C64 Repository

Matthew Gream
Version 0.90

The Commodore 64 scene experiences of Matt' in the Australian groups TOUR DE FUTURE, TOUR DE FUTURE+REFLEX and TERA for the years of 1987 to 1990.

Future changes
Commentary and review provided by TDF, TDF+RFX and TERA guys: verify my memory and fill in the gaps.
Locate further TDF, TDF+RFX and TERA software productions: from guys and contacts.
Locate further TDF, TDF+RFX and TERA paraphernalia: disk covers, photographs, drawings, etc.

Recent changes
Version 0.90 (December 2001) – the first substantial version of this document, having had minor review by TERA and TOUR DE FUTURE guys, but still needing further editorial work before becoming V1.00. As a first pass, there’

Contents

Figures

Acknowledgements

I have to thank the other guys in TOUR DE FUTURE, REFLEX and TERA for the times we had. The guys included RECKLESS ROB, BIZ, PARASITE, XLR8, MISTIC, SNOOP, HEX-HACKER, BOSS, BULLET, ELIMINATOR, STOREMAN, and KILENEMY. Special thanks to XLR8 who I've tried to keep better contact with in recent years.

Many of my contacts on the scene were an important part of it all, and it wouldn't be fair to single out names – but I'm especially reminded of HARRY of 711, JERRY of TRIAD, DEVIET of WOT and BUCCANEER of WOT. If I hadn't have put the scene behind me, I may have continued contact with these and other good guys.

Thanks to MARTIN FRECH for his C64DEMOS.COM and for providing me (very quickly!) with a lot of old TOUR DE FUTURE, TOUR DE FUTURE+REFLEX and TERA software. Other C64 repositories on the Internet filled in the gaps and completed the picture. It is great that a lot of collections are being archived on the Internet: hopefully this will continue, and my work will add another small part

Thanks to my parents for financing my activities as a teenager, yet still putting up with me: they have a lot of tolerance.

Introduction

From 1987 until 1990 I was actively involved in the Australian Commodore 64 scene as MATT' of TOUR DE FUTURE, TOUR DE FUTURE+REFLEX and TERA. My software productions played a key role in the development of these groups. We became highly regarded, especially for our packer and notemakers.

I left the scene in mid 1990 to pursue other interests, and only started to look back in mid 2000. I wrote a brief outline of my experiences, but little material was available to assist the activity. In mid 2001 I found some of my original productions on C64DEMOS.COM, and the ball started to roll.

From May to October 2001, I was able to recover further productions from various Internet repositories, and find disk magazines and other material to help put the pieces back together. This document evolved on the London Underground, the occasional evening, the odd early morning, and several weekends.

I have aimed to document my own experiences, and the history of the groups that I was part of. This is accompanied by extensive detail about my own productions, and complimented by background material about the system and the scene, including an extensive set of references.

At the time of writing this, more than a decade has passed since my activities with the Commodore 64 ceased. I expect that my memory will not have served me entirely well, so I am sure that the material is occasionally lacking or incorrect. I do hope to draw upon memories of others to ensure that the pieces are in the right places.

I can be contacted by email at *matthew.gream@pobox.com*, and would welcome feedback of any sort. I would also like to hear from old contacts.

Overview

The first set of sections (1-2) provides a general background to the Commodore 64 microcomputer and the scene that surrounded it. A comprehensive set of references offers avenues for further investigation.

The second set of sections (3) details my personal activities, and contains a history of the groups TOUR DE FUTURE, TOUR DE FUTURE+REFLEX and TERA. Excerpts from disk magazines, interviews and software products are included.

The third set of sections (4-11) extensively documents the software that I produced, with numerous screen captures, scroll texts and technical descriptions. VICE snapshots allow software to be brought to life, but must be installed on the system.

The final section (12) contains a number of D64 disks with all of the software productions mentioned throughout the document.

Section 1. C64 General Overview

1. Introduction

The Commodore 64 (C64) is an 8-bit microcomputer that was developed by Commodore Business Machines (CBM) in the early 1980s. It reached its height of popularity in the late 1980's before losing ground to 16-bit computers such as the Commodore Amiga and the Atari ST. The C64 was sold worldwide and became the preferred platform for computer game playing. It was a versatile multimedia platform, and a precursor to modern “consoles”.

2. The System

2.1. Introduction

There are many resources that focus on describing the Commodore 64 to varying degrees of coverage and quality. The summary provided here does not offer a comprehensive treatment, but does attempt to provide a reasonably broad picture. For further detail, other resources should be consulted.

2.2. Overview

The C64 can be decomposed as a system by considering the platform, the external enhancements, and the typical configurations.

There are a number of variations of the platform, consisting of different versions of the casing, motherboard and components. This platform provided a basic level of functionality, and a method of operation. Expanding the platform required the use of external hardware and external software. Although an instance of the system could be constructed in any way, there were common configurations.

2.3. Platform

2.3.1. Architecture

The platform consists of different versions of the unit, with the same features offered by similar – but evolved – motherboards and components.

2.3.1.1. Versions

The following versions of the system were made available.

- **C64**: the original version was released in January 1982, in a rounded beige case with the original motherboard.
- **C64C**: the revamped version was released in May 1986, in a more modern slimline white case with the original motherboard. It was re-released in 1987 or 1988 with the new short motherboard.
- **C64** "Golden Edition": the limited edition version of the original, released in December 1986. Only 200 were produced.
- **C64G**: the third generation version was released in August 1987 with the short motherboard.

2.3.1.2. Features

The following features can be found on the system. They are located on the (T)op of the case, the (R)ight-hand side of the case, and the (B)ack of the case.

- **Keyboard**: (T) 66 keys, with a standard QWERTY layout with 4 function keys.
- **Control ports**: (R) 2, 9-pin female Atari compatible “trapezoidal” sockets.
- **Cassette port**: (B) 1, proprietary edge connector [part of the motherboard].
- **Serial IEC port**: (B) 1, 9-pin round female DIN connector.
- **Expansion port**: (B) 1, proprietary edge connector [part of the motherboard].
- **User port**: (B) 1, proprietary edge connector [part of the motherboard].
- **Video/Audio port**: (B) 1, proprietary DIN connector.

- **RF output port**: (B) 1, female RCA connector (NTSC or PAL output).
- **Power switch**: (R) 1, small flip switch for on or off positions.
- **Power indicator**: (T) 1, red LED indicating on or off.
- **Power port**: (R) 1, 5-pin DIN round female connector.
- **RF output port selector switch**: (B) 1,small two-way switch.

2.3.1.3. Motherboard

There were two major versions of the motherboard, with various revisions.

- **Original**: the original 1982 motherboard, as used in the C64 and the earlier versions of the C64C. Contains the original SID & VIC-II chips with 12V references, and RAM constructed out of 8 x 64Kbit/1 pieces.
- **Short**: the revamped 1987 motherboard, as first used in the C64G, and then the later C64C. Contains a merged BASIC ROM & Kernel ROM, a new SID at 9V, and a new VIC-II at 5V.

2.3.1.4. Components

The motherboard contained many components, but a few of these provided the key functions of the system.

- **Central Processing Unit (CPU)**: MOS 6510 at ~1MHz (enhanced 6502-compatible). Main computing device.
- **Video Interface Controller (VIC)**: MOS 6569 (original), or MOS 8565 (short). Main video processing device.
- **Sound Interface Device (SID)**: MOS 6581 (original), or MOS 8580 (short). Main audio processing device.
- **Random Access Memory (RAM)**: 64KB RAM (can be shadowed by ROM).
- **Read Only Memory (ROM)**: 38KB BASIC ROM, 16KB Kernel ROM and 4KB Char ROM.
- **Complex Interface Adapter (CIA)**: 2 MOS 6526. Main utility device.

2.3.2. Functionality

The system functionality is provided by four major components, which operate with varying levels of hardware and software support.

- **Video system**: the graphics functionality is provided by the VIC chip. It offers 40x25 characters in character mode, 320x200 pixels in hi-resolution mode (with each 8x8 field containing a separate fore/back ground colour, out of 16), or 160x200 pixels in multi-colour mode (each 8x8 field containing pixels with four different colours, out of 16). In each of these modes, it can also display up to eight "movable object blocks" called Sprites. A Sprite has 24x21 pixel resolution, and can be in hi-resolution or multi-colour format. Horizontal and vertical smooth scrolling is supported. A light-pen is supported. Output is available through the RF output port as a VHF or UHF signal in NTSC or PAL format (respectively), and high quality output is available as a luminance/crominance video signal through the Video/Audio port.
- **Audio system**: the sound functionality is provided by the Sound Interface Device (SID) chip. It supports three independent voices, four different waveforms that can generate tones from 0-4KHz, and three different filters. Each tone can have its Attack/Delay/Sustain/Release times specified. It also contains two A/D converters, for two potentiometers (paddles) connected to the Control ports.

- **I/O system**: the I/O system is provided by the CIA chips, assisted by the Kernel ROM. These control the Keyboards, the two Control ports, the Serial IEC port and the User port. The Serial IEC port provides access to a number of Commodore peripherals. The User port is proprietary, but can be easily converted to be RS232C compatible.
- **Operating system**: The operating system is provided by the MOS 6510 CPU, assisted by the Kernel ROM, the BASIC ROM, and the Char ROM. The Kernel ROM provides I/O access for the various ports, such as the Serial IEC port. The BASIC ROM provides a simple, but effective, BASIC developed by Microsoft – but it contains no graphics or sound commands. The Char ROM contains the system character set. The system particular BASIC commands such as LOAD and SAVE use functionality in the Kernel ROM. The Char ROM is used by the VIC-II chip to render a font in 25x40 display mode.

2.3.3. Operation

The operation of the system consists of a start up phase, after which software is installed and executed.

- **Startup**: the default behaviour of a Commodore 64 was to display the familiar "blue screen" when it was powered on, indicating that the BASIC interpreter was available for use. If a cartridge was installed in the expansion port, then it might override the default behaviour and execute its own software.
- **Software Installation**: software can be installed through several I/O ports. The Cassette port can be used, as can the Serial IEC port. For these, the BASIC LOAD commands are needed – the media cannot load automatically. Software can be installed from a cartridge in the Expansion port, where it autostarts – as the cartridge maps into system memory space and overrides the default startup/reset vectors.
- **Software Execution**: when software was executed, it could switch out any of the system ROMs, to access the entire 64K of system RAM. From BASIC, it was not possible to switch out anything other than the Char ROM, however many assembly language programs switched out all ROMs. No ROMs were needed to access the graphics or sound hardware.

2.4. Expansion

2.4.1. Hardware

With the I/O ports on the unit, it is possible to connect the unit to external hardware. There are many peripherals available, and each can be roughly categorised by the port that it connects to.

- **Control port peripherals**: typically joysticks, but could also be paddles or a light-pen. Although joysticks were made by CBM, they were little used: there was a large market and many various joysticks available from different manufacturers, which were also compatible with the Atari systems (e.g. the 2600).
- **Cassette port peripherals**: typically the C2N (C1530) or C1531 Datassette, or compatible units sold by other manufactures specifically for the C64.
- **Serial port peripherals**: typically the C1541 5.25” SSDD disk drive (produced for the C64), but could also include the C1571 5.25” DSDD disk drive (produced for the C128, but compatible with the C64), or the C1581 3.5” DSDD disk drive. It was also possible to connect CBM disk drives. The port was also used for CBM printers, such as the MPS-801, MPS-803 or MPS-1200.

- **Expansion port peripherals:** typically a cartridge of some form, and usually either an application (e.g. a game), or a language extension (e.g. Turtle), or a utility cartridge (e.g. Datel Action Replay, Epyx Fast Loader).
- **User port peripherals**: typically modems and RS232C style devices.
- **Video/Audio port peripherals**: typically the Commodore monitors, such as the C1702 colour monitor.
- **RF output port peripherals**: typically a standard PAL or NTSC television capable of receiving UHF or VHF signals.
- **Power port peripherals**: typically the power supply!

2.4.2. Software

2.4.2.1. Installation

A large amount of software is available for the computer, and can be installed from the various peripheral devices.

- **Serial port peripherals**: which requires using the BASIC LOAD and SAVE commands, or something similar via an installed cartridge or extended DOS system (e.g. a directory listing required LOAD “$”,8,1; but the Datel Action Replay extended DOS provided the $ command).
- **Cassette port peripherals**: which requires using the BASIC LOAD and SAVE commands, or by hitting the COMMODORE-RUN keys together.
- **User port peripherals**: only if there was a custom cartridge allowing access to this device.
- **Expansion port peripherals**: either directly upon startup as hooked startup/reset vectors, or through some extended interface.
- **Custom internal ROMs**: such as with new BASIC ROMs, or an extended DOS system (the Dolphin DOS).
- **Keyboard entry**: by typing in a BASIC program, or perhaps in Assembler or another language.

2.4.2.2. Languages

The native language was assembler, but the system came with a BASIC interpreter.

- **Assembler**: the native language of the machine as directly interpreted by the CPU. The vast majority of software was implemented in this manner.
- **BASIC**: the default language of the machine as provided in the BASIC ROM set. However, it was slow: very few serious programs were implemented this way.
- **Others**: other languages were available on expansion cartridges, but were not very suitable for mass market software.

2.4.2.3. Functionality

A large variety of software was published for the system, and was typically categorised.

- **Game entertainment**: perhaps the large category of software for the C64. Games ranged from simple shoot-em-up’s, to graphics based adventures, text adventures, multi-level arcade style challenges, and so on.
- **Other entertainment**: included software such as trivial pursuit, karioki, cards or anything not really considered a game but still entertaining.

- **Home productivity**: included software such as certificate and banner makers, recipe programs, budgeting tools, catalogues, etc.
- **Office productivity**: included software such as word processors, spreadsheets, databases, bulletin boards, etc.
- **Utilities**: included software for file and disk management, printer utilities, tape and disk conversion, system expansion, etc.

2.5. Configuration

There are many possible system configurations, but it is possible to describe two general categories: the basic configuration, and the advanced configuration.

2.5.1. Basic

A very basic configuration of the C64 consists of the main unit with a TV as a monitor, and software installed from an expansion port cartridge. However, most C64s were sold with a cassette unit, which was connected to the Cassette port, so that software could be installed from tape. Joysticks were usually required to play games, and most systems included them for this reason. This was the configuration typically used for “inexpensive gaming”.

2.5.2. Advanced

The more advanced configurations add external disk drives, such as the C1541, the C1571 or the C1581. As the Serial IEC port supports daisy chaining of up to 4 devices, it is possible to have 4 disk drives installed. The port is also used for printers, such as the CBM MPS-801, CBM MPS-803 or the CBM MPS-1200. The expansion port can house a RAM expansion device, or – more typically – a fast load/save cartridge such as the Evesham Freeze Frame or the Datel Action Replay. Dedicated monitors such as the Commodore C1902 colour monitor provided a better quality output over the standard video RF output. There are other enhancements, including custom system ROMs that provided fast disk load/save, such as the Dolphin DOS system.

3. The Scene

3.1. Overview

The Commodore 64 did not exist as a technology in itself, it was the central artifact (or part of – depending upon your perspective) of a culture. This culture can be defined, and then described generally, and then a focus can be taken on the major elements: people, groups, and activities.

3.2. Nature

3.2.1. Definition

The C64 was a central part of a cultural scene that can be considered by its various perspectives. The perspective that I aim to describe is known by various names including "the demo scene", "the underground scene", "the group scene", or "the swapping scene". I will refer to it as "the group scene".

3.2.2. Arrangement

3.2.2.1. Overview

"The group scene" can be recognised by its primary elements: a large number of people, typically organised into groups, that produce software and trade software amongst themselves, mostly for personal status, and usually only secondarily for financial gain.

3.2.2.2. Geography

The scene was international, with an epicentre in northern Europe, particularly the Germanic and Scandinavian countries of Germany, Belgium, Switzerland, Netherlands, Denmark, Sweden and Norway, but also England and France. The scene stretched to all other continents, including the United States.

The leading elements of the scene (from any perspective or in any area of activity, etc) were typically located around the epicentre – most of the periphery was fairly average, and quite often very substandard. The home to *the* scene party was Venlo, Netherlands – conveniently located around the critical mass.

There were discernable differences in the nature of the various regions: the United States was the home of bulletin boards, modem trading, phone phreaking, but little production; whereas Europe was the home of production, parties and mail trading. Most of the rest of world was in someway connected to these mail or modem trading tentacles, and dabbled variously in all activities to different degrees.

3.2.2.3. Demography

The scene was mostly male and youth oriented, and there were many relationships with popular culture of the time, such as subway art, rap music, pop music, skateboarding, film and television, girls and relevant food products (Coca Cola!). There was a degree of machismo.

3.2.3. History

The scene started in early the early 1980s, but did not pick up into the mid-late 1980's and then matured to become increasingly sophisticated in the years of 1988 and 1989. It is a mark of maturity that the first scene magazine (SEX'N'CRIME by AMOK) appeared in early 1989. Advanced software techniques (higher resolution graphics, advanced digi-style music, superior manipulation of the machine) and refined stylistic approaches began in 1989, and seemed to have peaked in 1990 and 1991. It must be remembered that many elements of the scene began to migrate to the Commodore Amiga and 16-bit computers from 1989 onwards – sapping talent. A comprehensive study would likely find that the vast majority of technical innovations on the Commodore 64 were developed in the years 1989 to 1991: there was little left to do after that. Most of the momentum was lost in the mid 1990s, and although the scene continues until today, it is only a relic.

3.3. People

3.3.1. General

The participants in the scene were typically, but not exclusively, young - mostly under the age of 20.

3.3.2. Types

Most participants had various skills, and often defined themselves by their skills.

- **Musicians** wrote music, and sometimes they even coded music players and associated routines. They were often found in high status groups, or groups could become high status because they had musicians.
- **Graphics artists** created logos, character sets, sprites, images and various other displayable items that were used in software productions. Many coders were also

graphics artists, and virtually all groups had a graphics artist. The availability of tools meant that anyone could create graphics, with the result that quality varied widely.

- **Suppliers** obtained commercial software, either as company insiders, or from distribution companies, or even shops. Some even recracked software – but that was a big no-no.
- **Crackers** broke commercial software, and sometimes played a role in supplying. Many crackers were also coders, but often they preferred to only concern themselves with breaking software. They may have also trained and modified cracks, so some coding skill was required – but in a less than compositional manner.
- **Coders** create software, and brought together the use of music, graphics and code to create a product such as an intro or a demo. More advanced coders produced utilities or advanced system software. Virtually all groups had a coder, but quality varied widely.
- **Traders** were ubiquitous, and virtually any participant in the scene was a trader, which meant that they swapped software with other traders. The more professional groups had serious traders, who provided a distribution process for the cracking activities. Trading was mostly carried out by post, but modems and bulletin boards were extremely popular in the United States.

3.3.3. Status

There was a definite status hierarchy, in that 'musicians', 'graphics artists', ‘suppliers’, 'crackers', 'coders' and 'traders' where considered the most "elite" in that order. Eliteness corresponded to the sophistication of a skill, and also the scarcity of a skill (e.g. there were few musicians, but many traders). People could also be lame, which meant that they were stupid, bad or had attitude problems – a rough correlation to being “uncool”. There were many in the middle.

There were many legendary people of all sorts of skills – these people had very high status. BOD of TALENT was a legendary trader, MATCHAM was a legendary coder, and JCH was a legendary musician. Many commercial musicians and games designers achieved legendary status (e.g. Rob Hubbard for music, Jeff Minter for games design).

3.4. Groups

Participants typically formed themselves into groups, and often a group may have made a name for itself, while its members were less well known. Some groups were known for their cracking abilities, some for their trading, some for their demos and coding, and some for their nobility. There was at least one music only group (VIBRANTS), and many demo only groups. It was a sign of status to be in a good group, and a good group could attract talent. There were also wars, disagreements and various other dynamics between groups, including splits, resurrections, mergers. Some groups were international, but that was not always a sign of high status or quality.

Most groups traded software and/or produced software. Some groups produced magazines, and ran bulletin boards, or arranged parties. There are many similarities between groups and bands of people, such as commercial companies.

3.5. Activities

Activities on the scene were varied, but revolved around the production and distribution of software.

3.5.1. Producing

Software productions were typically released under the name of a group, and were usually a collaboration between different members for demos, but often individual efforts for intros.

- **Intros** were designed as attachments to games, and would execute before the game itself executed. When a game was cracked, the Intro would be appended, and the result would be distributed. Distributors and end users would know who the original crackers were - elevating their status. Most intros were compact, and consisted of a logo, a few lines of text, a scrolling text, music, and a few stylistic features. Some groups were mostly known by their intros (e.g. IKARI or TALENT) as they were cracking groups more than they were demo producing groups.
- **Demos** were designed as a display of technical virtuosity and group skill - although they often displayed the opposite! Demos came in different forms - either single or multi-part productions, distributed as a single or multi-part file, consisting of a series of disjoint screens, or with some overriding theme. Some groups produced consistently high output, and were known for their quality or virtuosity (e.g. HORIZON).
- **Tools** were made by more serious coders - and were something of a sign of status, because they were rarer than demos, and usually more complex to construct. The most famous tools were packers and crunchers: who can forget the TIMECRUNCHER, SLEDGEHAMMER and ZIPPER.

In general, leading coders were continually advancing the state of the art through their work. There were innovations in all parts of the production process. The VIC chip was exploited to provide full screen display and then higher resolution graphics modes. The SID chip was exploited to provide better music, including “digi” style music. Novel approaches to manipulation of sound and graphics resulted in increased ability and performance of demos. One of the most impressive demos that I saw made use of the C1541 6502 processor: at the top of the raster scan, vectors would be uploaded into the C1541 and algorithms would be carried out so that results could be downloaded at the bottom of the raster scan.

3.5.2. Cracking

Cracking commercial software was a major activity in the scene , but rejected by some who were more concerned with producing demos

Typically, commercial software would be provide to crackers by suppliers, who sourced software from inside companies, through distribution houses, or end stores (sometimes the software was purchased off the shelf). Some groups did have people who specifically labelled themselves suppliers, but in other groups it may have been an informal role.

The crackers were responsible for taking the "original" and cracking it. They produced a cracked copy that would be distributed. Cracking involved removing copy protection and liberating the software from its original media into a more easily reproducible format: this may also include extracting multiple files (e.g. levels) and perhaps aggregating them into a single file, or creating fast loaders, or compressing levels. Sometimes cracks needed to re-create original feature such as high score tables and save game options, and sometimes cracks added them when they weren't already there.

It became increasingly common for cracks to have "trainers", such as options for infinite lives, infinite energy and other approaches to help players. There were a few games for which crackers wrote level builders, and options to load in new levels (e.g. BOULDERDASH) - the same thing happened with Doom some years later. A crack with a trainer and other options was usually more likely to be archived and used, and also be in higher demand - there is much similarity to real world products.

3.5.3. Distributing

Software that had been produced or cracked, needed to be distributed. The item would be handed over to a trader (or the cracker may have carried out part of the trading function). The trader then "spread" the item. Some traders were intermediataries - for example, a trader in Australia may have many European contacts, and many Australian contacts, and could have relayed material between them. Many traders were a mix: relaying products from other groups, and distributing products from their own group.

Traders everywhere other than the United States typically swapped by mail trading in the post or by hand. In the United States, modem trading was popular, and some traders were known as "runners": their role was to run software between bulletin boards. For instance, a hot crack from Europe may be uploaded to a group HQ board in the United States, and then distributed by runners to other boards. Runners may also take the cracks from boards in the states back to Europe - this could be a less expensive approach if telephone calls were made by 'phreaking'. As local calls were free in the United States, it was also possible to transfer files long distances by hopping across local zones: sometimes known as relaying.

Intros were usually added by the cracking group, but sometimes also added by importing groups. An import intro was usually much smaller - or sometimes the import intro, or textual note, would be stored on the same disk. Intermediaries often performed a more simple activity: aggregating software items onto disks, or placing directories entries with their group name.

The end of the distribution chain mostly consisted of gamers - many of whom purchased tapes or disks of software from someone who was connected into the distribution network.

3.5.4. Miscellaneous

There were many other activities in the scene, including:

- **Parties**: were organised as an excuse to meet people, have a good time, swap software, and release new demos. They were held all around the world, but most predominantly in Europe: the largest party was held in Venlo, Netherlands on a yearly basis.
- **Magazines**: were produced by some groups, and AMOK made a name for itself with its SEX'N'CRIME scene magazine. These provided news, gossip, interviews and various other sorts of information to help keep people informed. They were often released on a monthly basis, and had a similar format.
- **Collectors**: were those groups or people that amassed large collections of software, and specifically set out to do so. In a way, many sceners were collectors, but some took it on as their primary activity.

Section 2. C64 General References

1. Introduction

There are many resources covering all manner of content relating to the Commodore 64. This collection either relates directly to the content of the repository, or provides a reasonable starting point for further research.

All of these resources were active at documentation time, and while preference has been given to resources that should persist for a considerable length of time, it is inevitable that some may be invalid.

2. General

These resources typically do not provide any information, but provide pointers and references to Commodore 64 material.

2.1. Directories

Directories provide collections of links to all sorts of resources, and typically include information about relevance and popularity.

- **Google** (http://directory.google.com/Top/Computers/Systems/Commodore/Commodore_64/) – The Google search engine directory with automated popularity markers. Good. Also look at the general Commodore category. The search engine finds obscure material.
- **Yahoo** (http://dir.yahoo.com/Computers_and_Internet/Hardware/Systems/Commodore/Commodore_64) – The Yahoo search engine directory. Good. Also look at the general Commodore category.
- **Lycos** (http://dir.lycos.com/Computers/Systems/Commodore/Commodore_64) – The Lycos search engine directory. Good. Also look at the general Commodore category. The search engine is also good.

2.2. Repositories

Repositories are specifically targeted to the Commodore 64 and contain various sorts of content.

- **COCOS – Commodore Computer Site List** (http://www.c64.cc) – An excellent directory with over 500 references in a wide variety of categories, including ranking pages for popularity.
- **Commodore Ring** (http://nav.webring.yahoo.com/hub?ring=cbmring) – Commodore ring, but with varying quality entries that are a hit and miss.
- **8-bit core** (http://www.cbm64.de) – A small site that has brevity and compactness in its favour.

3. Background

These references cover the history of Commodore Business Machines and the Commodore 64.

3.1. Commodore Business Machines

The Commodore 64 was produced by the company "Commodore Business Machines (CBM)".

- **A Concise Commodore History** (http://members.nbci.com/advc/COM64.HTM) – The history of Commodore Business Machines through its various generations from 1958 to 1993.
- **The History of Commodore Business Machines** (http://cbmhistory.homestead.com) – Documents the history – in multiple parts – from origins through to Amiga and death. The references and bibliography is very good.
- **Commodore History** (http://amiga.emugaming.com/commodore.html) – Good historical coverage that includes recent resurrection efforts.
- **Commodore company History** (http://www.classicgaming.com/area64/machine.htm) – Three articles including the byte article "R.I.P. Commodore 1954-1994".
- **Commodore [Jones Digital Century]** (http://www.digitalcentury.com/encyclo/update/commodr.html) – A good overview of Commodore Business Machines.

3.2. Commodore Computers

Information that is focused less towards either Commodore Business Machines or the Commodore 64, but looks at the spectrum of Commodore computer products.

- **A Look at the Incredible History and Legacy of the Commodore Home Computers** (http://www.oldsoftware.com/history.html) – An excellent extensive and detailed history of the C64, C128, SX64, VIC20, +4, A500 and so on – well written and researched.
- **Commodore History** (http://www.commodorehistory.com) – A fairly extensive site dedicated to history of various Commodore computer ranges.

4. The System

These resources are typically directed towards the Commodore 64 as a system, including its historical context, its nature (hardware, software & documentation), and specific issues such as development, publications, emulators and reference material.

4.1. Overview

Specific historical or descriptive content about the Commodore 64 as a system itself.

- **OLD COMPUTERS: The Museum – Commodore > C64** (http://www.old-computers.com/museum/computer.asp?c=98) – A museum dedicated to many computers, of which C64 is just one. The C64 is given good treatment with a number of links.
- **Commodore 64 Overview** (http://home.datacomm.ch/fmeyer/c64/c64.html) – A very brief overview of the machine and its history, good for its conciseness, includes reasonable hardware detail.

4.2. Hardware

These resources are focused more directly towards the hardware of the Commodore 64, from various sorts of perspectives.

- **Commodore 64 Hardware Overview** (http://www.student.informatik.tu-darmstadt.de:8080/~mjk/nepa/dev/c64.html) – Framed by a general overview, most of this resource is focused on the hardware to a high level of detail – notably various system revisions and particular component information. Includes an excellent description of various devices.
- **Commodore 64 Schematics (funet)** (http://www.nic.funet.fi/pub/cbm/schematics/computers/c64/) – Large number of schematics of the machine and some of the devices at the funet archives.

4.3. Software

These resources are focused towards the software of the Commodore 64, either that of the system itself, or external software.

- **c64.com** (http://www.c64.com) – The best source of c64 games – a very comprehensive site that is professionally constructed.
- **c64demos.com** (http://www.c64demos.com) – An excellent collection of c64 demos as collated by a demo scene participant.
- **The Banana Republic** (ftp://ftp.elysium.pl) – A scene based FTP site with a large collection of demos, intros, magazines, utilities and scene related software.
- **Gangsta's Paradise** (http://c64.rulez.org/chromance/bbs) – A scene based http BBS system with a large collection of software.

4.4. Documentation

These resources are focused towards documentation for the Commodore 64, either that of the system or external hardware and software.

- **Project 64** (http://project64.c64.org) – This project aims to create repository of hardware, games, peripheral documents with games maps, and so on. The collection is large and impressive.
- **Commodore 64 Documents (funet)** (http://www.funet.fi/pub/cbm/documents/index.html) – Good collection of documentation of various sorts – although not comprehensive, it is easy to search and access.
- **Commodore Internet Archive** (http://cia.c64.org) – Contains games, cover scans, adverts scans, and various bits of information, aimed towards storing documentation and manuals for the C64.

4.5. Development

These resources are focused towards software development for the Commodore 64.

4.5.1. General

- **C=Hacking** (http://www.ffd2.com/fridge/chacking) – A technical publication for c64 focused on advanced technical stuff and scene oriented production activities.
- **The Fridge** (http://www.ffd2.com/fridge) – A code storage facility for c64 and 6502 programmers, which is very good.
- **The Commodore Knowledge Base** (http://www.retrobits.com/ckb) – A large number of varying technical articles and information on the C64, some 673 articles. Very good.

4.5.2. Compressors, Compression and related tools

These resources are focused on various aspects of compression for the Commodore 64.

- **Compression FAQ, Jean-loup Gailly,** (http://www.faqs.org/faqs/compression-faq) – A good general FAQ on compression that introduces concepts and terminology.
- **Compression Basics, Pasi 'Albert' Ojala,** (http://www.cs.tut.fi/~albert/Dev/pucrunch/packing.html) – A general article on compression covering basic theory and approaches from an author who is involved in Commodore 64 compression.
- **Commodore & Compression Page, Pasi 'Albert' Ojala,** (http://www.cs.tut.fi/~albert/Dev) – An overview page for Commodore Compression with lots of links to related material.
- **Milestones of C64 Data Compression, Pontus Berg,** (http://www.ffd2.com/fridge/chacking/c=hacking16.txt) – An overview of the history of C64 compression, not especially well written, but important from the perspective of a scene participant.
- **The secrets of Fast LZW Crunching, Antitrack/Legend** (http://www.noquake.de/domination/lzw-secrets.htm) – An article on LZW crunching using 6510 source code as example, from a scene participant.

- **Crunched Files (.xZx)** (http://www.floodgap.com/retrobits/ckb/display.cgi?298) – Details of Crunched files that delves into LZW and Huffman.
- **What is RLE-packing** (http://www.floodgap.com/retrobits/ckb/display.cgi?413) – An extremely brief overview of Run Length Encoding.
- **Crunchers and packers** (http://www.floodgap.com/retrobits/ckb/display.cgi?146) – Some information about crunchers used for late 80s demo crunching.
- **Sledge Hammer 2.2** (http://starbase.globalpc.net/driven/SledgeHammer2.2.txt) – Readme style file that provides a bit of insight into one of the scenes most popular "char packers".
- **Commodore 64 Packers (funet)** (http://www.ibiblio.org/pub/micro/commodore/c64/packers) – An archive of many C64 packers.

4.6. Publications

These publications were related to the Commodore 64, and were reasonably influential for various audiences (gamers, developers, etc).

- **Zzap! 64** (http://www.zzap64.co.uk) – Perhaps the most widely read and popular gaming publication originating in the United Kingdom.
- **go 64!** (http://www.go64.de) – A more generally oriented publication.
- **Commodore Magazine Nostalgia: Ahoy!, RUN and Compute!'s Gazette!** (http://www.geocities.com/dunric/ahoy.html) – An overview of Ahoy!, run & Compute!'s Gazette which were leading development oriented publications originating in the United States.

4.7. Emulators

A number of emulators have been produced for the Commodore 64. This is not a comprehensive list, but specifically those that have been used in the development of this repository.

- **Commodore 64 Emulator FAQ** (http://www.faqs.org/faqs/by-newsgroup/comp/comp.emulators.cbm.html) – A slightly outdated FAQ covering C64 emulators.
- **How to use emulators** (http://www.xp64.emuita.it/manualeng/index.htm) – A short but good description on how to use emulators. Worthwhile read for emulator newcomers.
- **VICE** (http://www.cs.cmu.edu/~dsladic/vice/vice.html) – A versatile Commodore emulator for multiple machines including MS-Windows.
- **CCS64** (http://www.ccs64.com) – The Commodore 64 emulator for PC, which is good for DOS based systems.

4.8. Reference

These resources provide useful reference material.

- **Commodore 64 FAQ** (http://www.faqs.org/faqs/commodore/) – USENET FAQ's that do not seem to have been updated in a while. Includes the FAQ from comp.sys.cbm, general FAQ, user group list, supplier list, repair facility list, emulation FAQ.

- **C64 pedia** (http://www.xp64.emuita.it/c64pedia/mainidx.htm) – A good single point of reference for terminology relating to the Commodore 64.

4.9. News/Information

These resources provide news and information about the Commodore 64.

- **Commodore Wire** (http://wire.c64.org) – An excellent news archive that is continually updated with relevant and quality material on all aspects of the C64.

5. The Scene

These resources are directed towards the scene that surrounded the Commodore 64 often known as the "demo scene" or the "trading scene".

5.1. Background

These are resources that cover the scene from a general perspective, or from the perspective of other specialised instances (e.g. PC, Amiga, etc).

- **THE HACKER DEMO SCENE AND IT'S CULTURAL ARTIFACTS, George Borzyskowsski** (http://www.curtin.edu.au/conference/cybermind/papers/borzysko.html) – A widely referenced academic paper on the Amiga scene, but directly relevant to the C64 scene.
- **Introduction to Demos & The Demo Scene, Vincent Scheib** (http://www.gamasutra.com/features/20010216/scheib_01.htm) – Provides a general introduction to the scene, but mostly aimed at the PC scene.
- **Introduction to demoscene** (http://www.fortunecity.com/skyscraper/black/791/edemint.htm) – Covers pc, Amiga and C64 with specific reference to actual demos released, even includes the CPC scene!
- **scene.org** (www.scene.org) – "Dedicated to all scenes", and has a good newslog along with a repository of demos.

5.2. Overview

These resources cover the Commodore 64 scene specifically.

- **The C64 scene** (http://www.fortunecity.com/skyscraper/black/791/edemint.htm#c64scene) – An introduction to demoscene focus on the C64 arena.
- **c64.org** (http://www.c64.org) – This site aims to collect references to scene members and links and other points.
- **Hall of Fame** (http://www.agt.bme.hu/hof/index.htm) – *"The online museum of the C64 scene"* but it doesn't have a lot of content.
- **C64 Scenery** (http://exotica.fix.no/info/scenery) – A guide to groups, parties and releases on the C64/Amiga demoscene. There's a huge amount of documentation about activities each year from 1986 to 2001.

5.3. Magazines

Various aspects of the scene were often documented in scene magazines. These are an excellent historical perspective on events that occurred.

- **C64-MAGS.COM** (http://www.c64-mags.com/index.shtml) – A large collection of scene magazines, including Sex'n'Crime which was one of the most highly lauded.

5.4. Personal (MATT')

These references cover my personal involvement with the scene. Any personal software created was created as part of the groups that I was involved in – so should be found there.

- **Interview with Matt/TERA, Vandalism News #32** (http://www.onslaught.f2s.com/interviews/int20.asp) – A recent interview in Vandalism News as documented on the Onslaught site. The original disk edition is a worthwhile read (http://c64.rulez.org/scenecity/stuff/c64/mags/vandalism/vn32.zip).
- **C64 Repository** (http://members.tripod.com/matthew_gream/C64/Repository.htm) – Repository covering my personal involvement with the scene, largely devoted to technical aspects of software produced.

5.5. Group (TERA, TDF, TDF+RFX)

These references cover my group as part of the scene, including pointers to software productions and repositories.

- **TOUR DE FUTURE at c64demos.com** (http://www.c64demos.com/htmlindx/demos/t/tourdefu/index.htm) – A collection of TOUR DE FUTURE software.
- **TOUR DE FUTURE at gangsta's paradise** (http://c64.rulez.org/chromance/bbs/ftp.cgi?Demos/t/Tour_De_Future) – A collection of TOUR DE FUTURE software.
- **REFLEX at c64demos.com** (http://www.c64demos.com/htmlindx/demos/r/reflex/index.htm) – A collection of REFLEX software.
- **TERA at c64demos.com** (http://www.c64demos.com/htmlindx/demos/t/tera/index.htm) – A collection of TERA software
- **TERA noters at NOTE-MAKERS archives** (http://www.chez.com/hibisch/c64/down_noters.html) – A collection of TERA/TDF notemakers.
- **TERA at gangsta's paradise** (http://c64.rulez.org/chromance/bbs/ftp.cgi?Demos/t/Tera) – A collection of TERA software.
- **TERA demos at SCENE CITY** (http://c64.rulez.org/scenecity/64demos.html) – A collection of TERA demos.

5.6. Australia

These resources cover the scene as it relates to Australia. There is a very good history of the Australian scene in Onslaught's Vandalism News #32.

- **Scene City** (http://c64.rulez.org/scenecity/news.html) – The home page of DJB/ONSLAUGHT and his demos from old Australian groups.
- **Onslaught HQ** (http://www.onslaught64.cjb.net) – The headquarters of a largely Australian group and its "Vandalism News" publication.

Section 3. C64 Activities Overview

1. General History

1.1. Beginning: TOUR DE FUTURE grows

1.1.1. Beginnings of a group and a coder

I purchased a C64 somewhere around 1986 because friends at school had one. This critical mass meant that we could quite easily swap software with each other. In particular, my friends BIZ, RECKLESS ROB, XLR8 and PARASITE owned C64's before I did, along with a couple of other friends of ours (who had no "handles"). I remember when I only had a few tapes of software, and one of my most treasured tapes was a collection of demos with one that included the song "everybody wants to rule the world", and one called "the zoolook demo".

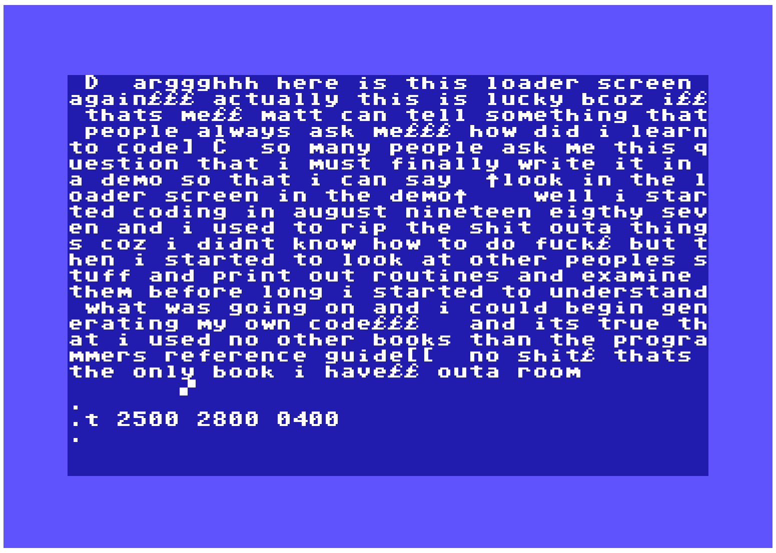
TOUR DE FUTURE (TDF) began somewhere between 1986 and 1987, with XLR8 and myself as the key drivers. The name was taken from a subway-graffiti group of similar name, as it had connotations of progress and was non-computer oriented – I remember being embarrassed at how the name was a bit of a rip-off. I called myself MATT', probably because I couldn't think of a good handle.

I was not really concerned about playing games, and I soon developed an interest in programming. I soon (late 1987, early 1988?) purchased a C1541 disk drive, a printer, and an Action Replay Cartridge. I do not remember the order of purchase, but it is likely that my parents paid for the disk drive and the printer (as it was used for school work), and I distinctly remember purchasing the Action Replay by mail order from Datel in the United Kingdom (XLR8 had purchased one earlier, and I was impressed). I also purchased the occasional C64 magazine (mostly those related to programming, such as Compute64! and Loader, rather than Zzap64!), and read books on programming at the library (both about BASIC and Assembler), and purchased a few odd books that were for sale in bookstores. I also remember going to Commodore 64 user groups, and I must have started swapping software by post around this time. XLR8 and I would make the occasional jaunt into the city in search of "cheap floppy disks".

I do not recall my first steps as a coder, but I do recall pouring over printouts of other code to understand how scrollers, colour bars and other routines worked. The Datel Action Replay cartridge had a monitor that allowed me to look at, and create, code. I remember writing my first TDF intro, which was quite horrible in retrospect, but it was the first step for me at the time. These were the nebulous steps of a group forming. RECKLESS ROB and BIZ were not really part of this activity, and they wandered off into other things (RECKLESS ROB was not really a "computer person", and BIZ dropped out of school for personal reasons). XLR8 was my trusty friend, and I remember a lot of time spent over at his house, playing with or stuffing around with software of some sort, and he would come over to my house. We did many things together at the time, and our Commodore 64 escapes were just one of them.

In the OHH MATE!! demo, I did write something about my starting steps, as extracted in the following screen capture.

Figure 3-1. Extract from OHH MATE!! about early coding days



```
 D  argggghhh here is this loader screen
again£££ actually this is lucky bcoz i££
 thats me££ matt can tell something that
 people always ask me£££ how did i learn
to code] C  so many people ask me this q
uestion that i must finally write it in
a demo so that i can say ↑look in the l
oader screen in the demo↑    well i star
ted coding in august nineteen eigthy sev
en and i used to rip the shit outa thing
s coz i didnt know how to do fuck£ but t
hen i started to look at other peoples s
tuff and print out routines and examine
them before long i started to understand
 what was going on and i could begin gen
erating my own code£££   and its true th
at i used no other books than the progra
mmers reference guide[[  no shit£ thats
the only book i have££ outa room

.
.t 2500 2800 0400
.
```

1.1.2. Developing contacts, reputation and status

It was in the year of 1988 that TDF began to make a name for itself in Australia. I continued to develop my contacts, and began build a reasonable collection of software (the collection totaled about 400 disks). XLR8 was also swapping software as well. It was through these contacts that we received new software of all sorts (demos, games and utilities), and I cannibalised a lot of them to learn the basics of coding. The first steps included learning how to rip music, code and graphics. The next steps involved being able to create my own code and graphics, but I never did take an interest in creating music. My contacts during this period began to include significant groups in Australia, and my valued contacts were DEVIET of WOT and BUCCANEER of WOT – but I also started to develop a good range of overseas contacts, mostly in Europe.

1.1.3. QCF Aussie Demo Competition in late 1988

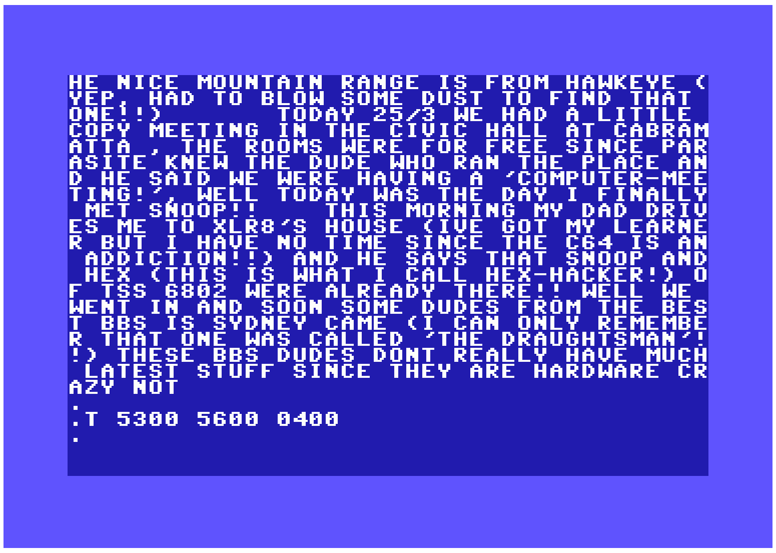
In November 1988, the QUEENSLAND CRACKING FORCE (QCF) proposed an “aussie demo comp” and I decided to submit an entry called BACK FROM BEYOND. I am not sure whether I had released any TDF productions before then (I was the only coder in the group so there was no one else to release anything) but that demo put TDF on the map and started to make my name as a coder. I think that it won the competition, but if it didn’t, then it rated highly. I think that I wrote the TDF NOTEMAKER V1.0 in the school holidays that bridged 1988 and 1989, and it was another “notable” product from our group. Various TDF intros were produced, and I think that we even cracked some originals. I remember that we had import intros that we would put on a lot of the software that we were receiving from our overseas contacts. All of these things continued to raise our profile, and I had many letters from people that wanted to swap.

1.1.4. Expanding our close friends in Sydney

We added MISTIC to the group in this period – he was involved in underground graffiti at the time, and I was a white-kid who was also peripherally involved (but as a hanger-on-er). We all had interests in rap and hip-hop music. MISTIC drew graphics and traded software, but I avoided using his graphics at first, as I did not consider them to be of good quality, but then he improved considerably, and I used his graphics in Solicitude when that reached a level that really impressed me. I always thought that MISTIC's major strength was as a trader, and as a team/group builder – for he was instrumental in taking the group to a larger stage (German division, and the USA board). Somewhere in the early part of 1989, we became friends with HEX-HACKER and SNOOP who had created their own double act called THE SOFT-SMASHERS 6802 (although, we may have even met them later than this – my memory is a little sketchy). We spent a reasonable amount of time hanging out with HEX-HACKER and SNOOP, especially in the formers Nissan Urvan (driving the streets, seeing movies, hanging out). We were a bunch of young male computer nerds unleashed on the world with a big white van – it's a scary thought.

The HUH demo was produced about this time. It documents a copy-party meeting that we had at the civic centre in Cabramatta – thanks to PARASITE and connections through his mother. We had known of HEX-HACKER and SNOOP through some other medium, and finally met them on that day. The following scroll text extract has the details.

