VOL 1 No 6 $3*

A Gareth Powell Magazine

The Australian COMMODORE REVIEW

raining cats a
press . . . at last Columbo is d
defence The Commodore 64 as a bu
Hot Stuff Where does
c-20 games review All
et Serious Program Di
ech Tips View from t
am Rumblings PC'84
dt's Sound Advice It'
– Part II 1541 Expres
r In the readers defe
TOTL picture Hot Stu
/Blagger Vic-20 games
Routine Get Serious Pro
en Blitz Tech Tips View
torial Ram Rumblings PC
Behrendt's Sound Advice
jump – Part II 1541 Expre
strator In the readers defen
the TOTL picture Hot Stuff
ack/Blagger Vic-20 games rev
t Routine Get Serious Program
lien Blitz Tech Tips View from t
ditorial Ram Rumblings PC'84 Show
Behrendt's Sound Advice It's raining
jump – Part II 1541 Express . . . at last Columbo is dead Tech sketch Micro i
rator In the readers defence The Commodore 64 as a business machine – gi
he TOTL picture Hot Stuff Where does all the memory go? The Hobbit Ba
/Blagger Vic-20 games review All about file loading Vic-20 hints Scree
outine Get Serious Program Directory Starting with disk drives – Pa
Blitz Tech Tips View from the Hold Wonder if it will do the washir
l Ram Rumblings PC'84 Show report Letters Inside Melbourne H
dt's Sound Advice It's raining cats and dogs Making BASIC ho
Part II 1541 Express . . . at last Columbo is dead Tech sketch
he readers defence The Commodore 64 as a business ma
ture Hot Stuff Where does all the memory go? The
20 games review All about file loading Vic-20
ous Program Directory Starting with di
ew from the Hold Wonder if it wi
'84 Show report Letter
bit I
creen
Part
ing up
House
p, skip
Micro il
ine – giv
bbit Bat A
s Screen In
ves – Part II
e washing up?
elbourne House
g BASIC hop, skip
OF THE
CAMELS
to th
Ca

Registered by Australia Post Publication No. NBG6656

*Recommended retail price

CodePro-64

Advanced Visual Programming Instruction For The Commodore 64.

Now you can learn BASIC and develop advanced programming skills with SPRITES and MUSIC — visually.

CodePro-64 is a professional-quality educational program designed to fulfill the need for quality tutorial and demonstration software for the Commodore 64.

Comprehensive Basic Language Tutorial

CodePro-64, the **most advanced and integrated instructional software** for the Commodore 64, guides you through structured tutorials on BASIC programming, using exclusive BasicView™graphics to visually illustrate program execution. **Interactive examples combined with graphics** let you step through and actually see the execution of sample program statements.

With CodePro-64, **you control your learning**. You use the **Main Menu** and **Topic Menus** to select different topics, or use **keywords** to select language elements to study. You can never get "lost." You can page back and forth between screens within a topic at the touch of a function key, and F1 always returns you to the Main Menu.

CodePro-64 lets you follow your interests and develop your BASIC programming skills by practicing with interactive examples, then lets you move on to more advanced concepts of SPRITE and MUSIC programming.

Sprite Generator & Demonstrator

With the CodePro-64 **sprite generator**, you define your own sprites on the screen and learn what data values correspond to your sprite definitions. You can experiment with different definitions and make changes to immediately see the effects, then save your sprite definitions on diskette and reload them with CodePro-64 or with the sample program provided.

We also help you learn to program with sprites by giving you a **sprite demonstrator**, so you can see sprites move around in a screen segment, change colors or size, and see the effects of your changes.

Music Generator & Demonstrator

CodePro-64 also gives you an **interactive music generator and demonstrator**. Using the **music generator**, you can set all your SID parameters, then **enter notes to play**. And we show your tune graphically as it plays, note by note, on the scale. You learn music programming by seeing and hearing the results of your input. And, you can **save your music on diskette** for replay by CodePro-64 or by the sample program provided.

Our **music demonstrator lets you experiment with various combinations of music programming parameters** and hear the results by playing the keyboard "organ." You can quickly modify any of the SID register values to hear the effects of the change. You can also change waveform and attack/decay values while holding all other SID values constant. By seeing your input and hearing the result, you quickly learn how to create new musical sounds and special sound effects.

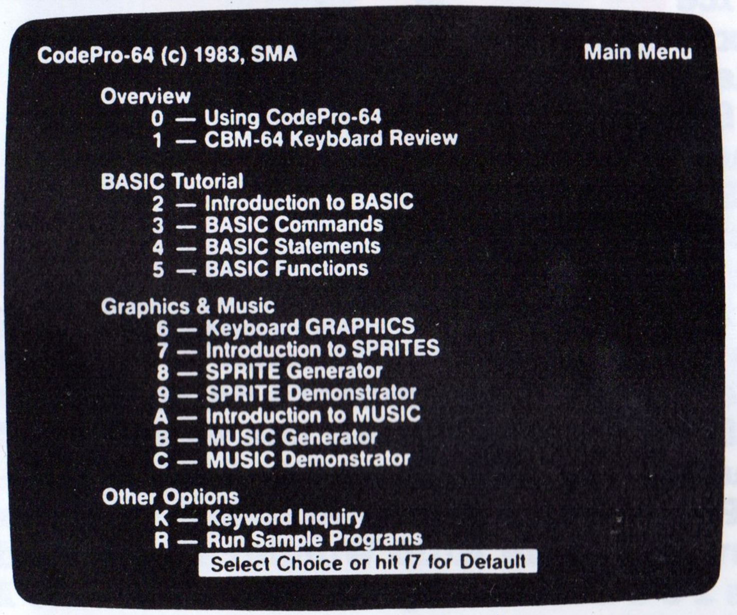
System Features

- Professionally designed and written programs
- 16 BASIC programs with source code provided
- Comprehensive 140 page User Reference Manual
- Program display screens cross-referenced to manual
- 2 diskette system, requires 1541 Disk Drive
- Executable SPRITE display sample program
- Sample SPRITE data file
- Executable MUSIC sample Program
- Sample MUSIC data file
- Automatic program loading from menu selections

User Features

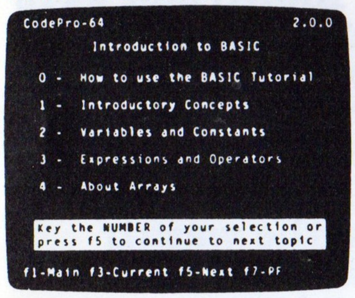
- Comprehensive BASIC tutorial
- All BASIC commands, statements and functions covered
- "BasicView" Graphics of statement execution
- Fully menu-driven, function keys used for menu selection
- Function keys used for scroll forward and backward
- Keyword selection for BASIC tutorial
- Fault tolerant, impossible to enter invalid data
- SPRITE Generator/editor for creating sprites
- SPRITE save, load to/from diskette
- SPRITE Demonstrator
- MUSIC Generator/editor music (99 notes per tune)
- MUSIC save, load to/from diskette
- MUSIC Demonstrator with keyboard organ

Main menu and 6 samples of the more than 150 screens available with this system.



