 is a completely different music editor and is a whole lot more difficult to learn. Compared to DMC 4, the composer looks does really basic, but to tell you the truth, it's looks are deceptive though because It has a whole lot more put into the application



As with DMC 4, there are no instruments ready made, so we are going to have to program these ourselves, however before we do, we'll take a look at the DMC 5 editor itself. Take a look at the screen below and you'll see what I'm talking about. If you understanding the whole manual of DMC 4, you will get on very easily with DMC 5. Remember, both editors have their own advantages and disadvantages.

As you can see, this is the main environment of the DMC composer. And it does look pretty dull doesn't it? Well don't worry, it's a very good composer ;o) You are able to download an updated version of DMC 5, from the Friends Downloads page, the software is distributed as Public Domain and is allowed to be spread, or copied freely. Some basic instructions are included on how to get yourself started with this editor (mainly the controls).

**3.2 The Sound Editor**
Before you try and get composing music, we'll need to do first use the Sound Editor to create our instruments. This isn`t an easy process; you would be required to have some sort of programming knowledge to work these sounds 100%. Below you'll see a screen shot and the



names for the parts of the sound editor. The size of this image is pretty meaty, but parts of the editor have been marked with white numbers, so I could indicate to you what these functions are and what purpose they serve.

1. Attack/Decay: (AD)
This controls the default volume and duration of your instrument.

2. Sustain/Release (SR)
A second control, which is to set a volume and longer, continuous duration of your sound.

3. Waveform (WV)
This function reads from the wave form table, to form instruments.

4. Pulse (PU)
A special function, which creates pulse sounds according to table, for when you use $4x or $5x as a sound inside the Wave form editor.

5. Filter (FL)
A special function which allows you to add a synthesized filter into your sounds. Please note that these will not work unless you set FLT.XX with/without FRQ.XX in the sector editor.

6. Vibrato #1 (V1)
Controls the vibrato delay.

7. Vibrato #2 (V2)
Same again, but this time with another vibrato.

8. Vibrato #3 (V3)
A special vibrato to create various modes.

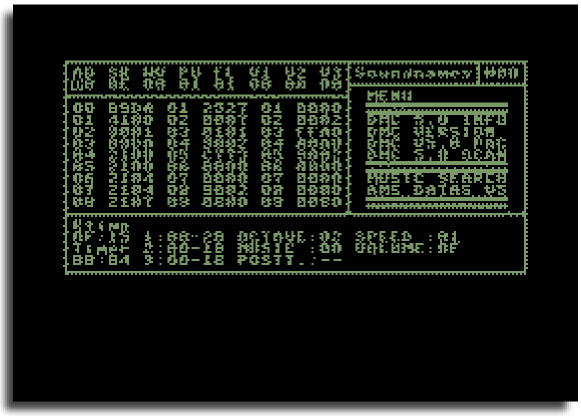
9. Sound Number (#000)
This indicates the instrument you are currently creating.

10. Waveform Table
A table where you program your own sounds.

11. Pulse Table
A table where you program your own pulse.

12. Filter Table
Where you program your filters.

13. Sound names (Not numbered)



If or however, you wanted to store your own instruments data then you can load and save all your sounds from the disk.

Now that's all the functions explained, so we are now going to create a few instruments.

Before we do actually create any instruments, I would advise you to read the section about Waveforms in Wavetable in DMC 4 manual. Same types will be used in the DMC 5 wavetable editor also.

Right, now we have these set up, we shall make some of our own instruments, although there are some sampled sounds supplied with DMC V5.0+, it would be very lame to use another person's instruments it is far better to create your own. We will make our first sound which is the main lead instrument. So copy the following from the diagram.

You may think 'Aaaah you're doing the same thing', but that is where you are wrong, because the waveform is pointing at the correct position of the wave table. The pulse is not required, as the C64 recognizes it as a saw tooth. And we still use $01 as the filter. What's happening here is that we're calling the right wave forms.

Please remember that there's more that meets the eye, when producing sounds and music using the DMC V5.0 player. Later on we will be showing you how to do filtered voices, but first, we'll work on the main sounds. We'll try and create a minor chord, using saw tooth. Where we were last left from, go to the next free sound (Where it's all blank) and then enter the following example below:

```
AD SR WV PU FL V1 V2 V3
0C 00 04 00 00 00 00 00
```

Now move your cursor underneath WV (Which represents the wave table) and then enter the wave table, by pressing the SHIFT button and RETURN. Now edit the wave table as below:

```
04 2100
05 2100
06 2103
07 2103
08 2107
09 2107
0A 9004
```

You probably wonder why I used 00,03,07 beside the wave form number, 21. This is because 21 represents the saw tooth and $00,$03,$07 represent the octave to make a minor chord. Using 9004 will make the sound jump back at $04 so we have a continuous minor chord. Which you should be familiar with in the future.

Our next step will be to create a major chord. So go to the next blank sound table make everything the same, except the WV, and put 0B (not zero eight) on to the wave table. We are now going to make a major chord. Follow this example below:

```
0B 2100
0C 2100
0D 2104
0E 2104
0F 2107
```

```
10 2107
11 900B
```

There we are, simple enough. We have created a major chord, using $00,$04,$07 as the octaves. Try experimenting more with the octaves beside the wave forms, you'll end up with some cool effects. I promise :)

3.3 Creating Your Own Drum Kit

After getting used to various sounds, we are going to start to go a little further into the music making scene. Before we actually compose any sort of music, using DMC 5 player, you cannot be without certain sounds. The next sounds which we are going to create is our own drum kit. As usual, in the sound editor, let's find the next blank sound, indicate attack decay (AD) as $08 and sustain release (RS) as $00, indicate the wave form table to jump to $12 and pulse table (PU) to jump to $05 and then follow the example below.

The Wave Table:

```
12 89FF
13 490B
14 4909
15 0900
16 9015
```

This was for our first drum. As you notice, we mix our sounds, to make it C64 drum like. At first you will find this task difficult, but it will get even more simpler once you have got used to this. Now lets create our own pulse table.

```
05 0800
06 0000
07 0000
08 9006
```

Here we are, our first drum. We have another drum to create, which should be simple enough. Find the next blank sound and then set the wave form to jump to $17 on the wave table, copy the pulse table as above, set the attack/decay as $09. Now follow this example for our wave table.

```
17 89FF
18 490F
19 81FD
1A 490D
1B 490B
1C 89FF
1D 901C
```

That's our second drum complete. We are nearly ready to get started using the track and sequence editors for composing music, but before you get started, try and experiment more creating sounds, etc. If you want to create your own filter tables, then use the same method with the pulse tables, and set up your own filters for your sounds. These filters wont work unless you set the filter in DMC 5's sequence editor, but we will be taking a look at this later on.

3.4 Making our first piece of music

If you remember taking a look at the DMC 4 tutorial, you would have familiarized yourself with the track editor. DMC 5's track editor is -exactly- the same. No instructions really needed to on how to use the track editor, except for use SHIFT+RETURN to enter the sector editor. You probably have the instructions with DMC 5 on how to access these editors. Now let's do a small piece of music, to get you started. Go to sector $00 and enter the following:

SECTOR 00:

```
00 DUR.03
02 SND.01
04 D-3
05 D-4
06 D-4
07 D-3
08 D-4
09 D-4
0A D-3
0B D-4
0C END
```

Now let's edit sector $01: this is where we have the drums and bass. xx will represent drum 1 and yy will represent drum SECTOR 01:

```
00 DUR.03
02 SND.xx
04 D-2
05 SND.00
07 D-2
08 D-3
09 D-2
0A SND.yy
0C D-2
0D SND.00
0F D-2
10 D-3
11 D-2
12 END
```

Now we have two sample, we'll leave the third track blank, so you can experiment with the sequence editor. This tutorial is only for DMC V5.0 beginners and will be updated later on, probably in a couple of month's time.

Now let's play with the track editor and make the two sequences sound like music.

```
TRACK1   TRACK 2   TRACK 3
00 TR+00  00 TR+00  00 -STOP-
02 --- 01 02 --- 00
03 --- 01 03 --- 00
04 --- 01 04 --- 00
05 --- 01 05 --- 00
06 TR+03  06 TR+03
08 --- 01 07 --- 00
09 --- 01 09 --- 00
0A --- 01 0A --- 00
0B --- 01 0B --- 00
0C TR-02  0C TR+02
0E --- 01 0E --- 00
0F --- 01 0F --- 00
10 --- 01 10 --- 00
11 --- 01 11 --- 00
12 TR+05  12 TR+05
14 --- 01 14 --- 00
15 --- 01 15 --- 00
16 --- 01 16 --- 00
17 --- 01 17 --- 00
18 END!  18 END!
```

19 --- 00 19 --- 00

There you go, a 2 channel piece of music. I was going to add more, but there really is no point, because this is a tutorial to get you all started using DMC. Anyway, here's some more interesting things before I leave you experimenting using DMC 5.