Figure 3-2. Extract from HUH about Cabramatta Civic Centre party



```
HE NICE MOUNTAIN RANGE IS FROM HAWKEYE (
YEP, HAD TO BLOW SOME DUST TO FIND THAT
ONE!!)        TODAY 25/3 WE HAD A LITTLE
COPY MEETING IN THE CIVIC HALL AT CABRAM
ATTA , THE ROOMS WERE FOR FREE SINCE PAR
ASITE KNEW THE DUDE WHO RAN THE PLACE AN
D HE SAID WE WERE HAVING A 'COMPUTER-MEE
TING!', WELL TODAY WAS THE DAY I FINALLY
 MET SNOOP!!     THIS MORNING MY DAD DRIV
ES ME TO XLR8'S HOUSE (IVE GOT MY LEARNE
R BUT I HAVE NO TIME SINCE THE C64 IS AN
 ADDICTION!!) AND HE SAYS THAT SNOOP AND
 HEX (THIS IS WHAT I CALL HEX-HACKER!) O
F TSS 6802 WERE ALREADY THERE!! WELL WE
WENT IN AND SOON SOME DUDES FROM THE BES
T BBS IS SYDNEY CAME (I CAN ONLY REMEMBE
R THAT ONE WAS CALLED 'THE DRAUGHTSMAN'!
!) THESE BBS DUDES DONT REALLY HAVE MUCH
 LATEST STUFF SINCE THEY ARE HARDWARE CR
AZY NOT
.
.T 5300 5600 0400
.
```

The HUH demo was destined for an Australian demo competition of some sort, apparently held by COLWYN of THE FORCE. I really do forget much of the details, but the demo is further advanced than BACK FROM BEYOND and shows increasing maturity of routines, and stylistic advances.

1.2. Consolidating: TOUR DE FUTURE + REFLEX merge

1.2.1. The merger of two complimentary groups

In early 1989, TDF had solidified its reputation. I created the TDF NOTEMAKER V1.0 somewhere around this period. It was very well received overseas, and gave us a name in Europe. We developed further contacts, one of which was a group in Victoria called REFLEX. They proposed a merger, and we accepted – it provided us with good traders, and them with an experienced coder. The merger was reported in SEX'N'CRIME 04 (May 1989). We chose to represent ourselves as "TDF+RFX" to promote the merger until we could come up with a better name, and we also used the merger as a way to remove underperforming members and consolidate "who we were". REFLEX provided BOSS, BULLET, ELIMINATOR & STOREMAN; whereas TOUR DE FUTURE provided MATT', XLR8 and MISTIC. We added KILENEMY of

QCF, as reported in SEX'N'CRIME 05 (July 1989). During 1989 I was receiving 10-15 parcels a week, and sending out just as many; most of my nights and weekends were spent with the guys, or carrying out these activities.

1.2.2. TEC-Illegal copy party in Sydney in mid 1989

TEC decided to hold a copy party in Sydney in early June 1989 called TEC-Illegal – eventually reported in SEX'N'CRIME 04. We decided to attend, and I created a demo called OHH MATE!!. The demo received favourable comments in SEX'N'CRIME 05 (July 1989), but I am not sure whether it won the competition at the party. I think THE FORCE demo won, but it won under dubious circumstances – but I could be wrong. SEX'N'CRIME 05 also reported us as #2 in Australia. The party also included graffiti antics by the other guys that almost landed them in trouble with the law!

OHH MATE!! is increasingly advanced from HUH, but does not reach out to the level of DAMNABLE. It is interesting to find that some of the DAMNABLE concepts are there, including activity based loading screens. Stylistically it is interesting, and clearly shows that I have started to become involved by modem activities. The demo is created on a modem theme. There are references to 1670's in the scroll text, so perhaps I already had one by then.

Around this time, I released a few Intros for the merged group, and BOSS and BULLET also released demos under the "TDF+RFX" banner, while ELIMINATOR created graphics. I was mostly concerned with producing code, and really did not take enough notice about our other group members and what they were up to, so I am not sure that I had a great deal to do with the guys in Victoria and Queensland. With hindsight, I release that I didn't really feel too much a part of the Sydney group either.

1.2.3. Reflections

The TDF+RFX phase was quite an important milestone for many reasons. It signaled the maturity of the group in the Australian scene, and it was the point at which the group, and my skills, reached a world-class level. That sounds a little self-serving, but I think that a fair analysis would show that our productions were on the same level as the top 20% (or better) of those produced in Europe: good style, good code, first class features. Both of our groups had matured from "ad hoc" collections of people into something more rigorously defined: members had to contribute rather than just "hang on". Roles and responsibilities were being taken more seriously. I remember that I offloaded some of my contacts to BOSS and BULLET so that I could concentrate on coding. Our trading was of increasingly better quality, as the REFLEX guys were very good traders, and in Sydney we also were improving our contacts.

1.3. Maturing: TERA is born

1.3.1. Defining and creating a new identity in Australia

After short integration period, TERA was born. I remember that the name was chosen because it had connotations of Australia ("Terra Australis" was an original name for Australia), but it was also a short and compact name with a graffiti background (a major "piece" along the train line near Redfern in Sydney – and one of the TERA demos includes a replica of the "piece"). SEX'N'CRIME 07 (Sept 1989) reported the formation of TERA. We were like many groups (music, business, etc) that reached a new point of maturity and redefined their image. We admitted SNOOP and HEX-HACKER at the same time. SNOOP mostly traded (and sold) software, while HEX-HACKER was a decent coder. He produced a few demos, and related utilities, and in general, he made worthwhile contributions to the group. BOSS and BULLET released a demo called ATMOSPHERIC after our formation. I thought BOSS, BULLET and ELIMINATOR were good solid coders – they were not innovative or technically sophisticated, but they had a nice style that seemed distinctly Australian.

1.3.2. The group takes a foothold overseas

The group, at least in Sydney, started to develop presence overseas thanks to the efforts of MISTIC. I had purchased a C1670 modem (perhaps as early as late 1988), and we frequented Bulletin Board Systems in the United States (and I called locally in Sydney, but under my real name). We were occasionally invited onto phone conferences (sometimes referred to as "bridges") by guys from "the states". Somewhere in here began our growing anger at COLWYN of ACU (later, COLWYN of THE FORCE), as we were unimpressed with his behaviour. He occasionally slagged us off for no good reason other than to further himself, and he patronised foreigners with idiotic stories about Australians (i.e. that we had kangaroos as pets). My intellectual curiosity started to find bulletin boards interesting, especially those that carried text files, such as RIPCO. I did not know it at the time, but thus began the activities that I would find more interesting than coding.

1.3.3. Continuing to release leading productions

Anyway, I did continue to produce and develop as a coder. Somewhere in this period, I developed BEEFTRUCKER V1.0 and NOTEMAKER V2.0. Both of these products were well received, and were partially created in order to "get our name out there". They illustrated my maturity as a coder, because producing something like an RLE packer was a notable achievement for a coder and even more so because BEEFTRUCKER broke a world-record, and generally presented itself as a well-rounded product. Towards the end of 1989, I had major school examinations to deal with, so I slacked off on trading and producing for a couple of months – although, I still remember talking on phone conferences only days before my exams. Because of our close friendship with WOT, we'd also had disk covers and a few other things drawn for us by one of their master drawers in Queensland. In retrospect, late 1989 and early 1990 were my high points as a coder.

1.3.4. THE FORCE copy party in Adelaide in late 1989

Our next big milestone occurred in late 1989, when we decided to attend THE FORCE COPY PARTY in Adelaide, held in the first weeks of December 1989. In the weeks leading up to this, I worked overtime to produce a demo called DAMNABLE. It was a proud achievement, as I had created all code (including fast loader and novel routines), and I had created all graphics (logos, sprites and character sets) – it was put together as a nice package, and even packed with our own packer (one of the reasons I avoided using graphics from other guys in the group was because of my ambition to produce a mostly self-created production). For me, DAMANBLE and BEEFTRUCKER remain the highlights of my coding career. Most of the work for DAMNABLE was carried out in Sydney in the weeks leading up to December. We were able to include a "last minute surprise": excellent MANIACS OF NOISE music ripped from TURBO OUTRUN, which would have only just arrived in Australia by the time we reached Adelaide. We had obtained it from the United States on the night of its release (a hot release). BOSS and BULLET also produced a demo for this party, called RIPOFF, but I really recall very little about this.

We left Sydney bound for Adelaide by train: MISTIC, HEX-HACKER, XLR8 and I. We took 12hrs to reach Melbourne, and then changed for a smoke-ridden overnight train to Adelaide to arrive there at 7am in the morning. I can still remember coming down through the Adelaide Hills. MADHACKER of IKON VISUAL took care of us. I do not remember the party very well, but I do remember that we finished and put the demo together as a releasable product. We may have even created the propaganda notes at the last minute during the party. I remember at least one event that made me feel a bit like an outsider, but I was going through a difficult time as my girlfriend had moved away from Sydney. HEX-HACKER and myself sat on a roundabout in Adelaide and moaned about our miseries. We traveled back to Sydney by bus, and I think that I slept most of the way. It was a long and arduous journey, but DAMNABLE had won the competition, and our leading name was gaining further strength.

1.4. Detachment: TERA peaks and fades

1.4.1. The group continues, but Matt' wanders off

As I had finished high school in November 1989, I had a lot of free time on my hands – but I also had the problem of a girlfriend who had relocated to Canberra (it didn't last as a long-distance-relationship). A number of parts for SOLICITUDE were completed over the following months, and our group spent a reasonable amount of time together as we all had a lot of free time with no work or study commitments. We ended our war with COLWYN of THE FORCE, as reported in SEX'N'CRIME 11 (Jan 90), and HEX-HACKER had problems with his employers as a result of our modem calling activities: this was reported in SEX'N'CRIME 12 (Feb 90). MISTIC obtained a board for us in the United States called THE JUNGLE: a dedicated TERA outpost. We also started finding an interest in our group for foreign members, and MISTIC shepherded the development of a German division. I really didn't recognise that this was a major new path of growth for us, the group was becoming international: this was a significant milestone.

During this time, I drifted out of my C64 interests, largely because I started calling bulletin boards in the United States, and collecting text files and learning about a "new world" called HPVAC. I idled more of my time away calling boards instead of coding – but I was also working and studying. There was some sourness in the end of TERA, because our access to calling cards meant that members seemed more interested in material gains, and I think that we were feeling the tensions normally felt a group where members are buying-in for different reasons. For me, October 1990 was the definite end: I was having a difficult time with university and my girlfriend, and ended up on the receiving end of a prank by the TERA guys that pushed me over the edge. I think that I reacted worse than I should have though.

HEX-HACKER released LOVESTRUCK in March 1990 as part of his infatuation with my sister. He thought that my estrangement with him and the group was because of this, but as I recall, it didn't really bother me, as I knew that my sister was destined for a different type of person. HEX-HACKER had also produced ENCAPSULATE in February 1990, and the WESTPAC DEMO around the same time. He also produced two NOTEMAKERS, which were strong derivations of my own – the first was released as early as late 1989, but I am not sure when he released the second one. In August 1990, SOLICITUDE was put together by BOSS and BULLET and released. It contains parts that I developed earlier in the year, and some of the scroll text mentions the Royal Easter Show (held in Sydney in March/April). It included parts by German members that I was not familiar with as I had left. The final demo released by TERA was in 1992, under the name GANGSTA. The note accompanying the demo indicates that TERA was DEAD. It definitely was by then.

1.4.2. Reflections

In hindsight, I can see that TERA was on the up in late 1989, and as the lead coder in the group, I left at the worst time for them. I realise now that it was a little unfair of me, and it cut the output of the group – but I had always been a solo coder, and a large amount of our reputation was built on my coding efforts. REFLEX explicitly stated that TOUR DE FUTURE offered an "experienced coder" into the merger. The productions that I released as part of TERA were widely considered to be world-class. After I left TERA, only BOSS and BULLET remained as leading coders, but HEX-HACKER also continued to produce output. The SOLICITUDE demo is telling: it is good, but it doesn't compare to the stylistic integrity and advanced technical achievements of DAMNABLE: the technically advanced parts of SOLICITUDE were my creations.

In any case, by late 1990, I had completely given up my C64 interest, and either before then, or not long later, I parted with all hardware and software. I did not start to resume contact with the TERA guys until 1997, and I did not return to look at my C64 productions until 2001.

2. Coding History

2.1. General development of a coder

2.1.1. Overview

"coding is an artform" – solicitude/tera

I don't remember whether I wrote that or not – but I believe it.

My coding on the C64 started with a monitor and a lot of reverse engineering. The Commodore 64 Programmers Reference Guide also helped, and was a bible for all coders.

To become a proficient coder required mastery of various things.

2.1.2. Obtaining production material, through ripping code, music and graphics

It was rare for anyone to be able to provide everything required for an intro or a demo. A coder needed to know how to rip code, music and graphics from other software, preferably commercial software, as it was considered "lame" to rip it from other intros or demos. Typically, most coders soon wrote all of their own code, and typically most groups had a graphics person (or, the coder was multi-skilled), and so a production could rely on all original code and graphics. Few groups had music, so nearly all music was ripped from elsewhere. A more advanced coder was able to relocate code, music and graphics, or to alter it in some way.

2.1.3. Assembling the production by relocating, linking, packing, crunching

This was needed to put together a complete package after coding. The coder usually required routines to relocate code, or to link different modules together, and then the ability to pack and crunch modules as well.

2.1.4. Creating the production by learning the basic technical skills.

The basic requirement for most coders was to master scroll text routines, logo placement routines, and raster manipulation routines. Colour bars were an early form of raster manipulation, and required precise timing to ensure visual stability. The same raster level routines could be used to keep side or top/bottom borders “open” for a full screen display. A basic intro or demo usually consisted of a logo display, scrolling text, moving sprites, music, and perhaps a few other special effects. Most of the activity was coordinated within ISR routines – so a coder had to master these basics. A coder also had to create sinusoid tables, and colour tables.

2.1.5. Learning advanced technical skills and compositional approaches.

With increased mastery, it became possible to do sophisticated things with the basics, such as multiplexed sprite routines, or manipulating various VIC registers, or creating scrolling texts with speed and size control or other special effects. In this phase, a coder typically also became better at composition: better stylistic integrity, a thematic or narrative consistency, matching appropriate sounds, images and effects, etc. This is where a coder moves to becoming a designer.

2.1.6. Mastering all parts of the machine and the production process.

Master coders could do just about anything, and they typically broke barriers by pushing the Commodore 64 hardware to the limits, or developing new sorts of approaches to intros or demos. Master coders could produce game level complexity, and could create (or at least understand) copy production, and system I/O, and typically had some mastery of the peripheral chips. There are a few groups on the scene that are known for their code mastery – their productions consistently broke new ground. At this level, there become differences between coders and designers. There were many coders who could do advanced technical routines, but were poor at overall design and composition; conversely there were good designers who were poor coders.

2.2. Personal development as a coder

2.2.1. Beginning

My early work involved ripped music and graphics, and even some ripped code, as was the case with all new starters. I was soon capable of producing my own code and graphics, with BEEFTRUCKER and DAMNABLE being two productions that I had create all code and graphics for.

2.2.2. Coding

I started with the Datel Action Replay monitor, and my early work often had a lot of code at $100 memory boundaries, as this was a much easier way of arranging different parts of the code. This meant that if I needed to expand something, then I did not need to relocate a lot of other code. Eventually, I migrated to Turbo Assembler: all releases made for TERA were created with Turbo Assembler, as were some of the later ones for TOUR DE FUTURE. Turbo Assembler allowed reuse, produced compact code, and provided label based branching and addressing. Ultimately, I began to write code that generated other code. The special effects in DAMNABLE and SOLICITUDE were made possible with pre-computed code. When a particular demo part starts, it creates a swathe of highly efficient code that is used during the execution of the demo. This was in-fact quite easy to do, and very enjoyable. I had also become adept at obfuscating code, to make it harder for rippers who wanted to take away my stuff.

2.2.3. Graphics

I started creating fairly basic logos and sprites, which were very bland. I created a number of utilities that would merge bitplanes, or "compact" graphics, or modify graphics. This allowed me to automate parts of the production process, or to come up with effects that would be difficult to create by hand. I forget the name of the tool that I used. It was primarily designed to create character sets, although I would use it to create logos, patterns, objects and various other items. Many of the logos were created in three phases: (1) create a shell construction, then (2) create a set of fill patterns, and then (3) apply the fill patterns. I needed to use my own routines for this purpose. I would also base the logos and character sets on mathematical similarities, so that I could easily cut and paste (e.g. corners, edges, curves, letter components, etc). It was all very pattern oriented. With BEEFTRUCKER and DAMNABLE I had created all graphics, and then in SOLICITUDE I went one step further and created the graphics "on the fly" as part of the demo's execution. There are many similarities between my logos and character sets, I often employed the same sort of basic style.

2.2.4. Composition

I learned to rip code, music and graphics very early. It was one of the most important tasks to come to grips with, even before learning to code. Eventually, I only needed to rip music, as I could create all code and graphics myself. I had also reached the point where I could break through sophisticated copy-protection. I remember making my way through protection that would continuously decode future instructions based upon current stack, register and status flags contents – all in an attempt to defeat cartridge monitors. I had learned how to relocate and modify other code and graphics. Graphics was easy to relocate, music was a little harder, but some code was more difficult. I was able to rip MATCHAM's packer for use in my notemaker products, and I also ripped a fast disk save routine from somewhere, and I built a fast load routine out of parts from other fast load routines.

I mastered all other parts of the production process. Packing and linking was easy, and I eventually wrote BEEFTRUCKER. I tried to take a lot of care with this process, as many groups just bundled different parts together in a horrible sort of manner, with "blue screen" resets between parts, and ugly unpacker junk on the screen. I thought that it was worth the effort to take care of these little details, because they really made the whole package happen at the end of the day. It was unfortunate to see a technically sophisticated part marred by a junky assembly approach.

2.2.5. Relationships

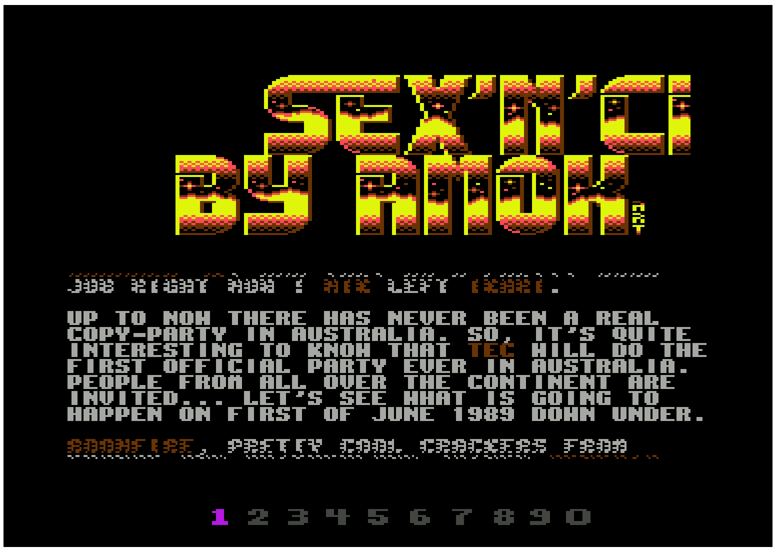
Most of my contacts were software traders, as I did not know any other coders. My first coder-friends were HEX-HACKER and SNOOP, more so the former than the latter. I suspect that I helped XLR8 learn to code, but I am not sure to what extent. I definitely spent some time sharing insights with HEX-HACKER. It was flattering to know that some of the other members in my group derived their own work from mine, as did other coders in Australia. I did have regular pen-friend contact with JERRY of TRIAD who once compared me favorably with MR Z of TRIAD.

3. Appendix

3.1. Reportage in Sex'n'Crime by AMOK

SEX'N'CRIME by AMOK was one of the premiere scene magazines in the late 1980's, and captured some of TOUR DE FUTURE, REFLEX and TERA activities, including the TEC-Illegal party in Sydney.

Figure 3-3. Sex'n'Crime #4 (06/89) – TEC-Illegal party announcement

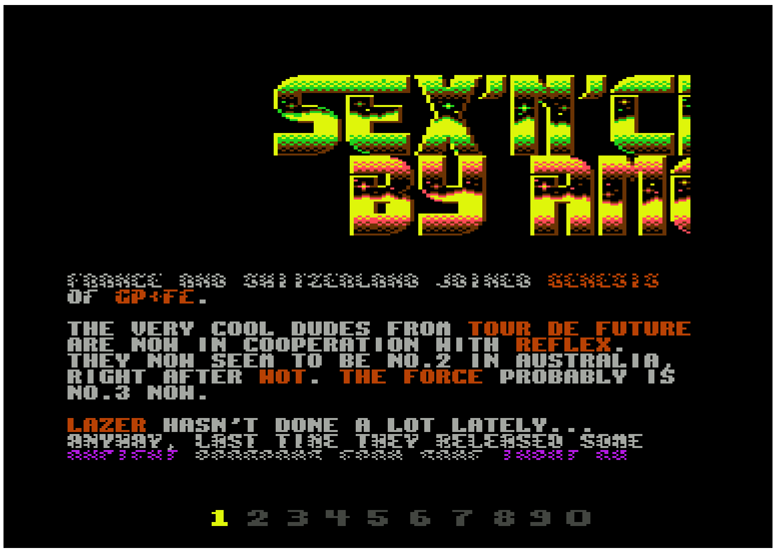


JOB RIGHT NOW ? [illegible] LEFT [illegible].

UP TO NOW THERE HAS NEVER BEEN A REAL COPY-PARTY IN AUSTRALIA. SO, IT'S QUITE INTERESTING TO KNOW THAT TEC WILL DO THE FIRST OFFICIAL PARTY EVER IN AUSTRALIA. PEOPLE FROM ALL OVER THE CONTINENT ARE INVITED... LET'S SEE WHAT IS GOING TO HAPPEN ON FIRST OF JUNE 1989 DOWN UNDER.

BONFIRE, PRETTY COOL CRACKERS FROM

Figure 3-4. Sex'n'Crime #4 (06/89) – TDF/RFX co-operation

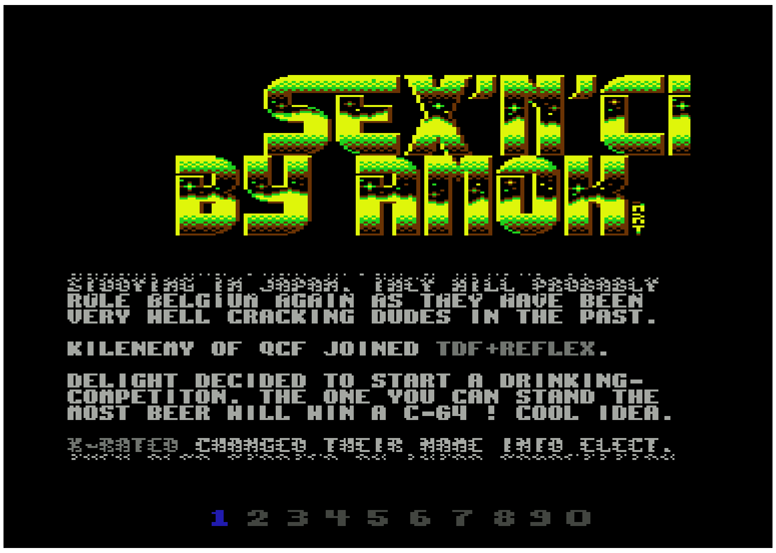


FRANCE AND SWITZERLAND JOINED GENESIS OF GP+FE.

THE VERY COOL DUDES FROM TOUR DE FUTURE ARE NOW IN COOPERATION WITH REFLEX. THEY NOW SEEM TO BE NO.2 IN AUSTRALIA, RIGHT AFTER HOT. THE FORCE PROBABLY IS NO.3 NOW.

LAZER HASN'T DONE A LOT LATELY... ANYWAY, LAST TIME THEY RELEASED SOME [illegible]

Figure 3-5. Sex'n'Crime #5 (07/89) – Kilenemy joins TDF+RFX



STUDYING IN JAPAN. THEY WILL PROBABLY
RULE BELGIUM AGAIN AS THEY HAVE BEEN
VERY WELL CRACKING DUDES IN THE PAST.

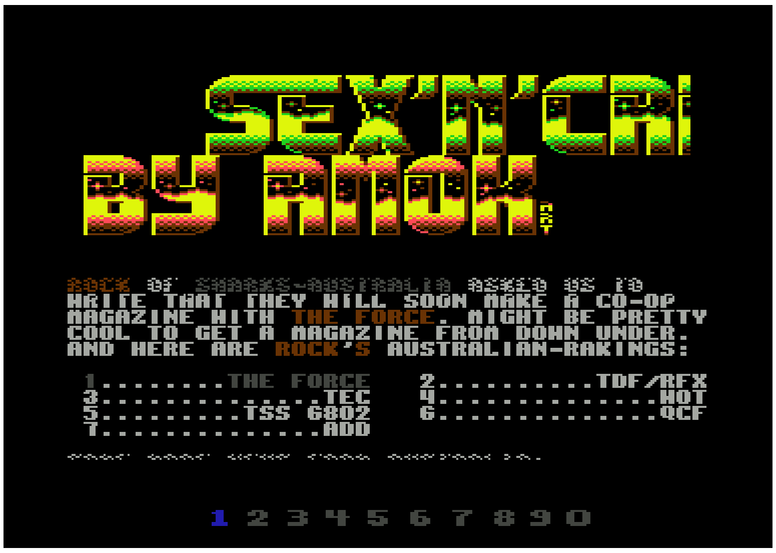
KILENEMY OF QCF JOINED TDF+REFLEX.

DELIGHT DECIDED TO START A DRINKING-
COMPETITON. THE ONE YOU CAN STAND THE
MOST BEER WILL WIN A C-64 ! COOL IDEA.

X-RATED CHANGED THEIR NAME INTO ELECT.

1 2 3 4 5 6 7 8 9 0

Figure 3-6. Sex'n'Crime #5 (07/89) – TDF/RFX rank in Australia



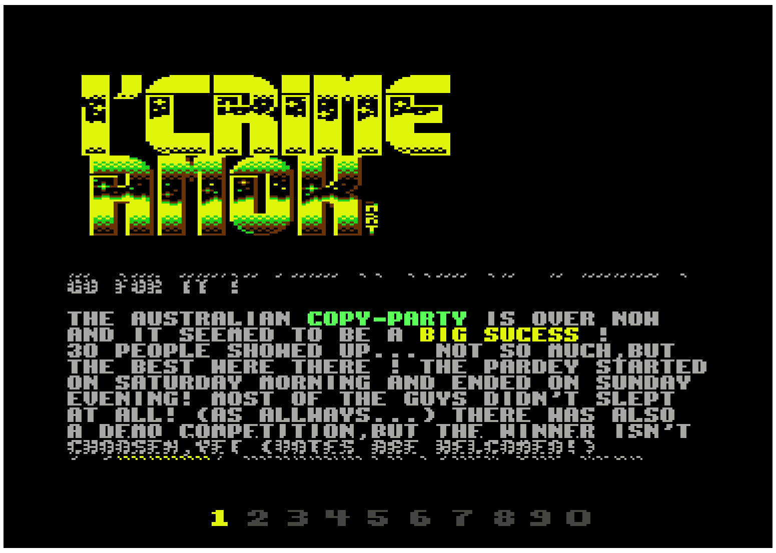
ROCK OF [illegible]-AUSTRALIA ASKED US TO
WRITE THAT THEY WILL SOON MAKE A CO-OP
MAGAZINE WITH THE FORCE. MIGHT BE PRETTY
COOL TO GET A MAGAZINE FROM DOWN UNDER.
AND HERE ARE ROCK'S AUSTRALIAN-RAKINGS:

```
 1.........THE FORCE   2..........TDF/RFX
 3...............TEC   4..............HOT
 5..........TSS 6802   6..............QCF
 7...............ADD
```

[illegible]

1 2 3 4 5 6 7 8 9 0

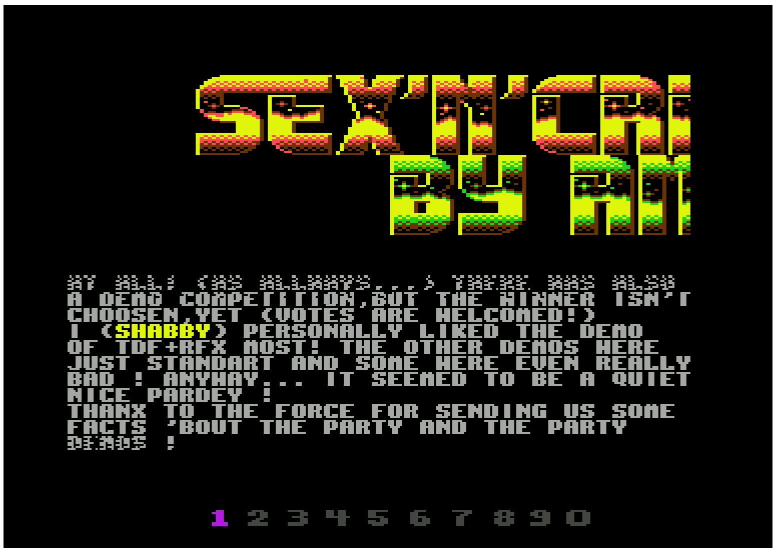
Figure 3-7. Sex'n'Crime #5 (07/89) – TEC-Illegal party report #1



[illegible]
GO FOR IT !

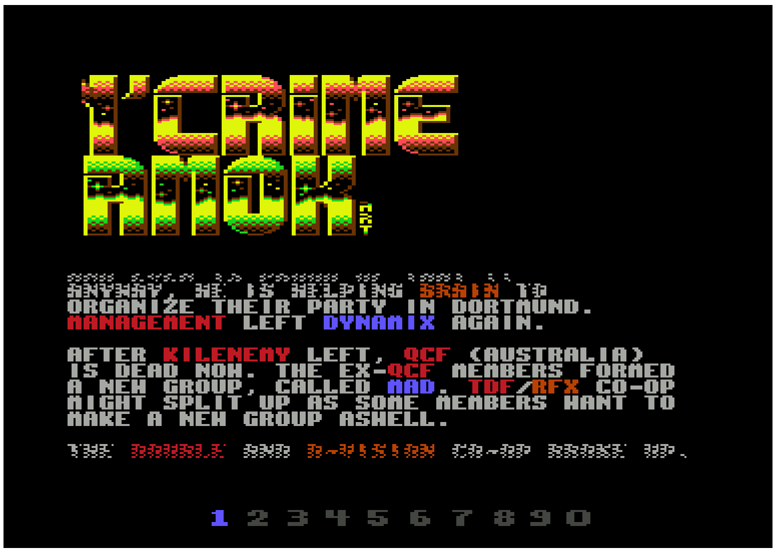
THE AUSTRALIAN COPY-PARTY IS OVER NOW
AND IT SEEMED TO BE A BIG SUCESS !
30 PEOPLE SHOWED UP... NOT SO MUCH,BUT
THE BEST WERE THERE ! THE PARDEY STARTED
ON SATURDAY MORNING AND ENDED ON SUNDAY
EVENING! MOST OF THE GUYS DIDN'T SLEPT
AT ALL! (AS ALLWAYS...) THERE WAS ALSO
A DEMO COMPETITION,BUT THE WINNER ISN'T
CHOOSEN,YET (VOTES ARE WELCOMED!)

Figure 3-8. Sex'n'Crime #5 (07/89) – TEC-Illegal party report #2



AT ALL! (AS ALLWAYS...) THERE WAS ALSO
A DEMO COMPETITION,BUT THE WINNER ISN'T
CHOOSEN,YET (VOTES ARE WELCOMED!)
I (SHABBY) PERSONALLY LIKED THE DEMO
OF TDF+RFX MOST! THE OTHER DEMOS WERE
JUST STANDART AND SOME WERE EVEN REALLY
BAD ! ANYWAY... IT SEEMED TO BE A QUIET
NICE PARDEY !
THANX TO THE FORCE FOR SENDING US SOME
FACTS 'BOUT THE PARTY AND THE PARTY
DEMOS !

Figure 3-9. Sex'n'Crime #6 (08/89) – TDF/RFX split rumours

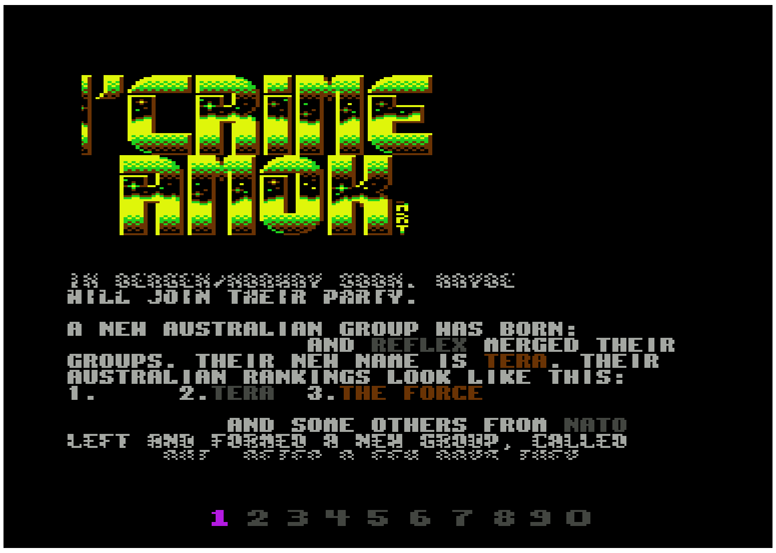


[illegible]
ANYWAY, HE IS HELPING BRAIN TO
ORGANIZE THEIR PARTY IN DORTMUND.
MANAGEMENT LEFT DYNAMIX AGAIN.

AFTER KILENEMY LEFT, QCF (AUSTRALIA)
IS DEAD NOW. THE EX-QCF MEMBERS FORMED
A NEW GROUP, CALLED MAD. TDF/RFX CO-OP
MIGHT SPLIT UP AS SOME MEMBERS WANT TO
MAKE A NEW GROUP ASWELL.

THE DOUBLE AND D-VISION CO-OP BROKE UP.

Figure 3-10. Sex'n'Crime #7 (08/89) – TERA born from TDF/RFX

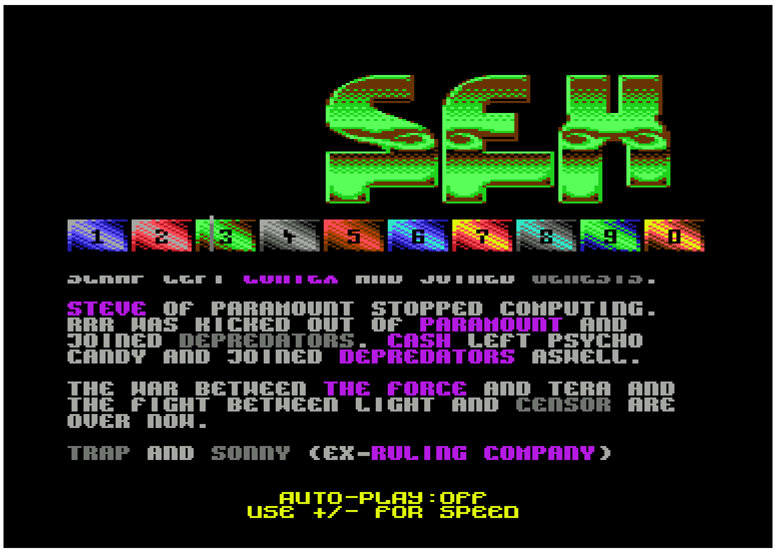


IN BERGEN/NORWAY SOON. MAYBE
WILL JOIN THEIR PARTY.

A NEW AUSTRALIAN GROUP HAS BORN:
AND REFLEX MERGED THEIR
GROUPS. THEIR NEW NAME IS TERA. THEIR
AUSTRALIAN RANKINGS LOOK LIKE THIS:
1.   2.TERA   3.THE FORCE

AND SOME OTHERS FROM NATO
LEFT AND FORMED A NEW GROUP, CALLED
[illegible]

Figure 3-11. Sex'n'Crime #11 (01/90) – TERA & THE FORCE war



SCRAP LEFT CONTEX AND JOINED GENESIS.

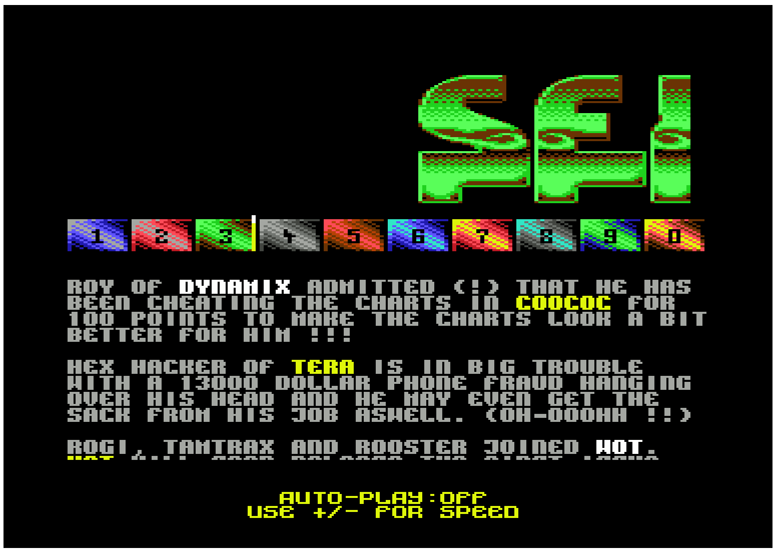
STEVE OF PARAMOUNT STOPPED COMPUTING.
RRR WAS KICKED OUT OF PARAMOUNT AND
JOINED DEPREDATORS. CASH LEFT PSYCHO
CANDY AND JOINED DEPREDATORS ASWELL.

THE WAR BETWEEN THE FORCE AND TERA AND
THE FIGHT BETWEEN LIGHT AND CENSOR ARE
OVER NOW.

TRAP AND SONNY (EX-RULING COMPANY)

AUTO-PLAY:OFF
USE +/- FOR SPEED

Figure 3-12. Sex'n'Crime #12 (02/90) – HEX-HACKER in trouble



ROY OF DYNAMIX ADMITTED (!) THAT HE HAS
BEEN CHEATING THE CHARTS IN COOCOC FOR
100 POINTS TO MAKE THE CHARTS LOOK A BIT
BETTER FOR HIM !!!

HEX HACKER OF TERA IS IN BIG TROUBLE
WITH A 13000 DOLLAR PHONE FRAUD HANGING
OVER HIS HEAD AND HE MAY EVEN GET THE
SACK FROM HIS JOB ASWELL. (OH-OOOHH !!)

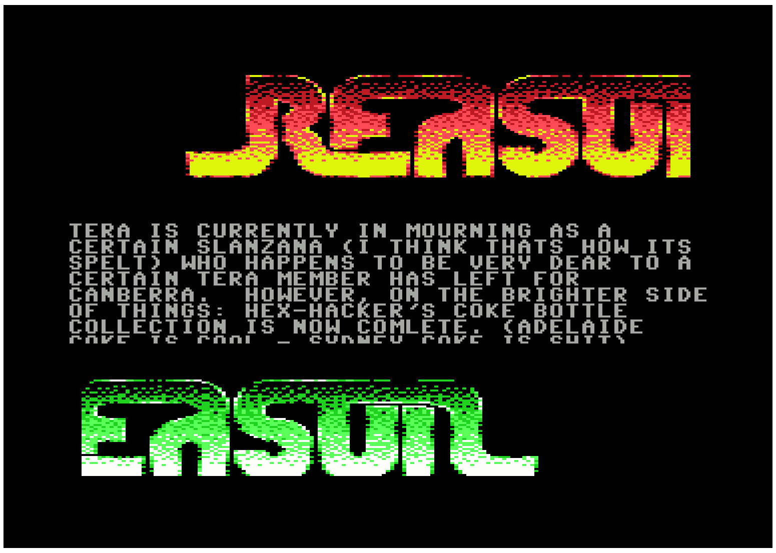
ROGI, TAMTRAX AND ROOSTER JOINED HOT.
[illegible]

AUTO-PLAY:OFF
USE +/- FOR SPEED

3.2. Reportage in Reason 4 Treason by IKON VISUAL

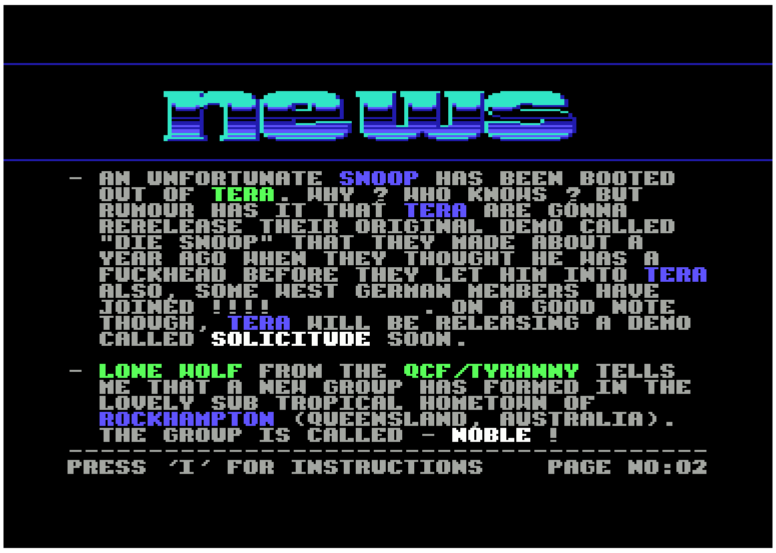
IKON VISUAL were another good group on the Australian scene in the late 1980's and early 1990s. There's some content in Reason 4 Treason relating to TERA.

Figure 3-13. Reason 4 Treason #1 – TERA loses valued inspiration



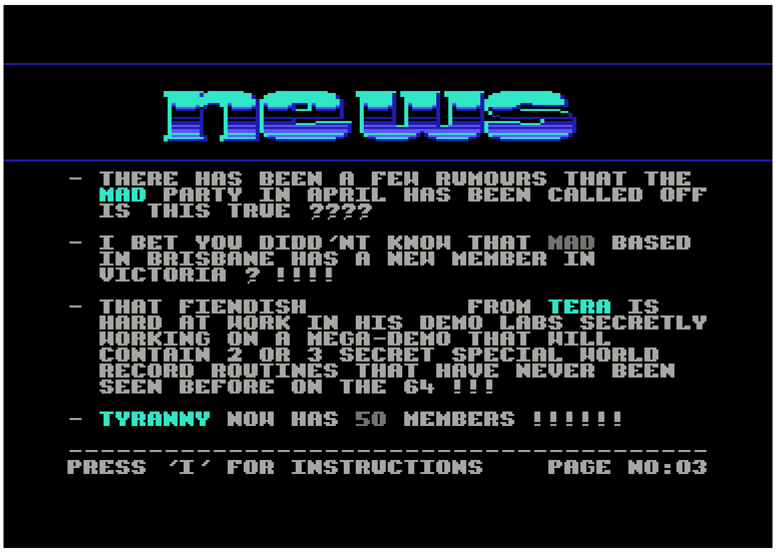
```
TERA IS CURRENTLY IN MOURNING AS A
CERTAIN SLANZANA (I THINK THATS HOW ITS
SPELT) WHO HAPPENS TO BE VERY DEAR TO A
CERTAIN TERA MEMBER HAS LEFT FOR
CANBERRA.  HOWEVER, ON THE BRIGHTER SIDE
OF THINGS: HEX-HACKER'S COKE BOTTLE
COLLECTION IS NOW COMLETE. (ADELAIDE
COKE IS COOL - SYDNEY COKE IS SHIT)
```

Figure 3-14. Reason 4 Treason #2 (a) – SNOOP and SOLICITUDE



```
 - AN UNFORTUNATE SNOOP HAS BEEN BOOTED
   OUT OF TERA. WHY ? WHO KNOWS ? BUT
   RUMOUR HAS IT THAT TERA ARE GONNA
   RERELEASE THEIR ORIGINAL DEMO CALLED
   "DIE SNOOP" THAT THEY MADE ABOUT A
   YEAR AGO WHEN THEY THOUGHT HE WAS A
   FUCKHEAD BEFORE THEY LET HIM INTO TERA
   ALSO, SOME WEST GERMAN MEMBERS HAVE
   JOINED !!!!             . ON A GOOD NOTE
   THOUGH, TERA WILL BE RELEASING A DEMO
   CALLED SOLICITUDE SOON.

 - LONE WOLF FROM THE QCF/TYRANNY TELLS
   ME THAT A NEW GROUP HAS FORMED IN THE
   LOVELY SUB TROPICAL HOMETOWN OF
   ROCKHAMPTON (QUEENSLAND, AUSTRALIA).
   THE GROUP IS CALLED - NOBLE !
 ----------------------------------------
 PRESS 'I' FOR INSTRUCTIONS    PAGE NO:02
```

Figure 3-15. Reason 4 Treason #2 (b) – TERA upcoming mega-demo



- THERE HAS BEEN A FEW RUMOURS THAT THE MAD PARTY IN APRIL HAS BEEN CALLED OFF IS THIS TRUE ????

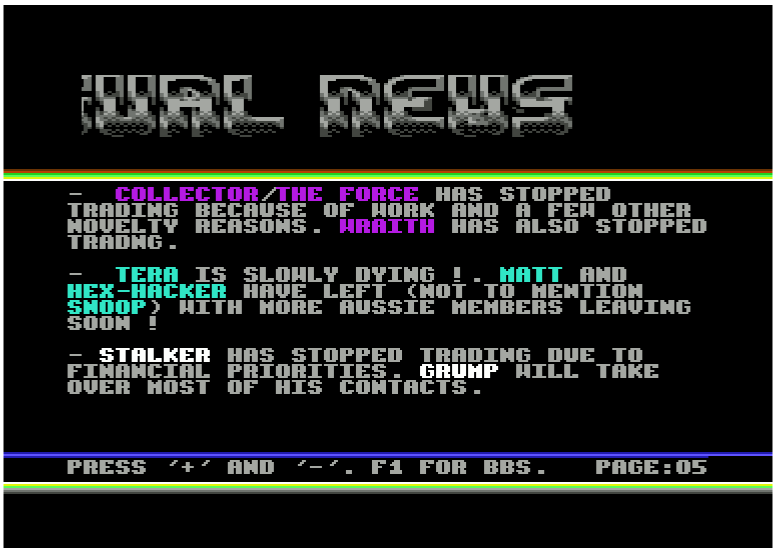
- I BET YOU DIDD'NT KNOW THAT MAD BASED IN BRISBANE HAS A NEW MEMBER IN VICTORIA ? !!!!

- THAT FIENDISH FROM TERA IS HARD AT WORK IN HIS DEMO LABS SECRETLY WORKING ON A MEGA-DEMO THAT WILL CONTAIN 2 OR 3 SECRET SPECIAL WORLD RECORD ROUTINES THAT HAVE NEVER BEEN SEEN BEFORE ON THE 64 !!!

- TYRANNY NOW HAS 50 MEMBERS !!!!!!

PRESS 'I' FOR INSTRUCTIONS PAGE NO:03

Figure 3-16. Reason 4 Treason #3 (a) – TERA dying, MATT leaves



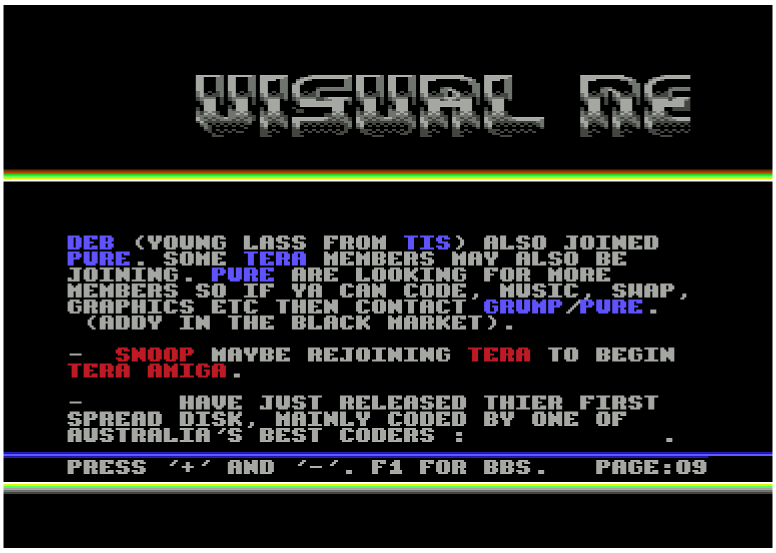
- COLLECTOR/THE FORCE HAS STOPPED TRADING BECAUSE OF WORK AND A FEW OTHER NOVELTY REASONS. WRAITH HAS ALSO STOPPED TRADNG.