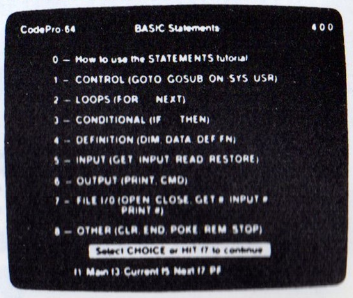
```
CodePro-64 (c) 1983, SMA                    Main Menu

    Overview
        0 — Using CodePro-64
        1 — CBM-64 Keybоard Review

    BASIC Tutorial
        2 — Introduction to BASIC
        3 — BASIC Commands
        4 — BASIC Statements
        5 — BASIC Functions

    Graphics & Music
        6 — Keyboard GRAPHICS
        7 — Introduction to SPRITES
        8 — SPRITE Generator
        9 — SPRITE Demonstrator
        A — Introduction to MUSIC
        B — MUSIC Generator
        C — MUSIC Demonstrator

    Other Options
        K — Keyword Inquiry
        R — Run Sample Programs
          Select Choice or hit f7 for Default
```

Introduction to BASIC



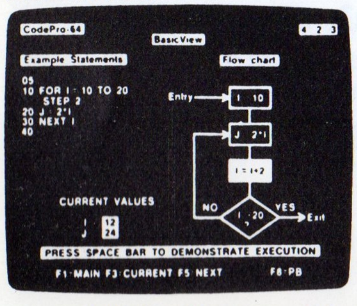
```
CodePro-64                          2.0.0
        Introduction to BASIC

  0 -  How to use the BASIC Tutorial
  1 -  Introductory Concepts
  2 -  Variables and Constants
  3 -  Expressions and Operators
  4 -  About Arrays

 Key the NUMBER of your selection or
 press f5 to continue to next topic

f1-Main f3-Current f5-Next f7-PF
```

BASIC Statements



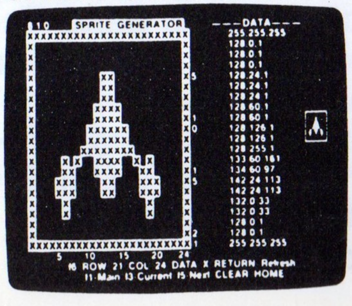
```
CodePro-64         BASIC Statements           4 0 0

    0 — How to use the STATEMENTS tutorial
    1 — CONTROL (GOTO GOSUB ON SYS USR)
    2 — LOOPS (FOR    NEXT)
    3 — CONDITIONAL (IF    THEN)
    4 — DEFINITION (DIM, DATA DEF FN)
    5 — INPUT (GET INPUT READ RESTORE)
    6 — OUTPUT (PRINT, CMD)
    7 — FILE I/O (OPEN CLOSE, GET # INPUT #
                  PRINT #)
    8 — OTHER (CLR, END, POKE REM STOP)
        Select CHOICE or HIT f7 to continue
     f1 Main f3 Current f5 Next f7 PF
```

BasicView



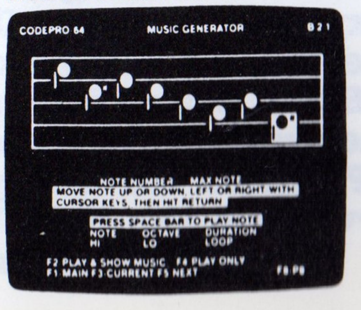
```
CodePro-64                                  4 2 3
                  BasicView
Example Statements              Flow chart

05
10 FOR I = 10 TO 20
   STEP 2
20 J = 2*I
30 NEXT I
40

     CURRENT VALUES
         I   12
         J   24
 PRESS SPACE BAR TO DEMONSTRATE EXECUTION
   F1 MAIN F3 CURRENT F5 NEXT          F8 PB
```

SPRITE Generator



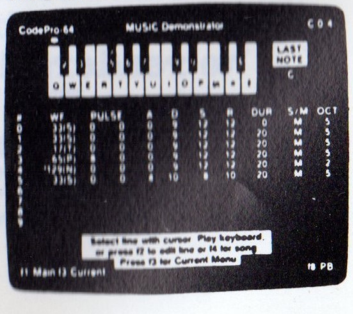
```
 1 0   SPRITE GENERATOR          ---DATA---
                                 255 255 255
                                 128 0 1
                                 128 0 1
                                 128 0 1
                                 128 24 1
                                 128 24 1
                                 128 24 1
                                 128 60 1
                                 128 60 1
                                 128 126 1
                                 128 126 1
                                 128 255 1
                                 133 60 161
                                 134 60 97
                                 142 24 113
                                 142 24 113
                                 132 0 33
                                 132 0 33
                                 128 0 1
                                 128 0 1
                                 255 255 255
   5    10    15    20    24
  f8 ROW 21 COL 24 DATA X RETURN Refresh
   f1 Main f3 Current f5 Next CLEAR HOME
```

MUSIC Generator



```
CODEPRO 64          MUSIC GENERATOR          B 2 1

        NOTE NUMBER     MAX NOTE
 MOVE NOTE UP OR DOWN, LEFT OR RIGHT WITH
 CURSOR KEYS, THEN HIT RETURN

      PRESS SPACE BAR TO PLAY NOTE
      NOTE       OCTAVE      DURATION
      HI         LO          LOOP

F2 PLAY & SHOW MUSIC   F4 PLAY ONLY
F1 MAIN F3 CURRENT F5 NEXT                F8 PB
```

MUSIC Demonstrator



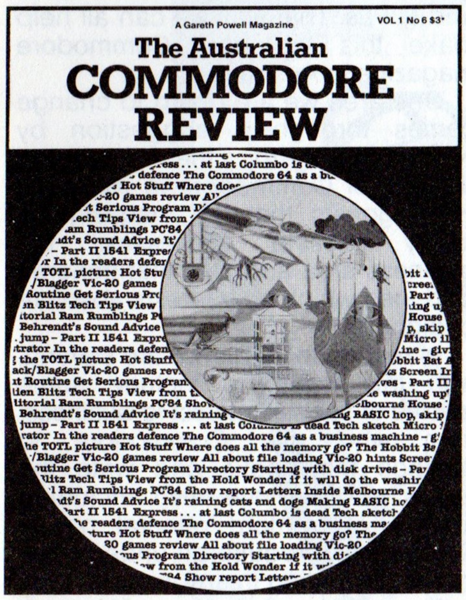
```
CodePro-64         MUSIC Demonstrator          C 0 4
                                            LAST
                                            NOTE
                                              C
     WF      PULSE     A    D    S    R   DUR  S/M  OCT
     33(S)   0    0    0    9    12   12   0    M    5
     33(S)   0    0    0    9    12   12   20   M    5
     17(T)   0    0    0    9    12   12   20   M    5
     65(P)   0    0    0    9    12   12   20   M    5
    129(N)   0    0    0    9    12   12   20   M    2
     33(S)   0    0    8    10   8    10   20   M    5

      Select line with cursor. Play keyboard,
      or press f2 to edit line or f4 for song
           Press f3 for Current Menu
f1 Main f3 Current                            f8 PB
```

𝔖.𝔓.𝔖. 𝔖𝔬𝔣𝔱𝔴𝔞𝔯𝔢 𝔓𝔯𝔬𝔡𝔲𝔠𝔱 𝔖𝔲𝔭𝔭𝔬𝔯𝔱 𝔓𝔱𝔶 𝔏𝔱𝔡
P.O. Box 229, CHATSWOOD, N.S.W. 2067. Phone: (02) 419 5879. Telex: 27642.

The Australian COMMODORE REVIEW

CONTENTS

2 Editorial
3 Ram Rumblings
5 PC'84 Show report
6 Letters
8 Inside Melbourne House
9 Bill Behrendt's Sound Advice
10 It's raining cats and dogs
11 Making BASIC hop, skip and jump – Part II
12 1541 Express . . . at last
13 Columbo is dead
14 Tech sketch Micro Illustrator
15 In the readers defence
16 The Commodore 64 as a business machine – giving the TOTL picture
20 Hot Stuff
22 Where does all the memory go?
23 The Hobbit
24 Bat Attack/Blagger
25 Vic-20 games review
26 All about file loading
28 Vic-20 hints
29 Screen Input Routine
30 Get Serious
32 Program Directory
33 Starting with disk drives – Part III
36 Alien Blitz
38 Tech Tips
39 View from the Hold
40 Wonder if it will do the washing up?