3.5 Other Bitz!
You have understood the basic bits using DMC 5. Now we shall go into the harder parts on using the DMC music editor 5. I'll show you how to use a music packer later on, but first of all let's familiarize ourselves with the sector editor a little more. You may notice that we only used simple commands, but you can also use other commands, such as VOL.xx, which is volume (Shift+V), SWITCH (Shift+X) ADSR.xx (Shift+Y), FILT.xy (Shift+F), FREQ.xx (Shift+Q) and -GATE- (£), GLD.xx (SHIFT+G), SLD.xx (SHIFT+S).

When you add a filter to your sounds, you must use track 3 for the particular sound to work a lot more properly. What you do is add the sound no, read from the sound table, then you at FLT.xy. You may wonder why I added xy. This is because 'x' represents the filter type (use 1,3,4 or 5) and 'y' represents the volume of the filtered sound. After using filter command, you can add a note. Also when using filtered sounds, you can add a frequency filter (FREQ.xx). You can use any value for the frequency sounds. ADSR can also do cool effects in your sounds.

To stop a note from playing, we use the -GATE- command. You can use -SWITCH- to add some effect to your sequence. You can also use the glide and slide commands (GLD+SLD) to make your note slide.

When you are composing music, you have to be extremely careful, because if the duration of your sounds do not calculate up to the same value, your song will go out of tune.

3.6 Packing your Music
There are two different sorts of packers for DMC 5. There is the original packer by Graffity, or the DMC 5 packer by Iceball/Motiv8. These two packers will raw DMC music files and then will pack these to a smaller size (just like an ordinary cruncher and packer). Don't attempt to use a cruncher or different packer for these tunes, else you'll just get a load of garbage in your machine code memory. If you use Action Replay, then I recommend that you use the GRAFFITY version of the packer, else if not. Then you can use Iceball's DMC 5 packer instead.

To use GRAFFITY packer, using Fastload or BASIC, load the packer like you normally load a normal C64 disk program. Next load your own music file like you do in BASIC, then enter M/C monitor and type G $2E00. Once done, wait a few secs then enter the address where you want to load your packed music. I recommend you use $1000 as start address so you can use the tune on various PD utilities, such as Old school Demo Makers, Note Writers, etc. Once done, your tune will relocate and pack to the address. Now save the tune, and play it (init $1000, play $1003) in your own programs or PD utilities ;).

To use Iceball's DMC 5 packer, load from BASIC, enter load name and then enter the address to pack to ($1000), wait and then save. Easy. That ends our DMC 5 music composing tutorial ;o). Maybe there will be more in the future.

## DMC 5 Commands

**THE EDITOR**

| | |
|---|---|
| SHIFT + Q | : ENTER ANALYZER |
| SHIFT + O | : DOS COMMAND |
| SHIFT + P | : DISPLAY STATUS |
| SHIFT + S | : ENTER SOUND MENU |
| SHIFT + D | : ENTER DISC MENU |
| SHIFT + C | : CHANGE COLOR |
| SHIFT + V | : MUSIC SETUP |
| SHIFT + M | : CHANGE SOUNDTRACK |
| SHIFT + RETURN | : ENTER SECTOR |
| ARROW UP | : COPY TRACK TO BUFFER |
| SHIFT + ALPHAZ. | : COPY BUFFER TO TRACK |
| SHIFT + E | : PUTS "END" |
| COMM. + E | : PUTS "STOP" |
| + | : PUTS "TR+00" (TRANSPOSE UP) |
| - | : PUTS "TR-00" (TRANSPOSE DOWN) |
| CLR | : CLEAR TRACK |
| HOME | : PUTS CURSOR AT POSITION 00 |

**THE SOUND EDITOR**

| | |
|---|---|
| SHIFT + R | : READ DIRECTORY |
| SHIFT + L | : LOAD SOUND |
| SHIFT + S | : SAVE SOUND |
| SHIFT + RETURN | : ENTER TABLE (AT WV,PU,FL) |
| +/- | : NEXT/PREVIOUS SOUND |
| CLR | : CLEAR SOUND, BUT NOT THE TABLE-DATA |

**THE SECTOR EDITOR**

| | |
|---|---|
| DUR (SHIFT + D) | SET THE DURATION OF A NOTE. |
| SND (SHIFT + S) | SET THE SOUNDNUMBER ($00-$1F) |
| FD+ (SHIFT + +) | FADE IN , THE VALUE IS THE SPEED (LOW - SLOW, HIGH - FAST) |
| FD- (SHIFT + -) | FADE OUT |
| GLD (SHIFT + G) | GLIDES NOTE |
| SLD (SHIFT + H) | SLIDE A NOTE, |
| ADR (SHIFT + A) | SET AD-REGISTER, YOU SHOULD USE THIS |
| SRR (SHIFT + Z) | SET SR-REGISTER, |
| FRQ (SHIFT + Q | SET THE BASIC FILTER-FREQUENCY, |
| FLT (SHIFT + F) | SET THE FILTERTYPE AND RESONANCE. |
| VOL (SHIFT + V) | SET THE VOLUME OF AN INSTRUMENT. |
| GATE (POUND) | RESET THE GATE-BIT |
| SWITCH (SHIFT + X) T | DISABLES THE HARD-RESTART, |
| END (=) | END SECTOR |

**EDITOR-COMMANDS**

| | |
|---|---|
| SHIFT + , | : TRANSPOSE 1 NOTE UP |
| SHIFT + . | : TRANSPOSE 1 NOTE DOWN |
| COMM. + + | : SWITCH TO NEXT SECTOR |
| COMM. + - | : SWITCH TO PREVIOUS SECTOR |
| 0-7 ON NOTE | : SET OCTAVE |
| 0-F ON COMMAND | : SET VALUE |
| AWSEDFTGYH... | : NOTES |
| ARROW UP | : COPY SECTOR TO BUFFER |
| SHIFT + ALFAZ. | : COPY BUFFER TO SECTOR |
| HOME | : GO TO TOP POSITION |
| CLR | : GO TO "END" |
| COMM + HOME | : CLEAR SECTOR |
| RETURN | : LEAVE SECTOR-EDITOR |

**ALWAYS ACTIVE KEYS**
F1 - PLAY MUSIC
F3 - STOP PLAYING

# In the Beginning Part 8
# \Lord Ronin from Q-Link\

Personal part here, I did say before that this is a personal look on the C=. One of the first programmes that I ever typed in was a game very similar to this upcoming guessing game. What I did was mess about with the colours and the word print statements. According to some of the big time programmers on the different Internet C= lists. that makes me a none programmer. Agree or not, the point of this story is that you should feel free to play with the programme, you are not going to hurt the C=.

```
1 rem number guessing game
2 ?"(shifted clear home)"
5 input"enter upperlimit for guess";li
10 nm=int(li*rnd(1))+1
15 cn=0
20 ?"i've got the number.":?
30 ?"what's your guess";gu
35 cn=cn+1
40 ifgu>nmthen?"my number is lower":?:goto30
50 ifgu<nmthen?"my nunmber is higher:?:goto30
60 ?"great! you got my number"
65 ?"in only ";cn;"guesses.":?
70 ?"do you want to try another(y/n)";
80 getan$:ifan$=""then80
90 ifan$="y"then2
100 ifan$<>"n"then70
110 end
```

Oh wow man that is a mess of meaningless drenn; lets try to take it apart. First thought some information about the use of spaces. The Commodore book shows the listing with spaces separating the individual parts or commands, Spaces reduce the amount of commands that can be entered on each line. Spaces also take up a byte or so in memory use, so no spaces help tighten the programme, and make it much harder for you and me to see the different parts. So lets go line by line, I'll suspect that you have typed the program in and run it already.

Line 1
the rem statement, This tells you what the programme is about for this listing, rem is REMARK and can be anything you like the programme just ignores the command. Here they are using Rem to show the title of the programme. I'm a lazy person and so don't do it myself, although it is good practice.

Line 2
The print command with the shifted clear home, the screen is clean up and putts the cursor at the top left.

Line 5
That input statement we covered a little bit earlier, you can see what will be written to screen by the text in quotes. Just like a regular print statement. Here what ever you type in will become the variable li. That li variable pops up again in line 10.

Line 10
The random number generator, not much different to our previous dealing with this command; Here we don't have a specific number locked into the formula. We are using the 0 variable li.

Line 20
The print statement that just writes to the screen it is ready.

Line 30
Input line for you put enter your number, this becomes the variable gu.

Line 35
adds one to the count.

Line 40
Takes your number as the variable gu, compares it to the generated number and sees if it is greater than the generated number. Programme commands to do that read if gu > than nm. Short and easy for the computer but a whole new language for us. Second part of the line is the print statement to tell you that the number is lower than your guess.