- TERA IS SLOWLY DYING ! . MATT AND HEX-HACKER HAVE LEFT (NOT TO MENTION SNOOP) WITH MORE AUSSIE MEMBERS LEAVING SOON !

- STALKER HAS STOPPED TRADING DUE TO FINANCIAL PRIORITIES. GRUMP WILL TAKE OVER MOST OF HIS CONTACTS.

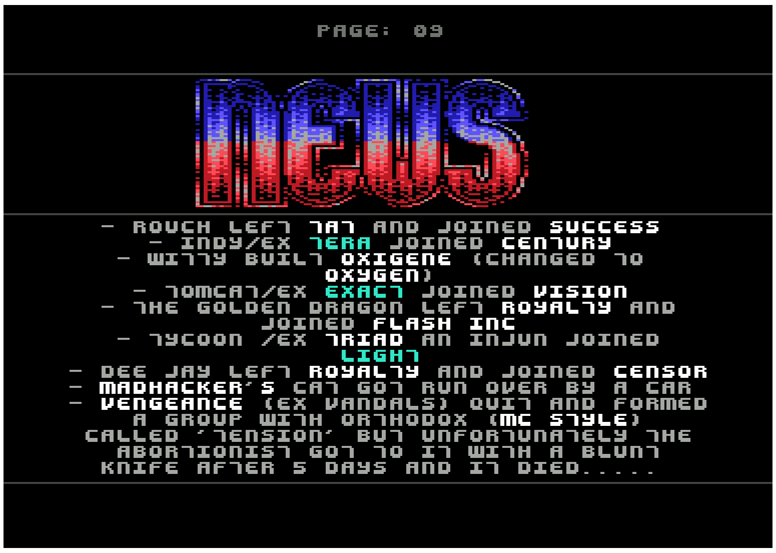
PRESS '+' AND '-'. F1 FOR BBS. PAGE:05

Figure 3-17. Reason 4 Treason #3 (b) – SNOOP and TERA Amiga



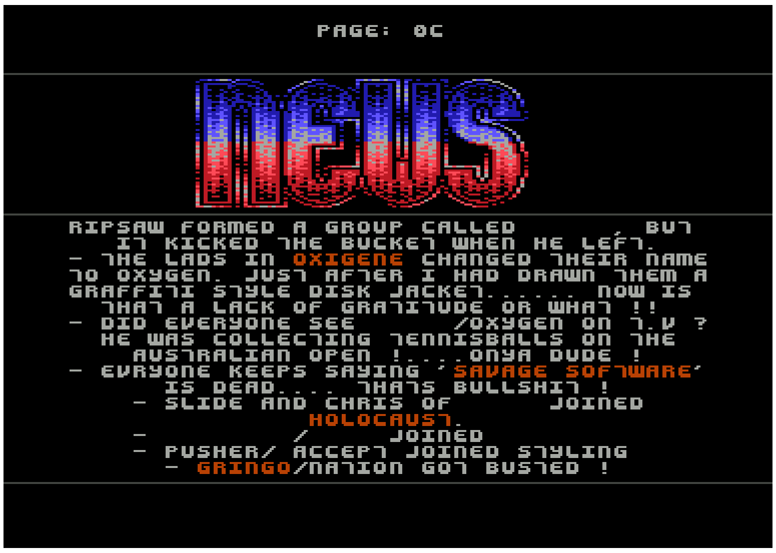
```
DEB (YOUNG LASS FROM TIS) ALSO JOINED
PURE. SOME TERA MEMBERS MAY ALSO BE
JOINING. PURE ARE LOOKING FOR MORE
MEMBERS SO IF YA CAN CODE, MUSIC, SWAP,
GRAPHICS ETC THEN CONTACT GRUMP/PURE.
 (ADDY IN THE BLACK MARKET).

-  SNOOP MAYBE REJOINING TERA TO BEGIN
TERA AMIGA.

-      HAVE JUST RELEASED THIER FIRST
SPREAD DISK, MAINLY CODED BY ONE OF
AUSTRALIA'S BEST CODERS :              .

PRESS '+' AND '-'. F1 FOR BBS.    PAGE:09
```

Figure 3-18. Reason 4 Treason #4 (a) – TERA loses INDY



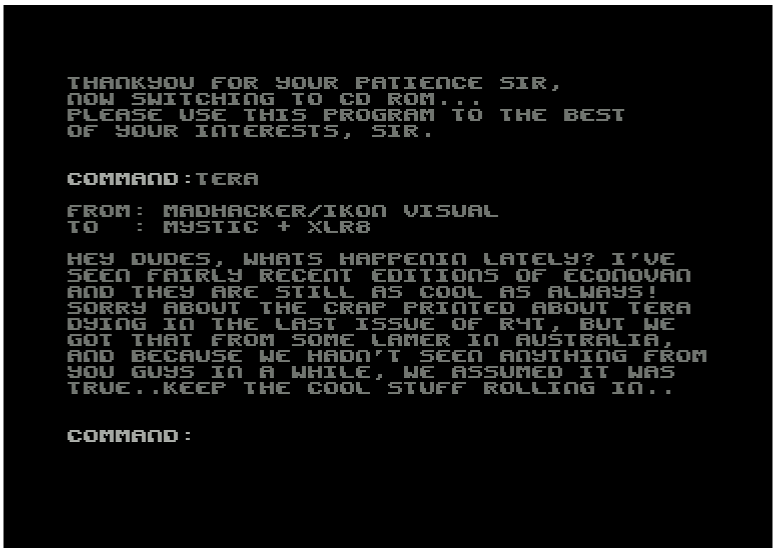
```
                 PAGE: 09

   - ROUCH LEFT TAT AND JOINED SUCCESS
      - INDY/EX TERA JOINED CENTURY
    - WITTY BUILT OXIGENE (CHANGED TO
               OXYGEN)
     - TOMCAT/EX EXACT JOINED VISION
   - THE GOLDEN DRAGON LEFT ROYALTY AND
          JOINED FLASH INC
    - TYCOON /EX TRIAD AN INJUN JOINED
               LIGHT
 - DEE JAY LEFT ROYALTY AND JOINED CENSOR
 - MADHACKER'S CAT GOT RUN OVER BY A CAR
 - VENGEANCE (EX VANDALS) QUIT AND FORMED
   A GROUP WITH ORTHODOX (MC STYLE)
  CALLED 'TENSION' BUT UNFORTUNATELY THE
   ABORTIONIST GOT TO IT WITH A BLUNT
  KNIFE AFTER 5 DAYS AND IT DIED.....
```

Figure 3-19. Reason 4 Treason #4 (b) – TERA loses more



```
RIPSAW FORMED A GROUP CALLED        , BUT
   IT KICKED THE BUCKET WHEN HE LEFT.
- THE LADS IN OXIGENE CHANGED THEIR NAME
TO OXYGEN. JUST AFTER I HAD DRAWN THEM A
GRAFFITI STYLE DISK JACKET...... NOW IS
  THAT A LACK OF GRATITUDE OR WHAT !!
- DID EVERYONE SEE      /OXYGEN ON T.V ?
  HE WAS COLLECTING TENNISBALLS ON THE
    AUSTRALIAN OPEN !....ONYA DUDE !
- EVRYONE KEEPS SAYING 'SAVAGE SOFTWARE'
      IS DEAD.... THATS BULLSHIT !
    - SLIDE AND CHRIS OF       JOINED
               HOLOCAUST.
    -       /     JOINED
    - PUSHER/ ACCEPT JOINED STYLING
      - GRINGO/NATION GOT BUSTED !
```

Figure 3-20. Reason 4 Treason #4 (c) – TERA greetings in the BBS

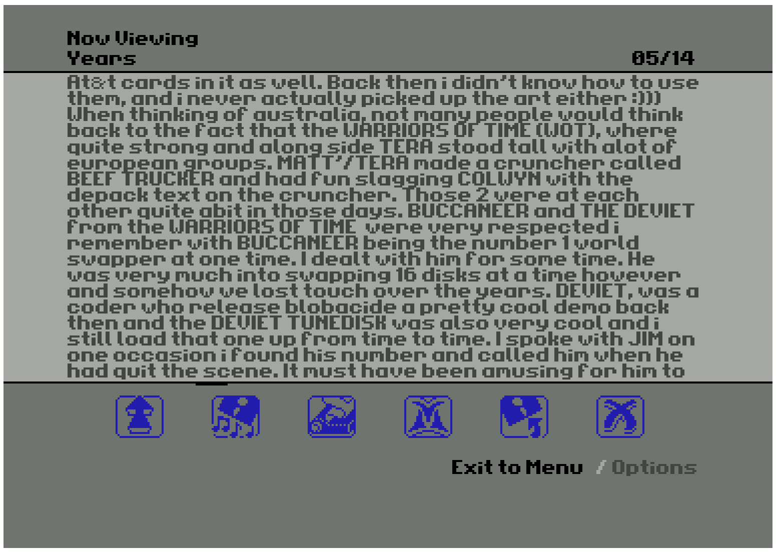


```
THANKYOU FOR YOUR PATIENCE SIR,
NOW SWITCHING TO CD ROM...
PLEASE USE THIS PROGRAM TO THE BEST
OF YOUR INTERESTS, SIR.


COMMAND:TERA

FROM:  MADHACKER/IKON VISUAL
TO   :  MYSTIC + XLR8

HEY DUDES, WHATS HAPPENIN LATELY? I'VE
SEEN FAIRLY RECENT EDITIONS OF ECONOVAN
AND THEY ARE STILL AS COOL AS ALWAYS!
SORRY ABOUT THE CRAP PRINTED ABOUT TERA
DYING IN THE LAST ISSUE OF R4T, BUT WE
GOT THAT FROM SOME LAMER IN AUSTRALIA,
AND BECAUSE WE HADN'T SEEN ANYTHING FROM
YOU GUYS IN A WHILE, WE ASSUMED IT WAS
TRUE..KEEP THE COOL STUFF ROLLING IN..


COMMAND:
```

3.3. History in Vandalism News #32 by ONSLAUGHT

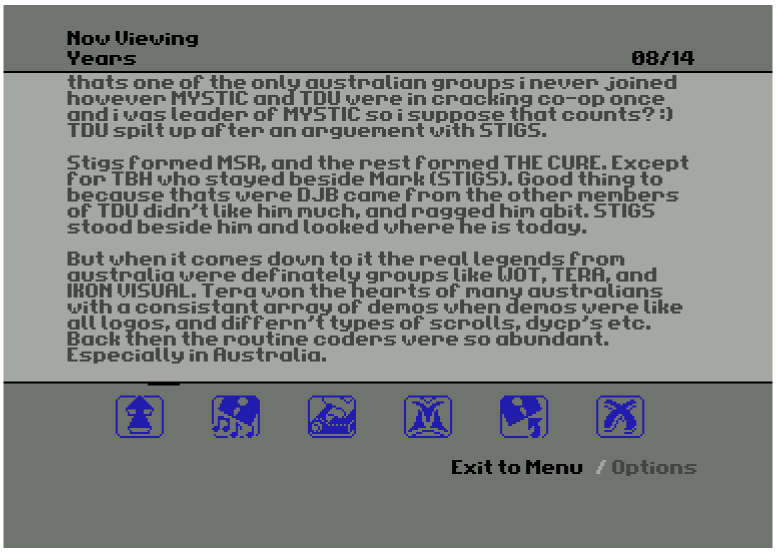
ONSLAUGHT's Vandalism News #32 (September 2000) carried a retrospective of the scene in Australia called "The Years" which mentions TERA and MATT'/TERA. There's also an interview with MATT'/TERA.

Figure 3-21. Vandalism News #32 (09/00) – "The Years" #1



At&t cards in it as well. Back then i didn't know how to use them, and i never actually picked up the art either :)))
When thinking of australia, not many people would think back to the fact that the WARRIORS OF TIME (WOT), where quite strong and along side TERA stood tall with alot of european groups. MATT'/TERA made a cruncher called BEEF TRUCKER and had fun slagging COLWYN with the depack text on the cruncher. Those 2 were at each other quite abit in those days. BUCCANEER and THE DEVIET from the WARRIORS OF TIME were very respected i remember with BUCCANEER being the number 1 world swapper at one time. I dealt with him for some time. He was very much into swapping 16 disks at a time however and somehow we lost touch over the years. DEVIET, was a coder who release blobacide a pretty cool demo back then and the DEVIET TUNEDISK was also very cool and i still load that one up from time to time. I spoke with JIM on one occasion i found his number and called him when he had quit the scene. It must have been amusing for him to

Figure 3-22. Vandalism News #32 (09/00) – "The Years" #2

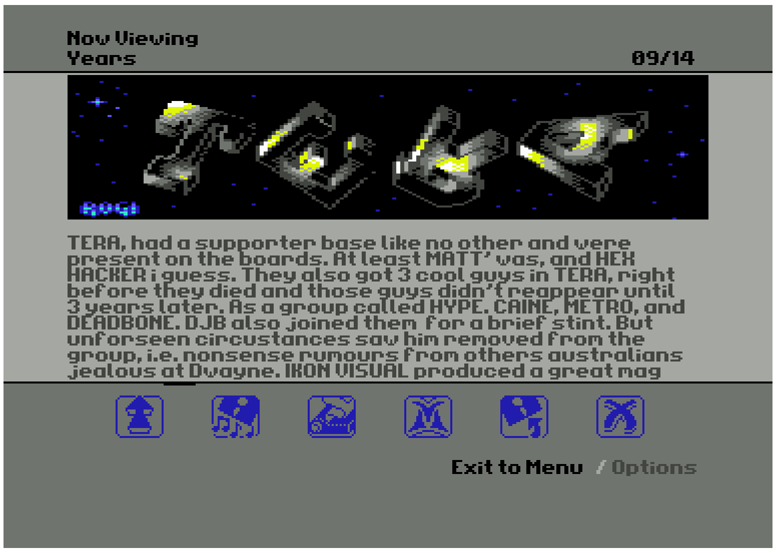


thats one of the only australian groups i never joined however MYSTIC and TDU were in cracking co-op once and i was leader of MYSTIC so i suppose that counts? :) TDU spilt up after an arguement with STIGS.

Stigs formed MSR, and the rest formed THE CURE. Except for TBH who stayed beside Mark (STIGS). Good thing to because thats were DJB came from the other members of TDU didn't like him much, and ragged him abit. STIGS stood beside him and looked where he is today.

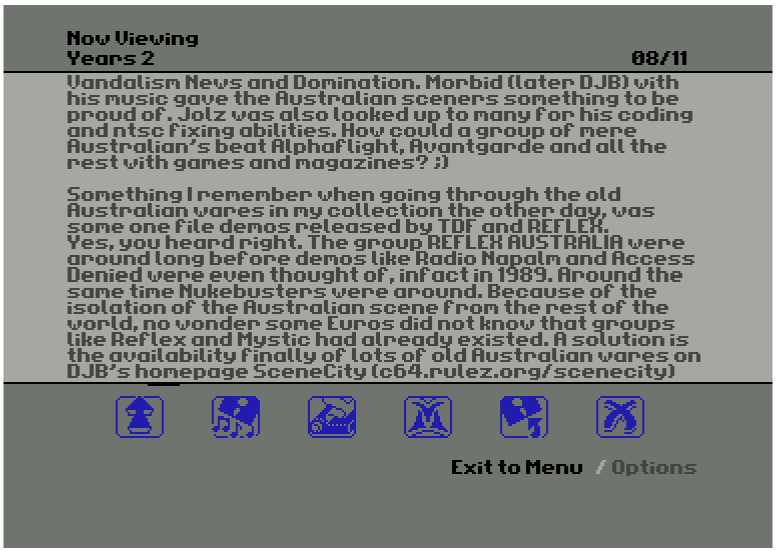
But when it comes down to it the real legends from australia were definately groups like WOT, TERA, and IKON VISUAL. Tera won the hearts of many australians with a consistant array of demos when demos were like all logos, and differn't types of scrolls, dycp's etc. Back then the routine coders were so abundant. Especially in Australia.

Figure 3-23. Vandalism News #32 (09/00) – "The Years" #3



TERA, had a supporter base like no other and were
present on the boards. At least MATT' was, and HEX
HACKER i guess. They also got 3 cool guys in TERA, right
before they died and those guys didn't reappear until
3 years later. As a group called HYPE. CAINE, METRO, and
DEADBONE. DJB also joined them for a brief stint. But
unforseen circustances saw him removed from the
group, i.e. nonsense rumours from others australians
jealous at Dwayne. IKON VISUAL produced a great mag

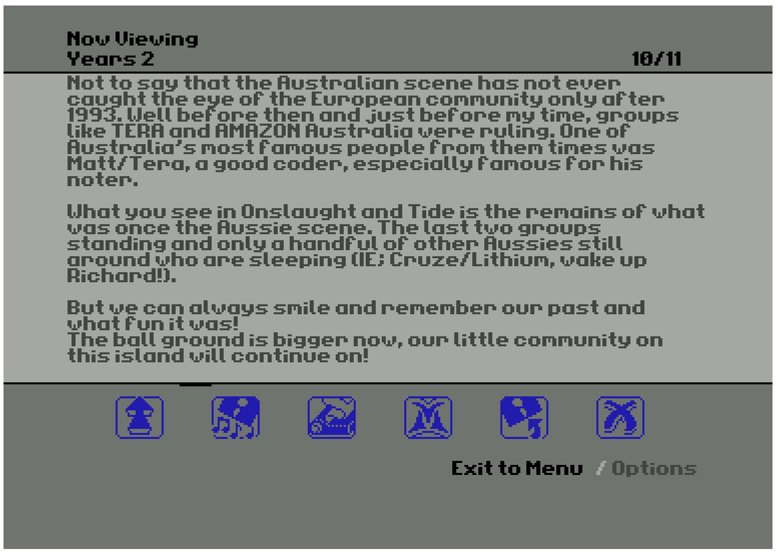
Figure 3-24. Vandalism News #32 (09/00) – "The Years 2" #1



Vandalism News and Domination. Morbid (later DJB) with
his music gave the Australian sceners something to be
proud of. Jolz was also looked up to many for his coding
and ntsc fixing abilities. How could a group of mere
Australian's beat Alphaflight, Avantgarde and all the
rest with games and magazines? ;)

Something I remember when going through the old
Australian wares in my collection the other day, was
some one file demos released by TDF and REFLEX.
Yes, you heard right. The group REFLEX AUSTRALIA were
around long before demos like Radio Napalm and Access
Denied were even thought of, infact in 1989. Around the
same time Nukebusters were around. Because of the
isolation of the Australian scene from the rest of the
world, no wonder some Euros did not know that groups
like Reflex and Mystic had already existed. A solution is
the availability finally of lots of old Australian wares on
DJB's homepage SceneCity (c64.rulez.org/scenecity)

Figure 3-25. Vandalism News #32 (09/00) – "The Years 2" #2

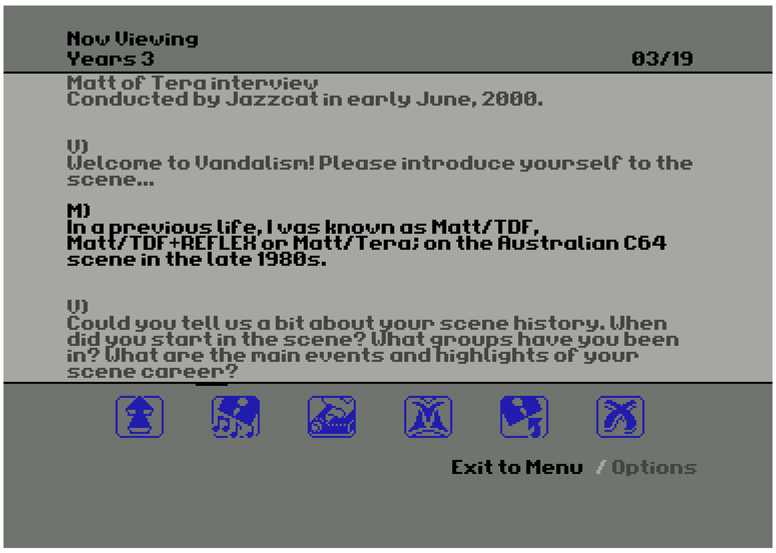

Now Viewing
Years 2 10/11

Not to say that the Australian scene has not ever caught the eye of the European community only after 1993. Well before then and just before my time, groups like TERA and AMAZON Australia were ruling. One of Australia's most famous people from them times was Matt/Tera, a good coder, especially famous for his noter.

What you see in Onslaught and Tide is the remains of what was once the Aussie scene. The last two groups standing and only a handful of other Aussies still around who are sleeping (IE; Cruze/Lithium, wake up Richard!).

But we can always smile and remember our past and what fun it was!
The ball ground is bigger now, our little community on this island will continue on!

Figure 3-26. Vandalism News #32 (09/00) – "The Years 3" #1


Now Viewing
Years 3 03/19

Matt of Tera interview
Conducted by Jazzcat in early June, 2000.

V)
Welcome to Vandalism! Please introduce yourself to the scene...

M)
In a previous life, I was known as Matt/TDF, Matt/TDF+REFLEX or Matt/Tera; on the Australian C64 scene in the late 1980s.

V)
Could you tell us a bit about your scene history. When did you start in the scene? What groups have you been in? What are the main events and highlights of your scene career?

3.4. MATT'/TERA interview by ONSLAUGHT

This interview was part of Vandalism News #32 (September 2000), but was reproduced on the ONSLAUGHT HQ site. My memory recall did not serve me well, so some of the things that I say about BEEFTRUCKER and C1650 are incorrect.

Interview with Matt / TERA

Printed in Vandalism News #32 – Olympic Edition
Interview Performed by Jazzcat

1) Welcome to the Vandalism News magazine! Please introduce yourself to the readers.

In a previous life, I was known as Matt/TDF, Matt/TDF+Reflex or Matt/Tera; on the Australian c64 scene in the late 1980s.

2) Could you tell us a bit about your C-64 history. When did you start in the scene? What groups have you been in? What are the main events and highlights of your scene career?

I started in 1983, self-taught programming and graphics design, producing intros, tools and demos in a group called Tour de Future (TDF) with school friends. We grew and added members from around Sydney, then began a co-op with a Victorian group called Reflex, before merging our names to become Tera. Tera eventually had an international reputation, affiliated boards in the states, a range of demos, intros and contacts. Our high point was probably early/mid 1989. I tended to be the most productive coder/graphics producer/etc, and did some trading, but others (e.g. Mystic) were stronger traders. I just liked making things and was too heads down in the code! My personal highlight was creating the worlds shortest RLE decoder, using undocumented opcodes and self-modifying code. Of course, the record was later beaten, but the whole package of Beeftrucker was great. And, I didn't realise it at the time, but the other guys in my group were very important, nothing could have happened without them.

3) What made you leave the scene in the end?

Lack of interest. I bought a c1650 modem and started calling bulletin boards, falling in love with phrack, cDc, underground bulletin boards, and a new world. At the same time, university started, and a primordal internet was there to be had. Programming moved to C, Unix and Linux (I ran a UUCP node on Linux 0.97 in 1991/1992).

4) When the Australian scene still existed it was hard to be recognized in the european scene. However, a lot of european sceners know of you, especially for the tools you made... what do you think made yourself be known to the scene? was it a particular demo or tool, presence on the boards?

That's cool to hear! Tools are the best thing to make, because they allow people to make even more stuff. All coders should make tools, it's liberating. I think we had a good range of contacts in Europe, and we had a few visible tools (our Notemaker and Compressor). The traders in our group (esp. Mystic) did an excellent job of getting us known and connected. I wonder if just being Australian singled us out as 'interesting' as well.

5) Tera were one of the most famous groups in the Australian scene. What was the time like in those glory days?

Busy! I think at the time it didn't seem like we were famous or in glory, we were just coding, trading and doing the things we liked to do; and that's perhaps the important thing.

6) What are your all-time favourites: (C64 only)

Unfortunately I'm going to let you down, as I've forgotten a lot of detail, since many things have happened since, but I'll try to answer as best as I can.

Demo Group: Horizon, because for a while they were coming out with innovative coding tricks.
Demo: I forget who it was by, but it used the 1541 processor to do parallel processing with the 6510, I remember hacking into the code to figure it out, it was cool!
Programmer: Mr Z and some of the triad guys for their hacking skills, but no particular one stands out.
Musician: sorry!
Graphician: sorry!
Game: Galaga (yes! simple but beautiful, ... I didn't play many games, maniac mansion I really enjoyed as well, with the memorable EagleSoft Loader!)

Can I add 'intro' ? I enjoyed the simplicity of Triad and 711 intros; and Triad inspired some of my own designs.

7) What's your view on the internet and how it has effected the way computer scenes and people communicate and produce?

The Internet is a phenomenal technology that has brought intellectual groups from across the world together in virtual communities: some amazing things are to come. However, there's nothing like sitting in a cafe with a bunch of friends and having a good time. With the birth of the internet, we've just reached another point in the continum of society. The information revolution can be seen as a continued outgrowth of post-renaissance western development, but in the 20th century it has flowered, and the internet has finally provided a uniform and global knowledge medium and the next bump in global history (consider the early, mid, late bronze ages, well now, the internet is a marker for the mid (or late perhaps?) information age). The coming years will be interesting. Readers should listen to some of recent comments by Bill Joy. Hey, and don't 'script kiddies' remind you of people using intro/demo-makers! The more things change, the more they stay the same ...

8) In your opinion, what is the most important element of a demo?

I tend to admire different demos for different purposes. A good overall composition of graphics, sound, text and everything is important. But then, some demos don't have this, but do amazing technical things (e.g. the first open top/bottom border scroll) and have to be admired for that. Some demos use no advanced techniques, but have a great layout/composition/music/interesting scroll text, etc.

9) Was a C64 a stepping stone for you in life or computer related activities?

In a way, I took to studying Computer Systems Engineering, because of enjoying C64 programming. I've made an impact in other areas since then, some times good, some times bad.

10) Do you still own a C64 and all of your old disks?

No, but sometimes I wish I did for memories sake. I have used a C64 emulator on a PC to reminisce about old games! They are still great and highly playable.

11) Are you in contact with Hex, Xlr8 or any of the other guys from Tera or old Australian scene?

I didn't have strong contact for some time, I tend to be too involved in my own work to the detriment of contact with people around me. Recently I've been keeping in email contact with XLR8, and when I visit home, will catch up with the guys.

12) Ever had any wars or disliking towards some group or person in the scene?

Colwyn of ACU, for various reasons that I cannot remember now.

13) Who do you think is the biggest lamer to walk the face of the C64 scene?

Can't remember to be honest; but mostly people with bad attitudes. Even if you're a great coder, if you have a bad attitude, that's pretty lame. And if you're not a good coder, but you're trying hard, and making improvements, you're not lame. But if you're producing sloppy work, low quality, etc when you could do better, then that's lame. It's about attitude.

14) Did you ever call the boards or were you ever involved in HP activities?

Yes.

15) What are your current activities these days? and what is your reaction when I say the C64 scene is still alive?

I have had a bumpy ride through internet activities, professional software engineering, and other things, and recently just completed studies in Art History a nd presently work for a technology consultancy in Cambridge UK. My technical areas of interest are now software engineering, cryptography, communications, computing, digital society, technical futures and a lot of related stuff; I'm an IEEE member, and have a reputation as a tech-head. I am surprised that the scene is still going, and amazed at what people are doing in recent demos! For me, it was some time ago. If you are in the United Kingdom, you need to visit the cryptologic museum at Bletchley Park, and the display of computing technology, there are C64s, Vic-20s and enough things to bring back the memories.

16) Please feel free to send any greetings to anyone you know...

One name stands out: Jerry of Triad, for the interesting pen-pal letters. I'd like to send greetings to the guys in my group; sometimes I was so busy with my work that I forgot how important they were!

17) Thanks for your time Matt, do you have any last comments to leave a final impression on the audience?

Do what you love to reach for your dreams and don't forget that family and friends are everything. Life is an interesting experience, best to remain philosophical, enjoy life, have fun, and aim for quality.

Matthew. aka Matt/Tera.

Section 4. C64 Activities Tools BEEFTRUCKER

1. Background

1.1. Compressors and their use

1.1.1. Need for compressors

Commodore 64 scene based software was almost always compressed before distribution, because modem transfer speeds were low (typically 1200bps or 2400bps), and disk storage sizes were small (typically 340KB per double sided 5.25" disk).

1.1.2. Use of compressors

The original production releases of commercial software were usually not compressed, however cracked versions (i.e. where copy-protection had been removed) almost always were. End users (e.g. game players) would often only distribute or archive the shortest version of software, so releasing the shortest version was part of the competition between cracking groups. For example, two groups may release a cracked version of a game within a week of the production release, but only the shortest release would survive. A more effective compressor could give one group an edge over another.

Typically, the original would be compressed with a "packer", and then an intro would be placed before the resulting packed image. The intro and the image would then be compressed with a "cruncher" to form the distribution image. When the distribution image was executed, it would decrunch and execute the Intro, which usually displayed a group logo and contained a list of greetings to other groups. When the Intro finished, the original image would depack and execute.

The arrangement could become complicated if the software had multiple parts. For example, a game may have multiple levels that need to load in as the game is in progress. The cracking process included extracting and compressing these levels, and storing them with the cracked distribution image. Whereas the original would load a raw binary image of the level, the cracked version would load a compressed binary image of the level, and decompress it while loading or once it had been loaded into memory. Specialised versions of “packers” or “crunchers” would be used to perform this “level-packing”. It became possible to store all levels in a single file, and read each level as a segment from that file.

Demos were usually always compressed. A single part demo may be “packed” and/or “crunched”, and a multi-part demo may load each succeeding part from disk. It was also possible to have multi-part demos with a single file, as each part would be compressed and appended to another part – so when each part finished execution, it would uncompress and execute the next part. Most other programs, such as tools, and notemakers were compressed.

1.1.3. Types of compressors

1.1.3.1. Overview

There were typically only two types of compressors, but the boundaries between these were blurred, and there were hybrids. Additionally, variants of these two types were often produced, such as a “level packing version” of a compressor. The two types of compressors corresponded roughly to two compression technologies: “run length encoding” and “dictionary encoding”.

1.1.3.2. Packers

Packers typically employed some form of run-length-encoding (RLE) as their primary compression technique. RLE involves taking sequences of equivalent bytes and representing them in a tokenised format (i.e. so that the decompressor can expand the token back into the original data). More complex variations used multiple tokens for different sorts and lengths of sequences, or limited dictionary encoding, and so started to approach the functionality of a “cruncher”. Sledgehammer, BYG Compactor, ECA Packer and Zipper were popular packers.

1.1.3.3. Crunchers

Crunchers typically employed some form of string-based-encoding, such as LZW, as their primary compression technique. LZW involves storing strings in their original representation, but substrings in a tokenised format (i.e. so that the decompressor can use the token to extract a substring from the string, and recreate the original data). Cruel Cruncher and Time Cruncher were popular crunchers.

“Crunchers” provided the best space performance, but compression and decompression time was high. “Packers” resulted in lower space performance, but compression and decompression time was low. Often, a “packer” would be used first, and a “cruncher” would follow: this could provide a more economical result. A good hybrid, such as Sledgehammer, would often be used by itself.

1.1.3.4. Variations

Other features distinguished “packers” and “crunchers”. Some could not compress all regions of memory whereas others couldn’t (typically, a cruncher could only compress $0800 to $FFFF, whereas a packer may be able to compress $0100 to $FFFF). Some employed fast loaders and savers, and others used more efficient decompression routines.

1.2. Motivations for writing a compressor

I cannot remember what the original motivations were for developing BEEFTRUCKER, but I do remember the origins of the name. It refers to a very large girl at school, who would sit and “squash” someone if they annoyed her. We referred to her as “beeftrucker”, and in general, we referred to any large girl by the same term. Anyway, it is likely that I experimented with designing the depack routine, only to find that I could beat the world-record. Then I would have decided about how to package it all together and provide a complete product.

2. Design goals

My goal with Beeftrucker was to develop a product that included:

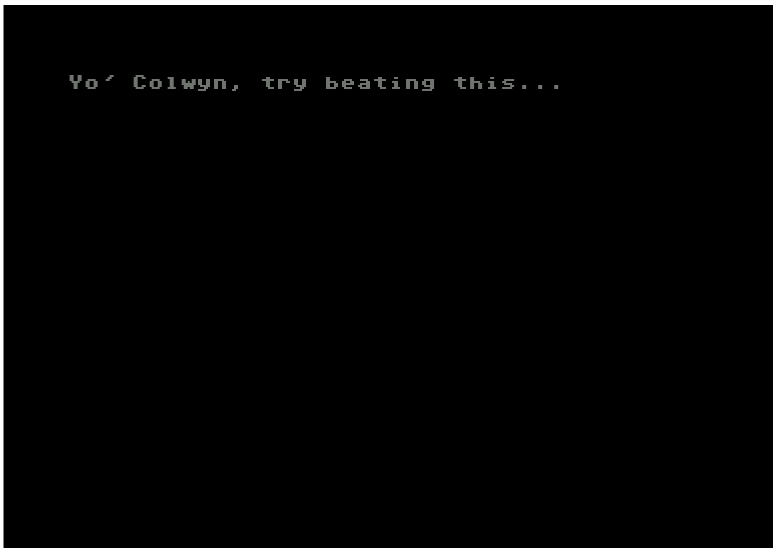
- **The world’s shortest depacking routine:** I broke the current world-record (held by ZIZYPLUS of ONEWAY with his $58 byte ZIPPER V1.0) by using a new approach. The depack routine overhead was only $56 bytes. This routine was responsible for relocating itself to another location in RAM, and then relocating the data to another location in RAM, before unpacking (decompressing) the data back to its original size, then setting various system registers, and finally executing the uncompressed data. I also set out to overcome a few limitations present in other packers.
- **A fast and efficient packing routine:** that used fast loading and fast saving routines to minimise the time required for the entire process.
- **A user-friendly interface:** allowing the user to view disk directory, execute disk commands, and configure options for setting the system state at execution time (such as $01, $2D/$2E, SEI/CLI and a JMP address).
- **A good overall product:** that included stylish intro screen, technical summary information, and the packer itself – with screen transitions that were “clean” and did not involve “blue screens”, or aborted music, or unpacker garbage. These are the little things that too many designers did not take care of.

3. Implementation package

3.1. Intro, with group oriented text

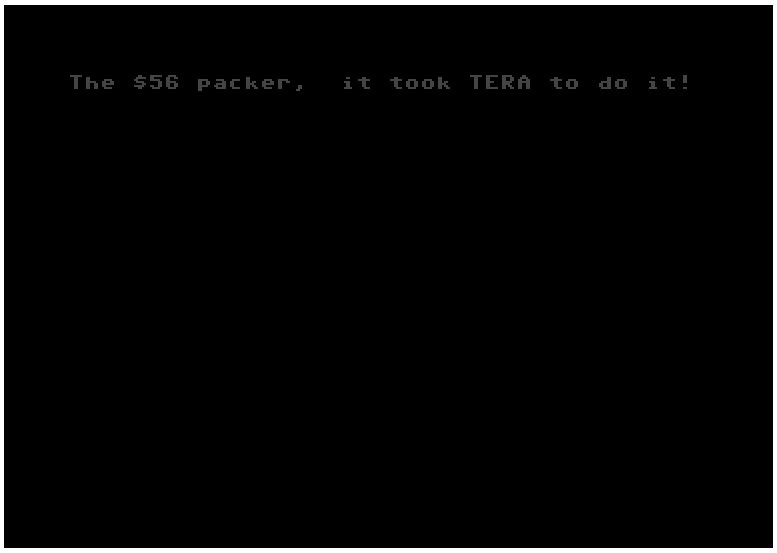
The program first starts with propaganda about our war with COLWYN of THE FORCE, and then a marketing spiel about how “it took TERA to do it!”

Figure 4-1. BEEFTRUCKER Intro - propaganda screen



```
Yo' Colwyn, try beating this...
```

Figure 4-2. BEEFTRUCKER Intro - marketing screen



```
The $56 packer,  it took TERA to do it!
```

The intro is basic, but mildly stylish. It fades in from black, and fades out to black when the SPACE key is pressed. I created the logo and the character set, and employed them in a blue and grey background with an overlaid “Australia” sprite for patriotic reasons. The 3x2 character set is one of my favourites, whereas the logo is somewhat pedestrian.

Figure 4-3. BEEFTRUCKER Intro - main screen

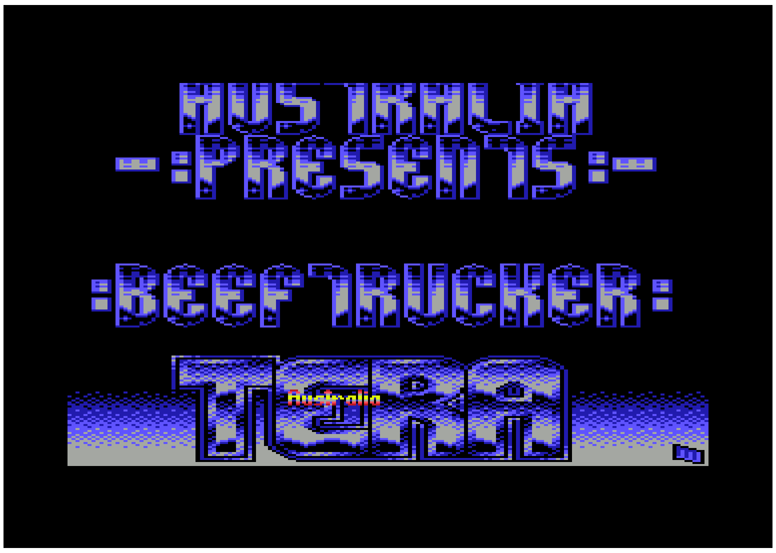


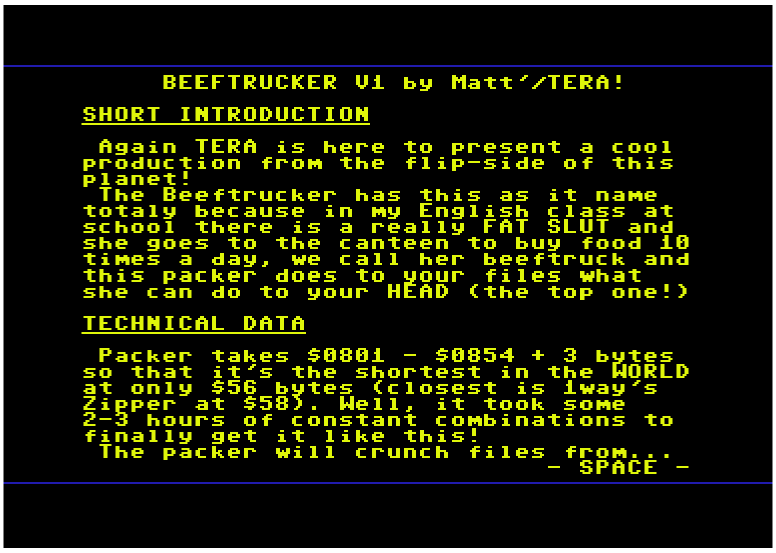
Figure 4-4. BEEFTRUCKER intro – scroll text

	TERA AGAIN IS	NO GREETS COZ	MISTIC	COMING SOON!
TERA	HERE TO KICK	THIS SCROLLER	BULLET	
AUSTRALIA	YOUR ASS!!	IS A PAIN TO	KILENEMY	LATER DUDES
-:PRESENTS:-		EDIT AS I HAVE	ELIMINATOR	
	WORD UP MAN!!	TO TYPE IT IN	STOREMAN(?)	.TERA FOREVER.
:BEEFTRUCKER:		THE ASSEMBLER.		+612-601-8691
	CREDITS FOR		REMEMBER…	
THE $56 PACKER	ZE INTRO HERE	ONLY HI’S TO	IN TERA WE	COLWYN/TF SUX
	GOTO MATT’ FOR	THE TERA DUDES	ARE A FORCE	
A COOL PACKER	ALL CODING &	XLR8	OF FRIENDS!!	
DONE BY	ALL GRAFIX.	HEX-HACKER		
MATT’/TERA	MUSIC BY LOS	BOSS	WATCH FOR…	
		SNOOP	‘SLUDGER V5’	

3.2. Overview, with technical descriptions

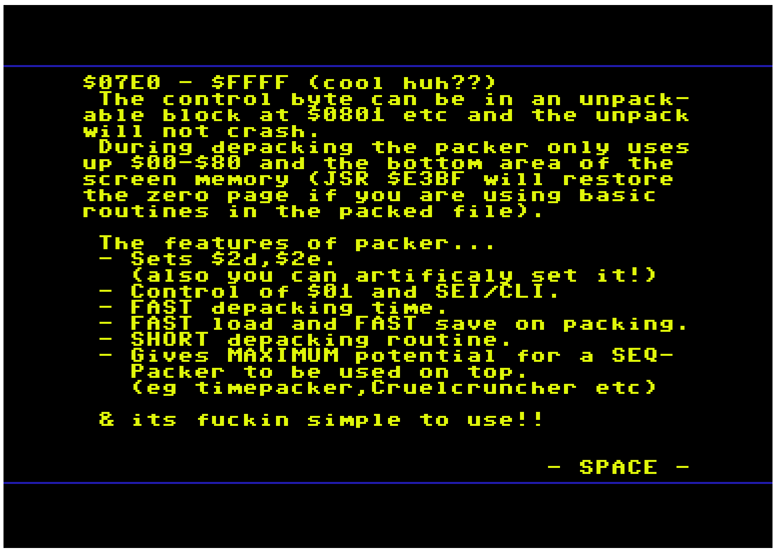
There are three overview screens that provide a detailed introduction to the packer; technical data; experimental comparisons with ZIPPER V1.0; and credits. The user can press RESTORE to skip each one, or SPACE to progress to the next screen. Each screen slides in from the right towards the left in a “smooth” manner.

Figure 4-5. BEEFTRUCKER Program overview (1)



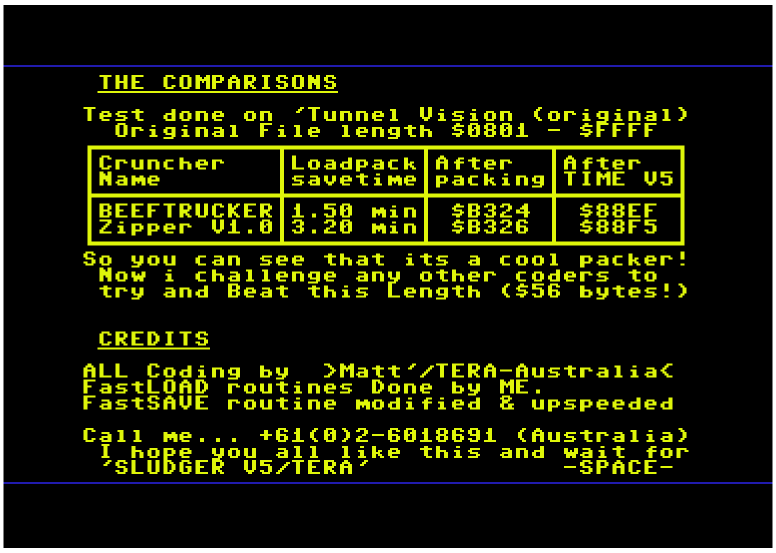
```
       BEEFTRUCKER V1 by Matt'/TERA!
SHORT INTRODUCTION

 Again TERA is here to present a cool
production from the flip-side of this
planet!
 The Beeftrucker has this as it name
totaly because in my English class at
school there is a really FAT SLUT and
she goes to the canteen to buy food 10
times a day, we call her beeftruck and
this packer does to your files what
she can do to your HEAD (the top one!)

TECHNICAL DATA

 Packer takes $0801 - $0854 + 3 bytes
so that it's the shortest in the WORLD
at only $56 bytes (closest is 1way's
Zipper at $58). Well, it took some
2-3 hours of constant combinations to
finally get it like this!
 The packer will crunch files from...
                              - SPACE -
```

Figure 4-6. BEEFTRUCKER Program overview (2)



```
$07E0 - $FFFF (cool huh??)
 The control byte can be in an unpack-
able block at $0801 etc and the unpack
will not crash.
 During depacking the packer only uses
up $00-$80 and the bottom area of the
screen memory (JSR $E3BF will restore
the zero page if you are using basic
routines in the packed file).

 The features of packer...
 - Sets $2d,$2e.
   (also you can artificaly set it!)
 - Control of $01 and SEI/CLI.
 - FAST depacking time.
 - FAST load and FAST save on packing.
 - SHORT depacking routine.
 - Gives MAXIMUM potential for a SEQ-
   Packer to be used on top.
   (eg timepacker,Cruelcruncher etc)

 & its fuckin simple to use!!


                              - SPACE -
```

Figure 4-7. BEEFTRUCKER Program overview (3)



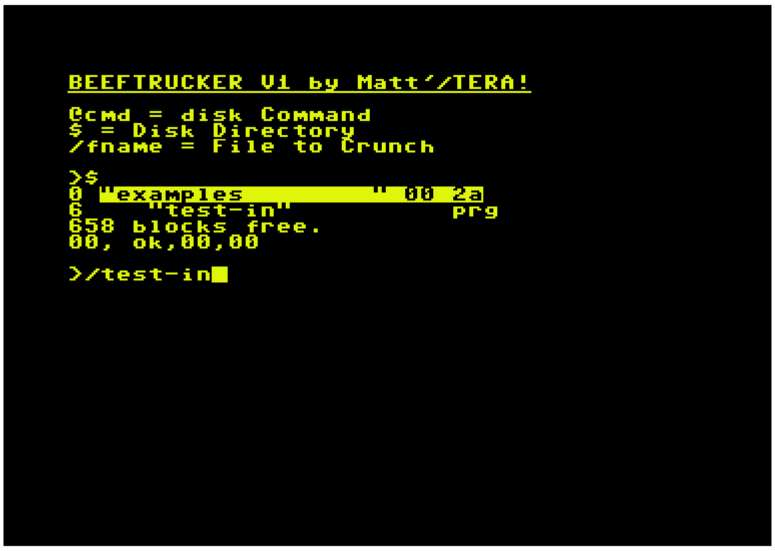
```
  THE COMPARISONS

Test done on 'Tunnel Vision (original)
  Original File length $0801 - $FFFF
 +------------+---------+--------+--------+
 |Cruncher    |Loadpack |After   |After   |
 |Name        |savetime |packing |TIME V5 |
 +------------+---------+--------+--------+
 |BEEFTRUCKER |1.50 min | $B324  | $88EF  |
 |Zipper V1.0 |3.20 min | $B326  | $88F5  |
 +------------+---------+--------+--------+
So you can see that its a cool packer!
 Now i challenge any other coders to
 try and Beat this Length ($56 bytes!)

 CREDITS

ALL Coding by  >Matt'/TERA-Australia<
FastLOAD routines Done by ME.
FastSAVE routine modified & upspeeded

Call me... +61(0)2-6018691 (Australia)
 I hope you all like this and wait for
  'SLUDGER V5/TERA'              -SPACE-
```

3.3. Compressor, with configuration options and information

3.3.1. Configuration process

The main compression program starts with a selection menu, allowing the user to execute disk commands, display a disk directory and choose the file to be compressed.