The Australian Commodore Review
Top Rear, 4 Carrington Rd,
Randwick, NSW 2031
Phone: (02) 398 5111

Published by Saturday Magazine Pty. Ltd.

Publisher: Gareth Powell

Editor: Andrew Farrell

Art Director: Roy Bisson
Production Manager: Sian Powell
Production Assistant: Gary Williams
Photographer: Gary Williams

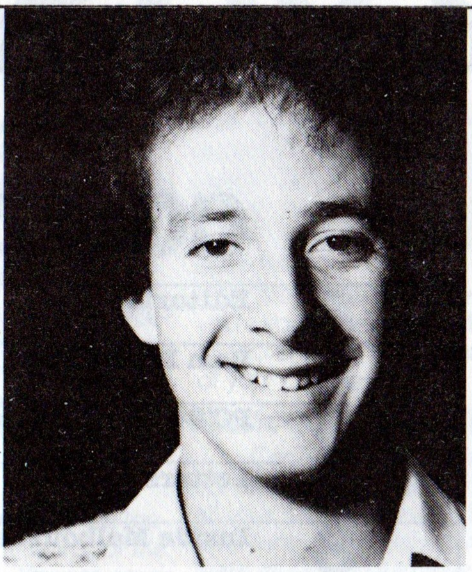
Advertising Enquiries:
Craig Hoskinson,
Top Rear, 4 Carrington Rd,
Randwick, NSW 2031
Phone: (02) 398 5111

Subscription Manager:
Tina Spathos
Phone: (02) 398 5111

Typesetting:
Printaction Pty. Ltd.

Distribution:
NETWORK

Cover photo:
Revenge of the Mutant Camels,
Distributed by Progressive Software.



This month we welcome several new writers to our ranks including a music teacher making his first attempt at entering the world of computing and an ex-MicroBee owner who has finally been converted. Our disk magazine is coming together but the flow of programs from you is barely a trickle.

How about it?

Written anything useful, a utility, game, anything?

Send it in. If it is good we will use it and we do pay.

It is coming up to that time of year again when everyone buys that once a year software purchase.

To help you choose better we are planning a huge comparison chart of all the top games and business programs. We also invite readers to send in any recommendations they may have for us to consider. There is every possibility we have overlooked a great game sitting quietly on the shelves of some remote computer store. Which brings me to another point.

We need your feedback.

So far I have had letters from people who cannot get the racing program to work and letters from hermits living in Yepoon who want to know where they can buy a program to count sheep.

What about the rest of you?

There are some 220,000 Commodore owners out there – would you all like to see more general stories, programming tips, programs, or more program reviews?

There is only one way to get what you want in the magazine. Write in and tell us. That way we can all help make this the best Commodore magazine in Australia.

One area we are going to change comes through a suggestion by Nigel Shepherd, the managing director of Commodore in Australia. He suggests that the magazine is totally missing addressing the business area and also that generalised area which so interests absolute beginners.

In response to this we are starting, with this issue, to include some articles of special interest to businessmen who use Commodore to balance the books and write the letters and we are also including articles of a more general nature which do not require an understanding of computing in any way.

If you enjoy them let us know. If not, likewise. □

DIRECT SUBSCRIPTION

The Australian Commodore Review

Please enrol me for ...... issues subscription to the Australian Commodore Review, commencing with the ...... issue. I enclose postal order/cheque for $ ....

Name: ....

Address: ....

....

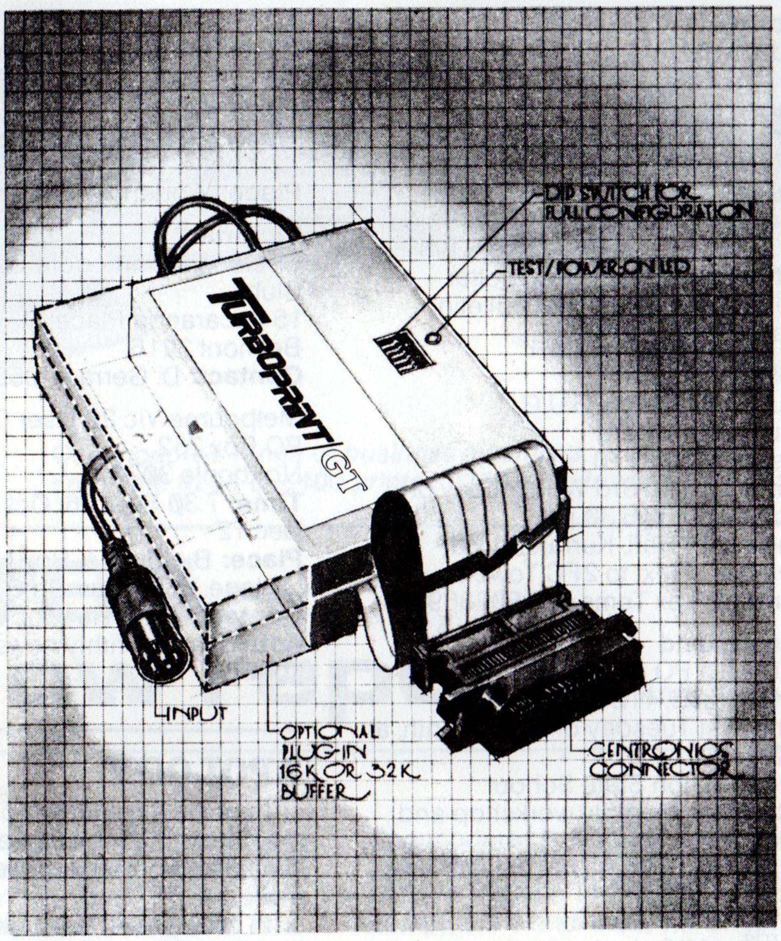
Postcode

(Please print in BLOCK letters)

Rates within Australia
Subscription rate for 6 issues $18.00
Subscription rate for 12 issues $36.00

The Australian Commodore Review
Top Rear, 4 Carrington Rd,
Randwick, NSW 2031

commodore


TURBOPRINT/GT
DIP SWITCH FOR FULL CONFIGURATION
TEST/POWER-ON LED
INPUT
OPTIONAL PLUG-IN 16K OR 32K BUFFER
CENTRONICS CONNECTOR

The Turboprint/GT from Interfaceware

OziSoft turbo charge

OziSoft are now using a turbo-cassette loader at the start of many of their top games. This move comes as no surprise after many other software companies made similar upgrades in recent months. Melbourne House are using the Pavlova fast loading system, whilst Progressive Software are using a version licensed from a German company.

Cassette users can no longer complain of back-ache or neck cramps from waiting for cassettes to load.

What about saving?

Melbourne house have gone one step further with a complete turbo-charged cassette operating system called ACOS. Full review coming soon.

VHF Modulator

Commodore 64 and SX-64 owners can now use a standard VHF television set for use as a monitor. The Computer Broker has released a VHF modulator which sells for $5Ø and plugs into the video-out port of either of the above machines.

Many SX owners were initially disappointed with the lack of a Modulator for use with a normal television set. The makers claim this one produces a very good picture and requires no modifications or nasty fixes. For further information call (Ø2) 525 94Ø6.

Apple Emulator

Early reports that an Apple emulator cartridge was available in the US have been confirmed. The unit sells for around $512 and turns your 1541 disk drive into an Apple drive and your Commodore 64 into an Apple.

Initially we were stuck for a good reason to own such a device.

Admittedly it is a cheap way to get an Apple – after all the plug in board must virtually be an Apple minus the memory.

Perhaps one day Apple owners will be able to buy a Commodore 64 emulator and really put their computers to good use. Watch out for more news soon.