Line 50
This is the reverse of line 40, both lines end with the programme taking you back to line 30, where you are doing the guess and the count of your guesses is being recorded.

Line 60
This is the line that gets to some people it looks wrong to them, I have been over this line many a times with users in our group. There are only three things that can happen, your number is higher or lower or right on Ie you guessed correctly. Line 40 & 50 cover the high and low part. Since the computer goes from the lowest line number to the highest <unless something moves it elsewhere> If line 40 isn' true, then the program goes to line 50, which if it isn't true, Well there is only one other option, that is you

guessed correctly. That is why this is after the other two possibilities.

Line 65
prints out text and as you see there is a break in the text where between the ; symbols there is the cn variable. This variable will be printed out as the number on the screen. Then more text appears, OK that print thing after the : symbol will do that drop down a line down again trick.

Line 70
Goes with the part to play again, Here the (y/n) is part of the print statement. As you see in lines 80-100 where the prg is checking to see your keypress from the KB. Line 80 keeps the programme stuck at that point, until there is a keypress, Line 90 gets the y from the keyboard and starts the game over again. Line 100 is expecting the n and if it doesn't get it, remember that if a y was typed we won't be at line 100. So if the input wasn't an n the Programme goes back to line 70. Putting up the text another try. This is an error trap, just making certain that the user has only the two listed options.

Go ahead and play with the text line numbers that you insert. Mess about with screen, border and even the different text colours. We have a bit more on this one. And Idid say that I would cover that weird part.

Well it is easier to see the programme functions as you play with it and then alter it a bit. However the book wants you to figure out how to set up a lower limit, now you can fight with this idea for a while, bash your head against the wall and the keyboard, then send me and e-mail that says I have done this better than the manual. <Stand by, having a laughing fit at that idea> Well that is the feeling every student in the user group had at this part, You see they want you to go back in the book to that weird line, the line that makes no sense as it is written.

Read in the book on page 50 as follows, and on one line.

```
nm=int(lowerlimit+(upper-lower+1)*rnd(1))
```

OK we can see from the guessing game that nm would be the variable, the rest almost appears to make sense.
You will need to recreate a new input line, where you ask for the lower limit; I made this the variable lo. So now you need to write that random number generator line, something like this should work.

```
nm=int(lo+(li-lo+1)*rnd(1))
```

A bit more understanding, but maybe if I substitute the variables into a set of numbers, you can see it better, I had to do that for members of the group.

```
nm=int(50(100-50+1)*rnd(1))
```

If you want to see how this works out for you, there is something I haven't really said yet. You can print out the value of nm, all you need to do is type on the screen just after you have written the above line press return and dont put this in the programme you are working upon.

```
? nm <press return>
```

The value will be printed right underneath the nm part you typed, right cursor up to the random generator line, press return. Go to back to the ? nm line and then press return key, now you get a different number each time.Right back to the guessing game programme if you didn't do it already well it is too late to save the programme. List it and see if it is still there and if so what was damaged,

Yeah I am tricking you. <VBG> I'll let you figure out where to place the input line for the lower limit, as to where the random number generator line should be. <SEG> Next type programme is small and I would not include it at this time but there is a trick on the user from the user book, and I want you to work with colour, the Book read as follows.

```
5 ? "Care to try your luck?"
10 ?"RED DICE  =";int(6*rnd(1))+1
20 ?"WHITE DICE  =";int(6*rnd(1))+1
30 ?"HIT SPACE BAR FOR ANOTHER ROLL":?
40 GETA$:IFA$=""then40
50 IFA$=CHR$(32)then10
```

Right save for the tightening of the programme lines, this is the way that the book has you type it in, So where are the tricks to you? Well first one that I saw was using the plural of DICE rather than the singular term which is DIE. You know one DIE two DICE, But that probably isn't the first thing that caught your eye. Line 5 is in lower case, that isn't my programme style or the way I typed out the line, that is the way it is written in the book on page 52.

I'm guessing that they wanted to automatically see it and do the C= and shift key press to move to lower case, but type the other lines in upper case. Did you notice that you aren't told to "new" the programme? If you didn't then the programme we typed in before would still be in the computer memory, makes the programme run crazy, that is if it will run at all.

What I want you to do is make the screen black and the border black, also I want you to make the words RED DICE <or for the purist DIE> as red and the words WHITE DICE as white, tough assignment isn't it? Well yes it is for those in the group that didn't take notes, you have this drivel in a form that you can use to read back a bit. on disk or on paper. However that was a few lessons ago and I will be here to lend a hand.

Written in lower case it looks like.

```
pO53280,0:pO53281,0
```

These 2 command make the screen and the border go black. What you need to do is think where in the programme the commands should be placed, Hint, the programme thinks in a straight line, from low line numbers to higher numbers.

How to make the words in different colours? Well this is hard, I mean really hard, after making the quote you press control key and the number 3 key. Write the words and close the quote and do the rest of the line. Same for the words white, although you make the quote and then press control key the number 2 for white.

Of course if you have some experience with this sort of thing, you may now want to try to have the number come out as a different colour. Thing here is to see that, what ever colour you last used will be the colour that will continue to be used, until you change it. You can change the colour with the command of pO646,X where X is a colour from 0-15. Or PRINT"<control and a number key" Or PRINT"<c= key and number key". Actually on a programme line it would look a little like ?" ". There will be a symbol between the quotes, call this for now a colour code. That will print as the colour for the text. Have fun playing around, and feel randy and see what you can do with reverse video.

New programme and type in

```
10 ?"<clear/home>"
20 ? chr$(205.5)+rnd(1))
40 goto20
```

Run that and see what happens on the screen.

If you have a mess of M and N show up, press shift and C= for upper case. Ah there is this weird maze looking thing that shows up, over and over and over again. The programme will run until you press that run/stop key. What is going on here? That was my question, all of that stuff on screen for just a few lines of programming.

We have already understood the first line as the shifted clear home to clear the screen. The Last line is the goto statement that loops the programme back again.

But all the action is created in line 20. But what is happening there? We can tell that there is a print command, we can see that this is going to print something to do with a sort of random number generator. A closer look at the screen shows that the things printed are the slash lines on the N and M keys. How doe two symbols appear with only one statement to print them? That is how it looks at first. Lets work things out a bit.

CHR$ stands for "character string", remember I said that there are 256, that is from 0-255, characters for the C=. Well each one of them has a character string, or a number. Real simple now, what is happening is that the chr$(205) is a one symbol. There is no chr$(205.5) the random number generator is basically making a number that will be between 205.5 and 206.5. chr$ will only accept the whole number. So you have a 50/50 chance of each of the two chr$ characters showing up and being printed on the screen, Rather fast for a 1mcps speed machine. <VBG>

Don't worry if my explanation of how that maze is being created doesn't make a lot of sense, I think that it is a bit too advanced for this level. It took me 7 times through the book working with others to even have a remote idea on how the process works. What the important point is that you can print the chr$ as well as the reverse. This is done as follows.

```
?asc("z")
```

Press return
90 will appear on the screen .

```
?chr$(92)
```

Press return
\ will appear on the screen .

Try a few of these on your own, there are books, reference cards and other places where the lists of the characters and codes are already listed. Still it is a way to have a little fun and get more of a feel for the system. A way to see the actual character from the keyboard as well as finding the ascii value. OK here we have a time problem, I quote from the book. "As you probably figured out, 'ASC' is another function, which returns the standard 'ASCII' code type for the character you typed."

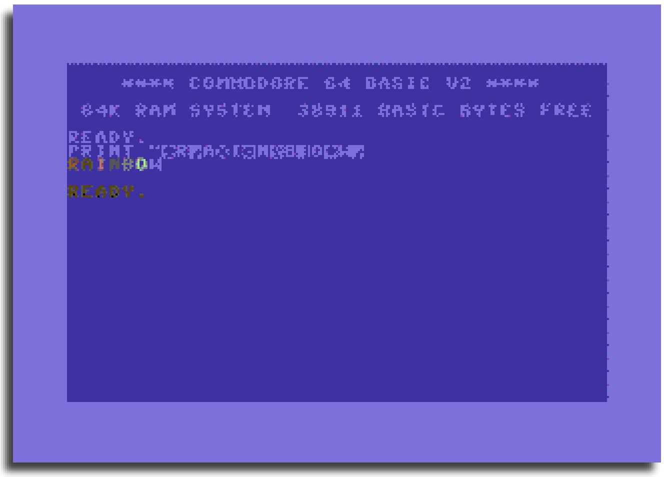
Early on in this collection of drivel I said something about the characters ascii and PETascii. Ascii and PETascii are different animals, lets not worry what the initials stand for in this case, only that there is a gigantic difference. Saying that we come to the point of what is really "Standard". Because, obviously there isn't a "real" standard, making a ruling now, PETascii is the Commodore standard and normal ascii is the standard used by the IBM pc! Clear for that, PETascii is what we are learning about, so it must be the only standard we need right? And yes I am biased towards the C=.