Figure 4-8. BEEFTRUCKER Compressor - file load screen



```
BEEFTRUCKER V1 by Matt'/TERA!

@cmd = disk Command
$ = Disk Directory
/fname = File to Crunch

>$
0 "examples          " 00 2a
6    "test-in"          prg
658 blocks free.
00, ok,00,00

>/test-in
```

After the first scan of the file (using a fast load routine) – to determine byte frequencies – the user is provided with a configuration screen. The Start, Finish and ControlByte values are displayed and cannot be modified. The ControlByte is chosen to be the byte that appears least in the file.

The user can configure the following information:

- **"Save Filename"**: the name by which the compressed program is saved to disk after compression has been carried out.
- **"Cli/Sei"**: specifies whether the interrupt register should be Set (Sei) or Cleared (Cli) upon execution of the original program. The default is "cli".

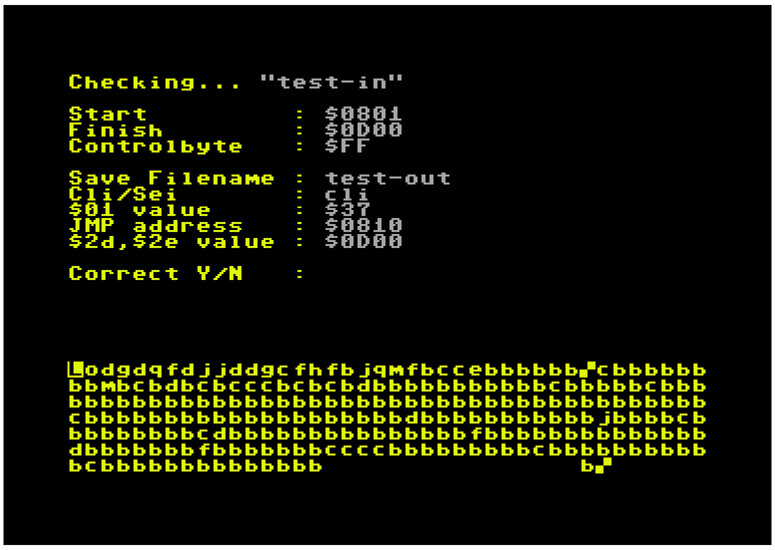
- **“$01 value”**: specifies what value should be placed into the $01 register upon execution of the original program. The default is “$37”.
- **“JMP address”**: specifies what address should be called to execute the original program. The default is “$080D”.

“$2d, $2e value”: is used by many programs as a marker for their ending address (where $0801 is the starting address). The default is the “Finish” value.

“Correct Y/N”: is where the user can go back and make changes.

The frequency distribution of bytes in the file can be seen at the bottom of the display memory, and there are also two bytes that indicate the frequency of the chosen ControlByte, and its value.

Figure 4-9. BEEFTRUCKER Compressor - configuration options screen

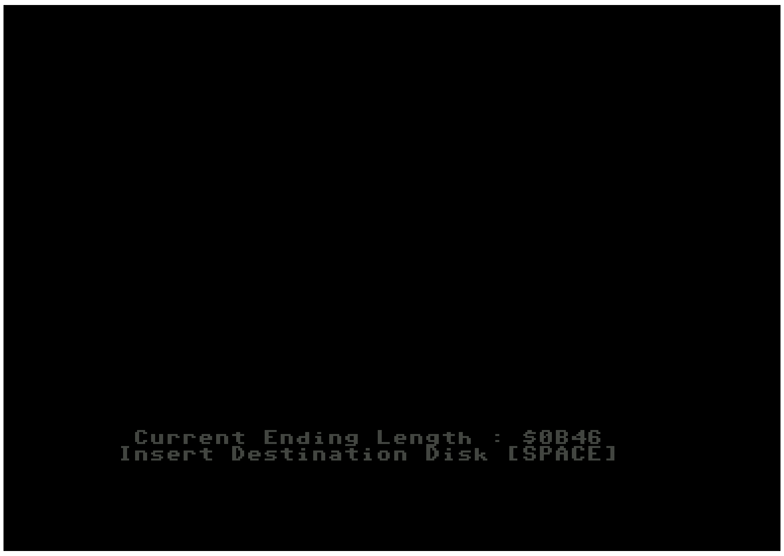


```
Checking... "test-in"

Start          : $0801
Finish         : $0D00
Controlbyte    : $FF

Save Filename  : test-out
Cli/Sei        : cli
$01 value      : $37
JMP address    : $0810
$2d,$2e value  : $0D00

Correct Y/N    :



▙odgdqfdjjddgcfhfbjqmfbccebbbbbb▞cbbbbbb
bbmbcbdbcbcccbcbcbdbbbbbbbbbbbcbbbbbcbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
cbbbbbbbbbbbbbbbbbbbbdbbbbbbbbbbbjbbbbcb
bbbbbbbbcdbbbbbbbbbbbbbbbbfbbbbbbbbbbbbb
dbbbbbbbfbbbbbbbccccbbbbbbbbbcbbbbbbbbbb
bcbbbbbbbbbbbbbbb                b▞
```

3.3.2. Compression process

After configuration, the file will be loaded and compressed. A running total of the compressed file end length is shown at the bottom of the screen. When completely loaded and compressed, the user is prompted to insert a destination disk and press the SPACE key. The key can be pressed multiple times for multiple saves, and each save uses a fast save routine. The saved file contains the compressed data, plus a header with the “$56 byte” decompression routine.

Figure 4-10. BEEFTRUCKER Compressor - file save screen



```
 Current Ending Length : $0B46
Insert Destination Disk [SPACE]
```

3.3.3. Compression method

The compression routine works by scanning the file for sequences of equivalent bytes, and turning those sequences into a reduced ("tokenised") format.

The "tokenised" format consists of a ControlByte, followed by a DataCount and a DataByte. A sequence of *N* bytes of value *M* is replaced by the sequence ControlByte(*X*)/DataCount(*N*)/DataByte(*M*), where *X* is the chosen ControlByte for that file. Obviously, this means that 128 bytes of uncompressed data can be represented as 3 bytes of compressed data.

Note that the decompressor reads from high memory to low memory, so this sequence is stored in memory as DataByte(*M*)/DataCount(*N*)/ ControlByte(*X*).

For example, if $FF was the chosen ControlByte, and the input file contained the sequence $01/$01/$01/$01/$01/$01, then the output file would contain the sequence $FF/$07/$01.

The compression routine will not compress a sequence of two or three equal bytes, unless it needs to "escape" three instances of the ControlByte. If it did convert three bytes into a tokenised format, it would only slow down the decompression process and result in no memory gain. When the compression routine reads a DataByte with the same value as the ControlByte, then it will replace it with the token ControlByte(*X*)/DataCount($01)/DataByte(*M*). If it did not do this, then the decompressor would mistake the DataByte as a ControlByte.

This means that the DataCount can be interpreted in the following ways:

DataCount	Meaning
$00	Used to represent a count of 256; as there is no reason to have a count of 0
$01	Used to "escape" the occurrence of the ControlByte in the data. If the compressor comes across the ControlByte, then it replaces it with the sequence ControlByte(*X*)/DataCount($01)/DataByte(*X*). If this did not occur, then the decompressor would mistake this DataByte as a ControlByte.

$02	Used to indicate the end of compressed data. When the decompressor reads a DataCount of $02, it reads the next DataByte, and then terminates. That last DataByte is then used as the value to be placed into the $01 zero-page address. This overload works as there is no other reason to have a count of $02 (it would be inefficient: because it would expand the output).
$03	Used to “escape” the occurrence of the ControlByte in the data (if there are 3 instances of the ControlByte in a sequence). If other sequences of length 3 were compressed, it would not make any difference to the length of output, yet it would increase the decompression time.
$04 - $FF	Used to represents normal “tokenised” format of sequenced bytes.

The decompression routine works by locating these sequences, and expanding the “tokenised” information back into an original form, using the rules above.

Enhancements

There are many possibilities to add extra intelligence into the process to further optimise the compression method. The most effective approach would be to collect more statistical information about the bytes in the file.

For instance, the lowest frequency ControlByte may not be the best choice, or multiple instances of compressed sequences could have been aligned to pre-empt a dictionary based compressor (e.g. if a 256 byte block contains the strings “*****” and “******”, then it may be better to encode it as CB(*X*)/DC($05)/DV(‘*’) and CB(*X*)/DC($05)/DV(‘*’)/DV(‘*’) rather than as CB(*X*)/DC($05)/DV(‘*’) and CB(*X*)/DC($06)/DV(‘*’): a cruncher may then reduce the first case to *TOKEN* and *TOKEN*/’*’ whereas it cannot reduce the second case).

3.4. Decompressor, with compact+efficient routines

3.4.1. Overview

The decompression routine is short and fast (that’s why it’s a world record!). The compressor places the decompression routine at the front of the compressed file, and also copies some of the leading compressed data to the end of the compressed file.

3.4.2. Decompression header

The decompression header is placed at the start of the compressed data, and contains a mixture of code and data.

Figure 4-11. BEEFTRUCKER Decompression header

```
0801 | 0B 08 41 12 9E 32 30 35 39 00 A2 00 78 BC 98 07 | ..A..2059.▄.■▀..
0811 | 94 81 CA 30 F8 4C 02 00 BD C6 0A 9D D4 07 CA 10 | ..└0─L..┘─..|.└.
0821 | F7 C8 CA 9D 01 0C D0 02 C6 0F 88 D0 F5 20 2F 00 | ─|└...┐.─..┐| /.
0831 | D0 EF 20 2F 00 A8 20 2F 00 C0 02 D0 E5 85 01 58 | ┐─ /.▒ /.─.┐|..X
0841 | 4C 10 08 00 0D A5 38 D0 02 C6 39 C6 38 AD C6 0A | L....|8┐.─9─8└─.
0851 | C9 FF 60                                        | ┐▦─
```

Figure 4-12. BEEFTRUCKER Decompression header

```
080B  A2 00     LDX #$00
080D  78        SEI
080E  BC 98 07  LDY $0798,X
0811  94 81     STY $81,X
0813  CA        DEX
0814  30 F8     BMI $080E
0816  4C 02 00  JMP $0002
0819  BD C6 0A  LDA $0AC6,X
081C  9D D4 07  STA $07D4,X
081F  CA        DEX
0820  10 F7     BPL $0819
0822  C8        INY
...
084E  AD C6 0A  LDA $0AC6
0851  C9 FF     CMP #$FF
0853  60        RTS
```

Location (incl)	Name	Description
$0801 - $080A	BASIC starting hook	Contains the BASIC starting hook which is “4673 SYS 2059”, corresponding to $080B. I am not sure why I chose 4673 ($41/$12): there must have been some specific reason.
$080B - $0818	Relocator for Decompression Routine	Transfers the decompression routine to address $0002. The coding is deceptive, as I exploit the wraparound of zero-page indexed addressing. Control is then transferred to $02. I do neatly overwrite $01 with the value $00 from $0818 as a side-effect.
$0819 - $0853	Decompression Routine	Contains the core routine that actually carries out decompression (once relocated) to zeropage.

3.4.3. Decompression routine

The decompression routine is relocated to zero-page memory, and is mostly code. By sitting in zero-page, it executes very fast.

Figure 4-13. BEEFTRUCKER Decompression routine

```
0002  BD C6 0A  LDA $0AC6,X
0005  9D D4 07  STA $07D4,X
0008  CA        DEX
0009  10 F7     BPL $0002
000B  C8        INY
000C  CA        DEX
000D  9D 01 0C  STA $0C01,X
0010  D0 02     BNE $0014
0012  C6 0F     DEC $0F
0014  88        DEY
0015  D0 F5     BNE $000C
0017  20 2F 00  JSR $002F
001A  D0 EF     BNE $000B
001C  20 2F 00  JSR $002F
001F  A8        TAY
0020  20 2F 00  JSR $002F
0023  C0 02     CPY #$02
0025  D0 E5     BNE $000C
0027  85 01     STA $01
0029  58        CLI
002A  4C 10 08  JMP $0810
002D  00        ???
002E  0D        ???
002F  A5 38     LDA $38
0031  D0 02     BNE $0035
0033  C6 39     DEC $39
0035  C6 38     DEC $38
0037  AD C6 0A  LDA $0AC6
003A  C9 FF     CMP #$FF
003C  60        RTS
```

Location (incl)	Name	Description
\$0002 - \$000A	Relocator for Decompression Data	Transfers the end of the file to the start of the file, so as to overwrite the unrelocated decompression routine, and to make the file a contiguous block again. This has the advantage of starting the compressed data at \$07D4, allowing extra space to allow for uncompressible data at the start of the file (some other compressors would have a problem with this) as the packer unwinds.
\$000B - \$0026	Decompressor Write	Writes the uncompressed data to memory, and also takes care of expanding tokenised sequences. The Accumulator value is written to memory at \$000D, and continues to loop around between \$000C and \$0016 (inclusive) until the Y register reaches zero.

		The X register is used as the write-memory offset index, and the data at $000E/$000F is used as the write-memory base address. The code at $0010 to $0013 (inclusive) ensures that if the X register has reached zero, then the base address is decremented by 256 bytes. This ensures that the write address continues to spiral downwards. It is important toe realise that this approach results in minimal clock-cycles, as opposed to using offset indexed addressing. The code at $0017 to $0026 (inclusive) involves fetching another input byte, and writing it (at $001A) if it is not a ControlByte: if it is, then the DataByte is fetched at $001C (and transferred to the Y register), and the DataCount is fetched at $0020. If the special DataByte of $02 is not detected at $0023, then execution branches back to $000B to write out the sequence (DataByte in Accumulator, DataCount in Y register).
$0027 - $002C	Decompressor Execute	Executes the uncompressed data, by storing the last DataByte (configurable at options menu) into the zero-page address $01, clearing the interrupt register (configurable at options menu) and calling the JMP address of $0810 (configurable at options menu).
$002D - $002E	End Of Program Register	The end-of-program registers marking the end of the uncompressed data. I cleverly arranged the code so that these two bytes were ideally located.
$002F - $003C	Decompressor Read	Reads the compressed data from memory, and takes care of checking for the ControlByte. Firstly, the read address is decremented by one at $002F to $0036 (inclusive), and then the data is read in at $0037. The comparison at $003A checks for the ControlByte, and leaves the result in the register flags, before returning from the routine at $003C with an RTS. By using a single load (LDA), this code executes fast.

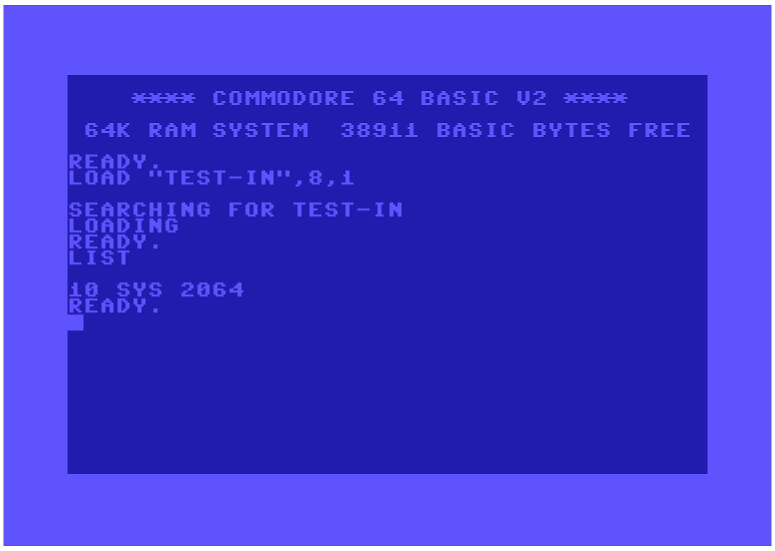
It may have been possible to use undocumented op-codes, or other side effects, to reduce decompression time and space, but this solution is quite close to optimal.

4. Operation by example

The *“test-in”* program has been used as an example to demonstrate the compressor and decompressor in operation.

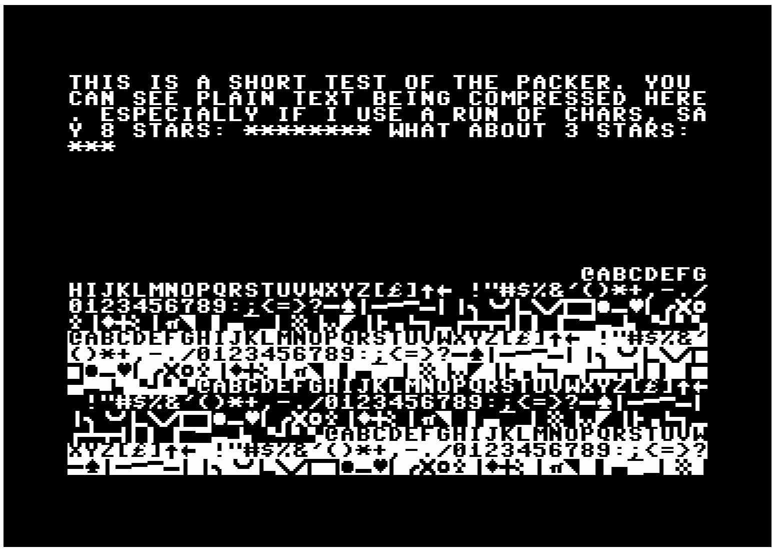
“test-in” is a small 6 block program that contains a short assembly language routine to copy 4 blocks of data onto the screen memory ($0400 to $07FF). The routine and BASIC “SYS” statement consume $0801 to $08FF, and the screen data consumes $0900 to $0CFF.

Figure 4-14. BEEFTRUCKER Example “test-in” loaded and listed



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

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
LOAD "TEST-IN",8,1

SEARCHING FOR TEST-IN
LOADING
READY.
LIST

10 SYS 2064
READY.
```

Figure 4-15. BEEFTRUCKER Example “test-in” executed



```
THIS IS A SHORT TEST OF THE PACKER. YOU
CAN SEE PLAIN TEXT BEING COMPRESSED HERE
. ESPECIALLY IF I USE A RUN OF CHARS, SA
Y 8 STARS: ******** WHAT ABOUT 3 STARS:
***
```

Figure 4-16. BEEFTRUCKER Example “test-in” contents

```
0801 | 0C 08 0A 00 9E 20 32 30 36 34 00 00 00 00 00 A9 | ..... 2064...../
0811 | 00 8D 20 D0 8D 21 D0 A8 A9 01 99 00 D8 99 00 D9 | .. ┐.!┐▒/...■..|
0821 | 99 00 DA 99 00 DB C8 D0 F1 B9 00 09 99 00 04 B9 | ..■..┼|┐┴─.....─
0831 | 00 0A 99 00 05 B9 00 0B 99 00 06 B9 00 0C 99 00 | .....─.....─....
0841 | 07 C8 D0 E5 78 4C 46 08 00 00 00 00 00 00 00 00 | .|┐|■LF.........
0851 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0861 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0871 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
```

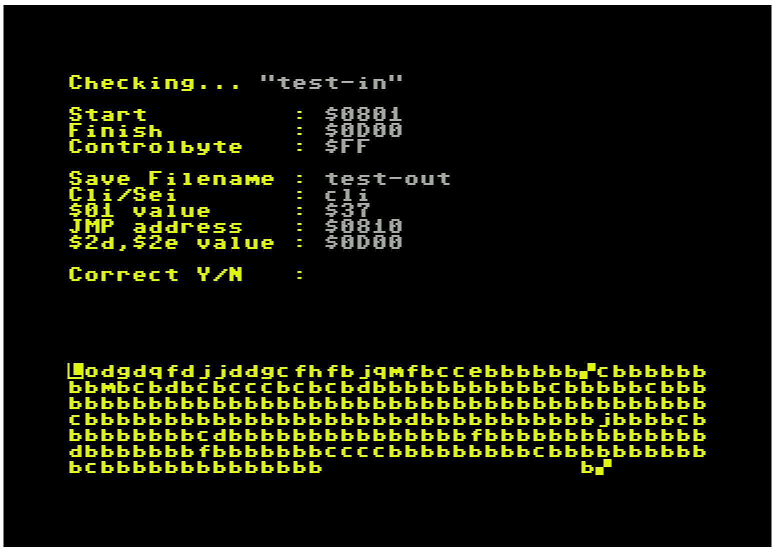
```
0881 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0891 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
08A1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
08B1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
08C1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
08D1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
08E1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
08F1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 14 | ................
0901 | 08 09 13 20 09 13 20 01 20 13 08 0F 12 14 20 14 | ... .. . ..... .
0911 | 05 13 14 20 0F 06 20 14 08 05 20 10 01 03 0B 05 | ... .. ... .....
0921 | 12 2E 20 19 0F 15 20 03 01 0E 20 13 05 05 20 10 | .. ... ... ... .
0931 | 0C 01 09 0E 20 14 05 18 14 20 02 05 09 0E 07 20 | .... .... ..... 
0941 | 03 0F 0D 10 12 05 13 13 05 04 20 08 05 12 05 2E | .......... .....
0951 | 20 05 13 10 05 03 09 01 0C 0C 19 20 09 06 20 09 |  .......... .. .
0961 | 20 15 13 05 20 01 20 12 15 0E 20 0F 06 20 03 08 |  ... . ... .. ..
0971 | 01 12 13 2C 20 13 01 19 20 38 20 13 14 01 12 13 | ..., ... 8 .....
0981 | 3A 20 2A 2A 2A 2A 2A 2A 2A 2A 20 17 08 01 14 20 | : ******** .... 
0991 | 01 02 0F 15 14 20 33 20 13 14 01 12 13 3A 20 2A | ..... 3 .....: *
09A1 | 2A 2A 20 20 20 20 20 20 20 20 20 20 20 20 20 20 | **              
09B1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
09C1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
09D1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
09E1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
09F1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A01 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A11 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A21 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A31 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A41 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A51 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A61 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A71 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A81 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0A91 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0AA1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0AB1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0AC1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0AD1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0AE1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |                 
0AF1 | 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 00 |                .
0B01 | 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 | ................
0B11 | 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 | ............... 
0B21 | 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 | !"#$%&'()*+,-./0
0B31 | 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 | 123456789:;<=>?
0B41 | 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 | ABCDEFGHIJKLMNOP
0B51 | 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 | QRSTUVWXYZ[£]^«—
0B61 | 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 | ■|────||┐└┘└\/┌┐
0B71 | 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 | ■─■|┌XO■|■┼▒|ò\.
0B81 | 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 | ................
0B91 | 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 | ............... 
0BA1 | A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 | ¦▄▔▁|▒|▒/|├▄┐▔┌
0BB1 | B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 | ┴┬┤|||───┘▞┘▀▦─
0BC1 | C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 | ■|────||┐└┘└\/┌┐
0BD1 | D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 | ■─■|┌XO■|■┼▒|ò\
0BE1 | E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 | ¦▄▔▁|▒|▒/|├▄┐▔┌
0BF1 | F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF 00 | ┴┬┤|||───┘▞┘▀▦.
0C01 | 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 | ................
0C11 | 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 | ............... 
```

```
0C21 | 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 | !"#$%&'()*+,-./0
0C31 | 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 | 123456789:;<=>?
0C41 | 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 | ABCDEFGHIJKLMNOP
0C51 | 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 | QRSTUVWXYZ[£]^«—
0C61 | 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 | ■|————||┐└┘└\/┌┐
0C71 | 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 | ■—■|┌XO■|■┼▒|ò\.
0C81 | 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 | ................
0C91 | 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 | ...............
0CA1 | A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 | ¦▄▔▁|▒|▒/|├▄┘▔┌
0CB1 | B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 | ┴┬┤|||———┘▗▀┘▀▦—
0CC1 | C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 | ■|————||┐└┘└\/┌┐
0CD1 | D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 | ■—■|┌XO■|■┼▒|ò\
0CE1 | E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 | ¦▄▔▁|▒|▒/|├▄┘▔┌
0CF1 | F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF    | ┴┬┤|||———┘▗▀┘▀▦
```

The compression process starts with the selection of *"test-in"* from the compressor menu. The compressor first "scans" the file to develop statistics that are shown at the bottom of the screen. The 256 byte "map" contains a count of each particular type of byte in the file, for instance, there are many "b", which indicate "2" as a result of the two blocks of screen memory that have a complete set of all characters (0 to 255 inclusive). In the very bottom corner, the two bytes indicate statistics for the selected control byte. Here it shows that there are 2 instances of $FF.

The configuration options reflect information about the program. Here it shows that the program is called *"test-in"*, and starts at $0801, and ends at $0D00. The control byte has been selected as $FF, as it has the least number of instances in the file. The user can configure the save filename (there is no default); the CLI/SEI value (CLI is the default); the $01 value ($37 is the default); the JMP address ($080D is the default); and the $2D/$2E value (program end address is the default). The user can select 'Y' or 'N'. I wish I had the opportunity to rewrite this, to improve the menu and the options. Also, the statistics collection process could be optimised (to select not just on pure byte count, but on post-compressed byte count).

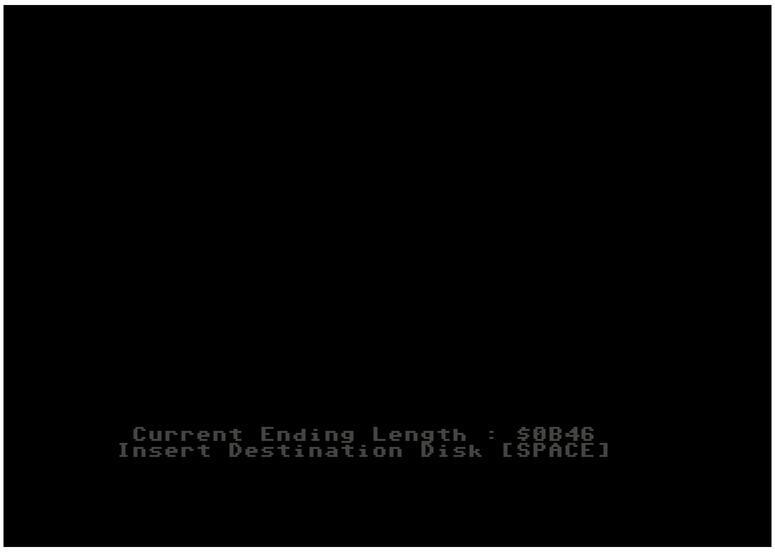
Figure 4-17. BEEFTRUCKER Example "test-in" compression config screen



```
Checking... "test-in"

Start          : $0801
Finish         : $0D00
Controlbyte    : $FF

Save Filename  : test-out
Cli/Sei        : cli
$01 value      : $37
JMP address    : $0810
$2d,$2e value  : $0D00

Correct Y/N    :



█odgdqfdjjddgcfhfbjqmfbccebbbbbb▞cbbbbbb
bbmbcbdbcbcccbcbcbdbbbbbbbbbbbcbbbbbcbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
cbbbbbbbbbbbbbbbbbbbbdbbbbbbbbbbbjbbbbcb
bbbbbbbbcdbbbbbbbbbbbbbbbfbbbbbbbbbbbbbb
dbbbbbbbfbbbbbbbccccbbbbbbbbbcbbbbbbbbbb
bcbbbbbbbbbbbbbbb                b▞
```

After verifying the configuration options, the second pass reads in and compresses the file, while displaying a running total at the bottom of the screen. Once complete, the user can press the SPACE key to save the file, and can continue to press it to save multiple copies.

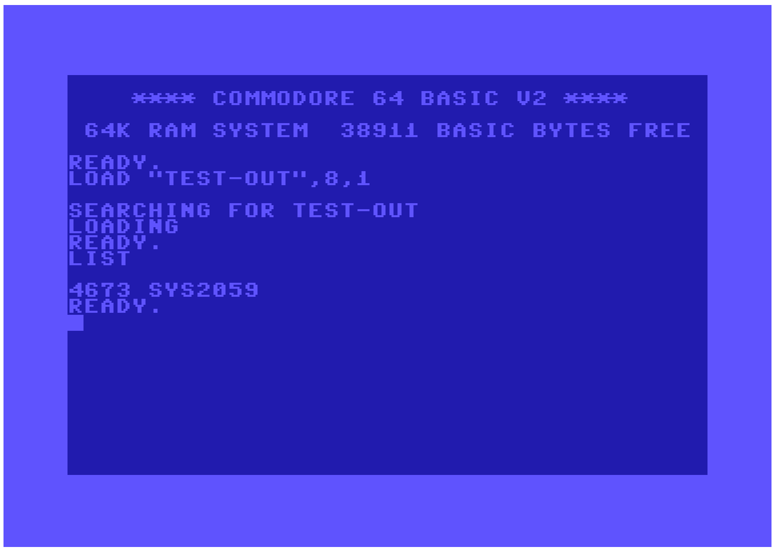
Figure 4-18. BEEFTRUCKER Example “test-in” compression save screen



```
  Current Ending Length : $0B46
 Insert Destination Disk [SPACE]
```

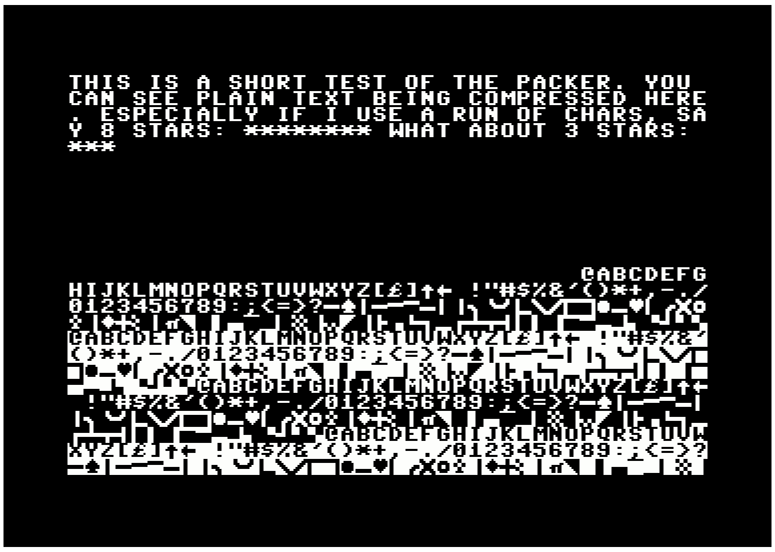
“test-out” is the resulting program, and is now 4 blocks long. Its length runs from $0801 to $0B46, an improvement of 35% over *“test-in”*, which runs from $0801 to $0D00. When the program is loaded and listed, it now reveals the decompression “SYS” statement, rather than that of the original program. There is an obvious difference between contents of the two programs: *“test-out”* definitely looks compressed.

Figure 4-19. BEEFTRUCKER Example “test-out” loaded and listed



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

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
LOAD "TEST-OUT",8,1

SEARCHING FOR TEST-OUT
LOADING
READY.
LIST

4673 SYS2059
READY.
```

Figure 4-20. BEEFTRUCKER Example “test-out” executed



```
THIS IS A SHORT TEST OF THE PACKER. YOU
CAN SEE PLAIN TEXT BEING COMPRESSED HERE
. ESPECIALLY IF I USE A RUN OF CHARS, SA
Y 8 STARS: ******** WHAT ABOUT 3 STARS:
***
```

Figure 4-21. BEEFTRUCKER Example “test-out” contents

```
0801 | 0B 08 41 12 9E 32 30 35 39 00 A2 00 78 BC 98 07 | ..A..2059.▄.■▀..
0811 | 94 81 CA 30 F8 4C 02 00 BD C6 0A 9D D4 07 CA 10 | ..└0─L..┘─..|.└.
0821 | F7 C8 CA 9D 01 0C D0 02 C6 0F 88 D0 F5 20 2F 00 | ─|└...┐.─..┐| /.
0831 | D0 EF 20 2F 00 A8 20 2F 00 C0 02 D0 E5 85 01 58 | ┐─ /.▒ /.─.┐|..X
0841 | 4C 10 08 00 0D A5 38 D0 02 C6 39 C6 38 AD C6 0A | L....|8┐.─9─8└─.
0851 | C9 FF 60 09 0E 20 14 05 18 14 20 02 05 09 0E 07 | ┐▦─.. .... .....
0861 | 20 03 0F 0D 10 12 05 13 13 05 04 20 08 05 12 05 |  .......... ....
0871 | 2E 20 05 13 10 05 03 09 01 0C 0C 19 20 09 06 20 | . .......... ..
0881 | 09 20 15 13 05 20 01 20 12 15 0E 20 0F 06 20 03 | . ... . ... .. .
0891 | 08 01 12 13 2C 20 13 01 19 20 38 20 13 14 01 12 | ...., ... 8 ....
08A1 | 13 3A 20 2A 08 FF 20 17 08 01 14 20 01 02 0F 15 | .: *.▦ .... ....
08B1 | 14 20 33 20 13 14 01 12 13 3A 20 2A 2A 2A 20 00 | . 3 .....: *** .
08C1 | FF 20 5D FF 00 01 02 03 04 05 06 07 08 09 0A 0B | ▦ ]▦............
08D1 | 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B | ................
08E1 | 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B | .... !"#$%&'()*+
08F1 | 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B | ,-./0123456789:;
0901 | 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B | <=>?
0911 | 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B | LMNOPQRSTUVWXYZ[
0921 | 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A 6B | £]^«─■|────||┐└┘
0931 | 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B | └\/┌┐■─■|┌XO■|■┼
0941 | 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 8A 8B | ▒|ò\............
0951 | 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B | ................
0961 | 9C 9D 9E 9F A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB | .... ¦▄▔_|▒|▒/|├
0971 | AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB | ▄└┐─┌┴┬┤|||───┘▄
0981 | BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB | ▀┘▀▦─■|────||┐└┘
0991 | CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB | └\/┌┐■─■|┌XO■|■┼
09A1 | DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB | ▒|ò\ ¦▄▔_|▒|▒/|├
09B1 | EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB | ▄└┐─┌┴┬┤|||───┘▄
09C1 | FC FD FE FF 01 FF 00 01 02 03 04 05 06 07 08 09 | ▀┘▀▦.▦..........
09D1 | 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 | ................
09E1 | 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 | ...... !"#$%&'()
09F1 | 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 | *+,-./0123456789
0A01 | 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 | :;<=>?
0A11 | 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 | JKLMNOPQRSTUVWXY
```

```
0A21 | 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 | Z[£]^«─■|────||┐
0A31 | 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 | └┘└\/┌┐■─■|┌XO■|
0A41 | 7A 7B 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 | ■┼▒|ò\..........
0A51 | 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 | ................
0A61 | 9A 9B 9C 9D 9E 9F A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 | ...... ¦▄─_|▒|▒/
0A71 | AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 | |├▄└┐─┌┴┬┤|||──
0A81 | BA BB BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 | ┘▄▀┘▀▦─■|────||┐
0A91 | CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 | └┘└\/┌┐■─■|┌XO■|
0AA1 | DA DB DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 | ■┼▒|ò\ ¦▄─_|▒|▒/
0AB1 | EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 | |├▄└┐─┌┴┬┤|||──
0AC1 | FA FB FC FD FE FF 37 02 FF 0C 08 0A 00 9E 20 32 | ┘▄▀┘▀▦7.▦..... 2
0AD1 | 30 36 34 00 05 FF A9 00 8D 20 D0 8D 21 D0 A8 A9 | 064..▦/.. ┐.!┐▒/
0AE1 | 01 99 00 D8 99 00 D9 99 00 DA 99 00 DB C8 D0 F1 | ...■..|..■..┼|┐┴
0AF1 | B9 00 09 99 00 04 B9 00 0A 99 00 05 B9 00 0B 99 | ─.....─.....─...
0B01 | 00 06 B9 00 0C 99 00 07 C8 D0 E5 78 4C 46 08 00 | ..─.....|┐|■LF..
0B11 | B7 FF 14 08 09 13 20 09 13 20 01 20 13 08 0F 12 | ─▦.... .. . ....
0B21 | 14 20 14 05 13 14 20 0F 06 20 14 08 05 20 10 01 | . .... .. ... ..
0B31 | 03 0B 05 12 2E 20 19 0F 15 20 03 01 0E 20 13 05 | ..... ... ... ..
0B41 | 05 20 10 0C 01                                  | . ...
```

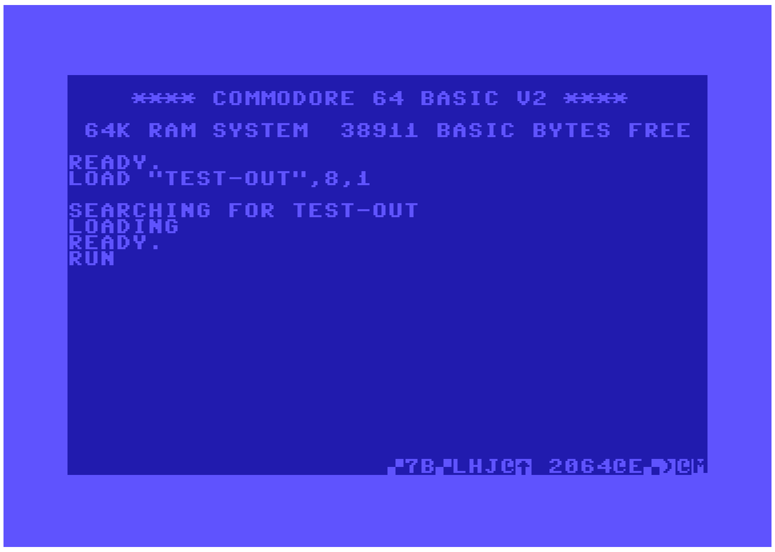
An inspection of the contents reveals the following:

- The **decompression routine** can be seen at the start of the *"test-out"* program. In fact, the start of the compressed data has now been appended to the end of the program, and can be seen starting from $0AC6. The decompressor, when executing, first copies its decompression routine to zero-page, and then copies the end of the program ($0AC6) to the beginning ($07D4), so as to recover the original ordering. It does this before it starts to "decompress" the data.
- The **run of 8 asterix characters** ($2A) which is present in *"test-in"* at $09A3 as "********" is now present in *"test-out"* at $08A4 as $2A/$08/$FF. This is the compressed pattern of ControlByte($FF)/DataCount($08)/DataByte($2A).
- The **run of 3 asterix characters** ($2A) which is present in *"test-in"* at $09A0 as "***" is now present in *"test-out"* at $08BC as $2A/$2A/$2A. This has not been compressed, as would not result in any reduction (i.e. otherwise it would be replaced by the string $2A/$03/$FF).
- The **run of >256 characters with no possibility of compression** is present in *"test-in"* at $0B01 to $0CFF. It is present in *"test-out"* at $08C4, and cannot be compressed. In fact, it has expanded, because the ControlByte is present. In *"test-in"* at $0BFF is the value $FF, which has been expanded in *"test-out"* at $09C4 as $FF/$01/$FF. This must be done so that the decompressor can distinguish legitimate control bytes.
- The **run of >256 characters with possibility of compression** is present in *"test-in"* at $09A3 to $0AFF and consists entirely of the value $20. The compressed tokens are present in *"test-out"* at $08BF as $20/$00/$FF and $20/$5D/$FF. This illustrates the 8-bit limitation of the string size. A more intelligent packer could scan the input file and decide whether to implement a 8bit+1 policy (i.e. the higher bit set could indicate that the string length is 15bits).
- The **special case "terminator"** is present in *"test-in"* at $0AC6 as $37/$02/$FF. This portion of data is copied to $07D4 onwards before decompression takes place, and the $02 signifies the end of the compressed

data – as it cannot be present as a value anywhere else. The data ($37) is used as the value to be placed into $01.

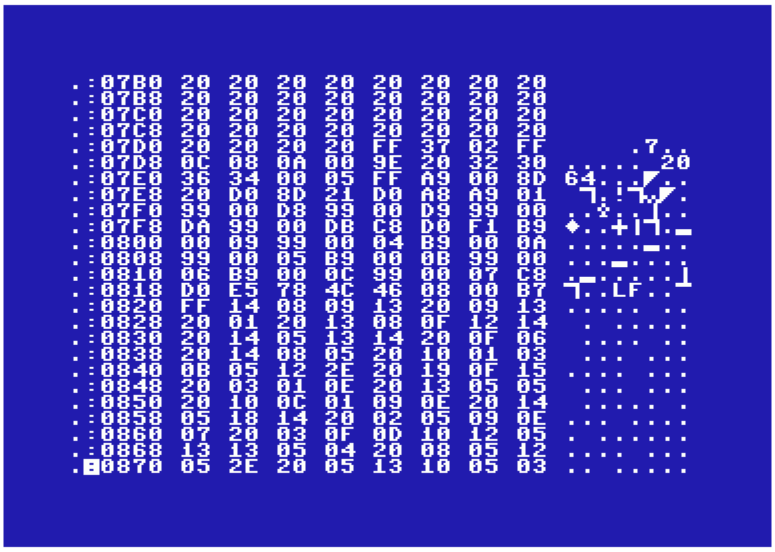
- If the decompressor is suspended, the **$07D4 information** can be seen at the bottom of the display memory. The **decompression routine** is also visible at $0002, and the $2D/$2E values are “magically” put into place.

Figure 4-22. BEEFTRUCKER Example “test-out” decompression stubs



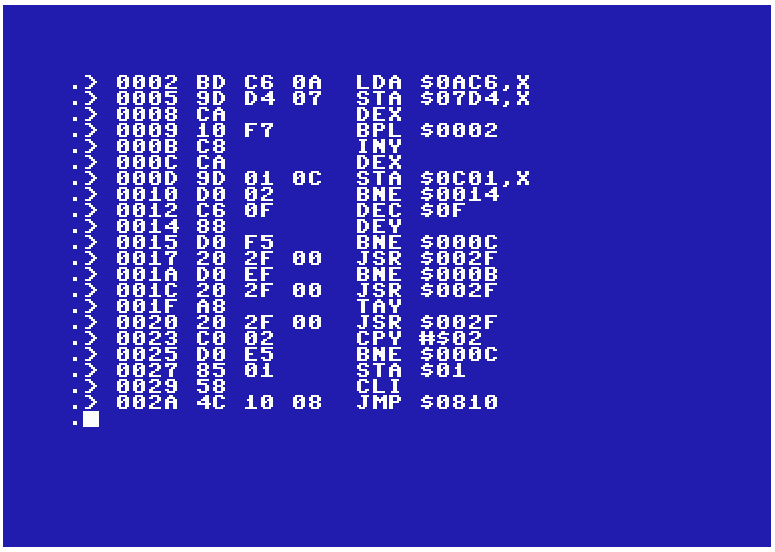
```
    **** COMMODORE 64 BASIC V2 ****

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
LOAD "TEST-OUT",8,1

SEARCHING FOR TEST-OUT
LOADING
READY.
RUN




                                        
                                        
                    ▖▘7B▖▘LHJC@↑ 2064@E▖▘)]@M
```

Figure 4-23. BEEFTRUCKER Example “test-out” decompression stubs



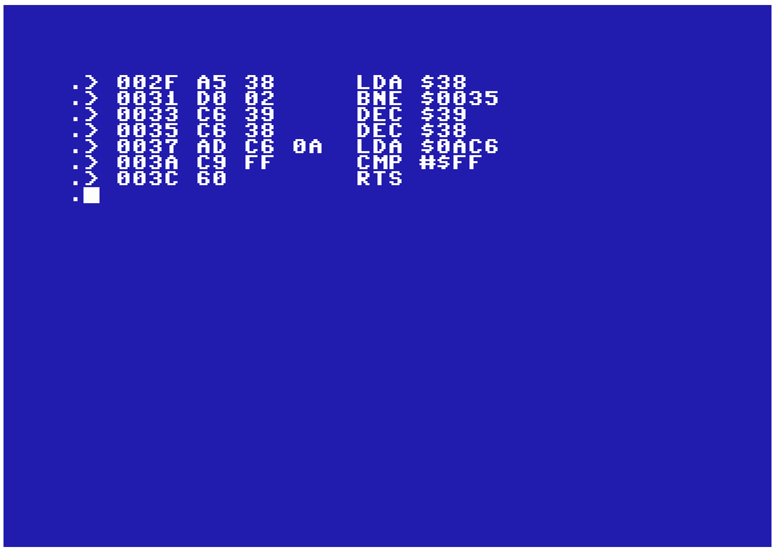
```
.:07B0 20 20 20 20 20 20 20 20
.:07B8 20 20 20 20 20 20 20 20
.:07C0 20 20 20 20 20 20 20 20
.:07C8 20 20 20 20 20 20 20 20
.:07D0 20 20 20 20 FF 37 02 FF     .7..
.:07D8 0C 08 0A 00 9E 20 32 30 ..... 20
.:07E0 36 34 00 05 FF A9 00 8D 64...◤..
.:07E8 20 D0 8D 21 D0 A8 A9 01  ┐.!┐.◤.
.:07F0 99 00 D8 99 00 D9 99 00 ..♤..┃..
.:07F8 DA 99 00 DB C8 D0 F1 B9 ♦..┼|┐.▁
.:0800 00 09 99 00 04 B9 00 0A .....▔..
.:0808 99 00 05 B9 00 0B 99 00 ...▁....
.:0810 06 B9 00 0C 99 00 07 C8 .▔.....┴
.:0818 D0 E5 78 4C 46 08 00 B7 ┬..LF..
.:0820 FF 14 08 09 13 20 09 13 ..... ..
.:0828 20 01 20 13 08 0F 12 14  . .....
.:0830 20 14 05 13 14 20 0F 06  .... ..
.:0838 20 14 08 05 20 10 01 03  ... ...
.:0840 0B 05 12 2E 20 19 0F 15 .... ...
.:0848 20 03 01 0E 20 13 05 05  ... ...
.:0850 20 10 0C 01 09 0E 20 14  ..... .
.:0858 05 18 14 20 02 05 09 0E ... ....
.:0860 07 20 03 0F 0D 10 12 05 . ......
.:0868 13 13 05 04 20 08 05 12 .... ...
.:0870 05 2E 20 05 13 10 05 03 .. .....
```

Figure 4-24. BEEFTRUCKER Example “test-out” decompression routine



```
.> 0002 BD C6 0A  LDA $0AC6,X
.> 0005 9D D4 07  STA $07D4,X
.> 0008 CA        DEX
.> 0009 10 F7     BPL $0002
.> 000B C8        INY
.> 000C CA        DEX
.> 000D 9D 01 0C  STA $0C01,X
.> 0010 D0 02     BNE $0014
.> 0012 C6 0F     DEC $0F
.> 0014 88        DEY
.> 0015 D0 F5     BNE $000C
.> 0017 20 2F 00  JSR $002F
.> 001A D0 EF     BNE $000B
.> 001C 20 2F 00  JSR $002F
.> 001F A8        TAY
.> 0020 20 2F 00  JSR $002F
.> 0023 C0 02     CPY #$02
.> 0025 D0 E5     BNE $000C
.> 0027 85 01     STA $01
.> 0029 58        CLI
.> 002A 4C 10 08  JMP $0810
.
```

Figure 4-25. BEEFTRUCKER Example “test-out” decompression routine



```
.> 002F A5 38     LDA $38
.> 0031 D0 02     BNE $0035
.> 0033 C6 39     DEC $39
.> 0035 C6 38     DEC $38
.> 0037 AD C6 0A  LDA $0AC6
.> 003A C9 FF     CMP #$FF
.> 003C 60        RTS
.
```

5. Deployment, and use in the field

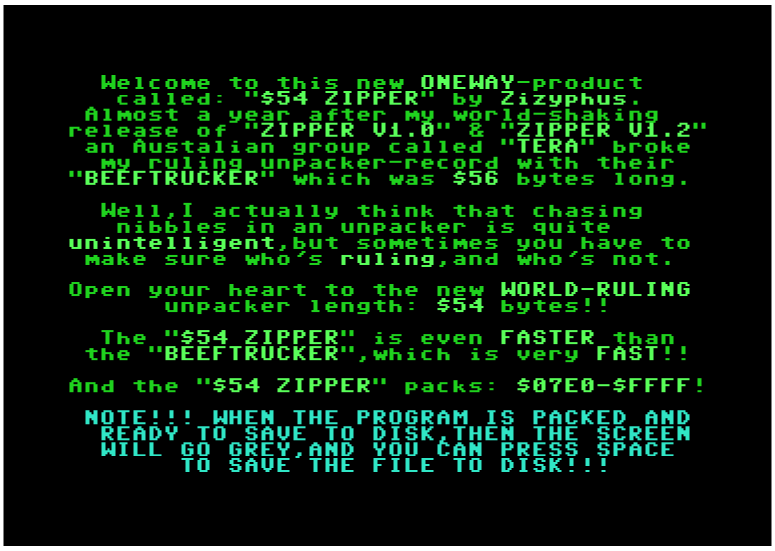
Once I had created BEEFTRUCKER, it was used in all of my subsequent productions, including DAMNABLE. Other TERA coders used it sporadically, but it was in no way mandated for use within the group, we didn’t set rules of that sort.