Cybex getting the Goodies

Cybex Computing in Adelaide is now the Australian distributor for Richvale Telecommunications of Canada. The reason that is good news is RTC have some very interesting products which are now finally available in Australia – such as the 1541 Express reviewed in this issue.

Other items include the C64-Link (an IEEE interface with DOS 4.Ø), PLINK (a parallel printer interface), Color 8Ø (8Ø columns from software) and Super Copy (backs up a disk in seven minutes). Script 64 and Scratch Pad are also available complete with 8Ø columns, user dictionary and full compatibility with the C64 Link. For further information call (Ø8) 267 5855.

Interfaceware

As the name suggests, Interfaceware are interface specialists. They are starting the ball rolling with two competitive entries to the world of Commodore 64 add ons. Turboprint/GT is a high performance centronics printer interface with facilities for enhanced Commodore graphics, high-resolution screen dumps, line buffering and an optional 16k or 32K buffer. Dip switches also allow you to select between various printer types, listing mode, tag mode (whatever that is) and graphics mode.

It is said to be compatible with a large range of Centronics printers including Epson, Star, Gemini, Brother and C-Itoh. Interfaceware are also releasing an IEEE interface allowing the Commodore 64 to be used with the CBM 8Ø5Ø, 825Ø, 4Ø4Ø and 2Ø31 disk drives and most other IEEE peripherals. Not all software will be fully compatible with the unit, however certain popular word processing packages and filing systems will run with the unit.

For further information call (Ø2) 46-4374.

Vic 2Ø and C64 computer stand

Pittwater Distributors have released a computer stand to snuggly accomodate either the Vic 2Ø or Commodore 64 computer along with several peripherals. This lockable unit stands 84Ømm (w) x 9ØØmm (h) x 49Ømm (d) and comes in a teak finish. Optional castors are available to allow your entire system to be wheeled from room to room. This would be most useful in institutions. The keyboard is placed on a sliding panel and when not in use may be retracted back into the main unit. A 1541 or 154Ø disk drive may be housed in its own compartment and the monitor and printer are then placed on top. All cables are channelled through a tidy hole in the rear.

For further information call (Ø2) 939 2858.

User Group Grapevine

Ever wondered how you can get access to all that public domain software everyone keeps talking about? A good start is to join your local User Group. Where they meet and who to contact will now appear regularly in the following column. If you run a User Group, let us know the details, along with any special up and coming events.

Sydney
Sydcom 64
PO Box 856
Mona Vale, 21Ø3
Time: 2nd Tuesday of each month at 7.ØØ pm.
Place: YWCA (4th Floor)
Contact: Secretary, Michael Stead on 99337Ø (between 4 and 6 pm).
Activities: Monthly newsletter – "Peripheral", electronic bulletin board system.

NSW
Southern Districts C.U.G.
3 Lucille Crescent
Casula NSW 217Ø
Time: 1st and 3rd Wednesday of each month at 6.3Ø pm.
Place: API Hall, Kurrajong Rd, Prestons. (Next to 2FC Tower)
Contact: Lex Toms (Ø2) 6Ø28691

Queensland
CCUG (Q) PO Box 274
Springwood, 4127
Time: 1st Tuesday of each month, at 7.3Ø pm.
Place: Milton State School
Activities: Monthly workshop and newsletter.

Upshot Electronics
135 Abbott St
Cairns
Contact: P McCartney (Ø7Ø) 519455

ACT
Commodore User Group (ACT)
PO Box 599
Belconnen 2616
Time: 1st Monday of each month at 7.3Ø pm.
Place: Woden Town Centre Library.

Victoria
Geelong Commodore Computer Club
15 Jacaranda Place,
Belmont 3216.
Contact: D. Gerrard (Ø52) 442863

Melbourne Vic 2Ø User Group
PO Box 252
Northcote 3Ø7Ø
Time: 7.3Ø Sept 26, Oct 24, Nov 28, Dec 12
Place: Building 4, Box Hill TAFE College, 465 Elgar Rd Box Hill.
Contact: John Ruddock, Secretary
Activities: monthly newsletter, tape library. □

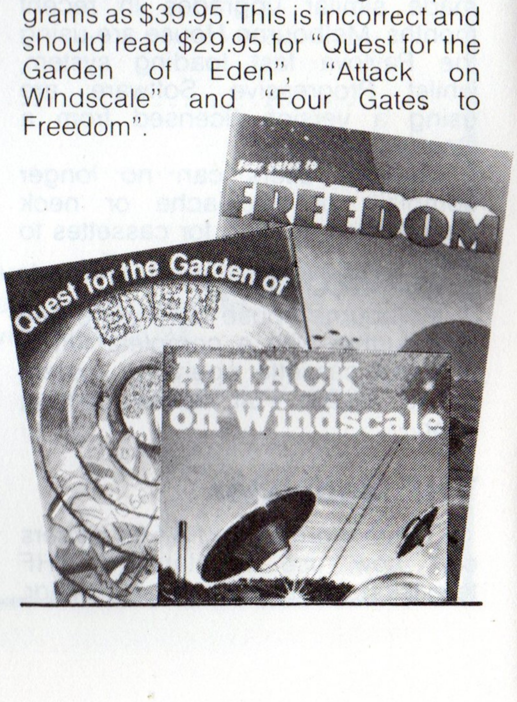
APOLOGY

In our last issue (No. 5) on page 20 we stated that the language cartridge "BC BASIC" was supplied by Micromail.

In fact it was supplied by INTERFACEWARE and enquiries can be made on (02) 46-4374.

APOLOGY

In our last issue (No. 5) on page 28 we stated the price of the "SOFTWARE CONCEPTS" twin cassette game programs as $39.95. This is incorrect and should read $29.95 for "Quest for the Garden of Eden", "Attack on Windscale" and "Four Gates to Freedom".



Commodore's range of business machines.
From left: 8296-D, 6400 printer, 8250 disk drive and the 8296.

PC'84 - Show report

by Andrew Farrell

Melbournes' World Trade Centre was a pleasant site for this years' PC 84 show held from July 18-21. Altogether nine publishers, 74 hardware and software distributors, and one lone retailer exhibited their works on three levels of glitter and glamour.

Commodore's Offering

Commodore was well represented with a fair slab of their Sydney and Melbourne staff busy holding the front-line. Some manned a well decorated stand complete with a collage of advertising from around the world and a cooling fountain in one corner. David Harvey was said to have been caught bathing in it on the last day after complaining of heat exhaustion and exposure. Meanwhile, others comforted confused buyers in the classroom situated upstairs providing much needed on-hands experience. The show also heralded the official release of their PLUS/4 machine (otherwise known as the 264) and the less familiar Commodore 16. Both computers have been surprisingly under-rated by some people as "nothing amazingly new" or "little to get excited about". Nigel Shepherd, Managing Director of Commodore Australia, feels otherwise. With the Vic 2Ø and Commodore 64 now the two top selling personal computers world wide, Mr Shepherd hopes that by the end of this year the PLUS/4 and Commodore 16 will help them collect all four top positions.

Predictions aside, what do these computers really have going for them that could possibly fill the dreams of an aspiring Commodore Manager?

Both the Vic 2Ø and Commodore 64 have had their fair share of criticism from dealers and users. Educational bodies used to complain about the availability of software to suit their needs. Business users often groan about the speed of the 1541 disk drive – doubtless they will be reading our review of the 1541 Express elsewhere in this issue.

Beginners continue to flounder in a mess of PEEKs and POKEs when using graphics or sound. Programmers would have liked a few programming aids built in – everyone would have liked to pay less.

So there it is. A short list of problems with existing Commodore machines. Perhaps they could be viewed as criteria for consideration before designing a new machine. Perhaps they already have.