Going by the book we just moved into Chapter 5 "Advanced Colour and Graphic Commands" Lets jump right in, even with the personal thought that we aren't really that advanced. I mean I have never thought that I was that "Advanced" any time I had reached this point in the manual. Here they start you off telling you the colours for the control and the C= key, we already covered this a bit earlier on in the series. You remember the bit about using the Control key and the number keys to make the first 8 colours, then the C= key and the numbers for the next 8 colours. In order to illustrate this in the book there is a little one line programme they want you to enter and run.

```
10 ?"RAINBOW"
```

What they want you to do is change the colour for each letter by pressing the control and keys 1-7. Doing that in front of each letter and using the numbers in that order, yeah they want control and 1 before the R control and 7 before the W. Then run the programme and you will see the word printed in different coloured letters on screen.

The rest of page 57 is taken up with a chart that shows you the character gfx symbol that appears in a print



```
    **** COMMODORE 64 BASIC V2 ****

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
PRINT "[illegible]
RAINBOW

READY.
```

statements, all 16 of them. Sorry the writing tool I am using doesn't have all those gfx characters in the set, since I am using the upper and lower case letters.

You can experiment on that at this point if you don't have a list, creating your own list; OK I don't bother with that myself. Only look at the word on the number key for the colour and press that key in the print statement, but I am a lamer programmer and lazy to boot. At this point in the book and therefore the series of articles we are going to run into this symbol problem in different ways, Seen by this short program they have you type in.

```
New

10 ?chr$(147): rem clear home
20 ?chr$(30);"chr$ (30) changes me To ?"

Run
```

Well you don't have a chart of the character strings, do you? I'm working on the assumption that you don't, at least at this moment. I believe that there is a place online that has the manual for Down Loading. This list of codes are in the appendix in he back of the manual and in the Programmers Reference Guide.

As well as in some of the type in books I have seen. Those of you who do have these charts, they are the ones that are ladled "Screen Display Codes", and the other one is "Ascii and CHR$ codes. You are lucky and I'd like you to thumb to that section from time to time. Right now we are dealing with the "Ascii and CHR$ Codes" one.

OK back to that little programme.

Line 10
This gives you the CHR$ for the shifted clear home, Note that the number is always in the () symbols.
Well my level and understanding of programming has me looking at that and counting the spaces. Counting the print command as the ? symbol, that is 9 spaces taken up.
Using the ? and the shifted clear home in the quotes takes up 4 spaces. Seems this last style is more effective in spaces. More experienced programmers can tell me if I am right or wrong on this thought. I have seen the chr$(147) in type in programmes and substituted the shifted clear home, didn't appear to me to do anything different in the prg.

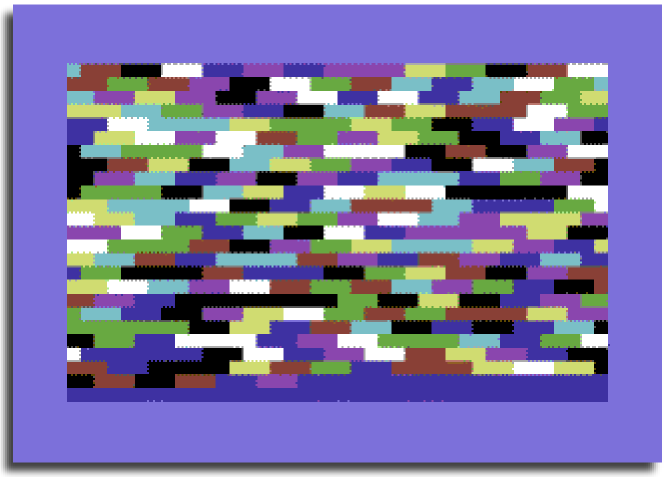
Line 20
This will print that character string and then the words in the quotes notice that there is nothing for you to do for a real answer, which is green, before I go to a complicated programme for you to type in. I should tell you something, a short hand way to write CHR$, which is cH. Well that is how it looks in lower case. Upper case would have the capital C and a vertical bar. Still have to type in the () things with the number between them though.

Next is a colour bar programme. I have a few things to say before you have a go at typing it into the C=. Running this will create a series of reverse video colour bars on the screen. They will run till you press the run/stop key, only does the first 8 colours of the commodore`s 16 and uses the chr$ codes. If you feel like experimenting and have the chart, feel free to expand this to the full16 colours. We have covered the rudiments for how to do that except for a command that is new to us at this point and I will get to that afterwards. I did make this as the 16 colours the tool that I use to test the colour chip and monitors when they come into my collection.

The final part is a lesson on editing; you will see that there are many lines here that are close to being the same. OK lines 40-110. I'll explain about editing, after you see the programme, but wait to type it in, and you will save some fingers.

Type NEW then press RETURN

```
1 rem automatice color bars
5 ?chr$(147)
10 ?chr$(18);"   ";
20 cl=int(8*rnd(1))+1
30 on cl goto 40,50,60,70,80,90,100,110
40 ?chr$(5);:goto10
50 ?chr$(28);:goto10
60 ?chr$(30);:goto10
70 ?chr$(31);:goto10
80 ?chr$(144);:goto10
90 ?chr$(156);:goto10
100 ?chr$(158);:goto10
110 ?chr$(159);:goto10
```



Before any explanation lets save your fingers editing time with the C=. Type in line 40 you can see how much it is the same as the other lines below it, so how to avoid typing all that drenn over and over again? Well after you type line 40.

Oh yeah make absolutely certain that you hit the return key [lightly of course] after you complete each line. Return takes you to just underneath the 4 in 40, cursor up to that 4, and now type a 5 over it. Right the line now reads 50, don't panic yet cursor over to the 5 in the () symbols. Use a shifted inst/del key this makes a space. Creating a two character area in between the () symbols. Type in the 2 and right over the 5 you see there type the 8. Now the line reads right for line 50. Press return and then cursor up, change the 5 to a 6. Type over the 28 making changing it to 30 press RETURN and you have line 60 done. Repeat this till you have all the lines completed.

Panic time? No not really, if you did it as above, then you will be sitting on line 110 with the cursor under the 1 sure you are panicking about the other lines, so I want you to type either LIST then press return. Why there is the complete programme! Editing here is that you log in or record the line as written with the line number, each time you press return you add the line to memory. You can also type in line numbers out of linear sequence but they will all show up in the programme in nice neat linear number fashion when the program is Listed to screen.

So what is happening in this programme? After you run it you will see the 5 space colour bars go across the screen and down one line, pretty fast considering the speed of the machine <VBG>. Well we have covered line 1, the REM statement. We have covered line 5 the clear home CHR$. Line 10 is a tad bit different, here you tell it to print a character string; this one being 18, that is the command to turn on reverse video with a character string.

A semicolon follows, so things will stay on the same line and be connected, that is why the colour bars are butted against each other, no spaces and no one colour bar, then it drops to the next line. In the quotes are 5 spaces, just a number I took from the book, you can make it larger or smaller for some interesting effects. At 20 we have that variable and a random number generator, as you see the number in there is 8, so we are creating 8 separate numbers. Line 30 has a new command, the ON command, before I go to the how it works, think what the programme would look like if you used IF THEN statements.

```
40if CL=1 then?chr$(5):goto10
```

Right you would have to do that for each of the 8 lines, of course you would be writing 8 lines of IF THEN and a line about getting the CL, and well you can see it is confusing. Using the ON command is a little hard and confusing at first, the majority of people in programming Basic that I respect, advocate the ON command. Saying that, here is how it works, as simply as I can state. The Line is saying that on the CL variable, the computer is to goto a specific line.