BEEFTRUCKER can be found on a number of compilation disks. There seem to be three versions in the field. Firstly, the original version with all three parts (Intro, Overview, Compressor), and then a stripped version with just the Overview and the Compressor, and also an even shorter version with the Compressor only. I did not create nor sanction these other versions, but I should have been aware that they would occur, as the first two parts do constitute a reasonable amount of baggage.

I received little feedback about the program. I have read one criticism: that it destroys zero-page, which it does, as everything from $02 to $3C (inclusive) other than $2D/$2E is destroyed. This can be a program with BASIC programs, but should not be a problem with any sensible assembly language program. I should have included configurable options to account for this.

ZIZYPLUS of ONEWAY did respond to my ante. He created a $54 ZIPPER which is 2 bytes shorter than BEEFTRUCKER. However, he merely found a way to shave 2 bytes from my routine, whereas I had created a new routine from scratch: the $54 ZIPPER can be considered a derivative of my work, and lacks real innovation. I have to have the last word though: chasing nibbles in an unpacker was not my real goal, my real goal was to produce a well balanced product, and put our group into the "major league". I think that I achieved this, as BEEFTRUCKER helped to solidify TERA as a "world class" group, and myself as a leading coder – hardly an unintelligent activity.

Figure 4-26. "$54 ZIPPER" by ZIZYPLUS/ONEWAY



```
   Welcome to this new ONEWAY-product
    called: "$54 ZIPPER" by Zizyphus.
  Almost a year after my world-shaking
 release of "ZIPPER V1.0" & "ZIPPER V1.2"
  an Austalian group called "TERA" broke
   my ruling unpacker-record with their
 "BEEFTRUCKER" which was $56 bytes long.

   Well,I actually think that chasing
    nibbles in an unpacker is quite
 unintelligent,but sometimes you have to
  make sure who's ruling,and who's not.

 Open your heart to the new WORLD-RULING
      unpacker length: $54 bytes!!

   The "$54 ZIPPER" is even FASTER than
  the "BEEFTRUCKER",which is very FAST!!

 And the "$54 ZIPPER" packs: $07E0-$FFFF!

  NOTE!!! WHEN THE PROGRAM IS PACKED AND
   READY TO SAVE TO DISK,THEN THE SCREEN
   WILL GO GREY,AND YOU CAN PRESS SPACE
        TO SAVE THE FILE TO DISK!!!
```

My regrets are: (1) that I put too much baggage in front of the Compressor; (2) that I did not shave off those extra 2 bytes; (3) that I did not create a packer that could pack from $0100 to $FFFF (rather than $07e0 to $FFFF); (4) that I did not put a lot more extra detail and options in the configuration menu; (5) that I did not make the fast load/save routines "optional". It is a pity that I did not get to produce BEEFTRUCKER V2.0. Life rarely offers second chances.

There are references to SLUDGER V5. This was a vague plan to create a cruncher (i.e. a dictionary based compression program). I am not sure why I called it "V5", but it was called SLUDGER because she was my "squeeze" at the time and that was her nickname because it related closely to her real name.

Section 5. C64 Activities Tools Notemakers

1. Background

1.1. Motivations

I cannot recall what inspired me to produce my first notemaker, but I think that I had seen something of a bad approach to a notemaker and thought that I could do much better. In particular, I wanted to develop a "standalone" notemaker, with a built in packer. I am fairly sure that I was the first to do this. The notemakers were also small and enjoyable creations, and not difficult to code.

1.2. The immature version

The first version of the notemaker was born in 1988 as *NOTEMAKER V1.0 by MATT'/TDF*. It was self-reproducing in form, as the same program allowed a user to read and write notes. If a user elected to read a note, they could move through multiple pages, while music was played in the background. If they chose to write a note, then they could create multiple pages of text, and then save a new copy of the program with the text. The program contained a compressor, so that it could compress itself when saving a new copy. My design goal was to provide a single integrated product that was presentable, but compact and efficient. I seem to remember that my notemaker was the first of this type. The concept was widely imitated in form, and even directly modified, such as with *NOTEMAKER V1.1 by ICE CUBE/*.

1.3. The mature versions

As a result of feedback and a desire to improve upon the original product, I created a second version of the notemaker as *Modified releases* of the notemaker were made, and they usually added new music, or new character sets, or allowed new music or character sets to be loaded in by users.

NOTEMAKER V2.0 by MATT'/TERA, which was born in 1989, while I was in TERA. It was rebuilt from scratch, but similar in form. The display screens were clearer and used a new character set, and new music. It was also possible to load new music. I may have introduced a new packing routine. The concept continued to be imitated, and direct enhancements were produced, such as *NOTEMAKER V2.1 by SPC/ATG*.

I wanted to create an "ultra" compact writer, as I thought that many of the notemakers were too large and heavy, and tried to be too stylish or neat. There seemed to be a place for a simple, streamlined, compact notemaker. *Modified* releases of the notemaker were made, and they fixed a few problems with music loading, or allowed new character sets to be used.

STUBBY-NOTER by MATT'/TERA was born. It used the ROM character set, a simple yet sweet display routine, and a short music routine. It also contained a modified packer so that it could save a new copy of itself in a compact form. In one of the Collections there is only one other notemaker anywhere near the same size. The concept was imitated, and direct enhancements were produced, such as *STUBBY-NOTER V2 by JACK ALIEN/* and *STUBBY-NOTER V4+ by SHRI SADHU/*.

I also created a super-short notemaker, that contained an ultra compact routine to uncompress the screen text from a 5 bit representation and then display that on the screen with a modified ROM character set. Modified releases of the notemaker were made, and they allowed other music to be used, or added a few features to the record/playback routines (e.g. changing the background colour).

4 BLOCK NOTER! by MATT'/TERA was created and used for the Damnable demo Release Note.

1.4. The derivative versions

Many other notemakers were released, some of which incorporated technology taken from my original V1.0 or V2.0 notemakers. For instance, *FUTURE NOTER V1.1 by FUTURE TECHNOLOGIES/* used my modified version of MATCHAM's cruncher. It is possible that many other notemakers used this technology, but without an attribution.

In late 1989, and across 1990, it became popular to create a notemaker, and there were many different variations on the theme. Some were poorly executed, in many different ways, but others were well done. Other members of TERA also produced notemakers, including *ECONOVAN (W)RIDER V1.0 by HEX HACKER/*, *HEXCELLENCE V1.0* by HEX-HACKER/ and *EASY NOTER by BULLET/*.

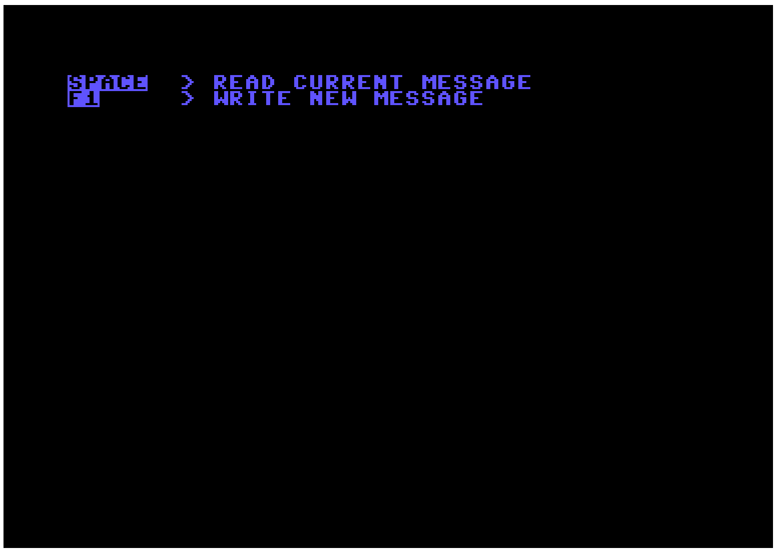
It is a sign of success that most There were many notemaker collections produced. Two are shown here. include versions of my notemakers. The EXTEND collection has most of my productions, and most other TERA productions. The HIBISCH collection indicates that it has "THE FAMOUS 'TERA' NOTERS!".

2. Productions

2.1. NOTEMAKER V1.0 by MATT'/TDF

When NOTEMAKER V1.0 decompresses, the user can choose between reading the current message, or writing a new one. If the user chooses to read, then music will start playing, and each page is rendered when the space key is pressed. If the user chooses to write, an instruction screen appears, and then it is possible for the user to edit multiple screens before saving. Before saving, the program is compressed.

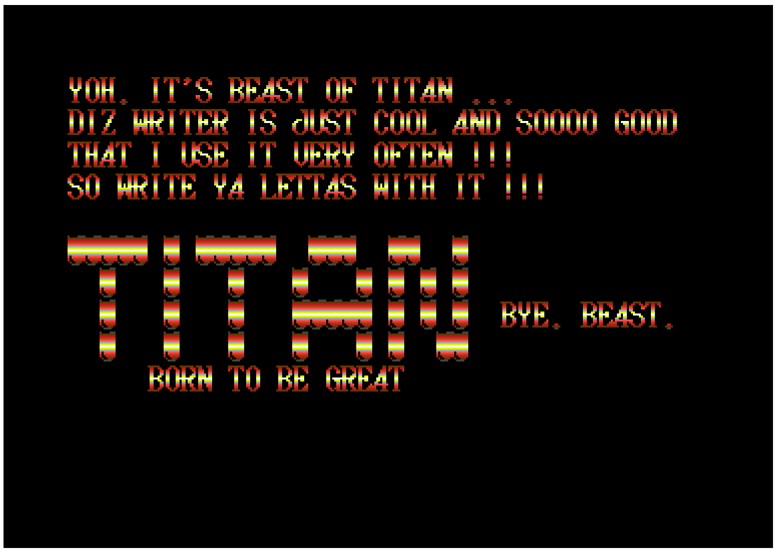
Figure 5-1. NOTEMAKER V1.0 – read/write select



```
SPACE   > READ CURRENT MESSAGE
F1      > WRITE NEW MESSAGE
```

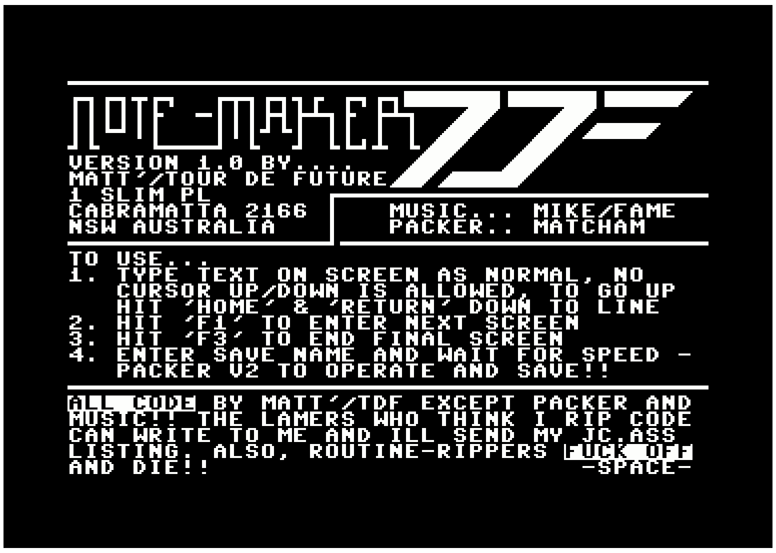
This was produced about the same time as my BACK FROM BEYOND demo, and illustrates the same level of coding and stylistic maturity. Colour bars and a "tinny" sounding music are the primary artefacts. I enjoyed creating the editor, and altering Matcham's packer and combining all of these into the one package.

Figure 5-2. NOTEMAKER V1.0 – read display screen



```
YOH, IT'S BEAST OF TITAN ...
DIZ WRITER IS JUST COOL AND SOOOO GOOD
THAT I USE IT VERY OFTEN !!!
SO WRITE YA LETTAS WITH IT !!!

TITAN   BYE, BEAST.
  BORN TO BE GREAT
```

Figure 5-3. NOTEMAKER V1.0 – write instruction screen



```
VERSION 1.0 BY....
MATT'/TOUR DE FUTURE
1 SLIM PL
CABRAMATTA 2166           MUSIC... MIKE/FAME
NSW AUSTRALIA             PACKER.. MATCHAM

TO USE...
1.  TYPE TEXT ON SCREEN AS NORMAL, NO
    CURSOR UP/DOWN IS ALLOWED, TO GO UP
    HIT 'HOME' & 'RETURN' DOWN TO LINE
2.  HIT 'F1' TO ENTER NEXT SCREEN
3.  HIT 'F3' TO END FINAL SCREEN
4.  ENTER SAVE NAME AND WAIT FOR SPEED -
    PACKER V2 TO OPERATE AND SAVE!!

ALL CODE BY MATT'/TDF EXCEPT PACKER AND
MUSIC!! THE LAMERS WHO THINK I RIP CODE
CAN WRITE TO ME AND ILL SEND MY JC.ASS
LISTING. ALSO, ROUTINE-RIPPERS FUCK OFF
AND DIE!!                        -SPACE-
```

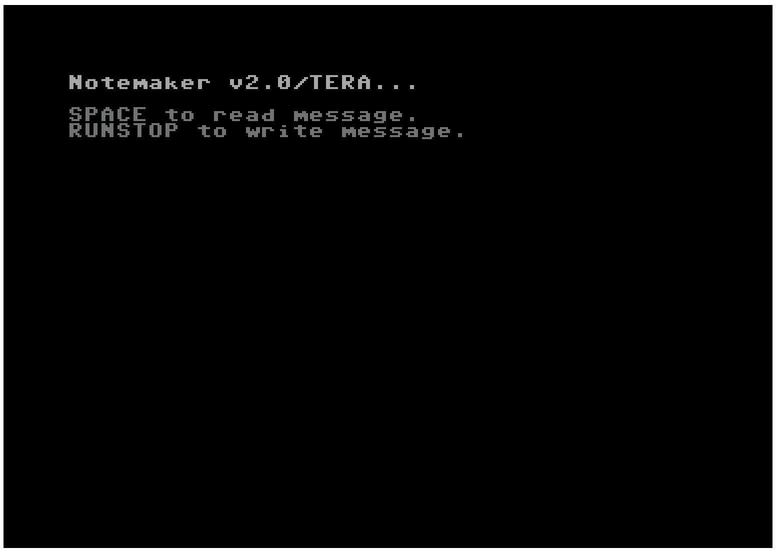
Modified releases of the notemaker were made, and they usually added new music, or new character sets, or allowed new music or character sets to be loaded in by users.

2.2. NOTEMAKER V2.0 by MATT'/TERA

The second version of the notemaker known as NOTEMAKER V2.0 was a complete rewrite, although it continued the same theme. There is an element of stylistic maturity, similar to the intros and demos that I was producing at the time. This version was written with a macro assembler, rather than by hand with a cartridge monitor (which is how the first version was created).

Once unpacked, the user can choose to read or write a message.

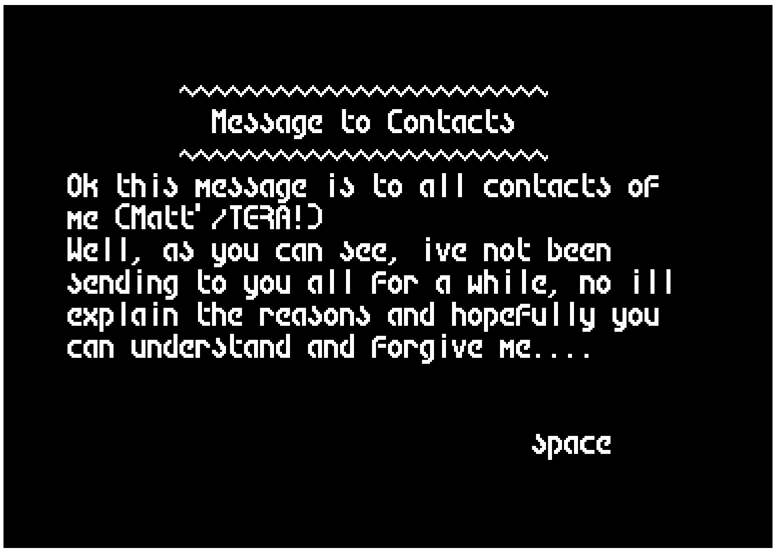
Figure 5-4. NOTEMAKER V2.0 – read/write select



```
Notemaker v2.0/TERA...

SPACE to read message.
RUNSTOP to write message.
```

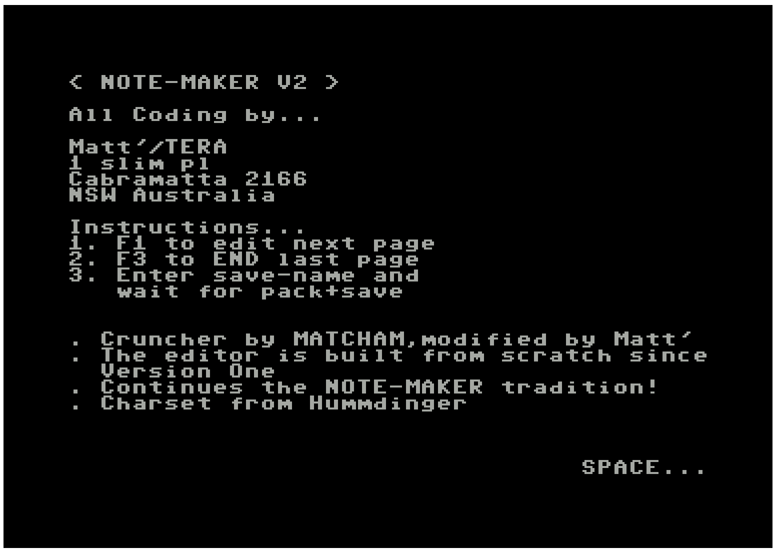
If the user reads, then the text screens are faded in and out in succession. The font is in a single colour and much clearer than the first version. If the user writes, then the instruction screen is followed by the option to load new music – a feature created since the first version. After creating the new text pages, the new copy of the program will be saved once it has "recompressed" itself.

Figure 5-5. NOTEMAKER V2.0 – read display screen



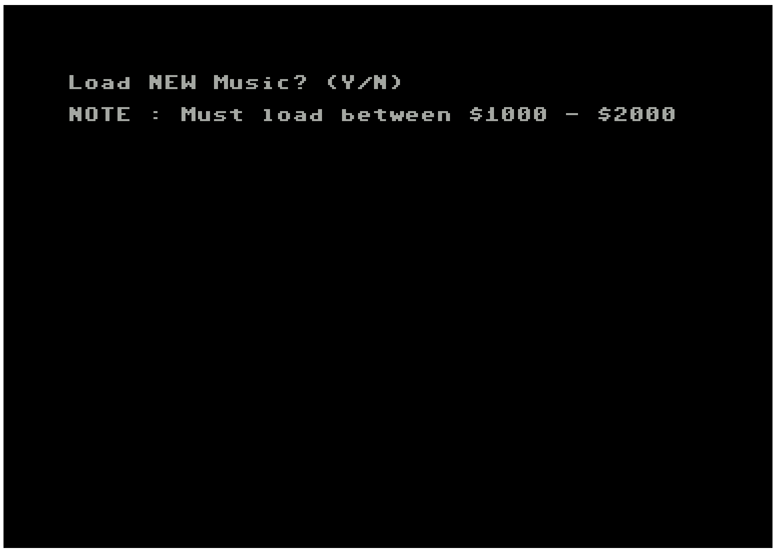
```
      ^^^^^^^^^^^^^^^^^^^^^^^
        Message to Contacts
      ^^^^^^^^^^^^^^^^^^^^^^^
Ok this message is to all contacts of
me (Matt'/TERA!)
Well, as you can see, ive not been
sending to you all for a while, no ill
explain the reasons and hopefully you
can understand and forgive me....


                             space
```

Figure 5-6. NOTEMAKER V2.0 – write instruction screen



```
< NOTE-MAKER V2 >

All Coding by...

Matt'/TERA
1 slim pl
Cabramatta 2166
NSW Australia

Instructions...
1. F1 to edit next page
2. F3 to END last page
3. Enter save-name and
   wait for pack+save

. Cruncher by MATCHAM,modified by Matt'
. The editor is built from scratch since
  Version One
. Continues the NOTE-MAKER tradition!
. Charset from Hummdinger

                                SPACE...
```

Figure 5-7. NOTEMAKER V2.0 – write load music option



```
Load NEW Music? (Y/N)

NOTE : Must load between $1000 - $2000
```

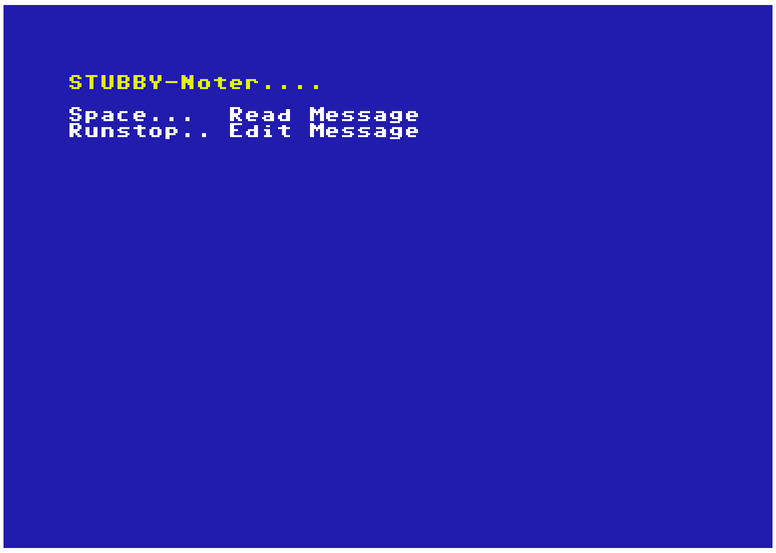
Modified releases of the notemaker were made, and they fixed a few problems with music loading, or allowed new character sets to be used.

2.3. STUBBY-NOTER by MATT'/TERA

The desire for compactness lead to the production of the STUBBY-NOTER. It was designed to be compact and efficient, but still include music. I should have allowed additional tunes to be loaded though. The implementation was carried out using a macro-assembler.

After the program was executed, and decompressed, the user can select to read or edit a message.

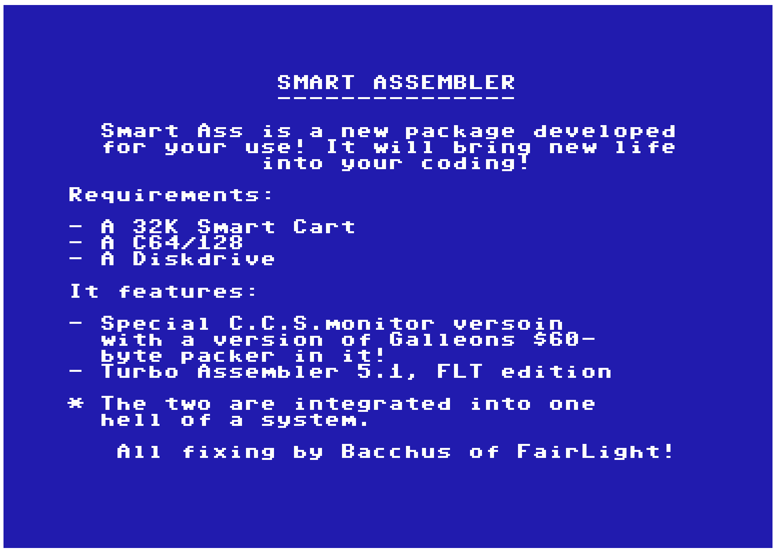
Figure 5-8. STUBBY-NOTER – read/write select screen



```
STUBBY-Noter....

Space...  Read Message
Runstop.. Edit Message
```

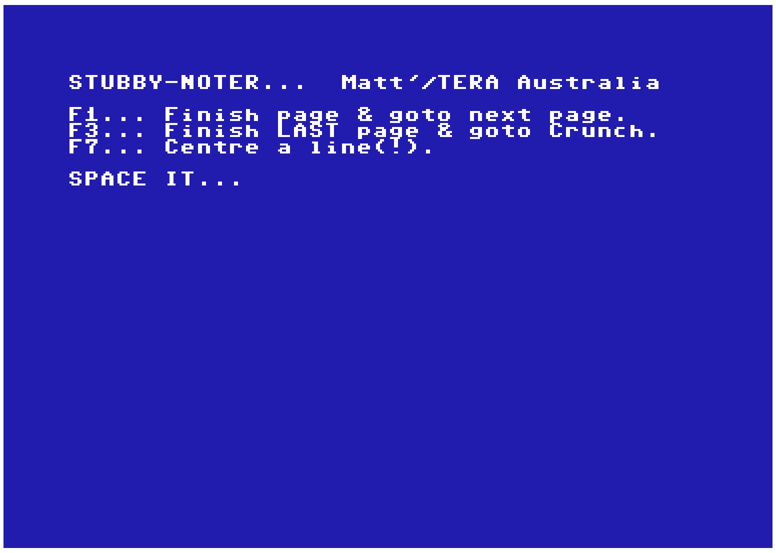
If the user chooses to read, then each page is faded in and out as the space key is pressed. A simple tune is played in the background to accompany the text. If the user chooses to write, then pages can be edited, before saving. Upon saving, the program is "recompressed". Another user can then create a new note from this: the notemakers were designed to be self-replicating.

Figure 5-9. STUBBY-NOTER – read display screen



```
              SMART ASSEMBLER
              ---------------

  Smart Ass is a new package developed
  for your use! It will bring new life
            into your coding!

Requirements:

- A 32K Smart Cart
- A C64/128
- A Diskdrive

It features:

- Special C.C.S.monitor versoin
  with a version of Galleons $60-
  byte packer in it!
- Turbo Assembler 5.1, FLT edition

* The two are integrated into one
  hell of a system.

   All fixing by Bacchus of FairLight!
```

Figure 5-10. STUBBY-NOTER – write instruction screen



```
STUBBY-NOTER...  Matt'/TERA Australia

F1... Finish page & goto next page.
F3... Finish LAST page & goto Crunch.
F7... Centre a line(!).

SPACE IT...
```

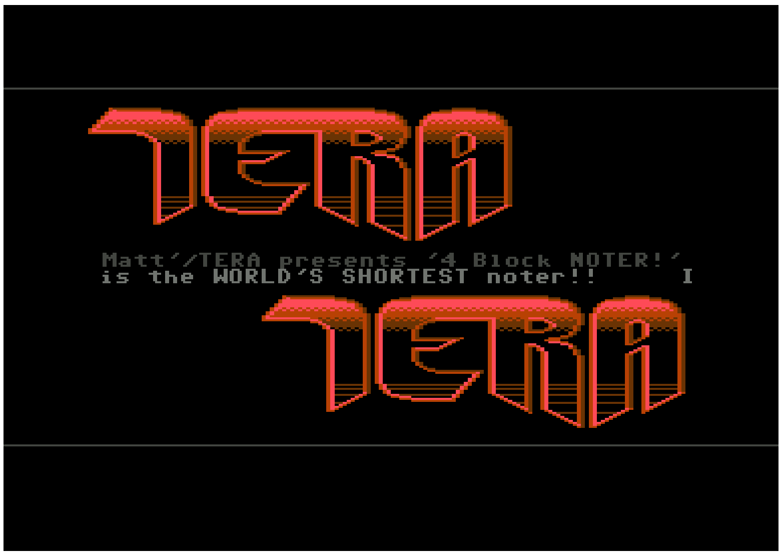
Modified releases of the notemaker were made, and they allowed other music to be used, or added a few features to the record/playback routines (e.g. changing the background colour).

2.4. 4 BLOCK NOTER! by MATT'/TERA

The 4 BLOCK NOTER! was designed to be ultra-compact, and is not a self-reproducing program. It was created to be small, compact and a world record, but I think I was still on a high from BEEFTRUCKER.

The intro is very short and compact, followed by a font selector screen, and then by screen editing instructions.

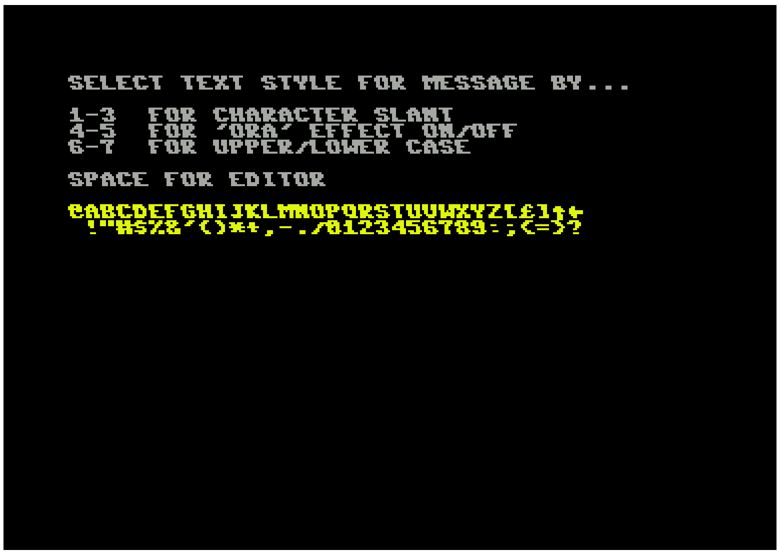
Figure 5-11. 4 BLOCK NOTER! – intro screen



```
Matt'/TERA presents '4 Block NOTER!'
is the WORLD'S SHORTEST noter!!      I
```

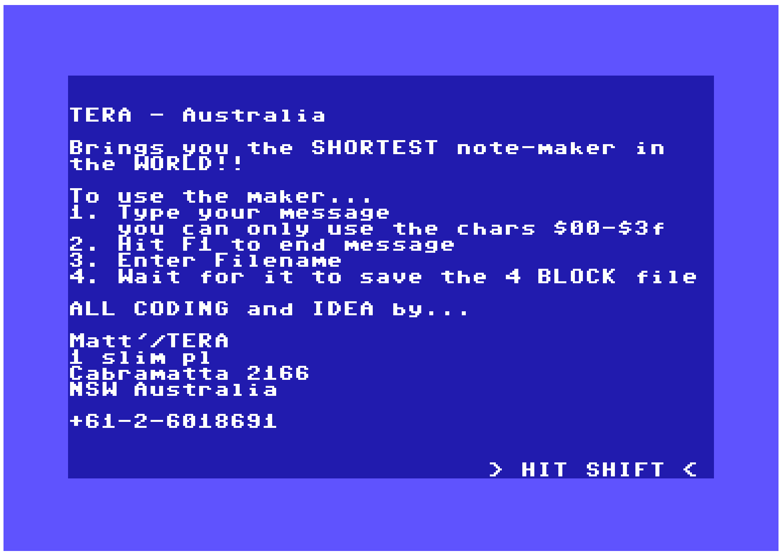
There is only one screen to edit, and then a filename entry to save the note. It was used to create one of the release notes that accompanied DAMNABLE.

Figure 5-12. 4 BLOCK NOTER! - select text screen



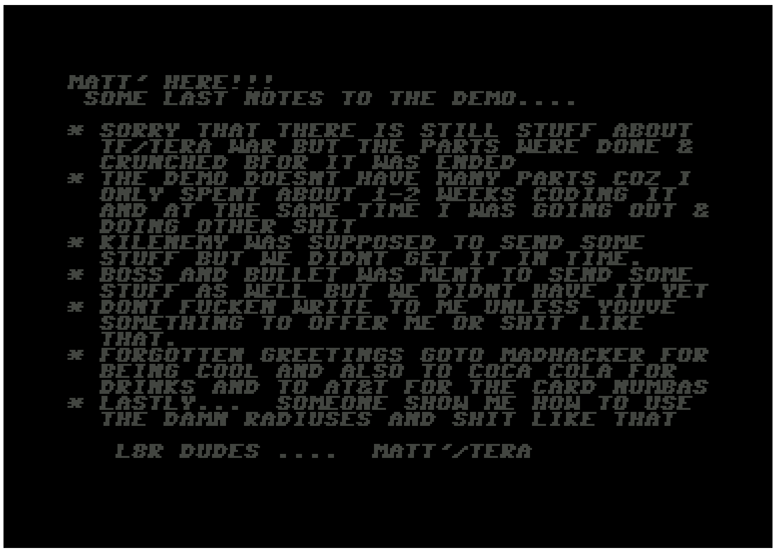
```
SELECT TEXT STYLE FOR MESSAGE BY...

1-3  FOR CHARACTER SLANT
4-5  FOR 'ORA' EFFECT ON/OFF
6-7  FOR UPPER/LOWER CASE

SPACE FOR EDITOR

@ABCDEFGHIJKLMNOPQRSTUVWXYZ[£]↑←
 !"#$%&'()*+,-./0123456789:;<=>?
```

Figure 5-13. 4 BLOCK NOTER! – write instruction screen



```
TERA - Australia

Brings you the SHORTEST note-maker in
the WORLD!!

To use the maker...
1. Type your message
   you can only use the chars $00-$3f
2. Hit F1 to end message
3. Enter Filename
4. Wait for it to save the 4 BLOCK file

ALL CODING and IDEA by...

Matt'/TERA
1 slim pl
Cabramatta 2166
NSW Australia

+61-2-6018691


                           > HIT SHIFT <
```

Figure 5-14. 4 BLOCK NOTER! – read display screen



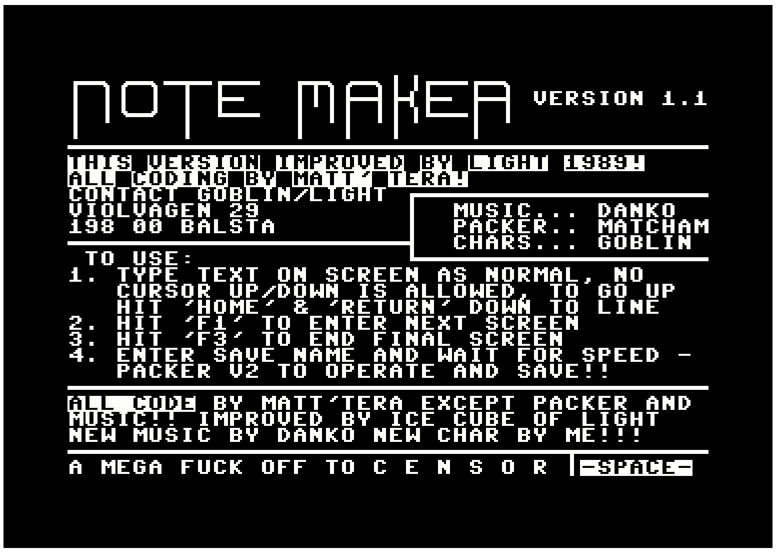
```
MATT' HERE!!!
 SOME LAST NOTES TO THE DEMO....

* SORRY THAT THERE IS STILL STUFF ABOUT
  TF/TERA WAR BUT THE PARTS WERE DONE &
  CRUNCHED BFOR IT WAS ENDED
* THE DEMO DOESNT HAVE MANY PARTS COZ I
  ONLY SPENT ABOUT 1-2 WEEKS CODING IT
  AND AT THE SAME TIME I WAS GOING OUT &
  DOING OTHER SHIT
* KILENEMY WAS SUPPOSED TO SEND SOME
  STUFF BUT WE DIDNT GET IT IN TIME.
* BOSS AND BULLET WAS MENT TO SEND SOME
  STUFF AS WELL BUT WE DIDNT HAVE IT YET
* DONT FUCKEN WRITE TO ME UNLESS YOUVE
  SOMETHING TO OFFER ME OR SHIT LIKE
  THAT.
* FORGOTTEN GREETINGS GOTO MADHACKER FOR
  BEING COOL AND ALSO TO COCA COLA FOR
  DRINKS AND TO AT&T FOR THE CARD NUMBAS
* LASTLY...  SOMEONE SHOW ME HOW TO USE
  THE DAMN RADIUSES AND SHIT LIKE THAT

   L8R DUDES ....  MATT'/TERA
```

3. Derivatives

3.1. NOTEMAKER V1.1 by ICE CUBE/LIGHT

A few enhancements to my original NOTEMAKER V1.0 to create NOTEMAKER V1.1. It is an update of my V1.0, and includes new music and a new character set.

Figure 5-15. NOTEMAKER V1.1 – write instruction screen

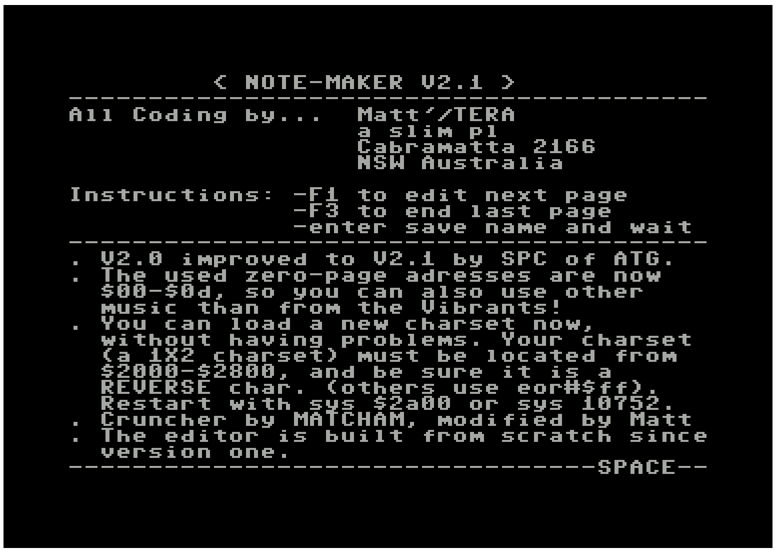


```
NOTE MAKER VERSION 1.1
THIS VERSION IMPROVED BY LIGHT 1989!
ALL CODING BY MATT' TERA!
CONTACT GOBLIN/LIGHT
VIOLVAGEN 29                   MUSIC... DANKO
198 00 BALSTA                  PACKER.. MATCHAM
                               CHARS... GOBLIN
 TO USE:
1. TYPE TEXT ON SCREEN AS NORMAL, NO
   CURSOR UP/DOWN IS ALLOWED, TO GO UP
   HIT 'HOME' & 'RETURN' DOWN TO LINE
2. HIT 'F1' TO ENTER NEXT SCREEN
3. HIT 'F3' TO END FINAL SCREEN
4. ENTER SAVE NAME AND WAIT FOR SPEED -
   PACKER V2 TO OPERATE AND SAVE!!
ALL CODE BY MATT'TERA EXCEPT PACKER AND
MUSIC!! IMPROVED BY ICE CUBE OF LIGHT
NEW MUSIC BY DANKO NEW CHAR BY ME!!!
A MEGA FUCK OFF TO C E N S O R |-SPACE-
```

3.2. NOTEMAKER V2.1 by SPC/ATG

A few enhancements to my original NOTEMAKER V2.0 to create NOTEMAKER V2.1. It is a modification of my V2.0, fixing a music load problem, and allowing the use of a new character set.

Figure 5-16. NOTEMAKER V2.1 – write instruction screen

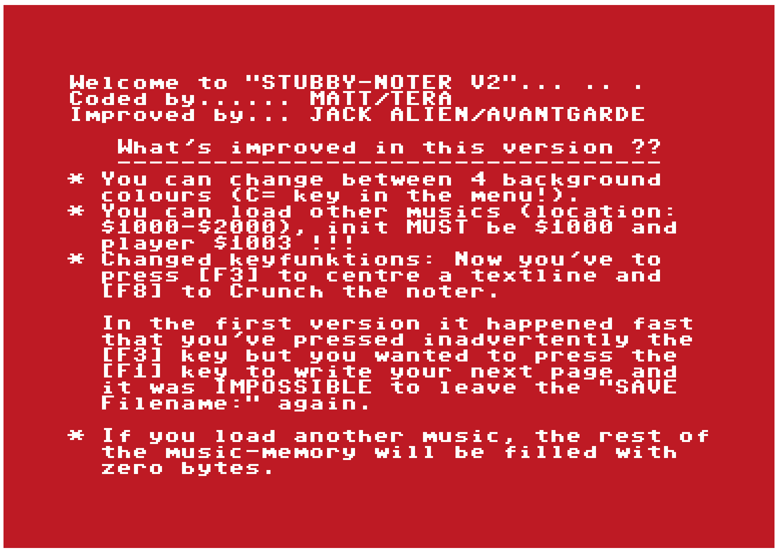


```
          < NOTE-MAKER V2.1 >
----------------------------------------
All Coding by...   Matt'/TERA
                   a slim pl
                   Cabramatta 2166
                   NSW Australia

Instructions: -F1 to edit next page
              -F3 to end last page
              -enter save name and wait
----------------------------------------
. V2.0 improved to V2.1 by SPC of ATG.
. The used zero-page adresses are now
  $00-$0d, so you can also use other
  music than from the Vibrants!
. You can load a new charset now,
  without having problems. Your charset
  (a 1X2 charset) must be located from
  $2000-$2800, and be sure it is a
  REVERSE char. (others use eor#$ff).
  Restart with sys $2a00 or sys 10752.
. Cruncher by MATCHAM, modified by Matt
. The editor is built from scratch since
  version one.
---------------------------------SPACE--
```

3.3. STUBBY-NOTER V2 by JACK ALIEN/AVANTGARDE

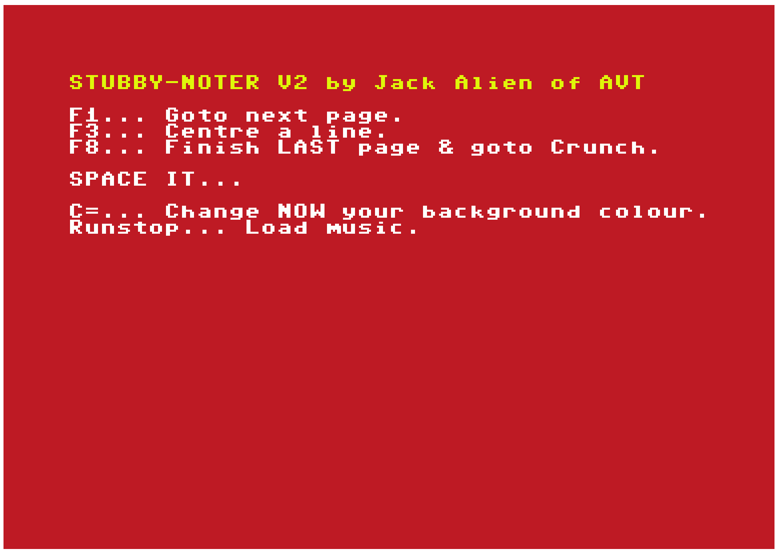
A few enhancements to my original STUBBY-NOTER to create STUBBY-NOTER V2. This modification of my noter adds a small amount of functionality, and it is now possible to change colours and load new music.

Figure 5-17. STUBBY-NOTER V2 – read display screen



```
Welcome to "STUBBY-NOTER V2"... .. .
Coded by...... MATT/TERA
Improved by... JACK ALIEN/AVANTGARDE

   What's improved in this version ??
   ----------------------------------
* You can change between 4 background
  colours (C= key in the menu!).
* You can load other musics (location:
  $1000-$2000), init MUST be $1000 and
  player $1003 !!!
* Changed keyfunktions: Now you've to
  press [F3] to centre a textline and
  [F8] to Crunch the noter.

  In the first version it happened fast
  that you've pressed inadvertently the
  [F3] key but you wanted to press the
  [F1] key to write your next page and
  it was IMPOSSIBLE to leave the "SAVE
  Filename:" again.

* If you load another music, the rest of
  the music-memory will be filled with
  zero bytes.
```

Figure 5-18. STUBBY-NOTER V2 – write options screen

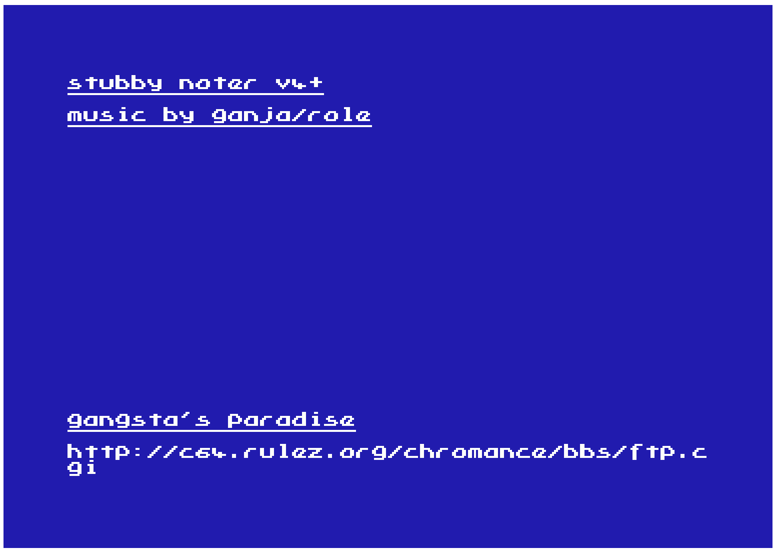


```
STUBBY-NOTER V2 by Jack Alien of AVT

F1... Goto next page.
F3... Centre a line.
F8... Finish LAST page & goto Crunch.

SPACE IT...

C=... Change NOW your background colour.
Runstop... Load music.
```

3.4. STUBBY-NOTER V4+ by SHRI SADHU/WILLOW

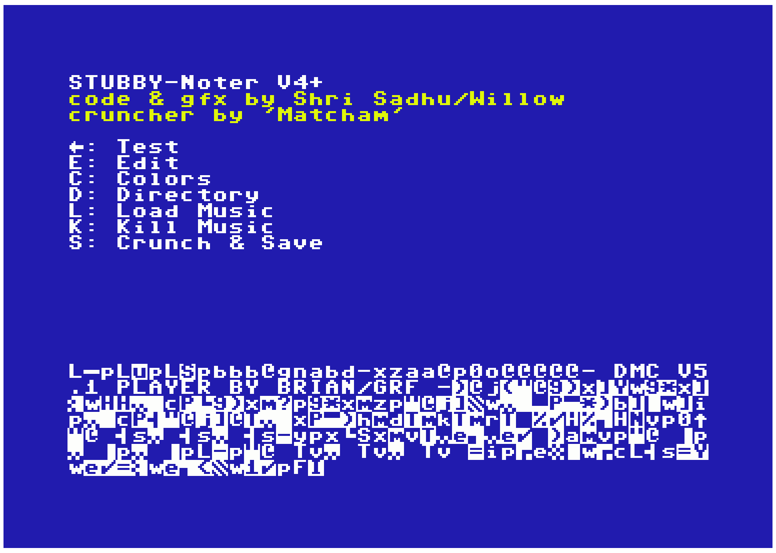
Major enhancements to my original STUBBY-NOTER – to create STUBBY-NOTER V4+ but without any attribution! This derivative of my original STUBBY-NOTER uses a non-system character set. The instruction screen includes many additional options, but the packer and concept are still derivatives of mine.