Both the Commodore 16 and the PLUS/4 have an extended BASIC which supports the extensive colour graphics, several disk operations and a few programming aids. There is a HELP key – invaluable to the raw beginner. Educational software can be built into the PLUS/4 with menu driven selection of the program you wish to use. Commodore expects a fast dual drive to appear for the PLUS/4 although initially it will use the existing 1541. Business software is included with the PLUS/4, in fact there are four packages available as soon as you power up. A reset switch has been added. Cartridges from the 16 are upwardly compatible with the PLUS/4. The PLUS/4 has a better keyboard similar to the SX-64 with separate cursor keys.

These are all simple improvements or additions. Various other features are an added bonus. Surely then, only one question remains. The Commodore 16 should sell for $199 and the PLUS/4 for around $599.

These machines are, in my opinion, the answer to many problems buyers have previously encountered. Once they are available the Vic 2Ø will fade from existence. The Commodore 64 will be a harder bird to kill and should be around for a long while yet as a top selling personal computer. So there will be a gap created by the Vic 2Ø's absence and a gap never fully filled by the Commodore 64. These gaps are full of low end beginners and the more serious business and professional users including schools. This is where the PLUS/4 and Commodore 16 should succeed. In fact if no-one else jumps in with anything amazing Mr Shepherd may well be right.

In the business area Commodore had a new daisy wheel printer (reviewed last month) and the 8296-D which features an inbuilt dual disk drive. The cabinet follows Porsche lines although it definately isn't a substitute.

Melbourne House

Australian software houses are few and far between, especially on the scale of Melbourne House. Apart from creating their own software they also import games from Anirog and several other UK companies. At PC-84 it was obvious Commodore was not their only line of business,

The Commodore 16, the new Vic-20?

with programs for almost everything that sells for under $1ØØØ.

I managed to have a quick play on one of their newer offerings called Space Pilot. Anyone who has ever played the arcade game Time Pilot will recognize it straight away. How they managed to get all those clouds, plains, jets, missiles and bullets to move all at once never ceases to amaze me. Other titles expected soon include Hexpert – a three dimensional jumping game, Jungle Drums – an arcade type adventure with great colour graphics and Cybotron which has something to do with robots.

Interfaceware

Amongst all the confusion of IBM PC lookalikes I just managed to find a new company on one of the higher levels who had some of the most useful pieces of hardware I've seen for some time. Most of you are probably sick of hearing me complain about disk drives. Until Commodore rectify the problem by providing an alternative to the existing drive I probably won't shut up. There are however several alternatives now being offered by the crowd I mentioned a couple of sentences back.

At the moment the only problem with the alternative is getting it. Demand is far exceeding supply as could well be imagined. Well, what have they·got? For the sceptical buyer who refuses to buy a 1541 the MSD single disk drive will be an easy sale.

It boasts greater reliability, an improved drive door and best of all, an IEEE port. For the business minded, a dual version is also available which mirrors closely the capabilities of Commodores recently discontinued 4Ø4Ø disk drive. The unit is very compact and yet retains a solid look about it.

Gareth (I need 8Ø columns) Powell will like this next one. The Data 8Ø cartridge provides 8Ø columns of readable text and is supported by a wordprocessor and spreadsheet which will be included in the deal. Interfaceware have also started importing an RS-232 interface which may well replace the one Commodore seem surprisingly short of at present.

Next year

Several big guns were missing from the Melbourne show including Imagineering and OziSoft. After speaking to Kevin Bermeister, Managing Director of OziSoft, I got the impression they no longer believed that exhibitions are worth their while.

Several other exhibitors have also expressed concern at the high cost of exhibiting coupled with the fact that people pay to gain entry. □

LETTERS

Dear Sir,

I do not own a computer, but since the appearance of your magazine – and largely because of the appearance of your magazine – I was contemplating the purchase of a Commodore 64.

Was contemplating, that is, until I read three very interesting letters in your issue no 2, in April. The first letter gave information on the gross unreliability of the 1541 disk drive, the second complained of the disk drive and complained also of the excessively slow servicing of his 1526 printer, while the third letter – from a dealer – really emptied the bucket over Commodore management. It seems that they have a good computer, terribly designed peripherals, and no idea whatsoever how to run a service department.

Your own editorial comment at the end of all this said that you were presenting these and many similar letters to Commodore management, and that they might or might not care to comment.

In the two issues since then there has been no further word on this extremely important subject. I (and I'm sure many other readers) have been waiting with bated breath for all that time, so please tell us – have Commodore simply refused to comment? Don't they care at all?

David Aron
Collaroy Plateau, NSW

Commodore has since rectified the problem with servicing their computers by streamlining their Sydney operation, and by setting up service departments in other states. The initial problem was due to the huge volume of computers sold over the Christmas period, which the service department was not initially set up to service.

As far as peripherals go, although it is fairly obvious problems do exist with some add-ons, other manufacturers provide very adequate alternatives such as the Single Floppy Disk drive from Interfaceware. Regarding printers see our article called "Printing On" in issue ± 4 page 14. – Ed. □

19 brand new brain busters for the Commodore 64.

Just $11.95*

These 19 Commodore 64 games are all new, all Australian, all original. From the fiendish brains of Australia's most brilliant young programmers. All games have multiple levels of difficulty, to keep you on your toes.

Programming instructions are easy and clear. Each game has been created specifically for the Commodore 64. Each makes the most of the 64's powerful colour graphics and sound capabilities. You will:

★ defeat dastardly alien scum
★ ski impossible slopes
★ blast your way through asteroid clouds
★ battle ghosts in graveyards
★ rule your own Roman Empire
★ win the Battle of Britain (you hope!)
★ doubt your IQ trying to win at noughts and crosses

and you'll never be the same again.

There are more where these came from: the "Games Commodore 64 Computers Play" book. If you could buy these games commercially you'd part with the best part of a thousand dollars. Our way costs a lot less. And it's a lot more fun. Even if it does scramble your brain!

* Recommended retail price.

Addison -Wesley

At the better bookshops and computer stores.

(Bookshop enquiries to Gordon & Gotch. Computer store enquiries to Computer Reporting Services.)

Inside Melbourne House

by Andrew Farrell

In recent months much dilemma has surrounded the Australian software industry at all levels. The copyright issue was being resolved in the High Court. One of the many statements made by groups such as Software Liberation was that software was overpriced, overprotected and generally not readily available to the right people. They also believe the Australian software industry is virtually non-existent.

So who is there on this bare island producing software? Ever played the Hobbit? Or perhaps you've read a book called the Commodore 64 exposed. Believe it or not, both were produced in Australia by a software house who, until recently, have been keeping a low profile locally. I for one was surprised to find they do in fact have a whole collection of programmers working for them – as rumour had it. Well, let's take a look inside.

Inside

After a brief struggle locating the right street we eventually arrived at the front entry. One wall was decorated with their huge range of software imported from overseas and produced locally. Very impressive. But are there really any programmers in here? Come this way, sir.

Scattered across the open room were a multitude of machines including Apples, Commodores, BBC's and Spectrums. All were manned by an interesting collection of University Computer Science graduates and several graphic artists hunched over graphic tablets. It must take a certain type of person to work in such an environment. Someone suitably qualified, dedicated to producing software of a marketable quality. To get a better idea of what's involved we interviewed Greg Barnet who is in charge of Melbourne House's Commodore section.

After 16 months spent with the company, Greg is now in charge of five other full-time programmers and several part-timers. I immediately expected to hear that everyone worked as a team. Not yet. "At present each programmer works on his own project. However, team work is now being implemented on certain large jobs which require the skills of a range of people".

"On smaller programs we do employ the skills of graphic designers and musicians – but each programmer is still in charge of his own work". By the sounds of things, Greg has all the help he'll ever need, but surely it must be difficult trying to coordinate the efforts of several people along one line of thought?