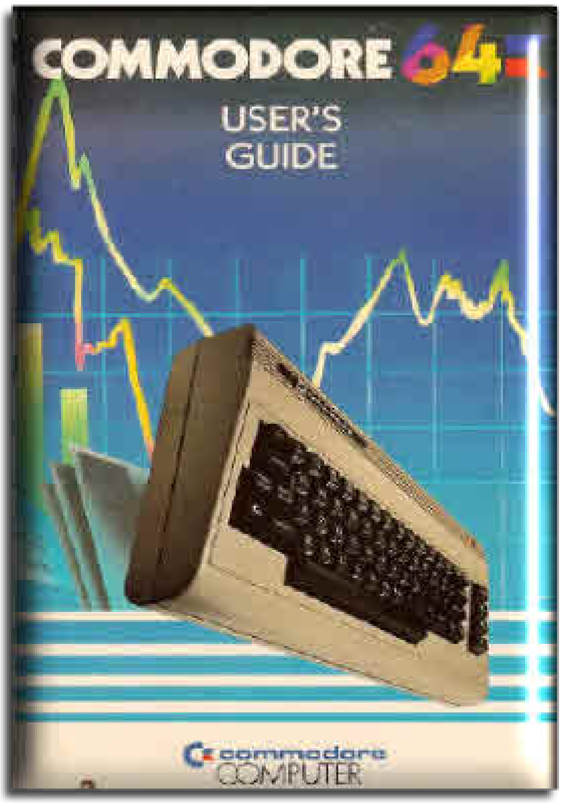
So let`s say that the random number generator creates for CL a 7, we do what and where? Right I understand the confusion, I had that problem for a long time a mental block. What helped me is to count the things in the rest of the line, here it reads 40-110. Yeah those commas are mandatory. Hmm seems to be that there are 8 of them, well the generator makes only 8 numbers. The Amount of things to goto must equal the amount created with the generator. Personally I got messed up with the numbers and other things for the goto part, blocking me from gaining even this little understanding.

Ok so back to my example, the generator has created for the variable CL a number 7, So on CL, or read that as ON 7 GOTO 100, this is because that is the 7th number in the line. Got it, generate a 4 and it goes to the 4th number in the line. 1 would be the first number, and so forth. Don't ask how long it took for me to understand that one. <S> OK you are brighter than me, bet you can see that in certain forms of games, you can have the ON statement take the programme to a print statement and have specific text appear.

After the above example, the book now on page 60 starts at the bottom with the idea of PEEKS and POKES do. Honestly this is a hard one for me, not that I don't know what they are, I don't know WHAT they are. Got you confused yet? <G> Well OK what I mean is that I know that a POKE or in short hand a pO, Sticks something into the computer memory, the book uses the idea of boxes. Other books talk about egg crates, and others the old fashioned roll top desk with all the little letter sized pigeon holes, all trying to give you a visual of a lots of storage areas that you can put a something into. A PEEK or in short hand pE. Is looking in as they say, to a box or pigeon hole, and shows you what is in there at that moment. Got that part in the bare basics, now what or where these places are is not in my understanding at this time.

Showing the vacuum of my brain above. We shall talk more on this in the next session

CONTINUED NEXT MONTH



# Commodore 64 .prg Generator

A handy Commodore BASIC programming utility for Windows

**Commodore 64 .prg Generator**
A handy Commodore BASIC programming utility for Windows
http://www.ajordison.co.uk/

What is it?
In a nutshell, the Commodore 64 Program Generator (called C64prgGen from now on) allows you to type a BASIC program using a nice Windows environment (let's face it, the C64 was a pain to use compared to today's modern kit) and then convert the application to a '.prg' file which you can run on an emulator, or even a real C64 if you're feeling brave and have the right kit to transfer the file.

What it isn't is a front-end for tok64, cbmcnvrt, bastext or any other tokeniser/detokeniser. It's been written completely from scratch.

C64prgGen has the following features:
-NEW! Program import (.prg, .T64 or .D64),
-Sprite editor,
-Character editor,
-Binary file import/export,
-Code snippets (often-used routines which can be pasted into code),
-A Screen Code Builder, for using those pesky print control characters,
-Code formatting and renumbering,
-Comprehensive help, including tutorials,
-Plus many other features.

Background
A while ago I was given a VB.NET project at work. I'm a C++ programmer by trade and I hadn't touched VB since version 3 (many years ago) so I was a bit rusty to say the least. Typically I wasn't given any investigation time and was expected to just get on with it, I'm sure any programmers out there are familiar with that! Anyway I was scrabbling about looking for a nice little project I could do in my own time to get me up to speed with VB.NET, at the same time I was into playing around with the various C64 emulators and tools particularly tok64. That's how C64prgGen came about, I noticed that there weren't many (if any) code generation tools with a GUI front end. Admittedly I hadn't looked very hard though, so after about six months of on-and-off (mostly off) work I finally released version 1.0.0

Future Developments
I'm writing a 6502/6510 assembler, which will be integrated into C64prgGen, which I'm hoping to release by the end of July. I'm open to any suggestions, so if you have anything you'd like adding to C64prgGen feel free to contact me!

News!
3rd June 2008, Version 1.3.1 released. NEW Features
Program Import You can now import programs either from .prg, .T64 or from a .D64 disk image. You can only import BASIC version 2 files though. This wasn`t something I was planning to implement into C64prgGen. I had some unexpected time off work and so I thought I'd have a stab it at. This is a 'work in progress' feature though so don't expect it to be bomb proof! It has been successfully tested with all of the sample programs which come with C64prgGen, i.e. the programs have been compiled into .prgs which have been imported again.

Binary Import
You can now import binary files and append them to your listing as DATA statements, useful for importing sprite and character data.

User Defined Tools
You can launch you own tools from within C64prgGen.

Sprite and Character Editors
These can now 'import' data from the BASIC listing by using the listings DATA statements.

Screen Codes
The screen codes, e.g. '{left*8}', can now be configured to be compatible with other tokenisers/detokenisers.

Comments
You can now select a region of text and automatically turn it into a comment. See the 'Changes' section

Changes
The comment block tool has been removed as it shared functionality with the new way of turning text into a comment and was no longer useful.

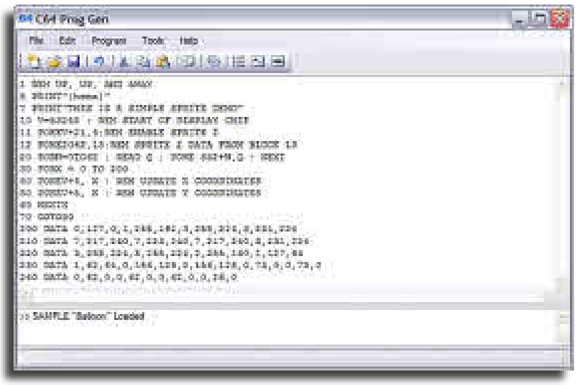
The rules regarding the format of screen codes have been relaxed. Previously a code string had to be in the format {code*multiplier}. Now it can be either way around, e.g {multiplier*code}.

Tweaks
Couple of minor interface issues.
You can now 'Find/Find Next' text in the BASIC listing.
You can now resize the status window. (Yay!)

Bugs fixed in this release
There was a problem if:
1) you used {pi} in your programs as it wasn't translated properly.
2) you tried to convert a .prg generated with C64prgGen back to BASIC text, e.g. with tok64.
3) you tried to renumber a program which had listing only ('!-') comments in them.

4) you closed the sprite or char editor and sometimes unwanted data was inserted into the program listing.
5) you tried to use the keyboard shortcuts for cut/copy/paste.

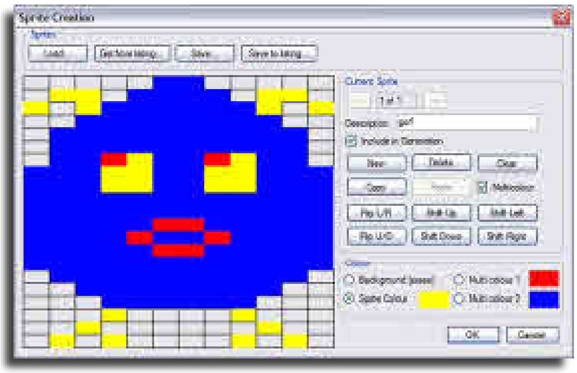
Screen shots
Main Application Window
Below is the main application window. This screen is showing the 'Balloon' sample program. A number of sample programs taken from the 'Commodore 64 User Guide' and 'Commodore 64 Programmer's Reference' are iincluded.



```
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 POKEV+21,4:REM ENABLE SPRITE 2
12 POKE2042,13:REM SPRITE 2 DATA FROM BLOCK 13
20 FORN=0TO62 : READ Q : POKE 832+N,Q : NEXT
30 FORX = 0 TO 200
40 POKEV+4, X : REM UPDATE X COORDINATES
50 POKEV+5, X : REM UPDATE Y COORDINATES
60 NEXTX
70 GOTO30
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
```

```
>> SAMPLE "Balloon" Loaded
```

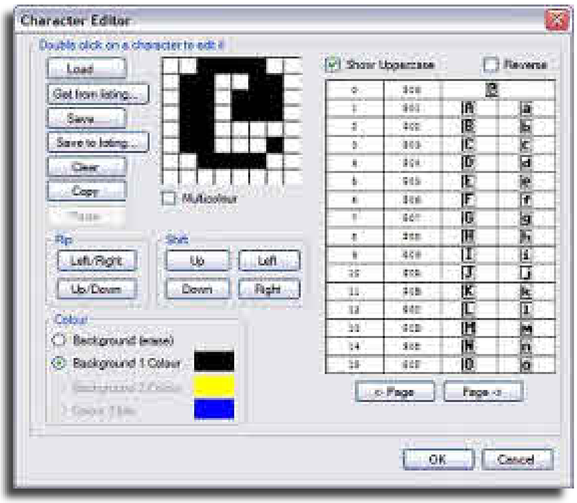
Sprite Editor
Below is the sprite editor dialog. The editor allows the creation, saving and loading of up to 128 sprites.
The sprite data can be generated as DATA statements directly into your BASIC program.