Figure 5-19. STUBBY-NOTER V4+– read display screen



```
stubby noter v4+
music by ganja/role

gangsta's paradise
http://c64.rulez.org/chromance/bbs/ftp.c
gi
```

Figure 5-20. STUBBY-NOTER V4+ – write instruction screen

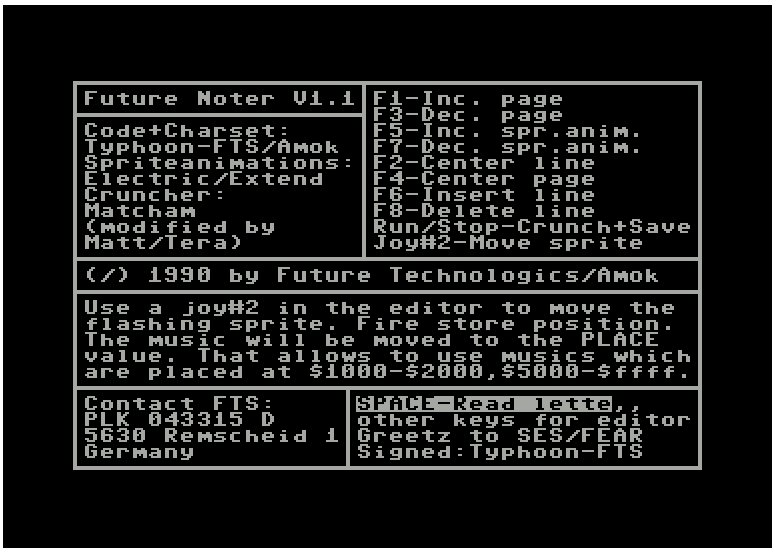


```
STUBBY-Noter V4+
code & gfx by Shri Sadhu/Willow
cruncher by 'Matcham'

←: Test
E: Edit
C: Colors
D: Directory
L: Load Music
K: Kill Music
S: Crunch & Save

L-pL[illegible]pL[illegible]pbbb@gnabd-xzaa@p0o@@@@@- DMC V5
.1 PLAYER BY BRIAN/GRF -[illegible]
[illegible]
```

3.5. FUTURE NOTER V1.1 by FUTURE TECHNOLOGIES/AMOK

My modified version of MATCHAM's cruncher was used in FUTURE NOTER V1.1 here. One could say they even appropriated my entire concept, but such is competition! As with many other notemakers, this one used my modified MATCHAM cruncher.

Figure 5-21. FUTURE NOTER V1.1 – write instruction screen

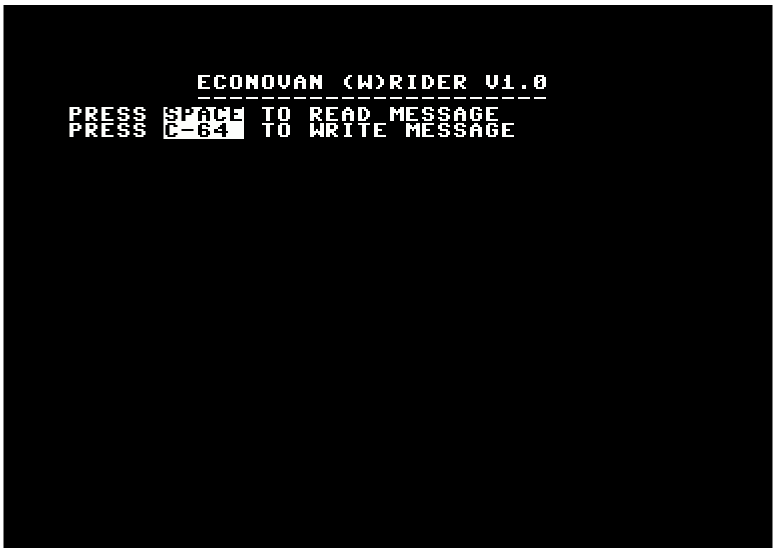


```
Future Noter V1.1      F1-Inc. page
                       F3-Dec. page
Code+Charset:          F5-Inc. spr.anim.
Typhoon-FTS/Amok       F7-Dec. spr.anim.
Spriteanimations:      F2-Center line
Electric/Extend        F4-Center page
Cruncher:              F6-Insert line
Matcham                F8-Delete line
(modified by           Run/Stop-Crunch+Save
Matt/Tera)             Joy#2-Move sprite

(/) 1990 by Future Technologics/Amok

Use a joy#2 in the editor to move the
flashing sprite. Fire store position.
The music will be moved to the PLACE
value. That allows to use musics which
are placed at $1000-$2000,$5000-$ffff.

Contact FTS:        SPACE-Read lette,,
PLK 043315 D        other keys for editor
5630 Remscheid 1    Greetz to SES/FEAR
Germany             Signed:Typhoon-FTS
```

3.6. ECONOVAN (W)RIDER V1.0 by HEX HACKER/TERA

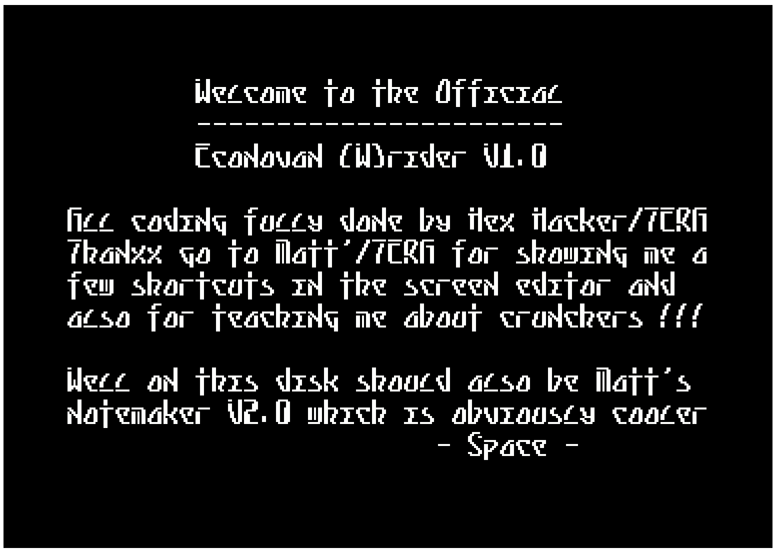
HEX-HACKER's first writer ECONOVAN (W)RIDER V1.0 was modelled closely on my notemakers. The selection menu is standard. Text is rendered in a single colour and fades in and out between pages.

Figure 5-22. ECONOVAN (W)RIDER V1.0 – read/write select



```
          ECONOVAN (W)RIDER V1.0
          ----------------------
PRESS SPACE TO READ MESSAGE
PRESS C-64  TO WRITE MESSAGE
```

Figure 5-23. ECONOVAN (W)RIDER V1.0 – read display screen

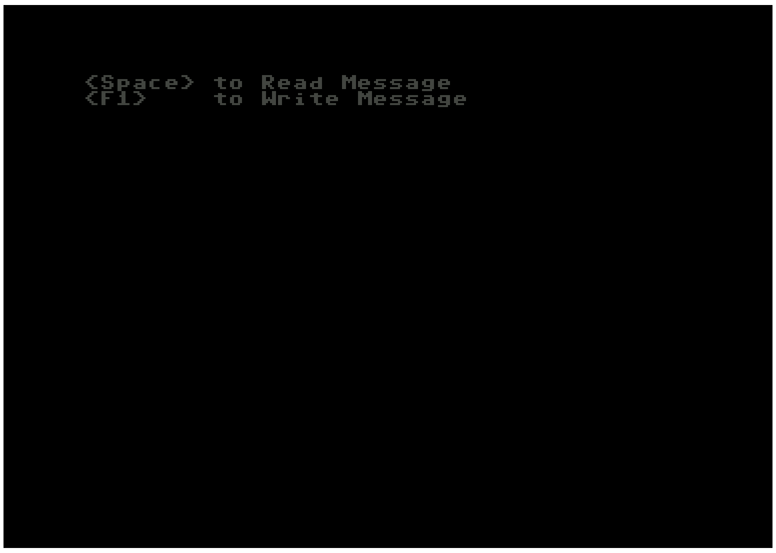


```
         Welcome to the Official
         -----------------------
         Econovan (W)rider V1.0

   All coding fully done by Hex Hacker/TERA
   Thanxx go to Matt'/TERA for showing me a
   few shortcuts in the screen editor and
   also for teaching me about crunchers !!!

   Well on this disk should also be Matt's
   Notemaker V2.0 which is obviously cooler
                             - Space -
```

3.7. HEXCELLENCE V1.0 by HEX-HACKER/TERA

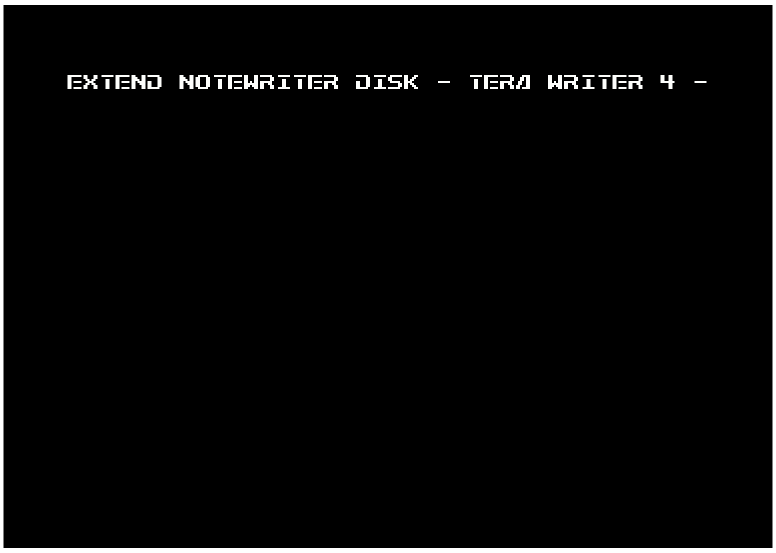
HEX-HACKER's second writer HEXCELLENCE V1.0 was more mature, but otherwise an extension of his first one. The display uses a single height character set. I do not have the original so this was taken from the EXTEND notedisk. The instruction screen is more verbose, and allows insertion of new music routines.

Figure 5-24. HEXCELLENCE V1.0 – read/write select screen



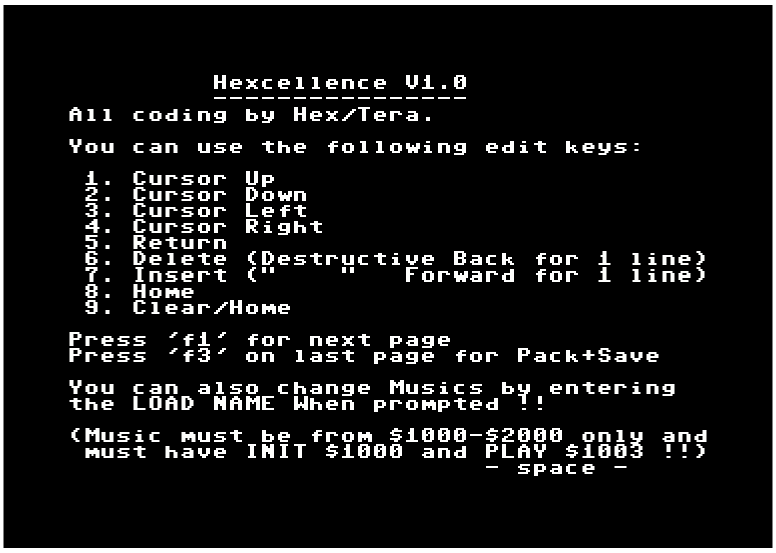
```
<Space> to Read Message
<F1>    to Write Message
```

Figure 5-25. HEXCELLENCE V1.0 – read display screen



```
EXTEND NOTEWRITER DISK - TERA WRITER 4 -
```

Figure 5-26. HEXCELLENCE V1.0 – write instruction screen

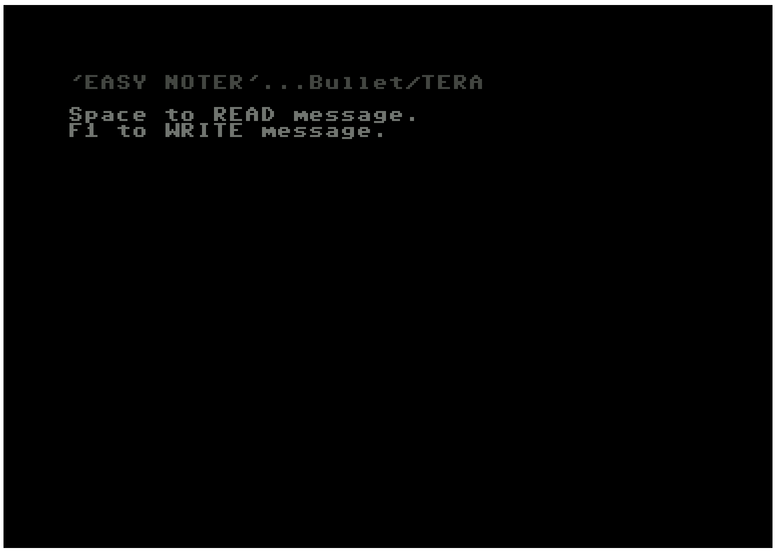


```
         Hexcellence V1.0
         ----------------
All coding by Hex/Tera.

You can use the following edit keys:

 1. Cursor Up
 2. Cursor Down
 3. Cursor Left
 4. Cursor Right
 5. Return
 6. Delete (Destructive Back for 1 line)
 7. Insert ("     "   Forward for 1 line)
 8. Home
 9. Clear/Home

Press 'f1' for next page
Press 'f3' on last page for Pack+Save

You can also change Musics by entering
the LOAD NAME When prompted !!

(Music must be from $1000-$2000 only and
 must have INIT $1000 and PLAY $1003 !!)
                              - space -
```

3.8. EASY NOTER by BULLET/TERA

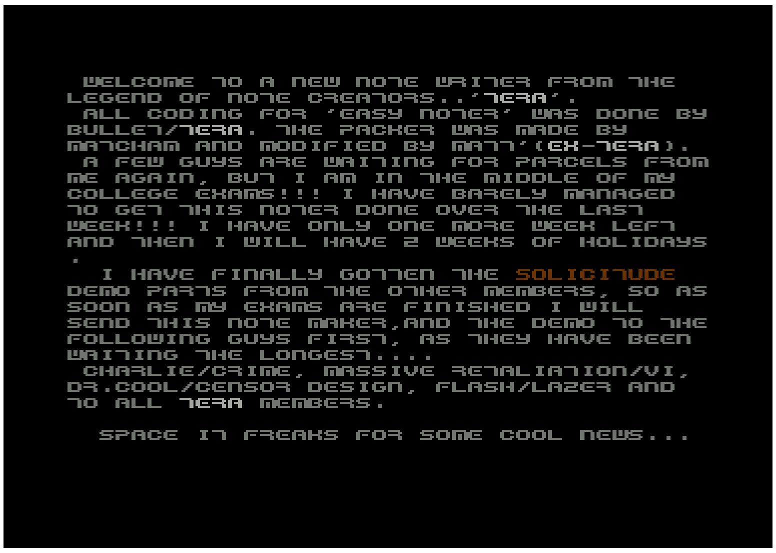
BULLET wrote a nice compact notemaker called EASY NOTER that is similar to the genre.

Figure 5-27. EASY NOTER – read/write select screen



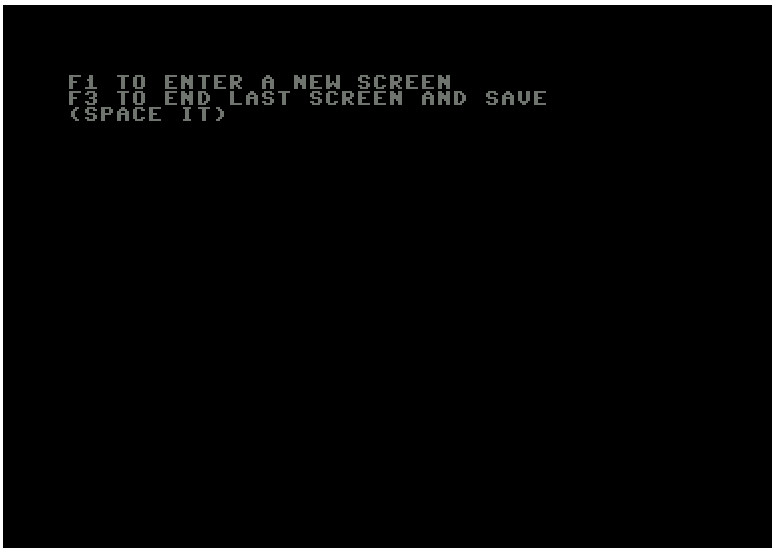
```
'EASY NOTER'...Bullet/TERA

Space to READ message.
F1 to WRITE message.
```

Figure 5-28. EASY NOTER – read display screen



```
 WELCOME TO A NEW NOTE WRITER FROM THE
LEGEND OF NOTE CREATORS..'TERA'.
 ALL CODING FOR 'EASY NOTER' WAS DONE BY
BULLET/TERA. THE PACKER WAS MADE BY
MATCHAM AND MODIFIED BY MATT'(EX-TERA).
 A FEW GUYS ARE WAITING FOR PARCELS FROM
ME AGAIN, BUT I AM IN THE MIDDLE OF MY
COLLEGE EXAMS!!! I HAVE BARELY MANAGED
TO GET THIS NOTER DONE OVER THE LAST
WEEK!!! I HAVE ONLY ONE MORE WEEK LEFT
AND THEN I WILL HAVE 2 WEEKS OF HOLIDAYS
.
   I HAVE FINALLY GOTTEN THE SOLICITUDE
DEMO PARTS FROM THE OTHER MEMBERS, SO AS
SOON AS MY EXAMS ARE FINISHED I WILL
SEND THIS NOTE MAKER,AND THE DEMO TO THE
FOLLOWING GUYS FIRST, AS THEY HAVE BEEN
WAITING THE LONGEST....
 CHARLIE/CRIME, MASSIVE RETALIATION/VI,
DR.COOL/CENSOR DESIGN, FLASH/LAZER AND
TO ALL TERA MEMBERS.

   SPACE IT FREAKS FOR SOME COOL NEWS...
```

Figure 5-29. EASY NOTER – write instruction screen



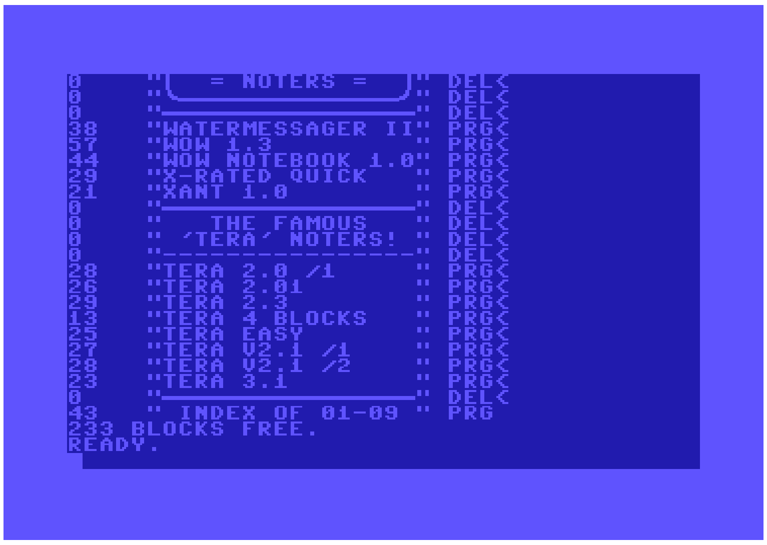
```
F1 TO ENTER A NEW SCREEN
F3 TO END LAST SCREEN AND SAVE
(SPACE IT)
```

4. Collections

There were many notemaker collections produced. Two are shown here.

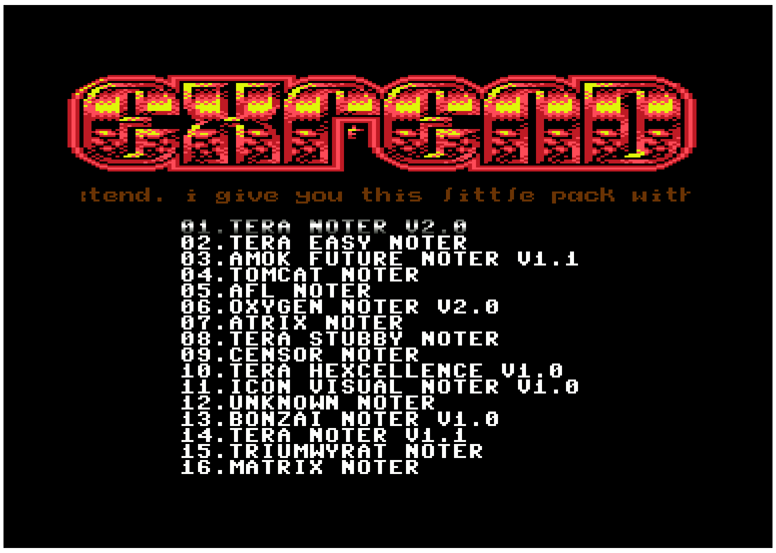
Quite a few of them contain a least one of my notemakers, either the NOTEMAKER V1.0, NOTEMAKER V2.0 or the STUBBY-NOTER. They often also include others that use my routines, especially the MATCHAM packer, and often without attribution. You can also find notemakers by IKON VISUAL and OXYGEN, who were both also Australian groups on the EXTEND disk.

Figure 5-30. HIBISCH noter-collection 09 disk listing



```
0    "   = NOTERS =     " DEL<
0    "                  " DEL<
0    "──────────────────" DEL<
38   "WATERMESSAGER II" PRG<
57   "WOW 1.3           " PRG<
44   "WOW NOTEBOOK 1.0" PRG<
29   "X-RATED QUICK     " PRG<
21   "XANT 1.0          " PRG<
0    "──────────────────" DEL<
0    "   THE FAMOUS     " DEL<
0    " 'TERA' NOTERS!   " DEL<
0    "__________________" DEL<
28   "TERA 2.0 /1       " PRG<
26   "TERA 2.01         " PRG<
29   "TERA 2.3          " PRG<
13   "TERA 4 BLOCKS     " PRG<
25   "TERA EASY         " PRG<
27   "TERA V2.1 /1      " PRG<
28   "TERA V2.1 /2      " PRG<
23   "TERA 3.1          " PRG<
0    "──────────────────" DEL<
43   " INDEX OF 01-09  " PRG
233 BLOCKS FREE.
READY.
```

Figure 5-31. EXTEND notedisk selection screen



```
:tend. i give you this little pack with
01.TERA NOTER V2.0
02.TERA EASY NOTER
03.AMOK FUTURE NOTER V1.1
04.TOMCAT NOTER
05.AFL NOTER
06.OXYGEN NOTER V2.0
07.ATRIX NOTER
08.TERA STUBBY NOTER
09.CENSOR NOTER
10.TERA HEXCELLENCE V1.0
11.ICON VISUAL NOTER V1.0
12.UNKNOWN NOTER
13.BONZAI NOTER V1.0
14.TERA NOTER V1.1
15.TRIUMWYRAT NOTER
16.MATRIX NOTER
```

Section 6. C64 Activities Intros

1. Background

1.1. Nature

Usually, intros were used as screens to be placed in front of cracked software. A game, or other item of software, would be broken from an original distribution (e.g. the copy protection may be removed, a disk image may be converted into a series of files, etc), and an intro would be appended. The intro provided details about the group, the software, and – more importantly – a list of greetings sent out to other groups, an important part of maintaining good relationships and "status" in the scene.

1.2. The group and the intro

I think I cracked one or two originals in my time, and even they were pathetic. We were not a cracking group (even if there was some pretension in that direction), and Australia did not have many software houses that produced anything worth cracking. However, intros were sometimes used for distribution purposes: we may have imported a hot release from a board in the states, so we would put on our distribution intro and spread the result to our contacts, and it would filter down through the chain. Every game player would see our intro, and we would develop a good reputation. In reality, we hardly even did this. Mostly, our intros were put in front of tools and other productions, such as graphics logos.

1.3. The coder and the intro

Personally, most of my intros were good excuses to experiment with new routines or use new graphics, or to create something new. I produced quite a few Intros for TOUR DE FUTURE, TOUR DE FUTURE+REFLEX and then TERA. I still remember my first intro for TDF, and it was shockingly immature compared to later TERA Intros. I first used ripped logos, character sets and code, but eventually matured to the point where I had created all code and graphics myself. I could not produce music, so it always came from elsewhere, as was generally normal in the scene. Usually I ripped music out of games rather than other intros or demos.

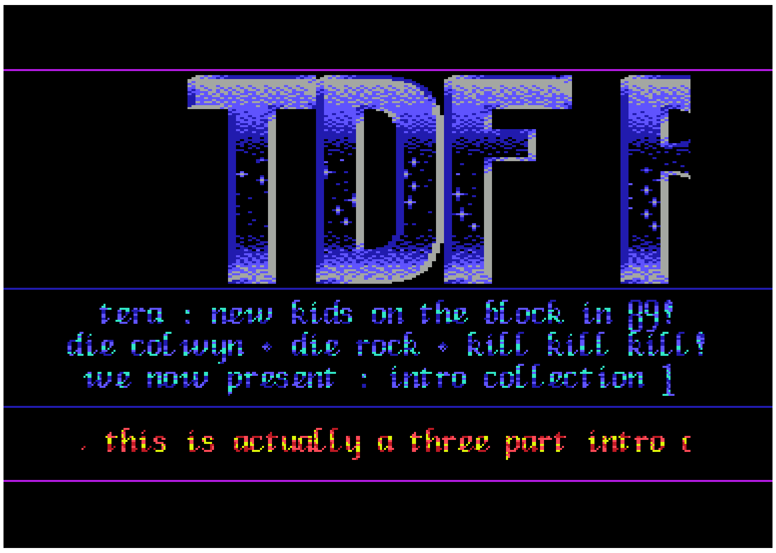
Some intros may not look too exciting to a user, but may contain novel/interesting code underneath (these are somewhat academic then). It was often important to keep an intro small – because collectors and users of games often kept the smallest version of any game that they received, unless larger versions had trainers or other features. These people were usually most concerned about playing the games, so wanted to skip over the intro screens as quickly as possible – short and sweet intros were a good goal.

2. Productions

2.1. TDF+RFX intro by MATT'/TDF+RFX

This is a sequenced intro. It starts with a blank screen and builds up in synchronisation with the music. The large logo fades into view and swings from side to side, with the banner text appearing, and the scroll text also comes into view. I used sprites as backdrops to a double height character set, and rolled the sprites to provide a ripple effect. When the space key is pressed, the Intro clears itself and proceeds onwards. I was usually careful about the small details: keeping a screen black and not resetting to system colours, for example. I created the logo, sprites and the code, but not the character set or the music. I had routines to create the sinus tables, and to merge bitplanes (so, for example, I drew the logo and its fill separately, then used my own utility routines to merge them together).

Figure 6-1. "TDF RFX" Intro with text ripple effect



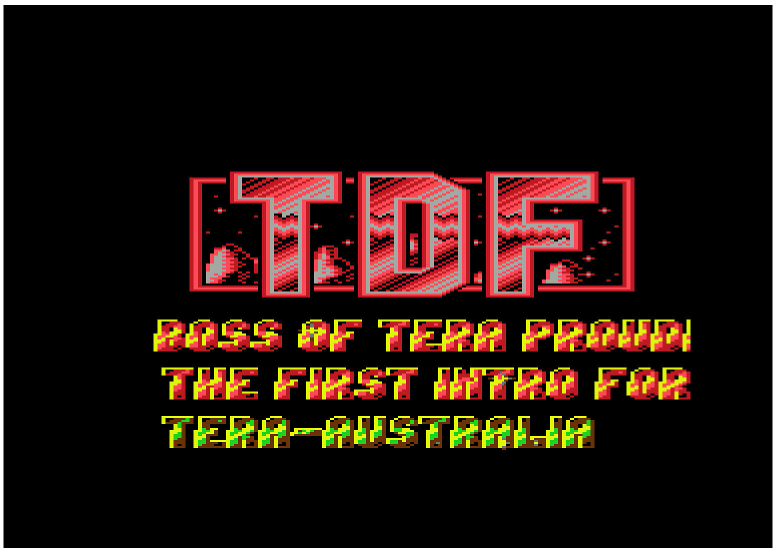
tera : new kids on the block in 89!
die colwyn + die rock + kill kill kill!
we now present : intro collection 1

. this is actually a three part intro c

2.2. TDF+RFX Intro by MATT'/TDF+RFX

This is a very quick Intro that used an alternating swinging logo, to show TDF or REFLEX. The two centre lines of text swing back and forth, while the scroll text moves across from the right, but has variable speeds and a stop capability. The music is short and sweet. I apparently created all code and graphics, including the character set. Honestly, I do not remember this Intro well, if at all. It was probably something that I created in a night, because I was able to quickly "knock out" easy Intros like this.

Figure 6-2. “TDF REFLEX” Intro with swinging logos

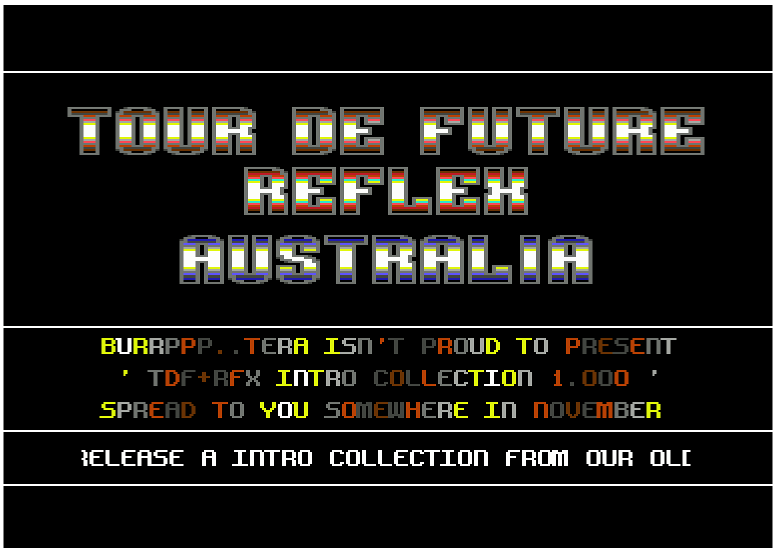


BOSS OF TERA PROUDL
THE FIRST INTRO FOR
TERA-AUSTRALIA

2.3. TDF+RFX Intro by MATT’/TDF+RFX

This Intro was another production for the merged group before we developed our new name of TERA. It is similar in style to my other intros, and includes a large group logo, and three lines of banner text with scrolling text at the bottom. Rippling colours provide a copper shimmering effect that cannot be seen well in the image capture. This was a very easy intro to produce, and I created the large character set, and possibly even the smaller character set.

Figure 6-3. “TDF+RFX Australia” Intro with shimmering text



TOUR DE FUTURE
REFLEX
AUSTRALIA

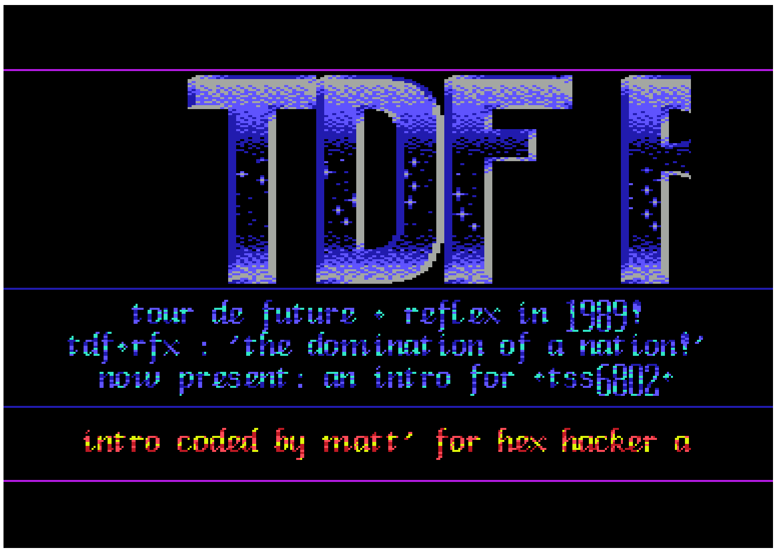
BURRPPP..TERA ISN'T PROUD TO PRESENT
' TDF+RFX INTRO COLLECTION 1.000 '
SPREAD TO YOU SOMEWHERE IN NOVEMBER

RELEASE A INTRO COLLECTION FROM OUR OLI

2.4. TSS6802 Intro by MATT'/TDF+RFX

As we moved from TDF into TDF+RFX and eventually TERA, we gained new group members, in particular, HEX-HACKER and SNOOP from a group called 'THE SOFT-SMASHERS 6802'. I created an intro for them that was proceeded by one of my own intros.

Figure 6-4. "TDF RFX" Intro with text ripple effect

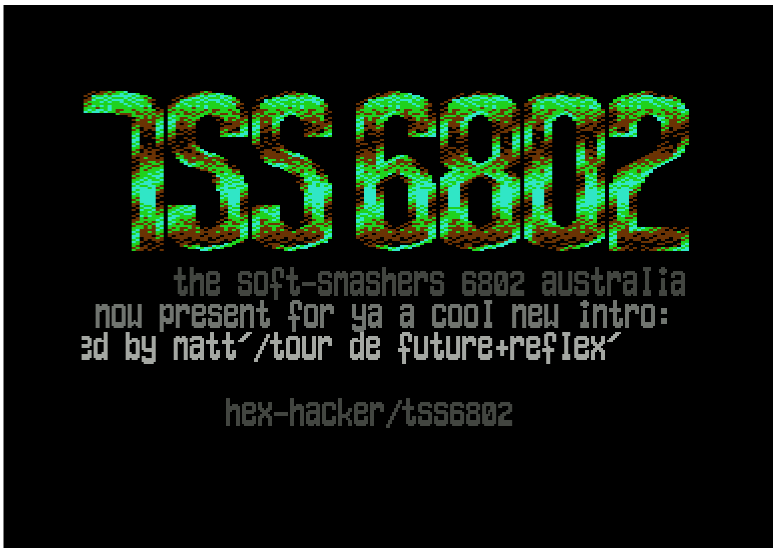


tour de future + reflex in 1989!
tdf+rfx : 'the domination of a nation!'
now present : an intro for +tss6802+

intro coded by matt' for hex hacker a

The TSS6802 Intro included a continuously fading in/out and colour changing logo for TSS6802, and three lines of text that swung back and forth. The scroll text swings up and down as it rolls in from the right, and a shimmer effect runs across it. The text speed can be changed (8 speeds), and halted, providing good control. The screen fades in and out to black very nicely at the start and after the user presses the space key. I created the graphics and the code, but the music and character set are from elsewhere. I would have used my merging routines to create the logo.

Figure 6-5. "TSS6802" Intro with colour changing logo

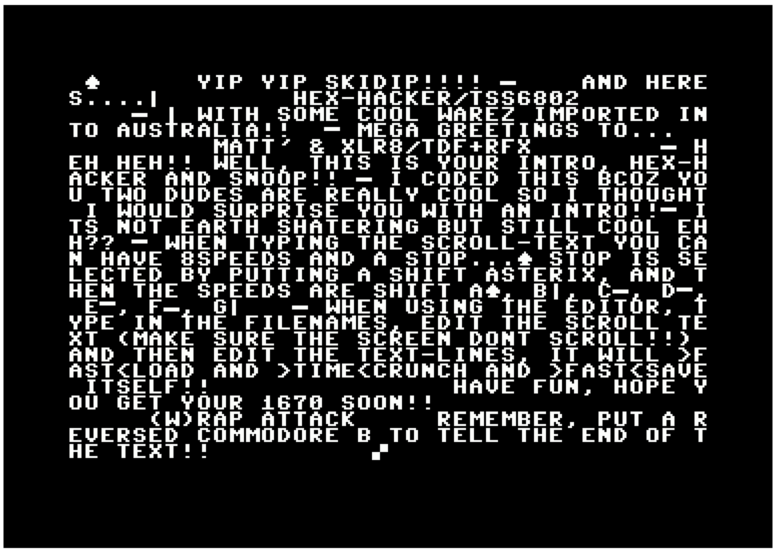


the soft-smashers 6802 australia
now present for ya a cool new intro:
ed by matt'/tour de future+reflex'

hex-hacker/tss6802

More importantly, this intro has an editor – allowing the user to edit the banner text, the scroll text, and then a file will be fast loaded, appended, packed and fast saved. I did this not so much because I didn't think that HEX-HACKER and SNOOP could figure out how to do it themselves, but I wanted to create a nicely presented and integrated package.

Figure 6-6. "TSS6802" Intro text scroller entry screen



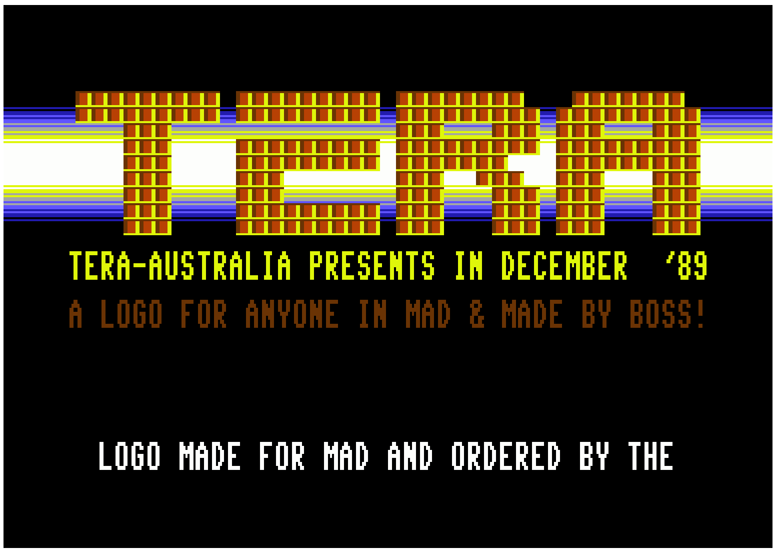
```
 ♠      YIP YIP SKIDIP!!!! —     AND HERE
S....|            HEX-HACKER/TSS6802
   — | WITH SOME COOL WAREZ IMPORTED IN
TO AUSTRALIA!!  — MEGA GREETINGS TO...
         MATT' & XLR8/TDF+RFX          — H
EH HEH!! WELL, THIS IS YOUR INTRO, HEX-H
ACKER AND SNOOP!! — I CODED THIS BCOZ YO
U TWO DUDES ARE REALLY COOL SO I THOUGHT
 I WOULD SURPRISE YOU WITH AN INTRO!!— I
TS NOT EARTH SHATERING BUT STILL COOL EH
H?? — WHEN TYPING THE SCROLL-TEXT YOU CA
N HAVE 8SPEEDS AND A STOP...♠ STOP IS SE
LECTED BY PUTTING A SHIFT ASTERIX, AND T
HEN THE SPEEDS ARE SHIFT A♠, B|, C—, D—,
 E—, F—, G|   — WHEN USING THE EDITOR, T
YPE IN THE FILENAMES, EDIT THE SCROLL TE
XT (MAKE SURE THE SCREEN DONT SCROLL!!)
AND THEN EDIT THE TEXT-LINES, IT WILL >F
AST<LOAD AND >TIME<CRUNCH AND >FAST<SAVE
 ITSELF!!                    HAVE FUN, HOPE Y
OU GET YOUR 1670 SOON!!
     (W)RAP ATTACK      REMEMBER, PUT A R
EVERSED COMMODORE B TO TELL THE END OF T
HE TEXT!!           ▞
```

2.5. TERA Intro by MATT'/TERA

TRIAD were one of the aristocratic groups in the scene, and for a while, I was a good pen-friend of JERRY of TRIAD. TRIAD coders such as MR Z of TRIAD impressed me, and I was very impressed by one of their short and sweet intros, so I decided to create my own super-compact intro (this was an inspired production).

This Intro only takes up $0800 to $0c20 (not including two lines of banner text, and scroll text) for all code and graphics. The Intro generates the double height character set from the system character set, and renders the TERA logo from a compressed bitvector, and then runs an interrupt routine to maintain the colour bars, cycle the banner text colours, and swing the scroll text while rolling it from right to left. TRIAD used silver squares rather than gold squares. There is no music.

Figure 6-7. "TERA" Intro inspired by TRIAD


TERA
TERA-AUSTRALIA PRESENTS IN DECEMBER '89
A LOGO FOR ANYONE IN MAD & MADE BY BOSS!
LOGO MADE FOR MAD AND ORDERED BY THE

2.6. Others

There were other Intros that were placed on specialised productions, sometimes the Intro was created for one release only. The examples of this are BEEFTRUCKER and the 4 BLOCK NOTER!

Section 7. C64 Activities Demos BACK FROM BEYOND

1. Background

Across 1988, I was developing my skills as a coder, and broadening my contacts around the country. In late 1988, a demo competition was arranged, and I decided to produce an entry. I came up with something of a space theme, arranged in multiple parts. At this stage of development as a coder, I was still obsessed with colour bars and other VIC effects, such as using multiple sprites by reprogramming the VIC as it progressed down the screen. My design skills were still undeveloped, as the TDF logo illustrates! I was becoming increasingly skilled at removing fonts, sprites, music and code from commercial software, and apart from the code, just about everything else in here was taken from other programs. I was also learning about multiple part demos, packing, relocating, interrupt routines, colour schemes, sinus effects, graphics modes, scrolling routines, and program sequencing.

As far as I remember, this was judged to be the winning entry, and I was beginning to develop a significant reputation as a coder. But it is clear that my style was lacking coherency, and needed to mature.

2. Packaging

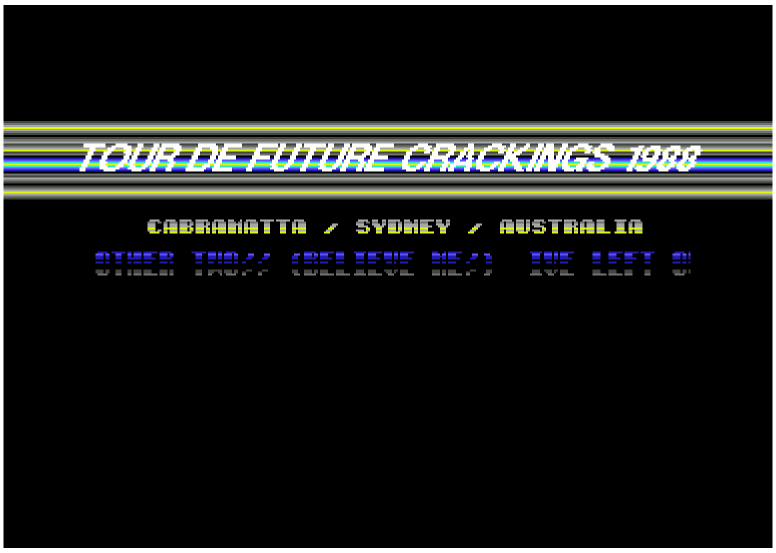
The release was packaged as a standard single-file multi-part demo. Each part would execute, and then unpack and execute the next part. The single-file was distributed with the name “BACK FROM BEYOND”.

3. Parts

3.1. Intro screen

The action starts with a fairly simple intro screen – although, I was learning at the time, so this was my personal “state of the art”. The colour bar routines were one of the first things to master, as they required finely tuned timing. My innovation was to make the blue bar move in front of, as well as behind, the logo. The colour bars moved around the scroll text. The style is very metallic and comes across as somewhat depersonalised and superficial. I created the logo and the code, but not the character set, nor the music.

Figure 7-1. BACK FROM BEYOND Intro screen, with early 1988 info


TOUR DE FUTURE CRACKINGS 1988

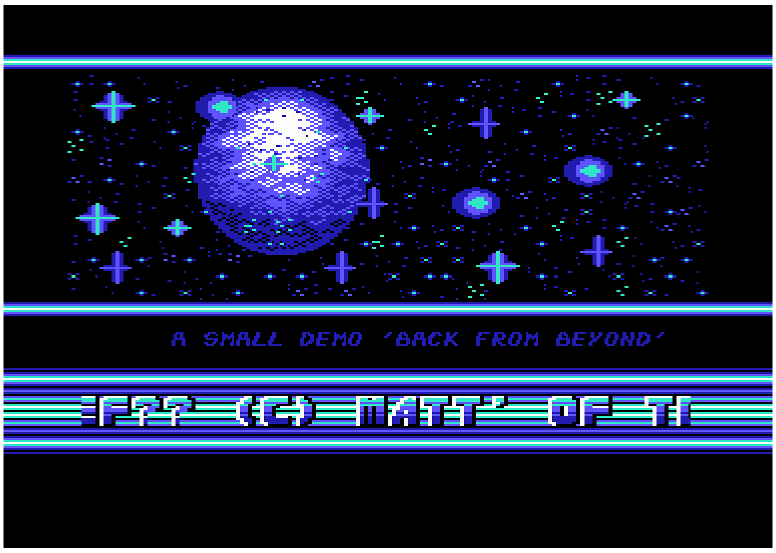
CABRAMATTA / SYDNEY / AUSTRALIA

OTHER TWO// (BELIEVE ME/) IVE LEFT O

3.2. First part, with rotating orb

The first part starts a space theme. The top section contains two bounding colour bars, and within that, a moving sphere, looking like a planet, with stars in front. The sphere is made out of more sprites than could be shown at once, as I redefine the sprite coordinates and locations along the raster scan lines. The small banner text waves back and forth, and the scroll text sits in front of colour bars, and has its own colour cycle – it has a metallic feel to it. The scroll text is worth reading, as it describes the reason for the demo (the competition) and a comment about the origins of our name (TOUR DE FUTURE). I created the code and the star field graphics, but not the planet, nor the fonts, nor the music. The sinusoid tables were created by my own routines.

Figure 7-2. BACK FROM BEYOND First part, with rotating orb - space theme

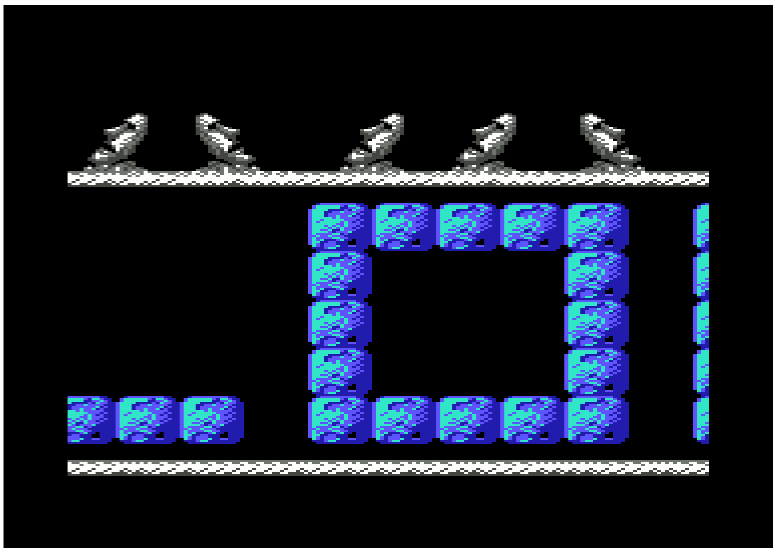

A SMALL DEMO 'BACK FROM BEYOND'

[illegible] (C) MATT' OF TI

3.3. Second part, with large ancient text

The second part is designed to have a sort of ancient feel about it, with the Easter island statues, and the text starts with “a long long time ago, there lived a race …”. I chose the music to give a sort of mysterious quality. The interesting part about this part is that it takes text and turns it into a very large scroll text with bits represented by “stone” looking blocks. The part nicely fades in and out. I created the code and the meta-character-set, but not the graphics, nor the music.