"We've found that by discussing ideas and passing ideas on to others we are able to solve difficult problems. But as far as simple games are concerned, it's better to have just one programmer in charge who can get the graphics from the graphics designer and music from the musician".

"Especially now as the expected quality of graphics and sound is much higher it is important to get the best possible results." You must all be wondering what amazing program they use to design graphics and sound. Greg believes that as yet there is nothing much in the way of good commercial programs which can be used to design graphics effectively. Often they must write their own utilities programs to create the desired graphics or sound. "Often the required design may be made on another computer such as the Spectrum and then transferred to the Commodore for use at a later date. Many of our programs were not originally written on the Commodore so this goes on quite a bit".

At the moment Greg is in the final stages of completing a Sound Synthesiser which will help design music or sounds. Anyone who has played Horace goes Skiing will also have seen his name on the credits for this program, which was originally designed on the Sinclair Spectrum.

What language do they use? BASIC will not suffice – except perhaps for the games included in many of their books. "Assembly language is the only way to write a good fast game." At this point I could hear Gareth Powell screaming from Sydney that there is a better way – you see, Andrew, there is this great language called "C" which lets you write in understandable English that compiles into full machine code.

Compiles? I thought you might ask. A compiler is something which changes a high level language into machine language which is the true tongue of all computers. Most compilers aren't very efficient and as a result they are not as fast as a true machine code program. But, according to the irreverent Mr Powell, this one does and it's transportable.

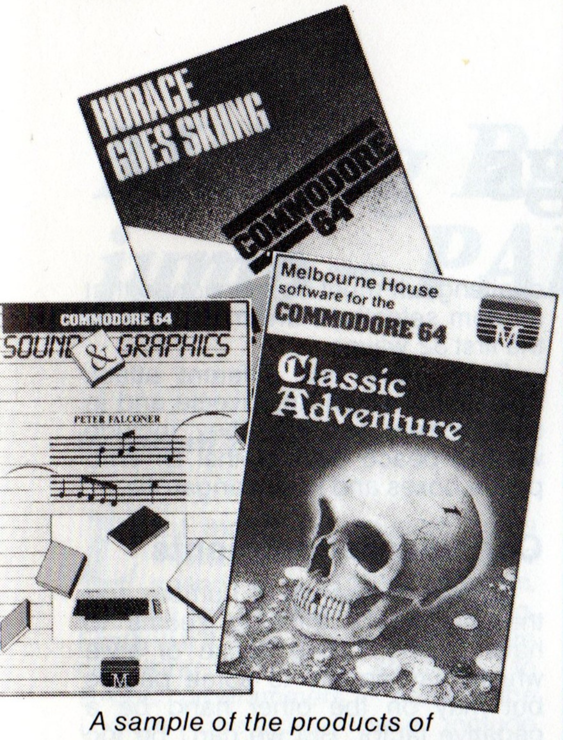
What – you mean it's not glued to the side of your computer? No, you can take it from one machine and stick it in another. Sounds nice. What does Greg have to say about all this?

"We do use 'C' for a lot of our utility programs, but it is definately not the answer to all your problems. 'C' is a big step forward from most other compilers but it still does not replace machine code programming. Being transportable has made it useful in getting a simple program to run on any computer, but that's about where it ends."

How disappointing. We are still using Assembly language after all that. How long does an average game take to complete? "There are two areas of programming going on here. Simple programs for our books might take two or three days. With an average arcade game we aim for three months. Often it takes much longer. The Hobbit, which was a team effort involving three people, took almost two years to complete. I later worked on the conversion to the Commodore 64".

Where can you start?

With so many budding young programmers out there, I asked Greg what he would recommend they do to be able to write good software. Greg has been playing with machine code for close to two and a half years now, so he has a slight edge on most of us.



A sample of the products of Melbourne House

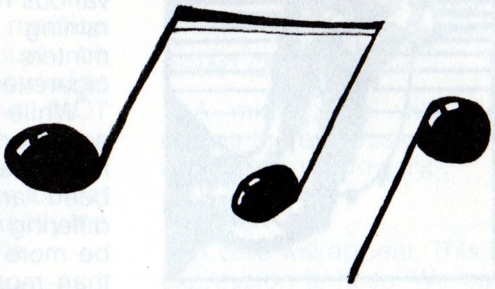
"I started at home. I first had an Atari and later the Commodore 64 and that's where I learnt machine code. I've also done a computer science degree at University, although that alone didn't get me this job here. Really, it was because I had experience with 65Ø2 machine language."

Well, now you can program, how do you market your software? Greg was lucky in getting a permanent job as he prefers the security of a wage each week rather than working on royalties. For most people the royalty is a good way to go. Melbourne House usually work on around ten percent of the wholesale price for each program sold. In dollars and cents that means you might get about $1.5Ø per program. If you are set on becoming part of a team, Greg stressed that many important programming principles are covered at the University courses.

Future of Games

Many experts believe people will soon get sick of buying the type of games which are available now. In the US, the current trend is for simulations, games which require a lot more use of the mind and less random action. Greg believes that Melbourne House will soon be moving into this area by taking their existing expertise with adventure games a little further and adding three dimensional moving graphics. □

Bill Behrendt's Sound Advice



A review of his book "Music and Sound for the Commodore 64"

Recommended retail price $23.95

According to the introduction, this book is designed to link the three worlds of music, synthesisers and computers. As you look into the book you'll find he does this with enthusiasm and zest. So prepare yourself for one fast moving book. He also makes you wonder what his first love is, because he has covered each area equally well.

Music

Bill has given us information that ranges from the basics of what sound is (scientifically) to some of the finer points of reading written music. I found particularly interesting the topic of sound timbre which refers to the actual qualities of the sound. A flute has a very pure timbre while a saxophone has a dirty, distorted timbre.

Also worth a mention is the way in which written music theory is explained . . normally a real killer subject to understand.

Synthesising sound

It becomes obvious we are being aided to understand things in a certain pattern. Firstly he establishes what music is, then proceeds to show us how to synthesise or electronically produce a duplication of the sound. And yes! . . . both analog voltage controlled synthesisers and digital synthesisers are covered, for all those electronic music buff's out there!

Explaining how each part of a sound is synthesised is a long winded affair, so instead we are given examples with accompanying explanations, in the hope we get an idea of what's going on through repetition and practical experience.

The 64 and sound

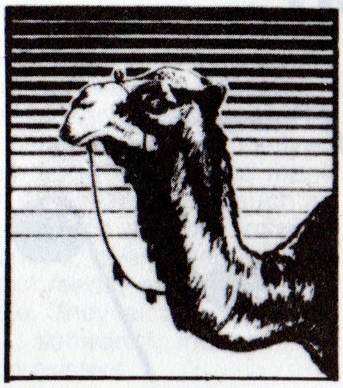
The next step is to get a computer to control the synthesiser. We are helped to see how to link notes together to make a melody, to arrange notes in harmonies and even compose or write music on our 64. This area shows just how flexible the 64 is. To help us use what we have learned, an appendix of programs and preset sounds takes up the last quarter of the book. For those of us who would like to take an easier path, the disk for this section will be available in October.

Conclusion

As I have said before, I feel Bill moves a bit fast in some sections of the book. The claim that the information will appeal to musician, synthesiser player and user is fairly well grounded, although I feel the main advantage of this book lies with people who are interested in the basics of all three fields.

Available through Prentice-Hall Brookvale. □

It's raining cats and dogs



Review of "Revenge of the mutant camels" – by Ric Richardson

For those of us who enjoy 15 minutes of warming up and thumb stretching excercise while waiting for a game to LOAD, you're about to be disappointed. In fact you'll hardly get a chance to find your joystick and connect it before Revenge is ready to play. This is all due to the amazing turbo loading facility which makes a cassette program load at the same speed as disk.