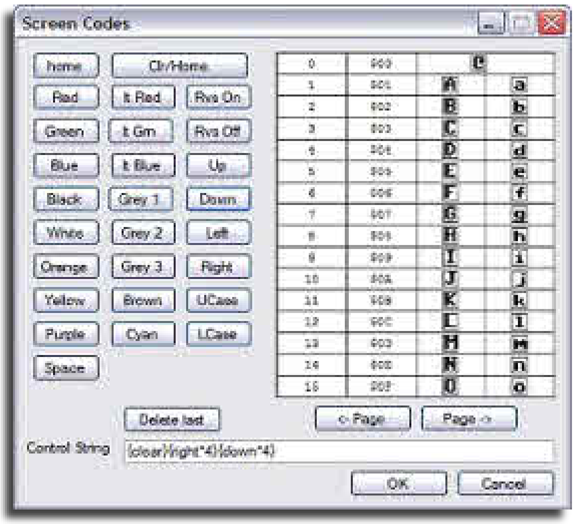
Character Editor
Below is the character editor dialog. The editor allows the creation, saving and loading of any character from the upper and lower case set. As with the sprite editor data the character data can be generated as DATA statements directly into your BASIC program.

Screen Codes
The Screen Code Builder dialog lets you build up strings of screen control codes. No more having to remember what the code to clear the screen is! The real reason for this is many listings found on the net use non standard codes, and this dialog is used to translate them.

Code Snippet Tool
The Code Snippet dialog is used to select and paste common, often used code snippets, e.g. clearing the SID chip into your code. You can also create and save your own snippets.

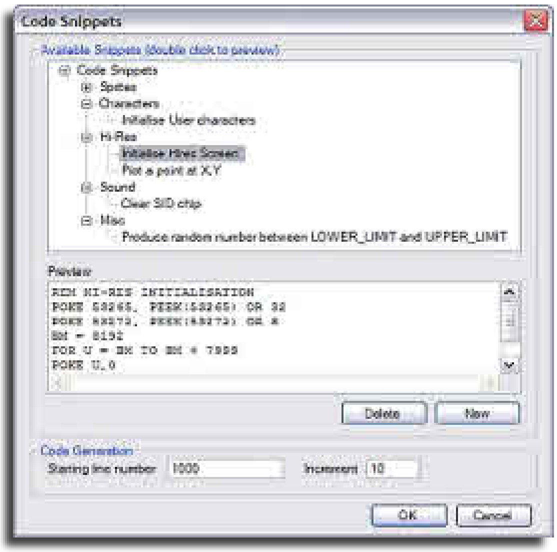


| | | | |
|---|---|---|---|
| 0 | $00 | @ | |
| 1 | $01 | A | a |
| 2 | $02 | B | b |
| 3 | $03 | C | c |
| 4 | $04 | D | d |
| 5 | $05 | E | e |
| 6 | $06 | F | f |
| 7 | $07 | G | g |
| 8 | $08 | H | h |
| 9 | $09 | I | i |
| 10 | $0A | J | j |
| 11 | $0B | K | k |
| 12 | $0C | L | l |
| 13 | $0D | M | m |
| 14 | $0E | N | n |
| 15 | $0F | O | o |



| | | | |
|---|---|---|---|
| 0 | $00 | @ | |
| 1 | $01 | A | a |
| 2 | $02 | B | b |
| 3 | $03 | C | c |
| 4 | $04 | D | d |
| 5 | $05 | E | e |
| 6 | $06 | F | f |
| 7 | $07 | G | g |
| 8 | $08 | H | h |
| 9 | $09 | I | i |
| 10 | $0A | J | j |
| 11 | $0B | K | k |
| 12 | $0C | L | l |
| 13 | $0D | M | m |
| 14 | $0E | N | n |
| 15 | $0F | O | o |

Control String: {clear}{right*4}{down*4}



```
REM HI-RES INITIALISATION
POKE 53265, PEEK(53265) OR 32
POKE 53272, PEEK(53272) OR 8
BM = 8192
FOR U = BM TO BM + 7999
POKE U,0
```

Starting line number: 1000
Increment: 10

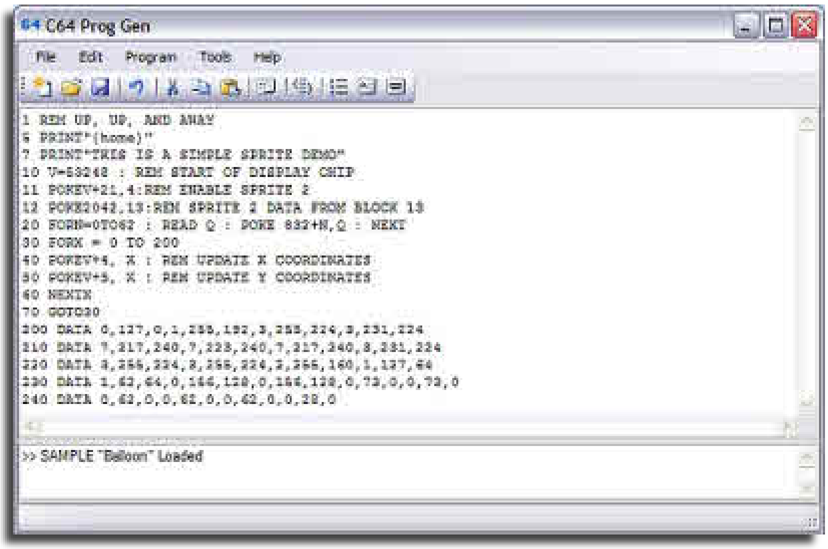
# Interview with Arthur Jordison creator of Commodore 64 .prg Generator

Commodore Free
Please introduce yourself to our reader

Arthur Jordison
I guess, like most programmers around my age, I started programming when I was in my teens. Back then you were either in the Commodore camp or the Sinclair camp, luckily I chose the right one! I was only casually interested in programming at the time though as my main interest was games, this is why I still like the emulator scene today. I kind of drifted into programming professionally really, ending up with a BSc degree in Information Technology and a Masters degree in 3D graphics, neither of which I use for work! Now I'm a C++ programmer by trade and I work for a company which develops traffic control and monitoring systems. I live in the north east of England with my wife and (almost) two children.

CF. When did you first hear about Commodore machines

AJ. I was lucky enough to get a VIC-20 not long after they were released, so I'll have been about 13.



```
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 POKEV+21,4:REM ENABLE SPRITE 2
12 POKE2042,13:REM SPRITE 2 DATA FROM BLOCK 13
20 FORN=0TO62 : READ Q : POKE 832+N,Q : NEXT
30 FORX = 0 TO 200
40 POKEV+4, X : REM UPDATE X COORDINATES
50 POKEV+5, X : REM UPDATE Y COORDINATES
60 NEXTX
70 GOTO30
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
```

```
>> SAMPLE "Balloon" Loaded
```

CF. What machines do you own Commodore or otherwise

AJ. I've had a number of Commodore 64s (and Amiga`s) over the years but the only Commodore machines I now own now are calculators. I used to have quite a lot of Commodore hardware but sold most of them a couple of years ago. I've still got some old 8-bit Ataris somewhere in the house though. At home I do all of my development on my wife`s laptop, which isn`t very exciting.

CF. Do you still use Commodore machines, if so what use do you put the machines to

AJ. Like I said, I still have some old calculators, one of which I still use at work. I wish I had the space to own some actual machines, so mostly I use emulation these days.

CF. can you explain what the "Commodore 64 .prg Generator" is

AJ. Basically it an applications that allows you to type in (or load from a text file) a BASIC program into a windows environment and then generate a .prg file which you can use on an emulator or a real 64. It also let's you create and edit sprites and characters and create the DATA statements you need for your BASIC programs. The application also comes with most of the sample programs taken from the Commodore user guide and programmer`s reference manuals. There are also some simple formatting tools, such as program renumbering built into the application too. I`ve just added the ability to import files, either .prg, .T64 or from a D64 disk image. The application can also extract sprite and character data directly from a BASIC listing.

CF. So does this application run on a PC only or are other systems supported?

AJ. PrgGen only runs on a PC with WinXP or windows Vista installed. I haven't tried it under Vista though so I'm assuming it works.

CF. What was the motivation to create such an application?