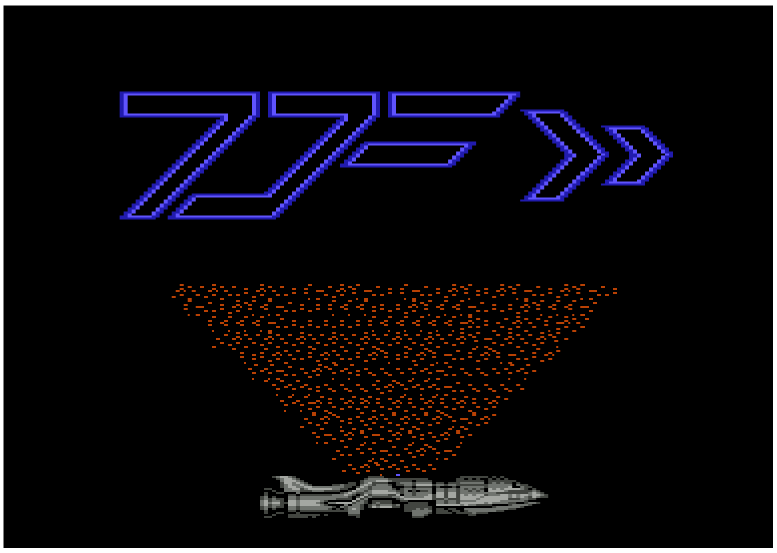
Figure 7-3. BACK FROM BEYOND Second part, large text - ancient theme



3.4. Third part, with unfolding narrative

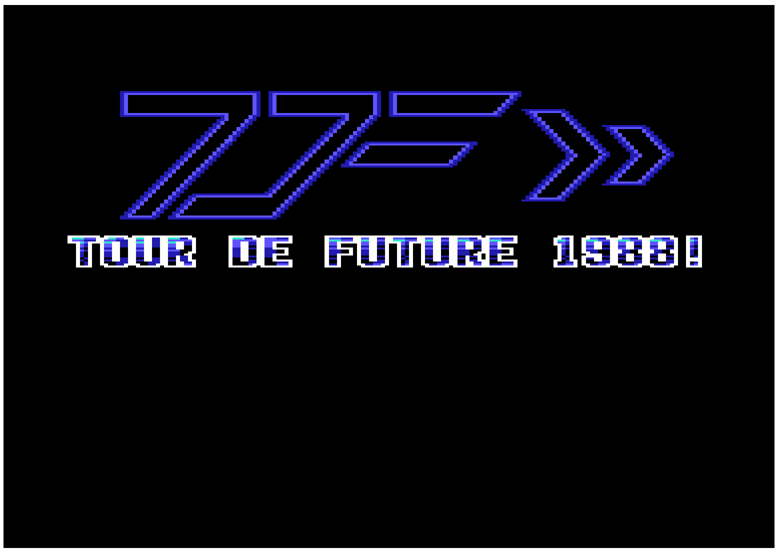
The third part is designed to “unfold”, like a narrative, and starts with a spacecraft that moves in from the left and uses a “beam” to create the TDF logo. The “beam” retracts, and the spacecraft moves off to the right hand side. I stole the space ship from somewhere, but created the code, the logo and the “particle beam”.

Figure 7-4. BACK FROM BEYOND Third part with ship provided logo



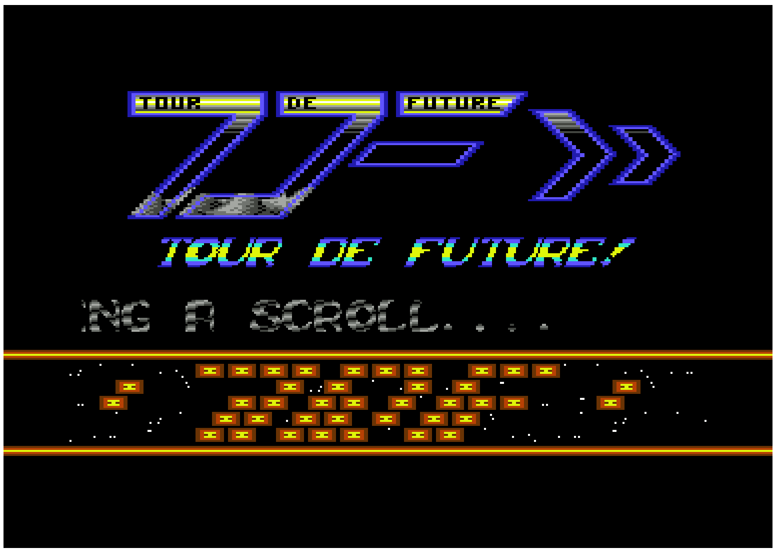
The narrative continues with a short display of text that announces the group name, the name of the demo, the author, and that it was created for the “aussie demo comp” in November 1998. I am not sure where the character set came from.

Figure 7-5. BACK FROM BEYOND Third part, with introductory text



The third part ends with a fairly simple screen with a rotating star field at the bottom, and a lunar landscape within the logo. The scroll routine handles multiple speeds and directions. I created the logos, and the colour bars and sprites, including the star field. I did not create the character sets, nor the music.

Figure 7-6. BACK FROM BEYOND Third part, with space background



Section 8. C64 Activities Demos HUH

1. Background

I do not have any personal memory of this demo, but can only provide details based upon what I read in the scroll-texts. The HUH demo was created early in 1989 (no earlier than March), which is after the BACK FROM BEYOND demo (our first real demo that set us on the scene in Australia), and before our OHH MATE!! demo (which opened us up to a much wider audience). HUH was created at a time when we were growing our base in Sydney – moving from an ad hoc assembly of school friends, into contact with other guys such as SNOOP, HEX HACKER and MISTIC. We were becoming a real group (in European terms). We did have Australian and International contacts at this point though, but did not have a lot of releases to our name.

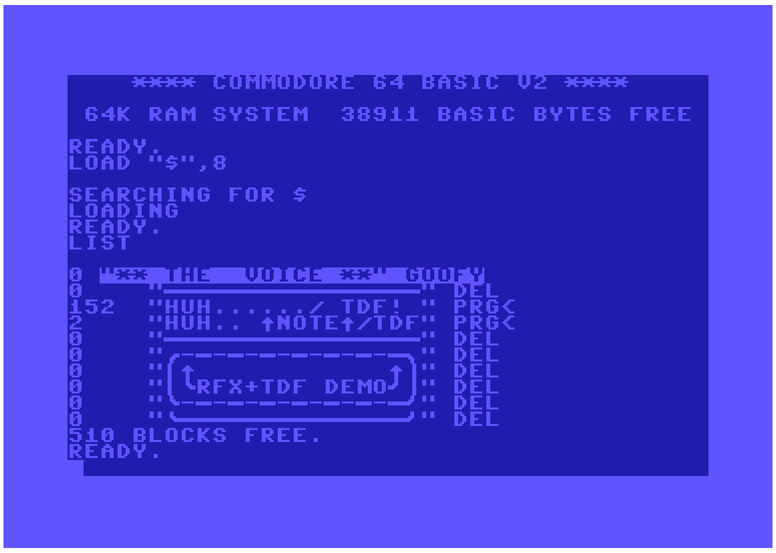
The demo is important as it documents a copy-party we organised at the Cabramatta civic centre (the demo may have been released directly after the party). The other people at this copy-party were from a couple of local BBS's (that I mentioned were cool – but they were not really cool in retrospect). I am sure that I had been frequenting bulletin boards, but perhaps only for a short time.

The copy-party was our first physical meeting with SNOOP and HEX-HACKER, and from here they became good friends. I am not sure of how much we had to do with MISTIC at this time, because the demo is largely about XLR8, PARASITE and myself. RECKLESS ROB and BIZ had long dropped out of the picture. The "group" was largely just XLR8 and I, with me doing a lot of coding and swapping. PARASITE and XLR8 were good friends, although PARASITE was a gamer with no scene interest at all – he didn't have much involvement in the group after this time.

2. Packaging

The demo was released as a single file, with an attached note. This version has had "TDF + RFX" added, when the demo had been released months before the co-operation.

Figure 8-1. HUH disk packaging



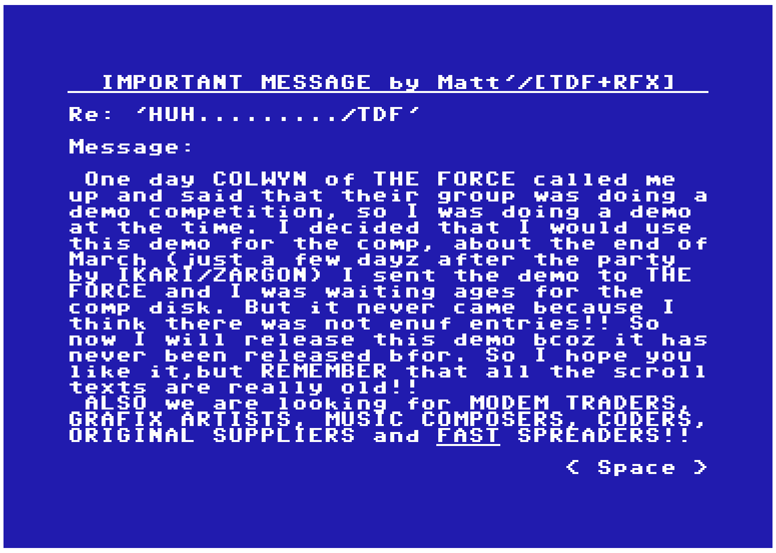
```
    **** COMMODORE 64 BASIC V2 ****

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
LOAD "$",8

SEARCHING FOR $
LOADING
READY.
LIST

0 "*** THE  VOICE ***" GOOFY
0     "────────────────" DEL
152   "HUH....../ TDF! " PRG<
2     "HUH.. ↑NOTE↑/TDF" PRG<
0     "────────────────" DEL
0     "                " DEL
0     " ↑RFX+TDF DEMO↑ " DEL
0     "                " DEL
0     "                " DEL
510 BLOCKS FREE.
READY.
```

3. Parts

3.1. Text announcement, an important message

The first screen indicates that the demo was originally designed to be part of a competition arranged by COLWYN of THE FORCE. I'm not sure if there ever was such a competition, or it was just a ruse – because on other occasions he was known to bend the truth.

Figure 8-2. HUH Text announcement, an important message

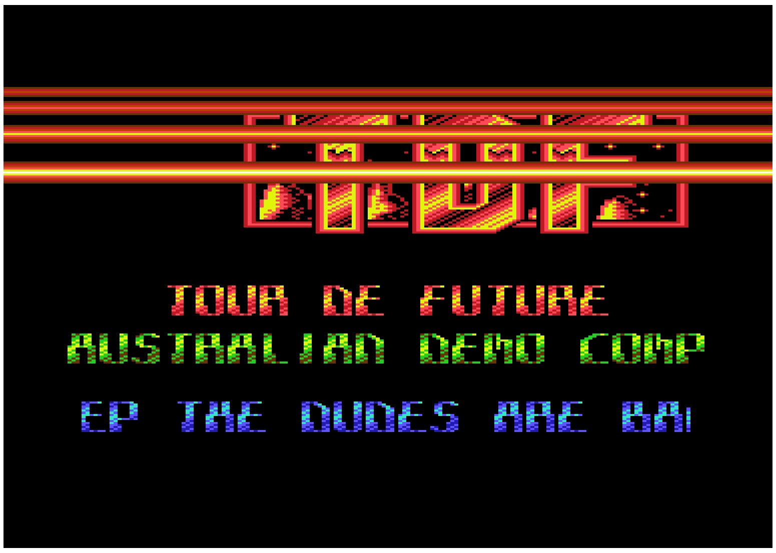


```
  IMPORTANT MESSAGE by Matt'/[TDF+RFX]
Re: 'HUH........./TDF'

Message:

 One day COLWYN of THE FORCE called me
up and said that their group was doing a
demo competition, so I was doing a demo
at the time. I decided that I would use
this demo for the comp, about the end of
March (just a few dayz after the party
by IKARI/ZARGON) I sent the demo to THE
FORCE and I was waiting ages for the
comp disk. But it never came because I
think there was not enuf entries!! So
now I will release this demo bcoz it has
never been released bfor. So I hope you
like it,but REMEMBER that all the scroll
texts are really old!!
 ALSO we are looking for MODEM TRADERS,
GRAFIX ARTISTS, MUSIC COMPOSERS, CODERS,
ORIGINAL SUPPLIERS and FAST SPREADERS!!

                              < Space >
```

3.2. Intro screen, for the Australian Demo Comp

The intro is very basic, but illustrates stylistic advances since BACK FROM BEYOND. The coding routines are not very innovative, and were fairly standard for the time. There is only some minor innovation in having colour bars circle around the logo. These routines were at the cutting edge of my personal envelope.

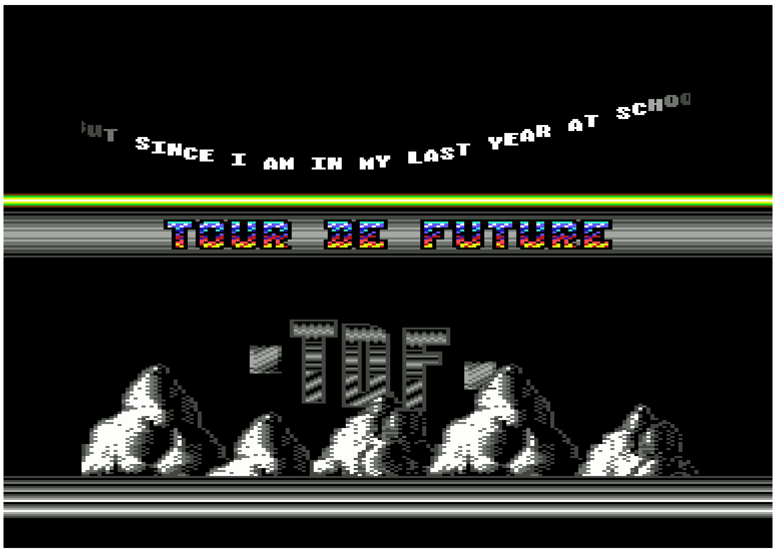
Figure 8-3. HUH Intro screen, for the Australian Demo Comp



3.3. First part, text DYCP'er

This part has a non-smooth DYCP'er, and shows a continued fascination with colour bars and raster routines. The graphics are largely (entirely?) ripped. This part has the importance scroll text dating our meeting with SNOOP and HEX-HACKER at 25 March 1989.

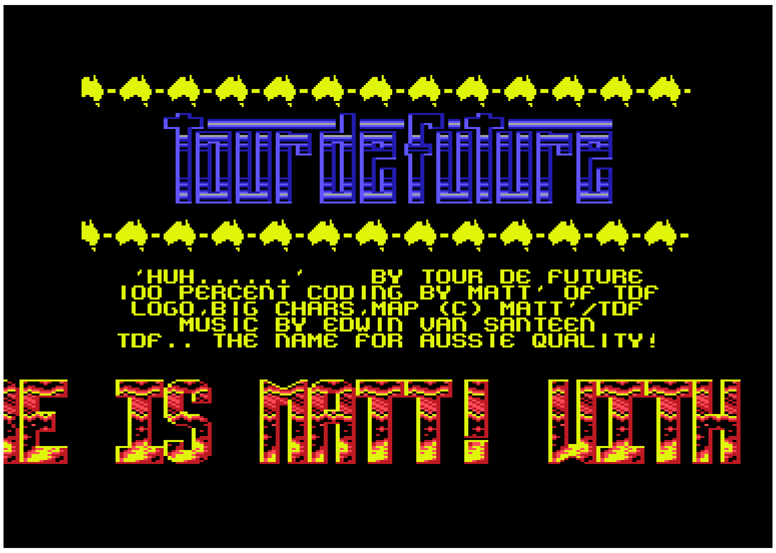
Figure 8-4. HUH First part, text DYCP'er



3.4. Second part, open border scroll and patriotic symbols

This part is also not highly advanced, but shows further code mastery with the use of open border routines for the large scrolling text. The big character set illustrates the growing skill at creating graphics.

Figure 8-5. HUH Second part, open border scroll and patriotic symbols



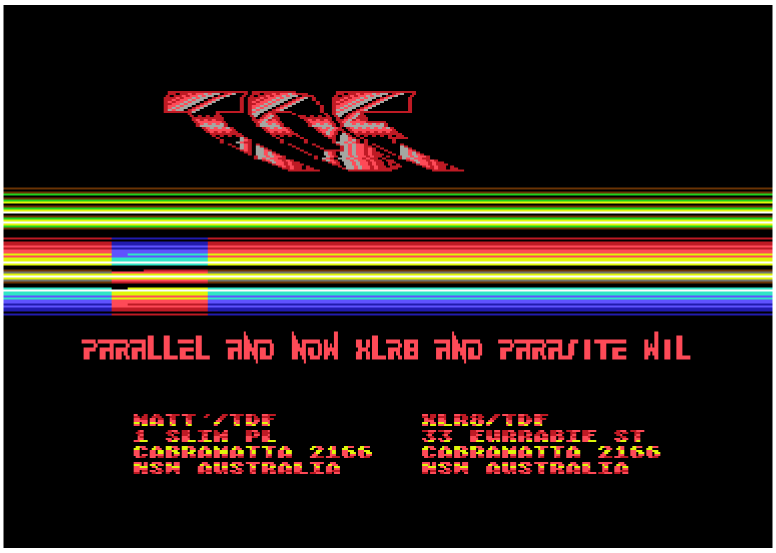
```
  'HUH......'     BY TOUR DE FUTURE
 100 PERCENT CODING BY MATT' OF TDF
  LOGO,BIG CHARS,MAP (C) MATT'/TDF
     MUSIC BY EDWIN VAN SANTEEN
 TDF.. THE NAME FOR AUSSIE QUALITY!
```

E IS MATT! WITH

3.5. Third part, wibbling logo and contact screen

The final part is perhaps the part with the most technical complexity in the demo, notably the fine scrolling of the logo, which requires a semi-real-time rewrite of the logo – consuming some reasonable amount of computing time.

Figure 8-6. HUH Third part, wibbling logo and contact screen



PARALLEL AND NOW XLR8 AND PARASITE WIL

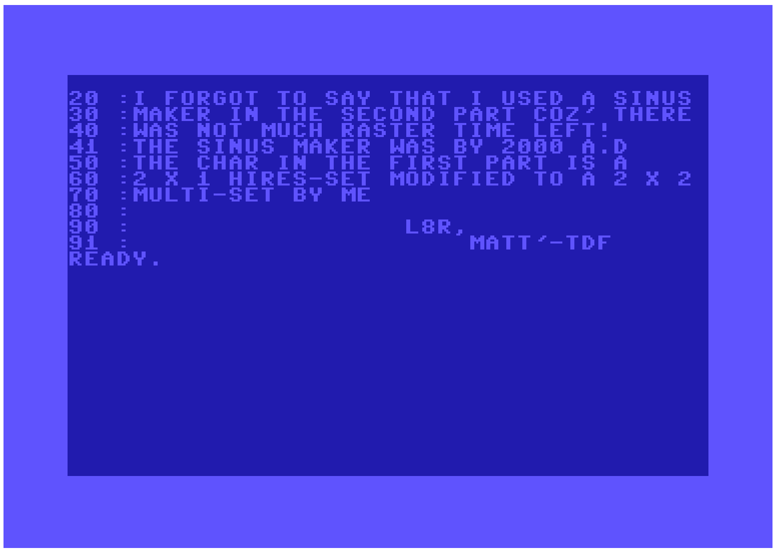
```
MATT'/TDF            XLR8/TDF
1 SLIM PL            33 EURRABIE ST
CABRAMATTA 2166      CABRAMATTA 2166
NSW AUSTRALIA        NSW AUSTRALIA
```

4. Attachments

4.1. Release note, a few outstanding credits

A small note was provided with the release. I indicated the source of sinus tables, but I was feeling a little guilty that at this time, a lot of my code was still highly reverse engineered, so while not ripped, derived from what I had learnt and understood of others.

Figure 8-7. Release note, a few outstanding credits



```
20 :I FORGOT TO SAY THAT I USED A SINUS
30 :MAKER IN THE SECOND PART COZ´ THERE
40 :WAS NOT MUCH RASTER TIME LEFT!
41 :THE SINUS MAKER WAS BY 2000 A.D
50 :THE CHAR IN THE FIRST PART IS A
60 :2 X 1 HIRES-SET MODIFIED TO A 2 X 2
70 :MULTI-SET BY ME
80 :
90 :                 L8R,
91 :                     MATT´-TDF
READY.
```

Section 9. C64 Activities Demos OHH MATE!!

1. Background

This demo is probably the most mature of the TOUR DE FUTURE group, at a time when we had grown into "a real group", had merged with REFLEX (most parts are TDF+RFX, but there are one or two exceptions). The demo was prepared for the TEC-Illegal copy-party, held in Sydney in the first week of July in 1989. The demo was created for the party, and pays homage to bulletin boards in a thematic manner. It is a multi-part demo, with neat loading activities.

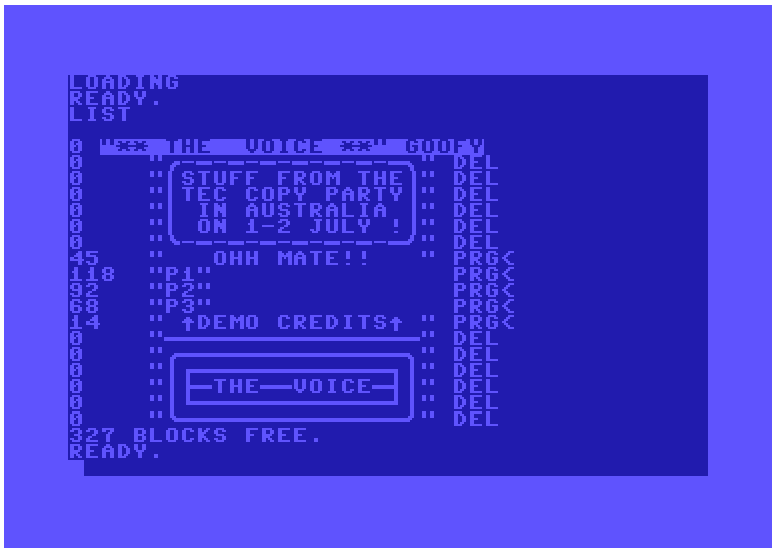
The party was a major event for us – we had released BACK FROM BEYOND and HUH as demos from TOUR DE FUTURE, along with a number of intros and tools. We had recently merged with REFLEX, and become friendly with a number of contacts in Sydney. The party was our first major scene event, and the demo was well received in disk mags (e.g. SEX'N'CRIME).

The coding illustrates some maturity in overall thematic style and construction, but the parts are still disjoint. There are novel routines, but they are novel for myself as a coder, and not really for the scene. It does illustrate a high level of skill, and definitely a leading production for Australia at the time. What it doesn't have is a lot of home grown graphics – as most graphics and music are ripped. The code is entirely home-grown, but still highly derived from reverse-engineering activities. It's not until the next major demo, DAMNABLE, under the TERA name, that I become self-sufficient as a coder and graphics artist.

2. Packaging

The demo was provided in multiple-parts, with custom fast loading routines to bring in each part. A release note provides information about credits.

Figure 9-1. OHH MATE!! disk packaging



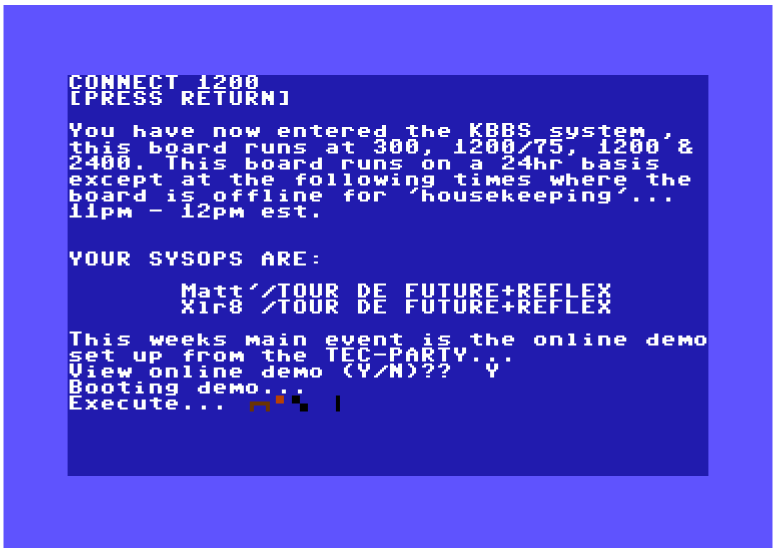
```
LOADING
READY.
LIST

0  "** THE  VOICE **"  GOOFY
0       "              "  DEL
0       " STUFF FROM THE " DEL
0       " TEC COPY PARTY " DEL
0       "  IN AUSTRALIA  " DEL
0       "  ON 1-2 JULY ! " DEL
0       "                " DEL
45      "    OHH MATE!!  " PRG<
118     "P1"                PRG<
92      "P2"                PRG<
68      "P3"                PRG<
14      "  ↑DEMO CREDITS↑ " PRG<
0       "                " DEL
0       "                " DEL
0       "                " DEL
0       "  THE——VOICE    " DEL
0       "                " DEL
0       "                " DEL
327 BLOCKS FREE.
READY.
```

3. Parts

3.1. Screen one, ‘Modem Mania’

The screen emulates the idea of connecting to a bulletin board system, but rather than a plain text environment, envisages that the bulletin board drops out to execute an online demo. The text display emulates time delays, and garbage characters.

Figure 9-2. OHH MATE!! Screen one, ‘Modem Mania’

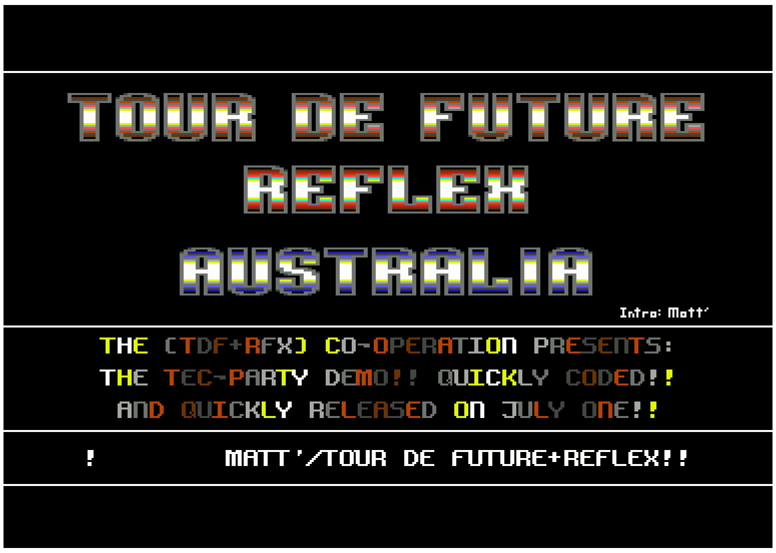


```
CONNECT 1200
[PRESS RETURN]

You have now entered the KBBS system ,
this board runs at 300, 1200/75, 1200 &
2400. This board runs on a 24hr basis
except at the following times where the
board is offline for ‘housekeeping’...
11pm - 12pm est.


YOUR SYSOPS ARE:

       Matt’/TOUR DE FUTURE+REFLEX
       X1r8 /TOUR DE FUTURE+REFLEX

This weeks main event is the online demo
set up from the TEC-PARTY...
View online demo (Y/N)??  Y
Booting demo...
Execute...
```

3.2. Screen two, ‘TDF+RFX Intro’

The intro pays homage to either IKARI or CONTEX, as it was inspired by them – but completely created by myself. It really does not say anything other than announce the coming of the demo.

Figure 9-3. OHH MATE!! Screen two, ‘TDF+RFX Intro’



TOUR DE FUTURE
REFLEX
AUSTRALIA
Intro: Matt'

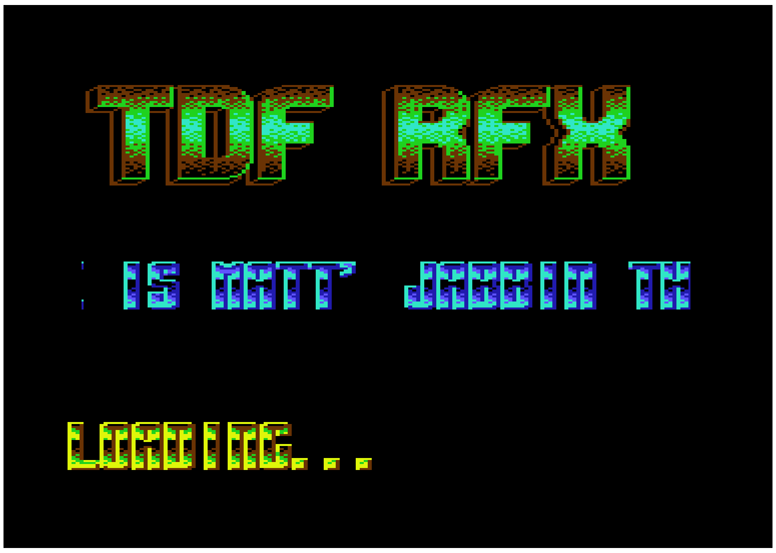
THE [TDF+RFX] CO-OPERATION PRESENTS:
THE TEC-PARTY DEMO!! QUICKLY CODED!!
AND QUICKLY RELEASED ON JULY ONE!!

! MATT'/TOUR DE FUTURE+REFLEX!!

3.3. Loader screen, ‘Loading…’

The screen is somewhat interesting in that it combines a standard demo like screen with a loading routine in the background, displaying a “Loading…” message during operation, and then a “Ready…” message when complete. The graphics were created by myself, but the loading routine was definitely ripped. This is where I was getting some of my experience prior to the loading routine for BEEFTRUCKER and DAMNABLE. The scroll text talks about my relations with HEX-HACKER and SNOOP, including a loan to them of my C1670 – I had definitely commenced modem activities by this time.

Figure 9-4. OHH MATE!! Loader screen, ‘Loading…’ (1)

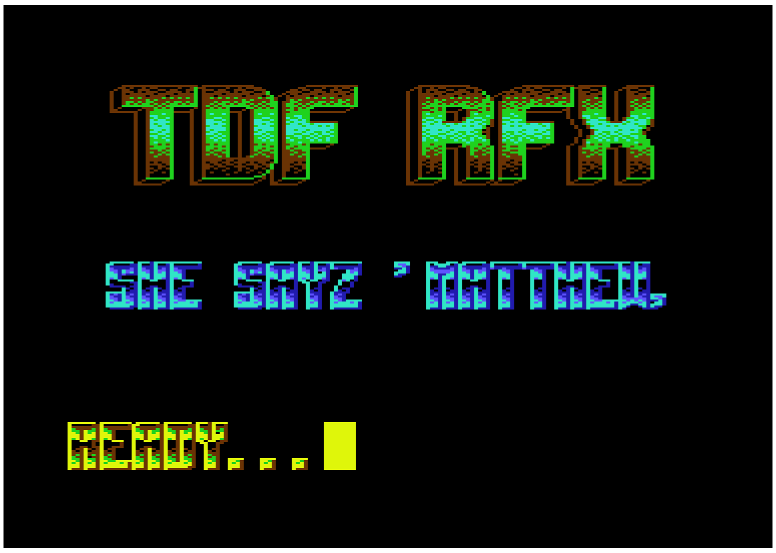


TDF RFX

IS MATT' JAGGIN TH

LOADING...

Figure 9-5. OHH MATE!! Loader screen, 'Loading…' (2)

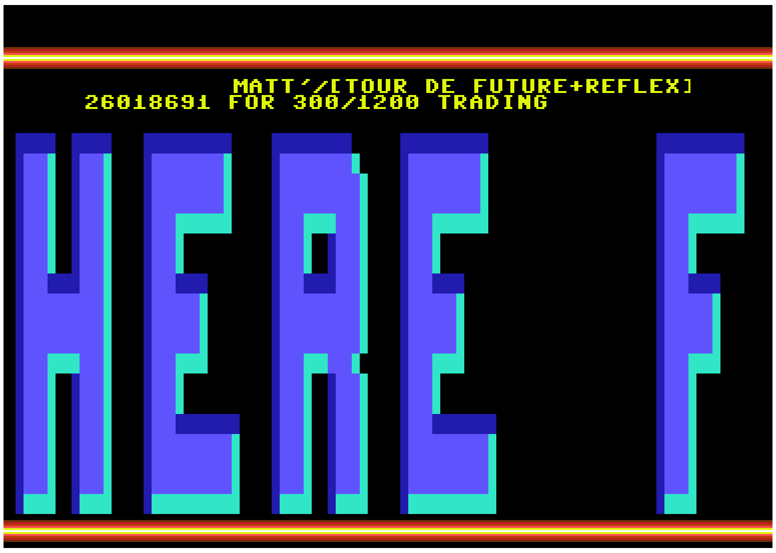


```
TDF RFX

SHE SAYZ 'MATTHEW,

READY...
```

3.4. Screen three, '$DO17 Flexer'

The novelty in this part is a super-large scrolling routine with stretched sprites and open-side borders. There would have been very little raster-time left for anything else, including the music. I was avoiding study for my mid-year exams by creating this part, and it was a rushed activity because apart from the code, nothing else was created by me : the sprites were taken from a HORIZON demo (in fact, the part smells of HORIZON a lot … perhaps it was inspired by them).

Figure 9-6. OHH MATE!! Screen three, '$DO17 Flexer'

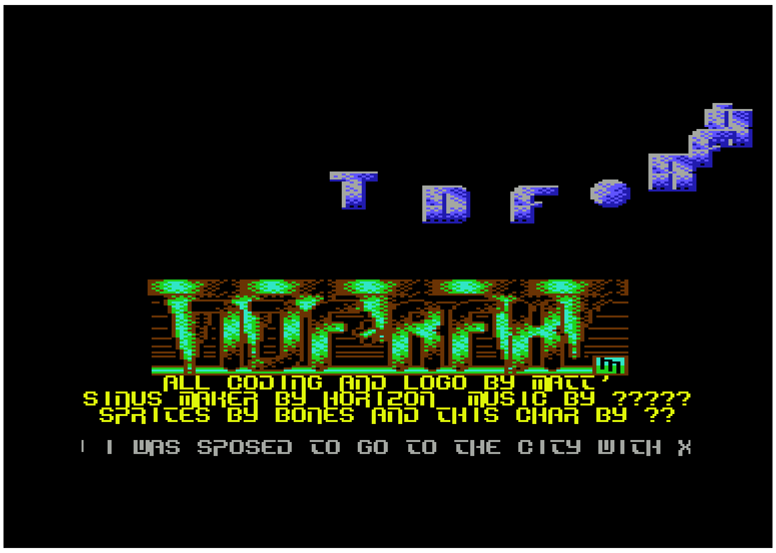


```
           MATT'/[TOUR DE FUTURE+REFLEX]
26018691 FOR 300/1200 TRADING
```

3.5. Screen four, 'No Sinus Limits'

I am impressed with this part as it has a nice feel to it, but is quite plain. I did not create any of the graphics other than the logo (but does illustrate increasing development of graphics skills). The open border sprite movement was probably a novel next step in my development as a coder.

Figure 9-7. OHH MATE!! Screen four, 'No Sinus Limits'



ALL CODING AND LOGO BY MATT'
SINUS MAKER BY HORIZON MUSIC BY ?????
SPRITES BY BONES AND THIS CHAR BY ??

I I WAS SPOSED TO GO TO THE CITY WITH X

3.6. Loader screen two, 'Loading is Fun'

Although the screen is dead boring (an embarrassing part of the demo), the music is quite good, and consumes a lot of raster-time. The loading routines occur while the music is playing.

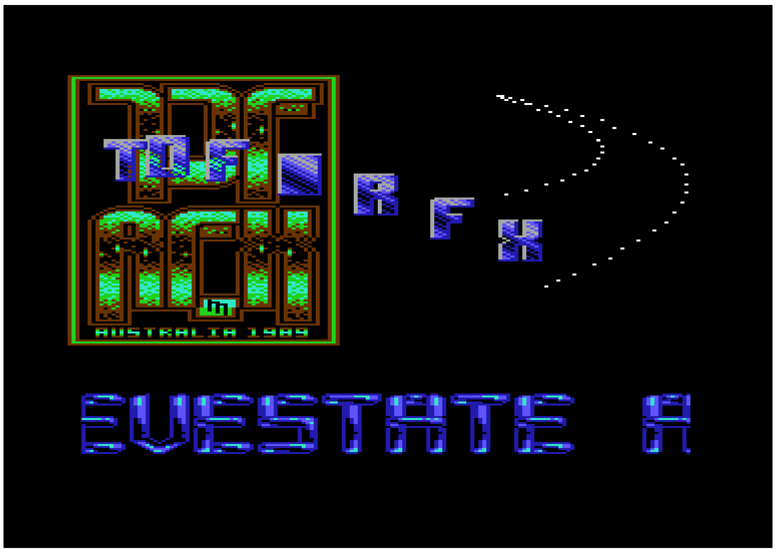
Figure 9-8. OHH MATE!! Loader screen two, 'Loading is Fun'

LOADING THE NEXT PART OF 'OHH MATE!!'...

3.7. Screen five, 'Sinus all round'

The major innovation in this part is the pixel plotter – but the logo and the TDF+RFX sprites are also kind of neat. The music, graphics, pixel plotter and scroll text all work together to provide a sort of funkiness. The scroll-text has a rhetorical theme to it.

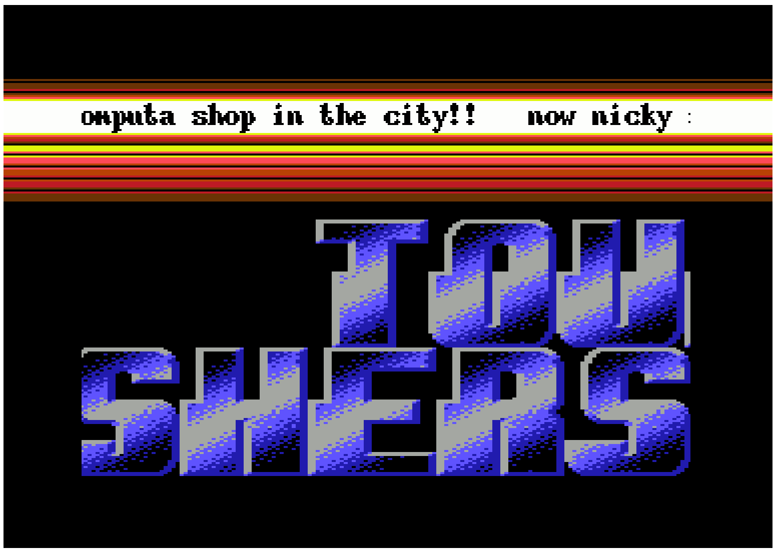
Figure 9-9. OHH MATE!! Screen five, 'Sinus all round'



3.8. Screen six, 'Cool dudes meeting'

As a tribute to our new friends, this part concerns myself, XLR8, HEX-HACKER and SNOOP and it was likely coded at least partially at HEX-HACKERs place. There is still little mention of MISTIC in these parts. There are greetings in here which illustrate our level of international contacts. The coding is not amazing, but there are some neat features: the large font used to create the swinging logos, and some of the effects in the synchronised scroll-text movement.

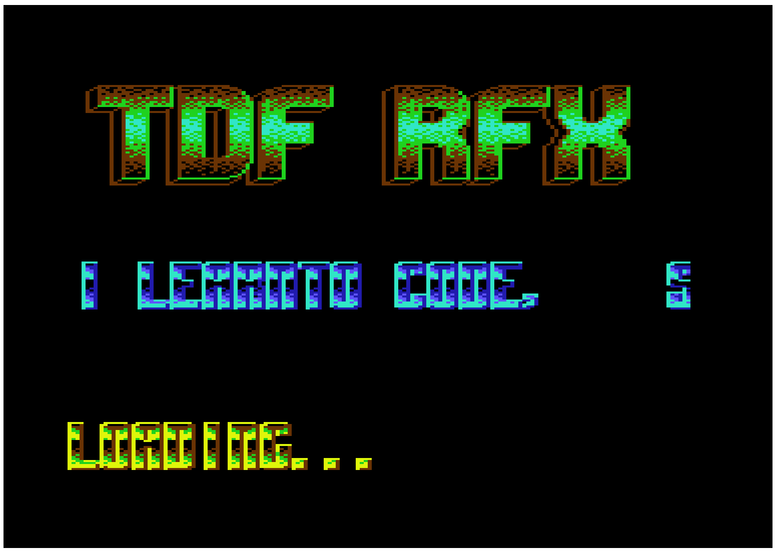
Figure 9-10. OHH MATE!! Screen six, 'Cool dudes meeting'



3.9. Loader screen three, ‘Loading…’

The loading screen re-appears, and importantly contains scroll-text about my starting steps as a coder.

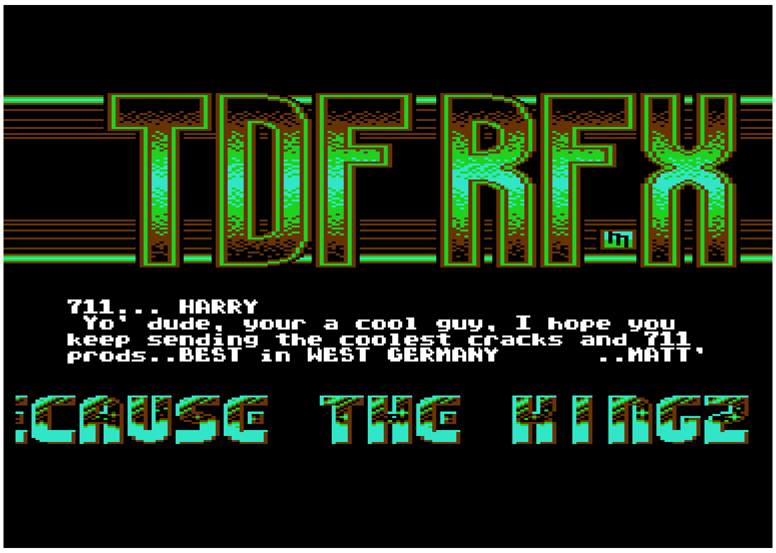
Figure 9-11. OHH MATE!! Loader screen three, ‘Loading…’



3.10. Screen seven, ‘No side limits!’

This is my favourite screen, as I created the upper logo and mapped it partially onto sprites so that it could swing into the side-border. The colour of the logo and the scroll text, plus the open-border nature of them gives the part a kind of nice feel. The central part of the screen displays a sequences of messages to our contacts, and we had a few good ones at this time.

Figure 9-12. OHH MATE!! Screen seven, ‘No side limits!’

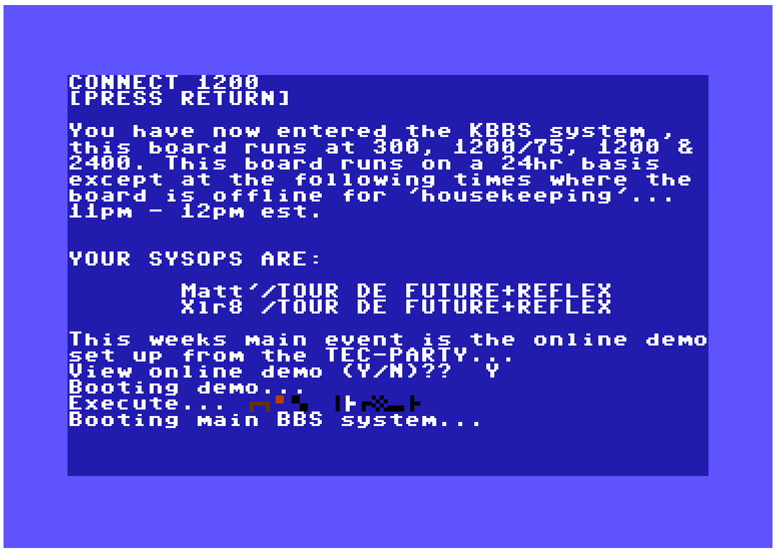


```
711... HARRY
 Yo' dude, your a cool guy, I hope you
keep sending the coolest cracks and 711
prods..BEST in WEST GERMANY        ..MATT'
```

3.11. Screen eight, 'Log off…'

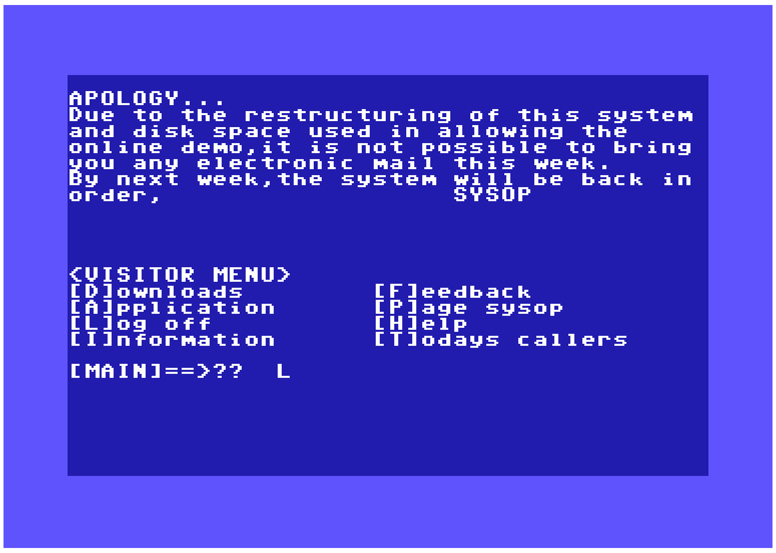
The final screen emulates the return to a bulletin board system, and the logoff phase which leads onto a reset of the computer.