Don't value the quiet solitude of a blank screen either. The introductory comments and credits give us an opportunity to get used to the game. These appear at the bottom of your screen much as the Lotto results interrupt your favourite Clint Eastwood movie. There is a special credit for the young man who wrote the music for the game and this is well deserved as I will explain later.

Of Camels and Kangaroos

To the left of the screen is "our camel" walking on legs that have a realistic stride. He walks along as the background scrolls from right to left at a refreshingly fast pace. A major feature of the design of the background is Jeff's attempt at giving us the impression of perspective, with the camel placed in the foreground while at horizon level we have the different landscapes. These include mountain ranges, space-age cities that feature futuristically coloured buildings and pyramids with winking heiroglyphic eyes. This all takes place under a star filled sky. Meanwhile you are attacked by various nasties including giant birds, raining cats and dogs, maniac minters and even toilets and cigarettes.

While your eyes are feasting, your ears aren't doing so bad either. There is quite a selection of tunes that have been arranged in two voices, in differing harmonies. These seem to be more musical and less obtrusive than most, although sometimes you can hear a few off-putting time changes.

Starting and We're Off.

By using the joystick we can get the camel to squat, move forward, stop in his tracks and even "jump" (something you have to get used to.) These functions are all smooth and realistic. The joystick also controls the direction in which we fire, which is a bit cumbersome as it would be good if we could manoeuvre and fire in different directions. Another surprise! For once "our man" can take a bit of a beating. The only problem is that it's easy to lose track of how much damage he has taken. You must remember to keep an eye on your Nuetronium Shield levels.

But as the saying goes "all good things must come to an end" – so the time soon comes when our fifth camel passes away and we must start again.

Attack Waves

Most of our attackers bounce across the screen in three to five bounds – a new challenge which gives making a target of them a rather hard task. In all there are 42 attack waves, which is to date (I think) the largest number in a game of this type.

Selection of these attack waves can be done progressively or at random. This is good for the novice and experienced player alike. Firstly, it gives a beginner the chance to find out what the characters are like in the higher waves and secondly it gives the expert a continuing challenge. It must be mentioned that random selections are made only in the first 32 waves.

Each wave has different attack characters, which are diverse and in some cases twisted! They include warring peace signs, lethal Dr Who police boxes and killer kangaroos.

Conclusive comments

Overall, the sense of humour and the manner of thinking seems to have a very strong sense of style which may appeal to some people but may on the other hand be a negative factor. But we can't be too hard on him – after all Jeff did include killer kangaroos to keep his Aussie fans happy. Credit must go to him for the amount of imagination that went into the concept of the game, the music and the use of perspective in the games graphic form.

I think it's definately worth having a look at. Our review copy came from Progressive – the Australian distributor for Llamasoft. Revenge sells for around $32 on cassette. □

Making BASIC Hop, skip and jump – PART II

by Andrew Farrell

Last month we covered some interesting ways to improve the speed of a BASIC program. Here's a few more to keep you going. Firstly we'll look at one of the more confusing functions available to BASIC programmers – namely the Logical Operators. These include AND, OR and NOT.

You'll often see them in the IF THEN statement. Used properly they can save a lot of space and execution time. A complex decision based on several variables can be made without too much fuss. Here's an example:-

```
1Ø INPUT "Start ";s
2Ø INPUT "End  ";e
3Ø If (S>E) or (S>1) or (E>100) then
stop
4Ø PRINT "The rest of the program"
```

Nice and simple. We collected two variables 'S' and 'E'. Then in line 3Ø we make sure that 'S' and 'E' do not exceed the limitations placed on them and that 'S' is not greater than 'E'. A typical use of the above statement may be in a database when you're asking for the range of records to delete or copy. We wouldn't want a value higher than the last record or lower than the first. Also the start of the range would have to be less than the end.

In the above case, use of the 'OR' function within an IF THEN statement was fine. Now let's look at another example:-

```
1Ø GET R$: IF R$=""THEN 1Ø
2Ø IF R$="(crs left)" THEN X=X+1
3Ø IF R$="(crs right)" THEN X=X-1
4Ø IF R$="(crs down)" THEN Y=Y+1
5Ø IF R$="(crs up)" THEN Y=Y-1
6Ø IF X>20 then X=1
7Ø IF X<1 THEN X=2Ø
8Ø IF Y>23 then Y=1
9Ø IF Y<1 THEN Y=23
1ØØ POKE SC+X+(Y*4Ø),81
11Ø GOTO 1Ø
```

Very messy, not to mention frightfully slow. There must be another way. With a bit of knowledge about the AND, OR and NOT statement you could probably narrow things down a bit. But there's an even better way. Have a close look at the next program – it does exactly the same as the previous example:-

```
1Ø POKE 198,Ø: WAIT 198,1: GET
R$
2Ø X=X+(R$="(crs left)")-(R$="(crs
right)")
3Ø Y=Y+(R$="(crs up)")-(R$="(crs
down)")
4Ø X=X+(20*(X<1))+(20*(X>20))
5Ø Y=Y+(23*(Y<1))+(20*(Y>23))
6Ø POKE SC+X+(Y*4Ø),81
7Ø GOTO 1Ø
```

As you can see it's a whole lot shorter than our first attempt – well it should move a ball around the screen. Of course we would also have to add another line to make sure our ball is visible but let's not worry about that.

Line 1Ø is different. POKE 198,Ø says make sure there are no characters in the keyboard buffer. This stops any nasty accidental key presses slipping through. WAIT 198,1 means wait until location 198 contains the value one. Remember location 198 is the place where our computer stores the number of characters in the keyboard buffer waiting to be processed by your program or the operating system.

Next we grab a character just like normal with GET R$ Now's where the fun begins. I've scrapped all those IF statements and replaced them with a much faster statement which uses the results of relationship tests to decide what to do. Mr Powell is about to perform a cranium removal . . . be patient – I was just about to explain.

Any given comparison, whether it is performed by a relationship operators such as '<', '>', '=' or a boolean operator such as 'AND' or 'OR', produces an arithmetic true/false result. Try the following:-

```
PRINT (2=2)
```

A minus one will appear. This means the expression was evaluated as being true. Now try:-

```
PRINT (2=3)
```

A zero will appear. This means the expression is false. We can use this evaluation in programs. In fact, the above example uses plenty of them which is why we are so confused at this very moment. The evaluation may also take place in string comparisons such as line 1Ø. Try this:-

```
R$="FRED"
PRINT (R$="FRED")
```

The expression is true so the result is minus one. Thus in line 1Ø X=X+ the result of the first expression minus the result of the second. Now try this:-

```
R$="(crs left)"
X=5
X=X-(R$="(crs left)")+(R$="
(crs right)")
PRINT X
```

Since R$ is equal to cursor left X will be equal to X-(-1). Don't forget a positive minus a negative is a positive. So X=X+1. If none of that makes sense, one of us is failing miserably. Yes Mr Powell, I know it's probably me so I'll keep trying.

Line 2Ø modifies the Y value in much the same way as line 1Ø does to the X value. Lines 3Ø-4Ø check for illegal values and causes the ball to wrap around if it does exceed any of the limitations set. Line 6Ø sticks the ball on the screen. Now for line 65:-

```
65 POKE 55296+X+(Y*4),1
```

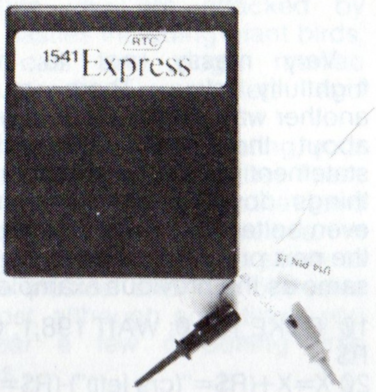
and line 5:-

```
5 X=1:Y=1
```

Now that you've got the whole program, type it in and try it out. For a fair comparison of the speed differences run the first program just for fun. □

1541 Express . . . at last

by Andrew Farrell



Complain no more. If it's already working, the 1541 Express will make it slave up three times faster. Yes folks, not a moment too soon, the 1541 can be put to work at a speed which is more acceptable to those of us used to a real disk drive.