AJ. I liked tinkering around with various emulators and tools but they tended to be old DOS or windows applications that hadn't been updated in years. I was mostly playing around with tok64 and using it to generate small programs, it's at this point when I started thinking about creating a modern window's tool which did the same job. I didn`t however want to just put a windows front-end on tok64. So about six months ago I wrote a simple .prg generator application and since then I've been improving the .prg generation and bolting on extra features and tools.

CF. From the Website you say you were given VB as a project at work so how did the project develop from there?

AJ. Well I'm sure you understand but I can't really talk about my work as it's a bit sensitive, let's just say that having VB at work allowed me to do some extra lunchtime coding! C64prgGen has nothing to do with work though.

CF. Do you intend to develop the software for other platforms or re-write the application in any other language

AJ. I've been thinking about using a product called realBASIC which let's you develop code for Linux and Mac platforms. I don`t have access to either Linux or a Mac though, so testing would be a nightmare. I might write a version of C64prgGen using this but it's really a question of it being worth it. I may be wrong but I don't think that there are that many Linux or Mac users who would be interested!

CF. Is this the final version, and will there be bug fixes tweaks etc in the future, I guess I am trying to say will you continue support for the application?

AJ. No this isn't the final version, in fact I've just released a new version with some new features. I'll continue to release bug fixes as time allows, I do however have a busy family life, so I'm very limited to how much time I can spend programming.

CF. Do you plan to support other Commodore machines

AJ. I'm going to add support for other 64 BASICs (don't ask me when!) and maybe even the VIC-20 as the BASIC is almost identical. I'm not sure about that though as C64prgGen focuses on the 64, as the name suggests. If I do add other machines then C64prgGen will need to be re-written as something else and I'm reluctant to do this in the near future.

CF. So this is an application to help BASIC programmers develop BASIC applications?

AJ. It only helps in the sense that it merges some common tasks (e.g. program creation/generation, sprite and character editing etc.) into one application, saving you the hassle of using separate tools. It also has some built-in code snippets, which are sets of pre-defined functions for using sprites, music etc. I did get an e-mail from one person who wants to use C64prgGen to teach BASIC to his students so I'll be interested to see how he gets on with that.

CF. Can you explain basically how our reader would use the application?

AJ. The application is very simple to use.. Basically you can either type (or load) in your program and just hit F6. C64prgGen will then generate a .prg file and if it's valid the file will be sent it to your emulator. You will need to configure C64prgGen properly to use this function though. It does much more than that though, there are a few tutorials in the help file which shows you how to do some common tasks.

CF. Do you have any other projects planned Commodore or otherwise?

AJ. I don't have any other Commodore related projects in the pipeline, I can't think of anything I'd like to write that hasn't already been done before; though I am open to suggestions.

CF. Is the application FREE to our readers?

AJ. Yes it's completely free, much to my wife's disappointment....

CF. If our reader is wanting to contribute to the project how can he or she help?

AJ. I'm open to any (sensible) suggestions and criticism about the application. C64prgGen has a couple of threads on the Lemon64 forum, so that`s a good place to contact me. I`ve had lots of help and suggestions from people in that forum. You can also contact me via email. I'd like to let users leave feedback using my website but I haven't a clue how to do that, I had to learn some HTML just to get a basic web site going. It's fair to say it's definitely basic! Any tips on how I can improve the website would also be appreciated. On a more technical note one thing I`d like C64prgGen to do is syntax highlighting. Syntax highlighting is where BASIC keywords are coloured differently from other text and variables to make them stand out. I had a brief play around with this but all of the things I`ve tried are very slow. If any of your readers know how to do this efficiently (in VB) I`d like to know!

CF. Your website says your wife is due to give birth, has your wife had the baby and is everything ok?

AJ. The baby was born on the 30th May

CF. I guess everything will be on hold for a while with a recent addition to the family?

AJ. Yes, the computer (and almost everything else) will be off limits for a while I expect. Except for my two year old who loves it!

CF. Do you have any question you would have liked to have been asked?

AJ. No, I think you've covered everything.

CF. Thank you for supporting Commodore machines and good luck with the application

AJ. You're welcome. I'm just glad you wanted to do this article. I'm very keen to get the application out to as many enthusiasts as possible. Currently I get around three or four downloads a day and hopefully after your article I'll get a few more. I spent a lot of time on this application and a reward is knowing that people will use the application for creative purposes.

commodore computer club

# Membership forms for the Commodore Computer Club (UK) (CCC(UK)).

Please sign me up for a membership to the CCC (UK) - I wish to be a member for:
[ ] 6 months* at £3 [ ] 1 year* at £5 [ ] life member at £30

My personal details:

**Name:

**Address:

**Postal code:

Country (if outside of the United Kingdom)**:

**Date of application:

[FOR INTERNAL USE: Date membership fee received: ]

Email address:

Please tick the box below once you have read and understood the club's rules and regulations:

**[ ] I declare that I, the named applicant above, have read and understood the CCC (UK) rules and regulations, and agree to abide by them fully and co-operatively. I understand that I am joining this club on a personal level, and not as a representative of any group, developer or vendor that I belong to.

Please tick which computers you own:
[ ] CBM/PET 40 columns
[ ] C64GS
[ ] C64DTV
[ ] CBM/PET 80 columns
[ ] SX-64
[ ] C16/116
[ ] Other Commodore 8-bit
[ ] VIC/VC 20
[ ] Plus/4
[ ] Other Commodore 16-bit
[ ] C64/64c
[ ] C128/128D

If you have ticked 'Other Commodore 8-bit' or 'Other Commodore 16-bit', please list these machines below:

Please tick from the list below your interests from the following:
[ ] Gaming
[ ] GEOS
[ ] Collecting
[ ] BASIC programming
[ ] JOS/WiNGs
[ ] Archiving/preservation
[ ] Machine language coding
[ ] Tech/scene demos
[ ] Other applications

If you have ticked 'Other applications', please give details below:

Please tick if you use any of the following peripherals below:
[ ] Datasette
[ ] FD2000/4000
[ ] Commodore REU
[ ] Other accelerator
[ ] RR-Net or FB-Net
[ ] Action Replay
[ ] Other speed loader
[ ] 1541 compatible disk drive
[ ] CMD HD or RAMLink
[ ] Other RAM expansion
[ ] MMC/Retro Replay
[ ] Other networking device
[ ] Trilogic Expert Cartridge
[ ] Commodore VDU
[ ] 1581 compatible disk drive
[ ] 1351 mouse or compatible
[ ] SuperCPU 64/128
[ ] SwiftLINK/Turbo232
[ ] 1541Ultimate/+
[ ] Other cartridge upgrade
[ ] Other hardware

Please list any items not mentioned above which you have (especially for other Commodore 8-bit machines):

Thank-you for taking the time to fill in this membership form. Your membership will be processed on receipt of the relevant membership fee as stated.

* Membership fees for 6 months or 1 year will be back-dated to the beginning of the month that the membership application has been made and relevant fee has been cleared into our account. This can be paid either by personal cheque, postal order or PayPal. Membership fees are currently £3 for six months, £5 per year or £30 for life membership.

** It is essential that you fill these details, otherwise your membership will not be processed

# Commodore Computer Club (UK) (CCC(UK)).

## Club rules:

(1) Membership subscriptions, raising funds and re-selling items.
(a) All members will pay a membership fee as follows: £3 for six months, £5 per year and life membership at £30. This fee will entitle the holder to free entry to the meetings, and special limited areas of the website, such as private forums and exclusive downloads, should we get anything exclusive to download. Membership will always be back-dated to the start of the month in which the member took out the subscription, so that everyone joining in the month of April for one year will see their membership expire on the 31st of March the following year.

(b) We should have a 'Commodore Computer Club Shop', which will stock all of the latest hardware mods and sods for Commodore computers. To stop the 'Maurice Randall' effect, in which the club will have to repay people for not receiving their goods because they haven't been delivered but have been paid for, items will only be on sale if they are in stock.

(c) There will be two prices, one for members (cost of item + postage and packing + 10%), and one for none members (cost + postage and packing + 20%).

(d) Any members that do work for the club, organising events, donating items for auction, coding, or are otherwise active, with exception to posting on forums and turning up to meetings, will be considered for free membership and/or lifetime membership on merit based only on work they have done for the club. In certain instances, will include what they have contributed overall to Commodore computing or gaming during their life-time, should any 'Commodore legends' show sufficient interest to join the club.

(2) Events, software and other developments.
(a) Any money that is raised by the club should be used primarily for setting up events, or bolting onto other events as appropriate. This is to go towards, or cover costs with van hire, hotels, food and drink, so the person or people who are willing to travelling to these events, man stalls and generally promote the club and its work are not be out of pocket as far as possible.