Figure 9-13. OHH MATE!! Screen eight, 'Log off…' (1)



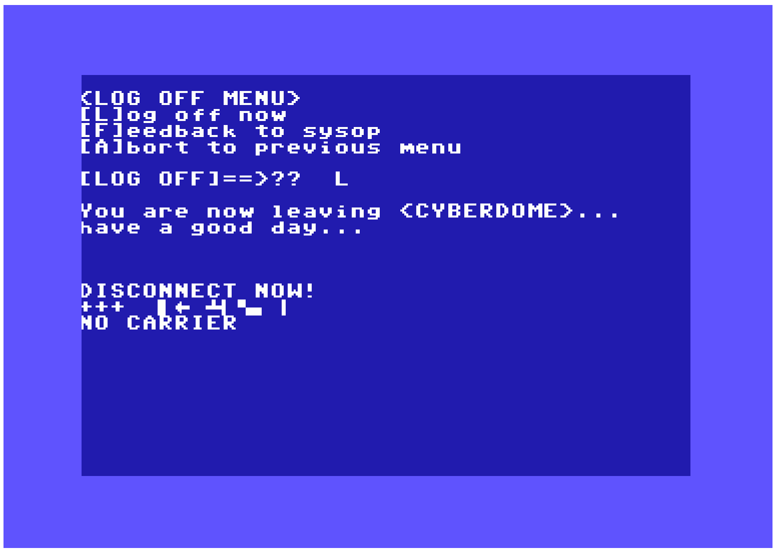
```
CONNECT 1200
[PRESS RETURN]

You have now entered the KBBS system ,
this board runs at 300, 1200/75, 1200 &
2400. This board runs on a 24hr basis
except at the following times where the
board is offline for 'housekeeping'...
11pm - 12pm est.

YOUR SYSOPS ARE:

           Matt'/TOUR DE FUTURE+REFLEX
           X1r8 /TOUR DE FUTURE+REFLEX

This weeks main event is the online demo
set up from the TEC-PARTY...
View online demo (Y/N)??  Y
Booting demo...
Execute...
Booting main BBS system...
```

Figure 9-14. OHH MATE!! Screen eight, 'Log off…' (2)



```
APOLOGY...
Due to the restructuring of this system
and disk space used in allowing the
online demo,it is not possible to bring
you any electronic mail this week.
By next week,the system will be back in
order,                               SYSOP


<VISITOR MENU>
[D]ownloads             [F]eedback
[A]pplication           [P]age sysop
[L]og off               [H]elp
[I]nformation           [T]odays callers

[MAIN]==>??   L
```

Figure 9-15. OHH MATE!! Screen eight, ‘Log off…’ (3)



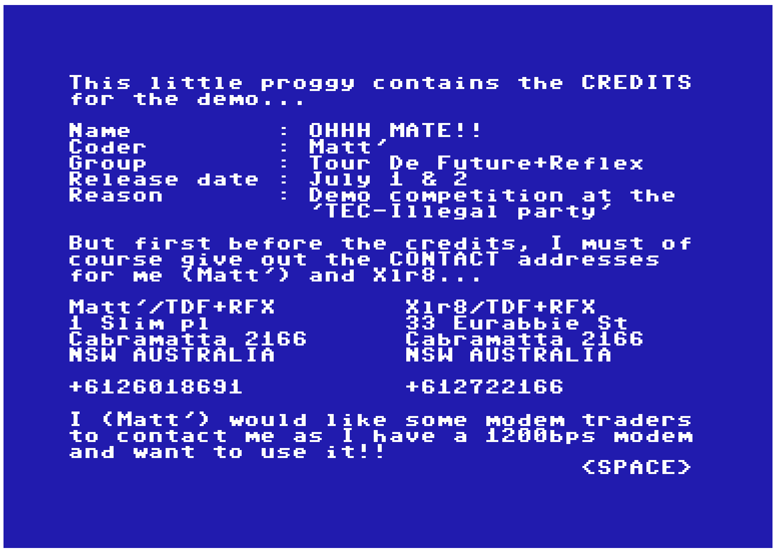
```
<LOG OFF MENU>
[L]og off now
[F]eedback to sysop
[A]bort to previous menu

[LOG OFF]==>??  L

You are now leaving <CYBERDOME>...
have a good day...


DISCONNECT NOW!
+++  ▌← ┤▚▬ |
NO CARRIER
```

4. Attachments

4.1. Demo credits

The demo credits were provided in a separate file, and are reasonably comprehensive.

Figure 9-16. OHH MATE!! credits



```
This little proggy contains the CREDITS
for the demo...

Name          : OHHH MATE!!
Coder         : Matt'
Group         : Tour De Future+Reflex
Release date  : July 1 & 2
Reason        : Demo competition at the
                'TEC-Illegal party'

But first before the credits, I must of
course give out the CONTACT addresses
for me (Matt') and X1r8...

Matt'/TDF+RFX           X1r8/TDF+RFX
1 Slim pl               33 Eurabbie St
Cabramatta 2166         Cabramatta 2166
NSW AUSTRALIA           NSW AUSTRALIA

+6126018691             +612722166

I (Matt') would like some modem traders
to contact me as I have a 1200bps modem
and want to use it!!
                                <SPACE>
```

Section 10. C64 Activities Demos DAMNABLE

1. Background

1.1. The party

Towards the end of 1989, a large party was organised in Adelaide – to be held by THE FORCE and the other guys. We were not on good terms with THE FORCE, but we were on good terms with MADHACKER of IKON VISUAL so decided to attend with his hospitality, and we also needed to create a demo for the party. It was usual for groups to create a demo to release at a party, and most had competitions. This demo won the competition at the party (easily!).

1.2. The production

At this stage in my maturation, I had a reasonable mastery of graphics. I created every item of graphic in this demo – other than the system font. I had tools to assist me in merging bitplanes, and automating parts of the process. Most of my graphics were font oriented, or used patterns in some mathematical form. I was not good at "freestyle" drawing.

I also created all the code. There is a particular emphasis on pre-computed code here, as a lot of the routines are processor intensive, or required fine timing, so the parts generate code as they start. For instance, individual load/stores require less clock-cycles than indexed load/stores, so runtime code is created that uses individual stores. I had recently completed BEEFTRUCKER, and had created my own fast loader (it was a highly derived work…) which was used to link parts together. Naturally, all parts were packed with BEEFTRUCKER!

There is no particular overall theme to this production though. It is “just an assembly of parts”, but each part generally tries to do something outstanding. For instance, the “25 scrollers at once”, “full open border logo and DYCP”, “turbo outrun ripped music”, “sprite multiplexer and evolving character set”, “124 pixel plotter” and “disk fast loader” were all at the cutting edge of my own personal skills. I also consider the overall packaging to be well done, compared to my previous productions.

1.3. The process

Parts of the demo were completed in Sydney in November, prior to the party in Adelaide. For instance, we had only just obtained TURBO OUTRUN from boards in the states, and I used the freshly ripped music in a part. In Adelaide, I completed a few of the parts (and possibly created one from scratch … I have no particular memories), and filled in some of the scroll texts (a few were written in Sydney). The demo was packed and put together as a “release”. We had taken our computers along with us, which made this possible.

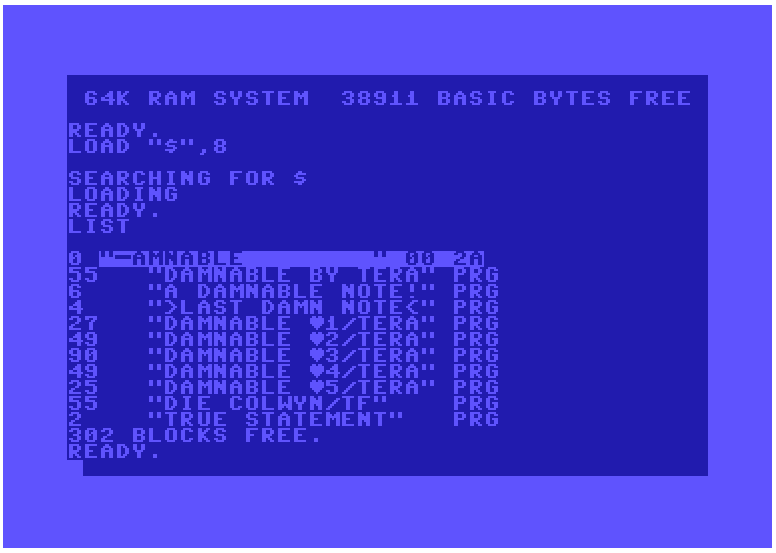
2. Packaging

The demo was released as a multi-part demo, but not in a single file. The reason for this is that some of the parts may be small, but require a large amount of memory to execute in, as they use pre-computed code or graphics.

It is possible to load each part individually, but a fast loader and loading screen is used if the demo is viewed in its entirety. The files are neatly structured and can be copied very easily. I am not a fan of “BAM copy only” productions with obscure file names: they usually make things more difficult and annoying.

The distribution included all parts, plus two release notes, plus two items of propaganda.

Figure 10-1. DAMNABLE disk packaging



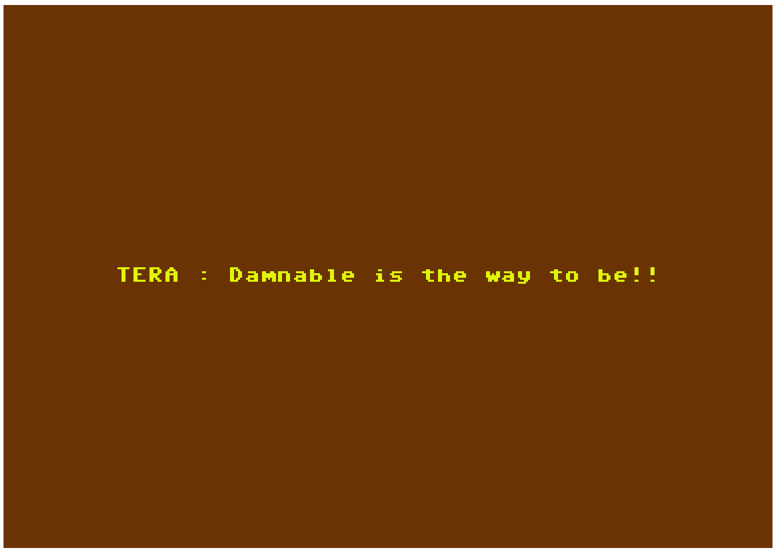
```
 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
LOAD "$",8

SEARCHING FOR $
LOADING
READY.
LIST

0 "—AMNABLE          " 00 2A
55    "DAMNABLE BY TERA"  PRG
6     "A DAMNABLE NOTE!"  PRG
4     ">LAST DAMN NOTE<"  PRG
27    "DAMNABLE ♥1/TERA"  PRG
49    "DAMNABLE ♥2/TERA"  PRG
90    "DAMNABLE ♥3/TERA"  PRG
49    "DAMNABLE ♥4/TERA"  PRG
25    "DAMNABLE ♥5/TERA"  PRG
55    "DIE COLWYN/TF"     PRG
2     "TRUE STATEMENT"    PRG
302 BLOCKS FREE.
READY.
```

3. Parts

3.1. Unpack screen

While unpacking, the user is provided with TERA “platitude” – there were numerous platitudes in this demo. You can consider them to be marketing statements!

Figure 10-2. DAMNABLE Initial unpack message screen w/ propaganda

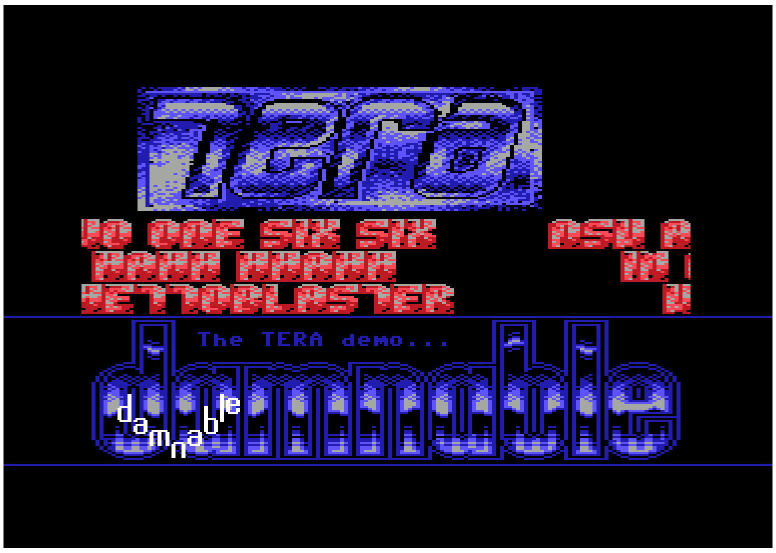


TERA : Damnable is the way to be!!

3.2. Intro screen

The intro screen fades in and out, and provides contact information. The code is not particularly innovative, but I am impressed with the graphics, especially the TERA logo. The scroll texts are embarrassing if you read them.

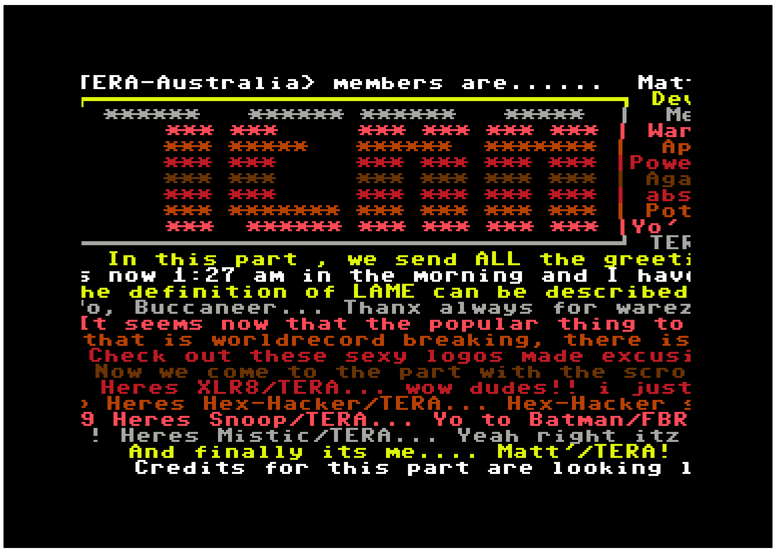
Figure 10-3. DAMNABLE Intro part, presenting DAMNABLE to the viewer



3.3. First part, with 25 scrollers

This is the part where we greet all of our friends and contacts. It is an innovative part, in that the 25 scrollers require a lot of computing cycles, and I must also alter values down the raster line. There is a lot of pre-computed code that performs the scrolling in minimal clock cycles. This demo has a large amount of pre-computed code and graphics, as it was a something that I had started to master.

Figure 10-4. DAMNABLE First part, with 25 scrollers occurring at once

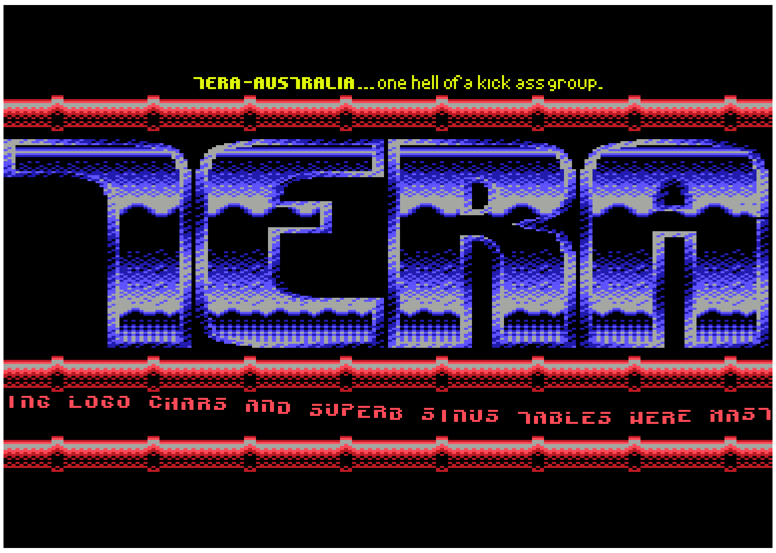


```
TERA-Australia> members are......   Mat
                                     Dev
                                      Me
                                     War
                                      Ap
                                    Powe
                                     Aga
                                     abs
                                     Pot
                                    Yo'
                                     TER
   In this part , we send ALL the greeti
s now 1:27 am in the morning and I hav
he definition of LAME can be described
'o, Buccaneer... Thanx always for warez
It seems now that the popular thing to
that is worldrecord breaking, there is
 Check out these sexy logos made excusi
  Now we come to the part with the scro
   Heres XLR8/TERA... wow dudes!! i just
> Heres Hex-Hacker/TERA... Hex-Hacker s
9 Heres Snoop/TERA... Yo to Batman/FBR
 ! Heres Mistic/TERA... Yeah right itz
     And finally its me.... Matt'/TERA!
      Credits for this part are looking l
```

3.4. Second part, with open borders

This is a part that I am pleased with because of its stylistic integrity. I created the logo and transferred some of into sprites that sit in the side border (I am not sure whether this transfer into sprites occurs as the part starts, or where I used my own routines to do it). The various fills in the logo were created by own bit-plane merging routines. I must keep the side border open along each raster line, for most part of the screen, which takes a lot of raster time. The open border scroll is a processor intensive 2 high DYCP, which is implemented using pre-computed code and graphics. There are interesting embedded messages at $0f00, $1a00 and $4600.

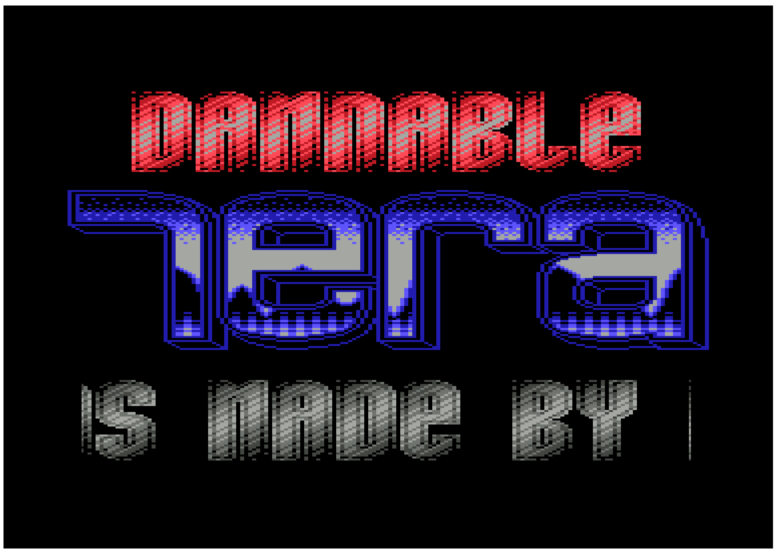
Figure 10-5. DAMNABLE Second part, with full open border action


TERA-AUSTRALIA ...one hell of a kick ass group.
TERA
ING LOGO CHARS AND SUPERB SINUS TABLES HERE MAST

3.5. Third part, with TURBO OUTRUN music

In the few nights before we travelled to the party in Adelaide, we downloaded TURBO OUTRUN from the states. It has just been released and cracked, and had not arrived via post – so it was extremely hot. The music was great and included digitised efforts, so with a bit of effort, I was able to rip out the music create a quick part to show off our speed and might. The music itself is processor intensive, and uses interrupts itself, so it was not possible to create a very complex part anyway. We also released an “import” of TURBO OUTRUN at the party. There are interesting embedded messages at $7100 and $7300. The scroll indicates that there was two weeks until the party, and I had only created two parts – in the last three days, I had only slept 10 hours.

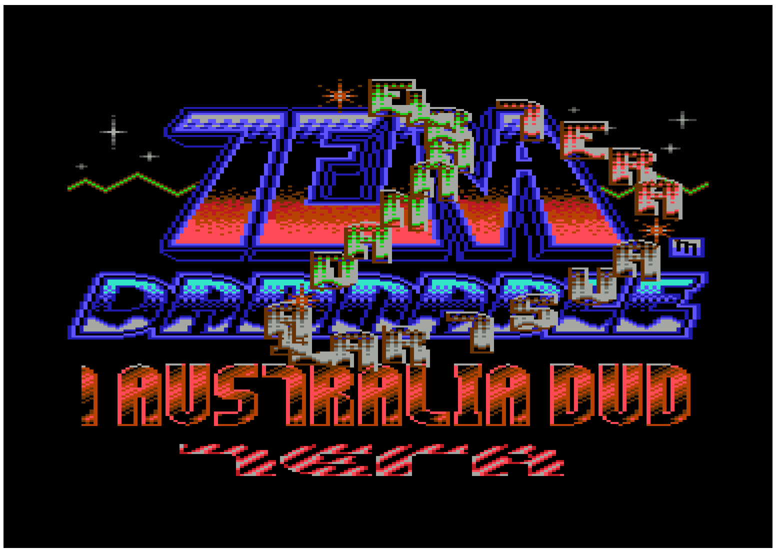
Figure 10-6. DAMNABLE Third part, with ripped TURBO OUTRUN music



3.6. Fourth part, with multiplexer and graphics

The impressive activity in this part is the >8 sprites, which is managed with fun raster based interrupt routines. It involves rewriting the sprite register and ensuring that the sinusoid tables always leave enough pixel distance at any particular point in time so that too many sprites will not overlap each other. This was not an easy task. The other feature of this part is the character set, which is redesigned in real-time, using pre-computed code that efficiently rewrites, and then switches the video bank registers. Although the code is neat, I think that there is a lack of stylistic integrity here. There are interesting embedded messages at $1600. There is a bug in this part which I am sure was not in the released version (but I think that it was … ugh), as some of the characters turn out wrong.

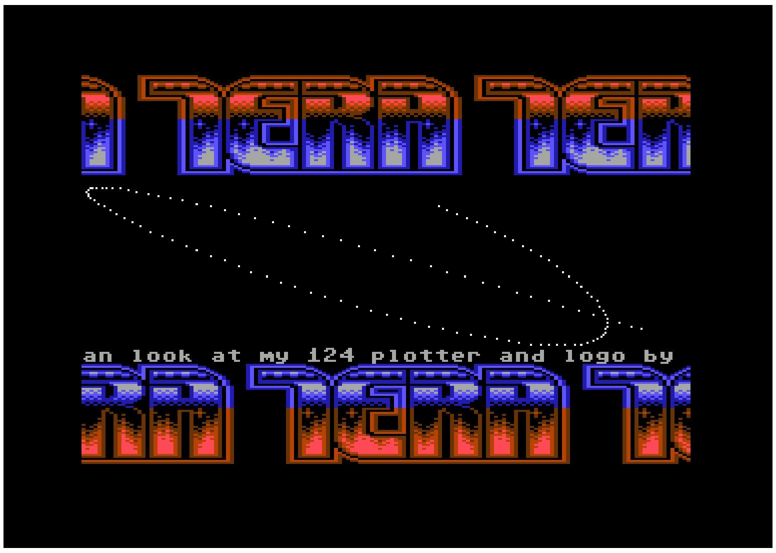
Figure 10-7. DAMNABLE Fourth part, with multiplexed sprite routines



3.7. Fifth part, with 124-point pixel plotter

Another use of pre-computed code is in this routine that needs to plot 124 pixels every frame, which is quite an intensive task. It is a little boring though. The scroll text was written at the party, and mentions our 2000KM train ride, and experience with coca-cola that tastes different.

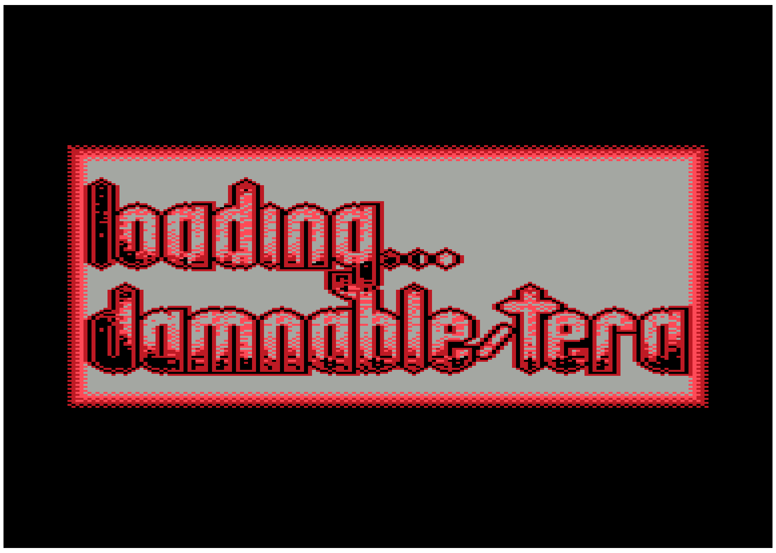
Figure 10-8. DAMNABLE Fifth part, with 124-point pixel plotter



4. Loader

One of best things I did with this demo (at least, in my mind!) was link the parts using a fast loader, and a loading screen. I also preserved colour integrity (no annoying escapes to the blue screens), so that a part would complete, and then the loading screen would fade in, and then fade out, before the next part kicked in. All activity in the background was transparent to the user. So many coders failed on these simple tasks. I created the fast loader (well, it was *heavily* based on other work – almost ripped, but not quite), and I created the graphics. The loader was not duplicated for each part, but remained stashed in memory – which is why parts that are run individually will exit to a blue screen (I should have had them exit more gracefully). There are interesting embedded messages at $1c00.

Figure 10-9. DAMNABLE Parts loader

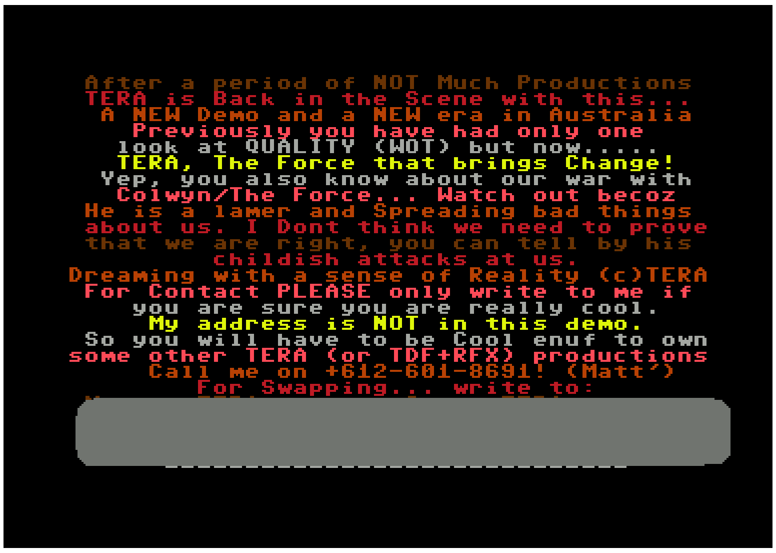


5. Notes

5.1. Formal release note

The demo was released with a formal note, which provides an introduction to the demo, and something about our contact addresses. I lied though, because my address is in the demo. I was less interested in postal trade. The display routine is pretty simple, but self-written.

Figure 10-10. DAMNABLE Release note, with contact addresses

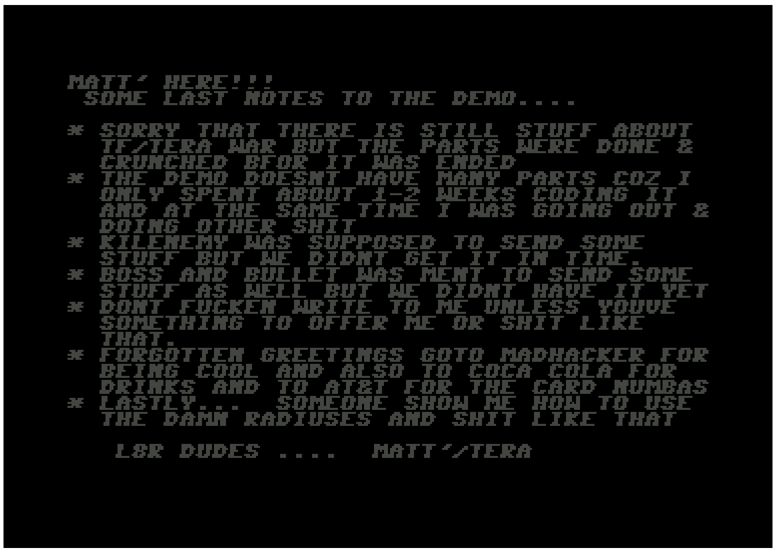


```
After a period of NOT Much Productions
TERA is Back in the Scene with this...
 A NEW Demo and a NEW era in Australia
   Previously you have had only one
  look at QUALITY (WOT) but now.....
  TERA, The Force that brings Change!
 Yep, you also know about our war with
  Colwyn/The Force... Watch out becoz
He is a lamer and Spreading bad things
about us. I Dont think we need to prove
that we are right, you can tell by his
         childish attacks at us.
Dreaming with a sense of Reality (c)TERA
 For Contact PLEASE only write to me if
   you are sure you are really cool.
    My address is NOT in this demo.
 So you will have to be Cool enuf to own
some other TERA (or TDF+RFX) productions
    Call me on +612-601-8691! (Matt')
       For Swapping... write to:
```
[redacted]

5.2. Informal release note from MATT'

This informal note provided some information about why I had not been very active, and some other information about the demo. I am not sure whether this was written in Adelaide, or elsewhere. It was created using the 4 BLOCK NOTER!

Figure 10-11. DAMNABLE Release note, with addendum information



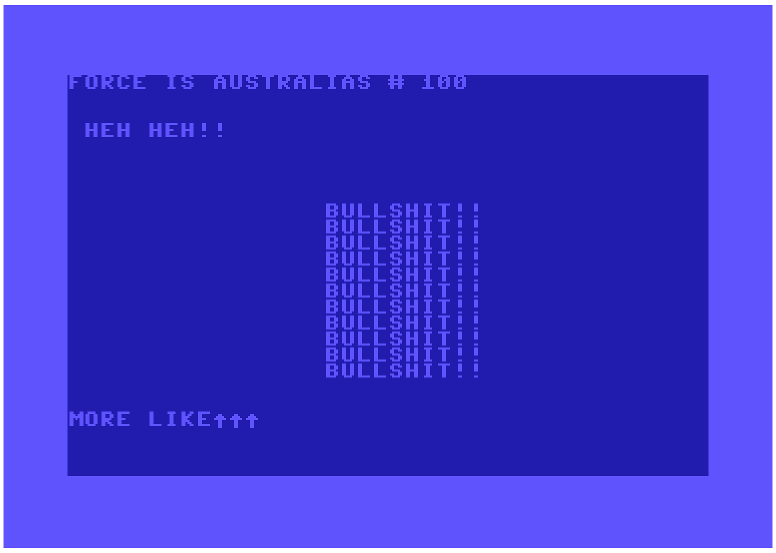
```
MATT' HERE!!!
 SOME LAST NOTES TO THE DEMO....

* SORRY THAT THERE IS STILL STUFF ABOUT
  TF/TERA WAR BUT THE PARTS WERE DONE &
  CRUNCHED BFOR IT WAS ENDED
* THE DEMO DOESNT HAVE MANY PARTS COZ I
  ONLY SPENT ABOUT 1-2 WEEKS CODING IT
  AND AT THE SAME TIME I WAS GOING OUT &
  DOING OTHER SHIT
* KILENEMY WAS SUPPOSED TO SEND SOME
  STUFF BUT WE DIDNT GET IT IN TIME.
* BOSS AND BULLET WAS MENT TO SEND SOME
  STUFF AS WELL BUT WE DIDNT HAVE IT YET
* DONT FUCKEN WRITE TO ME UNLESS YOUVE
  SOMETHING TO OFFER ME OR SHIT LIKE
  THAT.
* FORGOTTEN GREETINGS GOTO MADHACKER FOR
  BEING COOL AND ALSO TO COCA COLA FOR
  DRINKS AND TO AT&T FOR THE CARD NUMBAS
* LASTLY...  SOMEONE SHOW ME HOW TO USE
  THE DAMN RADIUSES AND SHIT LIKE THAT

   L8R DUDES ....   MATT'/TERA
```

6. Propaganda

6.1. THE FORCE, BASIC hate note

We were having a "war" with COLWYN of THE FORCE, so included this "hate note" written in BASIC. It is very childish, but then, we were young at the time. I am not sure who created this note – I think it may have been myself.

Figure 10-12. THE FORCE, BASIC hate note

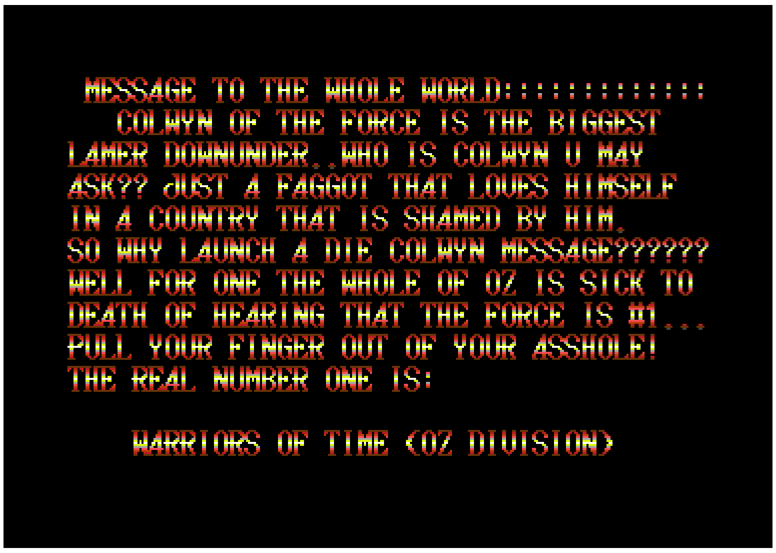


```
FORCE IS AUSTRALIAS # 100

 HEH HEH!!


                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!
                BULLSHIT!!

MORE LIKE↑↑↑
```

6.2. THE FORCE, extended hate note

This more detailed note was written by XLR8 and provides a lot of detail about our war with COLWYN of THE FORCE, and its history. There are some observations about how COLWYN didn't seem to know what he was doing as a coder – implying that he ripped a lot of code (we thought that he had).

Figure 10-13. THE FORCE, extended hate note



MESSAGE TO THE WHOLE WORLD:::::::::::::
COLWYN OF THE FORCE IS THE BIGGEST
LAMER DOWNUNDER..WHO IS COLWYN U MAY
ASK?? JUST A FAGGOT THAT LOVES HIMSELF
IN A COUNTRY THAT IS SHAMED BY HIM.
SO WHY LAUNCH A DIE COLWYN MESSAGE??????
WELL FOR ONE THE WHOLE OF OZ IS SICK TO
DEATH OF HEARING THAT THE FORCE IS #1...
PULL YOUR FINGER OUT OF YOUR ASSHOLE!
THE REAL NUMBER ONE IS:

WARRIORS OF TIME (OZ DIVISION)

Section 11. C64 Activities Demos SOLICITUDE

1. Background

As I did not construct this demo entirely, I will not cover all of its parts. There were other parts created by other members of TERA, including a credits screen at the end that clearly shows all responsibilities. The demo used a few parts that I had created myself, but it also contained parts that used some of my graphics (e.g. logos and fonts). It was fairly common for other TERA coders to use my products (e.g. Hex used some of my fonts and logos in LOVESTRUCK and ENCAPSULATE).

In early 1990, I created a few parts for our next group release, to be called SOLICITUDE. Unfortunately, I have little memory of the time, but there was something of a fall out between all of us guys [this is documented elsewhere in more detail!], and we were going our own way (in particular, I was busy at University, and starting to take an interest in bulletin boards). I probably gave the parts to the other guys, and then the whole demo was likely to have been stitched together by BULLET and BOSS. It also includes parts from German members, and other Australian members, but none from HEX-HACKER (I don't know why).

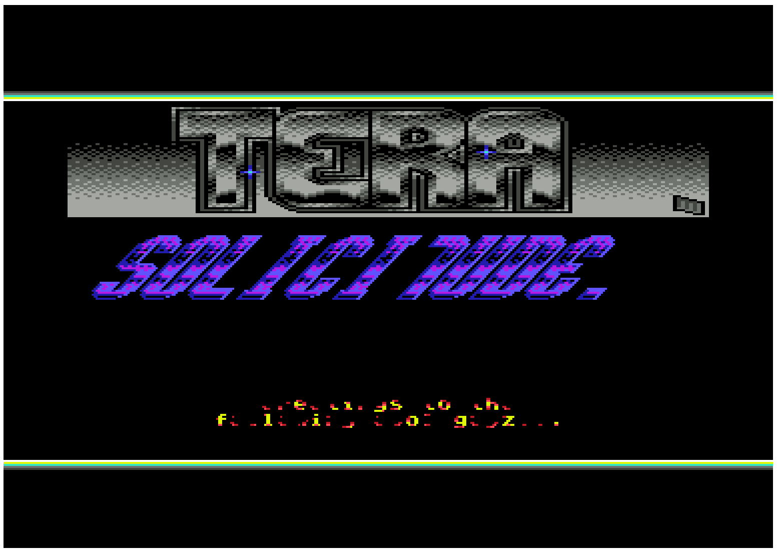
My contributions show maturity around the same level, perhaps just a little further advanced, than the DAMNABLE demo. These parts were probably created in the two months after DAMNABLE (as I had some time free late in 1989 and early in 1990 after I finished High School before I started University). Some of the scroll texts can be dated to early 1990.

2. Parts

2.1. Intro, with logo

I am responsible for the logo and the character set, however they had both been taken from the BEEFTRUCKER intro without my permission.

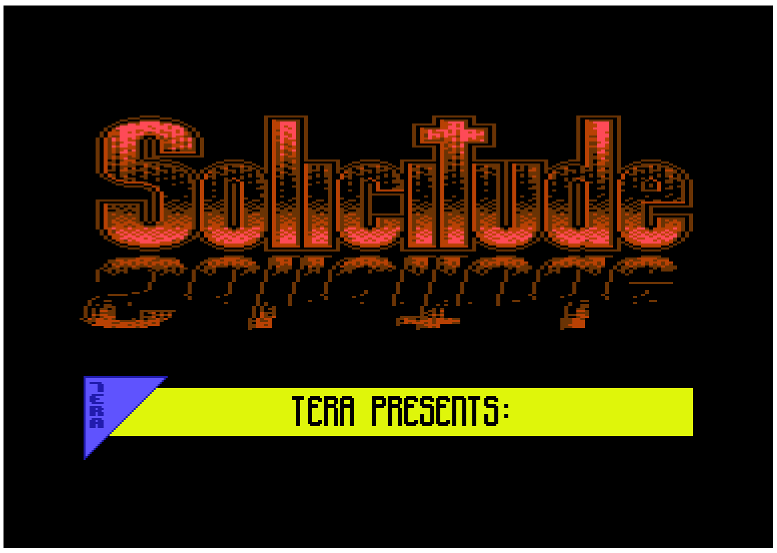
Figure 11-1. SOLICITUDE Intro screen, with logo



2.2. Opening screen, with cinema like effects

This part was designed to provide a cinematic opening to the demo. The logo fades into view, and the text fades in and out before it all collapses back into darkness. I created a few routines to make the shadow from the logo (why do it by hand when it is easier to code?). It was going to be the first part (i.e. the Intro). I did not write the text though: it was created later. As there is no music in this part, then I can claim 100% responsibility for it.

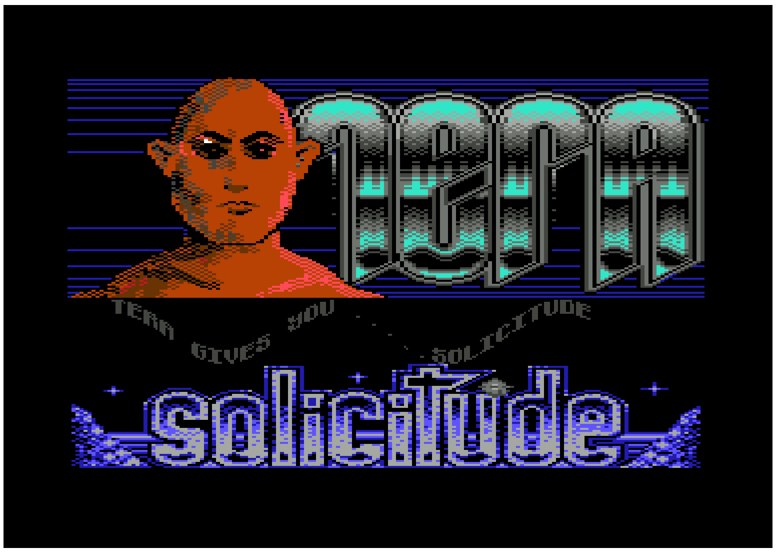
Figure 11-2. SOLICITUDE Opening screen, with cinema like effects



2.3. 5th Part, with logo and pixel DYCP

A local friend of ours had a C64, and he turned out to be a good graphics artist. The picture he drew of a bald headed man, needed to be somewhere, so this part was born. I created the logo and the background and fitted it in with his image (all rather nice, I think!). MISTIC created the lower logo (it is unusually good for his usual style). Embedded strings at $4000 indicate that I was author, and $5B00, $6F00 have other messages. The DYBP (! not DYCP as they called it) character set was by someone else. The scroll text (which is weirdly poetic) indicates that on Wednesday I am going to enroll in “Computer Systems Engineering”, which occurred in Feb/March 1990.

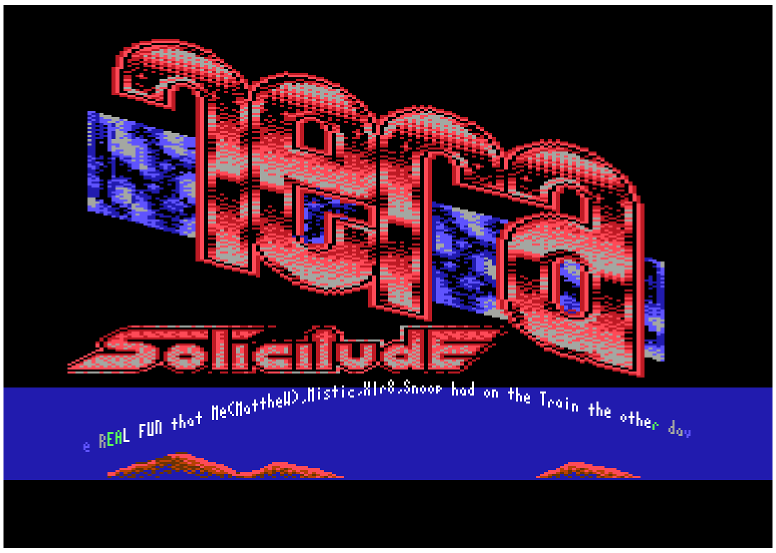
Figure 11-3. SOLICITUDE 5th part, with logo and pixel DYCP



2.4. 80 Column DYCP, with large logo

This is part that I am proud of. The 80 column DYCP uses my own font, and a set of pre-computed routines. I created the large TERA logo by drawing, but also by filling in with my own fill routines, and I also italicised it. The SOLICITUDE logo is also my creation (a poor one, however), but the character set was taken from an 80 column terminal program. MISTIC was responsible for the “mountains”. The scroll text advertises our board in the United States: THE JUNGLE, located in 215.

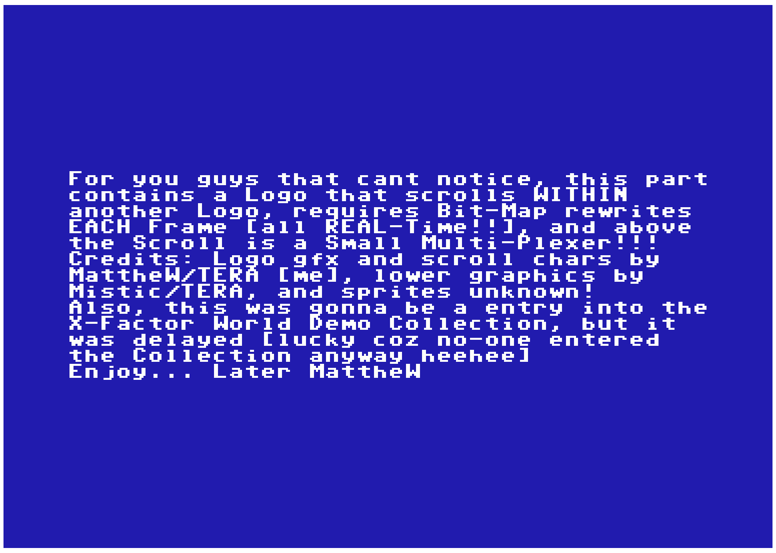
Figure 11-4. SOLICITUDE 80 Column DYCP, with large logo


Tera
Solicitude
e REAL FUN that Me(MattheW),Mistic,Xlr8,Snoop had on the Train the other day

2.5. Logo inside Logo, with introductory screen

I was also proud of this part, enough so that it comes with an introductory screen describing the main routine (a logo within a logo). The part was intended for a demo competition, but I do not remember any details about this. When the space key is pressed, temporary run time information is splattered on the screen: this is done on purpose, to show “technical junk” from the code pre-computer.

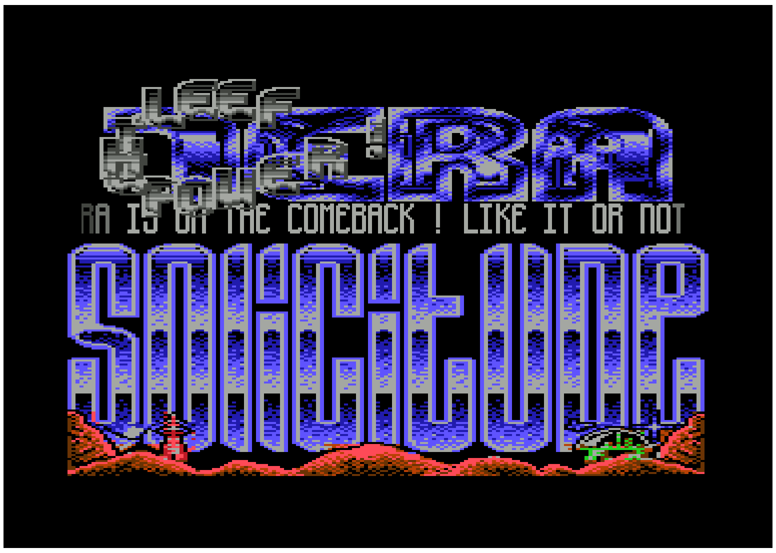
Figure 11-5. SOLICITUDE Logo inside Logo, introductory screen



```
For you guys that cant notice, this part
contains a Logo that scrolls WITHIN
another Logo, requires Bit-Map rewrites
EACH Frame [all REAL-Time!!], and above
the Scroll is a Small Multi-Plexer!!!
Credits: Logo gfx and scroll chars by
MattheW/TERA [me], lower graphics by
Mistic/TERA, and sprites unknown!
Also, this was gonna be a entry into the
X-Factor World Demo Collection, but it
was delayed [lucky coz no-one entered
the Collection anyway heehee]
Enjoy... Later MattheW
```

The logo within logo occurs as a result of pre-computed code, similar to many of the parts in the DAMNABLE demo. I created the character set as well. MISTIC was responsible for the lower SOLICITUDE logo, and wrote the scroll text.

Figure 11-6. SOLICITUDE Logo inside Logo, main screen



3. Loader

A loader was used to bind together the SOLICITUDE parts and it was my fast loader, as I had created and used for BEEFTRUCKER and DAMNABLE.

Section 12. C64 Software

1. D64 disk images

All software mentioned and screen-captured in the repository is provided in the following D64 disk images.

Name	Contents	D64 image
Tools	Includes BEEFTRUCKER, NOTEMAKER V1.0, NOTEMAKER V2.0, STUBBY-NOTER and 4 BLOCK NOTER!	
Intros	Includes various intros produced for TOUR DE FUTURE, TOUR DE FUTURE+REFLEX and TERA.	
Demos – BACK FROM BEYOND	Includes the BACK FROM BEYOND demo produced for TOUR DE FUTURE.	
Demos – HUH	Includes the HUH demo produced for TOUR DE FUTURE.	
Demos – OHH MATE!!	Includes the OHH MATE!! demo produced for TOUR DE FUTURE+REFLEX.	
Demos – DAMNABLE	Includes the DAMNABLE demo produced for TERA.	
Demos – SOLICITUDE	Includes the SOLICITUDE demo produced for TERA.	
Examples	Includes various examples created in this document.	

Index