(b) Profits made from items sold at events should contribute to cover the costs of attending, or hosting, and/or expenses accrued during the event. This will not include monies raised from membership subscriptions paid for during the event.

(c) The club should seek to raise money for the purpose of developing hardware and/or software that will benefit Commodore users in the UK and world-wide, and that could therefore be sold through the club Shop.

(d) Payments to developers who are commissioned to work on behalf of the club should not be made in advanced or up front unless otherwise agreed by the treasurer and chair-person, and any other two members. This should be openly discussed with all members either in private members areas of the site, or at an organised meeting as appropriate.

(e) Hardware that is commissioned on behalf of the club which reaches production should be sold at a small profit, and monies raised to put back into the clubs funds.

(f) If it is agreed that the club should commission entertainment software, the productions should be available to download for free from the site for members only. Real-media versions should also be sold through the shop with non-members able to buy copies, though at a higher price than members.

(g) Any software commissioned by the club will either be purchased outright, paying the programmer an agreed fee on completion, or paying a lesser fee and splitting the profits at an agreed rate. This should be discussed on a case-by-case basis. The chair-person and treasurer, and two other members, must agree which method should be implemented.

(3) Meetings and monies
(a) The club should hold an annual general meeting in which members have a say in its running, and are able to make suggestions and table official club business for the year ahead. Membership subscriptions should be reviewed at the annual general meeting, and any price increases must be agreed by the chair-person, treasurer and at least two other members.

(b) There should be an annual audit of the clubs finances, with a news-letter at least every three months. The audit should be published before the annual general meeting, and this and the news-letter should be available to current members online in the private member areas. Former members may request this information, which will be granted on a case-by-case basis.

(c) Members will be able to attend any events that organised and run by the club for free when ever possible, whilst non-members will pay a small signing in fee of at least £2.50. With agreement with other event organisers, and at events that the club is attending in an official capacity, we will work towards getting members a discounted entry fee.

(d) All monies raised will go back into club funds.

(4) End of line
(a) If it is apparent that the club is not running within its means to the extent that it is likely to fold, or that legal action against it

will lead to the club being dissolved, all club assets should be sold or auctioned off, and members will be refunded their current subscriptions based on the length of time they have been members. The longest-serving paying members will be refunded first as appropriate, either patricianly or fully depending on the financial circumstances at the time. The newer members, who will be dealt with last.

(b) Personal donations to the club's funds can never be fully refunded.

Club complaints procedure:
Phase 1: Where a complaint is made against the club, or one of the club members, there should initially be a private apology between the club or individual and the plaintiff. This apology should be for 'any undue harm or upset caused', and will not amount to an admission of guilt or a retraction in any way. The club will not be able to force any of its members to make this initial apology except in the instance that the individual has clearly and admittedly worked on the clubs behalf in the matter specifically relating to the complaint that has been lodged.

Phase 2: The matter should then be investigated to establish the facts. If it is deemed that an individual club member has not been acting on the clubs behalf with regards to the specifics of the complaint, then this becomes a personal matter between the two parties. The club should therefore stop any further investigations or involvement in the matter.

Phase 3: If the complaint lacks any real evidence, or it is felt that the findings are not conclusive, then the matter should be closed. Neither the club, nor any of its members, should therefore discuss the matter publically. All findings should be reported to the plaintiff, and the matter should be considered closed from the club's point of view.

Phase 4: Where a complaint is upheld, a public apology and/or retraction should be published through the official website, and in the newsletter. The club should also give the plaintiff the opportunity to give his or her point of view through the website and/or newsletter as appropriate. In this instance, the case will be considered closed from the club's point of view unless the plaintiff wants to take the matter further through due legal process.

Emergency phase: If at any point during this process the plaintiff feels aggrieved to the extent that he or she instructs a solicitor to take the matter up against the club or club members who have clearly being acting on behalf of the club in this instance, the club should then consider its legal position on the matter, and a meeting should be set up with the principle members of the club within two weeks of receiving legal notice to discuss the matter, and what to do next. Obviously, one would hope that any complaint would ever get to this stage.

## Commodore Computer club FAQ

**So what is the Commodore Computer Club (UK) then?**
It's a UK-based user group supporting all classic 8-bit Commodore computers, and the C64DTV, excluding the rare Commodore machines which are more collectible, such as the P500. As a user group, our primary interest is in keeping these aging machines useful for as long as possible, by supporting hardware and software developments across all machines.

**Does it cost to join?**
Yes. subscription fees start from £3 for six months membership, £5 for a year and £30 for life membership.

**What will I get for my money?**
The club is run by its members. An annual audit of club funds will take place, as well as an AGM at which all members will be able to have their say about the running of the club, and make suggestions for its direction, or future projects that the club should take an interest in. On the whole, it is one member, one vote, except if stipulated in the club rules.

**What will my money be used for?**
The club will use its funds to set up shows and meets across the UK, and commission hardware and software as appropriate to its membership. For instance, we are currently looking at creating a good quality VGA adapter that will adequately handle the various video signals created by Commodore 8-bit machines: composite, Luma/Chroma and RGBI. Once found, these units will be made up and sold through our club shop. Whilst anyone may purchase these, these is a surcharge added to the price for non-members.

**So, non-members will get all the benefits of membership by only paying a small surcharge?**
Not always. We're setting up a private members area to the club website, and certain items will not be available to non-members, as agreed by the club: the newsletter, for instance, will be members only. All club-organised events will be free to members, and non-members will have to pay a signing in fee of £2.50. In the instances in which the club is bolting onto another event organised by someone else, we will work with the event manager so that our members will get a discount on the ticket price for the day or weekend as appropiate.

**What's the difference between a CCC (UK) event, and me organising to meet some Commodore users in my local pub**?
If you think that you can arrange to meet other enthusiasts locally to you without the club, then that's great! But generally speaking, it's difficult to organise even the most informal meetings and even with the lure of beer. Our club events will have the benefit from being organised with a purpose. Not only will you get the chance to meet other enthusiasts, but also you can e.

thrash out hardware problems by bringing along your misbehaving piece of kit, or learn new skills through our workshops. Also, we have many spares in our arsenal, so if you have a flaky SID chip that needs replacing, or you need your drive head realligning, then bring it along and we'll see if we can help you! And whilst non-members will have to pay a signing in fee, they'll also have to pay for any fixes. As a member, you'll usually receive any easy fixes for fre

**Do you support Amiga?**
At the moment, we are only able to support Commodore 8-bits because we don't have members who have enough knowledge about Amiga range to adequately support that platform.

**So, how long have you been going?**
Not long. We started talking about a user group at the end of June 2008, held our first meeting on the 26th and 27th of July 2008, and started taking formal membership on the 1st August 2008. There are still some things that are under construction, but we're getting there.

**What sort of users are you attracting**?
At the time of writing, we have a rough 50/50 split between the so-called "Power users" and the average Commodore user. The word user is an important one: whilst we can offer help and advice for emulator fans, that's not the principle aim of the club.

**What work have you been doing behind the scenes then?**
We have agreement in principle with Cronosoft (www.cronosoft.co.uk) and Psytronik Software (www.binaryzone.org/psy.php) to resell their Commodore games on disk (3.5" and 5.25") as Cronosoft aren't very good at advertising that you can buy their games on disk, and Psytronik software don't currently do disks. We have also spoken to several individuals about manufacturing a SuperCPU-like accelerator for the C64/128 (one is looking likely to take on this challenge), and there is a distant possibility of a game released for the Commodore +4 and C64/GS on cartridge.

**Why aren't you just supporting the existing projects out there that are free?**
Whilst the club will always support any projects or publishers who largely work for free, there are still some gaps that need filling, and because of real life or other commitments, these projects have a tendency to stall every-so-often.

**You said that you have a shop?**
Yes. If you remember Commodore Scene's importing service, this will work in pretty much the same way. And whilst CMD hardware might no longer be available, there are other hardware projects that are. We have several benefits too: because we'll have some resources to play with, we could buy in several items at a time, meaning that we could save on shipping costs and pass that saving on to our members. There is also the issue of documentation, certainly with hardware projects out there. The manuals are often extremely lacking, or the English translation isn't clear enough. As a club, we can provide further documentation to hardware items sold through our shop, and hold workshops for people who are still having trouble.

**Are you trying to be another website like Lemon64?**
Absolutely not. There are lots of good websites for gamers, but if you're a GEOS user, for instance, where do you go for advice? Whilst we will happily report on and review any new software, we're not necessarily all about nostalgia, and have the added dimension of good programming and technical advice too. Plus, we're happy to support all Commodore 8-bits, rather than going to Sleeping Elephant for the VIC, EmuCamp for C16/+4, Lemon64 for the 64 and other sites for the 128 and PET, we want everything in one place if we can!

**http://www.commodorecomputerclub.co.uk/**