I could end this entire review right now, but I suppose you want to know a little more about it first. It's not a crank you connect to the side of your 1541 and you won't need a soldering iron to connect it. You will need to be able to count to eight and have reasonable eyesight. Apart from that, it works great – even with Easy Script.

When the Express first arrived at our office some were a little sceptical that a cartridge with two wires hanging out could possibly be so useful. Don't be too intimidated by the wires either, unless your computer is still under warranty. One connects to the 651Ø chip whilst the other goes to some other relatively unheard of beast, called, for anyone who is dying to know, the "74LS258".

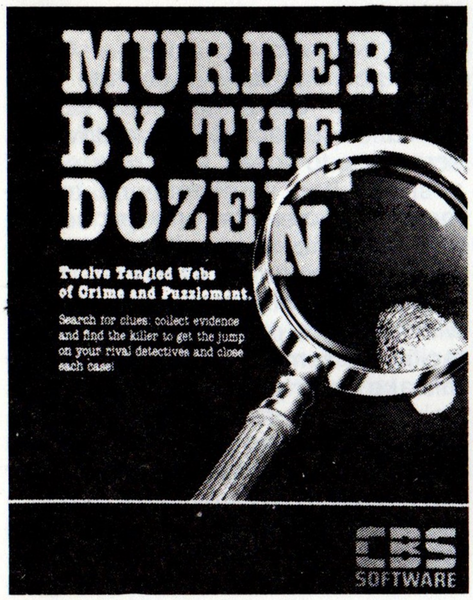
The wires connect using a special shielded hook which literally grabs hold of one of the I.C. legs and stays comfortably put during earthquakes and other such natural disasters. When you power up, the 64 responds with a message to indicate all is well, and the 1541 Express is ready to go in Fast mode.

Two modes are provided to aid compatibility with other peripherals such as printers which may affect the operation of the Express. Pressing CTRL and Function One switch to fast mode – CTRL and Function three switches to slow mode. These keys remain active even once commercial software is operational. The Express Cartridge further modifies the operation of your 64 by making the Shift-Run Stop key produce the equivalent to LOAD"O:*",8. This is a useful improvement also implemented on larger CBM machines.

A fast cable is also available which increases the speed of operation even further and also increases compatibility with serial printers.

The 1541 Express is a very useful, much needed cartridge which should sell plenty for around $99. Our review copy came from Cybex Computers who are the Australian distributors for Richvale Communications of Canada. For further information call (Ø8) 267 5855. □

Columbo is dead


MURDER BY THE DOZEN
Twelve Tangled Webs of Crime and Puzzlement.
Search for clues: collect evidence and find the killer to get the jump on your rival detectives and close each case!
CBS SOFTWARE

Review of the new CBS game "Murder by the Dozen" by Ric Richardson

Would you miss a whole hour of time on your Commodore 64 to catch the latest episode of Banecek or Columbo? Well, now you do not have to leave your Commodore at all. CBS has made sure of that with this well thought out little package. Let us, as they say in the television thrillers, case it out.

The Kit

The black box arrived on my desk at 4.ØØ pm sharp. After ultra-scanning (for explosives) the contents were investigated. They included: detective manual, mystery master worksheet pad and most conspicuously, an envelope marked "Confidential".

This envelope proved to contain a booklet of 7ØØ clues and an encoded solution booklet for later reference. For anyone following this case, it would do you well to make sure to read, thoroughly, the detective manual if you decide to get a black box like this. It is here that a great many mysteries unravel. The first being how to load the program which is the only simple part of the whole operation.

Mission near Impossible

The mission, should you accept this assignment, is not as obvious as you would think. The game is not solved by entering the name of a suspect to see if it is right. No – you have to do the same amount of work as a real detective to get a correct solution. Not only do you have to name the murderer, but you also have to give motives and hard facts, enough to get him convicted.

On top of this, your objective as a player is to do all this in the shortest time without sacrificing thoroughness in your investigation. The computer has an inbuilt game-clock that keeps track of the "time" you take to interview suspects and witnesses, search for clues or travel from location to location.

Although this is a text game, it has some interesting graphics that act as spacers between sections of the game. You are also frequently given the opportunity to look at Micropolis (the city of intrigue) from an overview, to work out your options for locations to investigate and also calculate an efficient way to travel and save time.

Enter the Mystery

Forward into the web of shady characters, false leads, descriptions and twisted trails. Soon after the disk is loaded you are presented with your master case selection. An assortment of 12 cases with accompanying pictograms. Although each case is equally complicated there are certain factors in the later cases that need to have been introduced earlier; like arranging yourself to spend time efficiently until an autopsy or lab report is ready.

The selection includes; hit and run victims, shootings, hangings and even an 82 year old socialite who was beaten to death with a hospital bedpan. After the case number is entered the phone rings at Homicide. The information (or case history) of your selection is then displayed. The mad rush to take notes, mark locations and develop solutions begins now. If you miss any information it is readily available in your manual.

Next you enter the number playing and their names, then you are posted at the first location, usually the scene of the crime. You are presented with seven choices. The first three being opportunities to interview people at this location. The next three involve examining physical evidence, searching rooms and the like.

The last choice is the option to go to another location which upon selection displays the map of the city if required. If you select any of the first six choices you will be given clue numbers that correlate to the numbers in your clue booklet. Each clue number refers to three minutes or five minutes respectively depending on whether you get them by interview, or searching a room for example. How many clues and locations will you need?

Well, I took three and a half hours of game time to get a right answer and even then I got rated as a "Befuddled Detective". The idea is that you get up enough information to prove your case then you privately check your answer. If you are wrong you are out of the game. If right you are given a sleuth rating. Be careful though – not having a full answer can rob you of the satisfaction of winning properly. Also it proves that "crime doesn't pay" because any attempt at cheating only backfires and leaves you without a challenge. Remember, it is the challenge not the solution that is the most important thing about the game.

▷

Conclusions

The impression you get is that the more you put into this game the more you get out of it. The game is constantly leading you. Never once do you feel as if you are on top. On recollection it is obvious there is a lot of ingenuity built in and also an intentional effort to make the user work to enjoy himself.

The only negative thing I have to say about the general rules of play is that they really do not present enough of a deterrent to help you from subconsciously or intentionally cheating. For example, when you think you have the answer to the murder the only thing stopping that deluge of curiosity is the task of opening a booklet and checking.

I would be more inclined to have the computer control access to the solutions. Maybe by a system of questioning that will either pronounce you successful or unsuccessful. I think the game would be a lot more fun if you played it with someone else or even the maximum four players possible. Also I found each section of the game a bit too streamlined. By this, I mean much of the superfluous things we have gotten used to have been done away with. We are left with a very to-the-point arrangement with a minimum of graphics.

On the other hand the thought given to the accuracy of the instructions, the concept of the game and the imaginative characters we meet, are exceptional. I wonder if Robert Ludlum had anything to do with it. When you have finished the game you feel as if you have been solving a real murder. The tension of being efficient with your time and being accurate in your observations, of thinking clearly and being decisive has been well designed into the game.

Finally I must warn you, if you decide to try and tackle this assignment the government will renounce all responsibility for you, your team or its activities. Good luck. – This article will self destruct in three seconds (if lit).

Imagination	*****
Concept	****
Graphics	***
Presentation	****

□

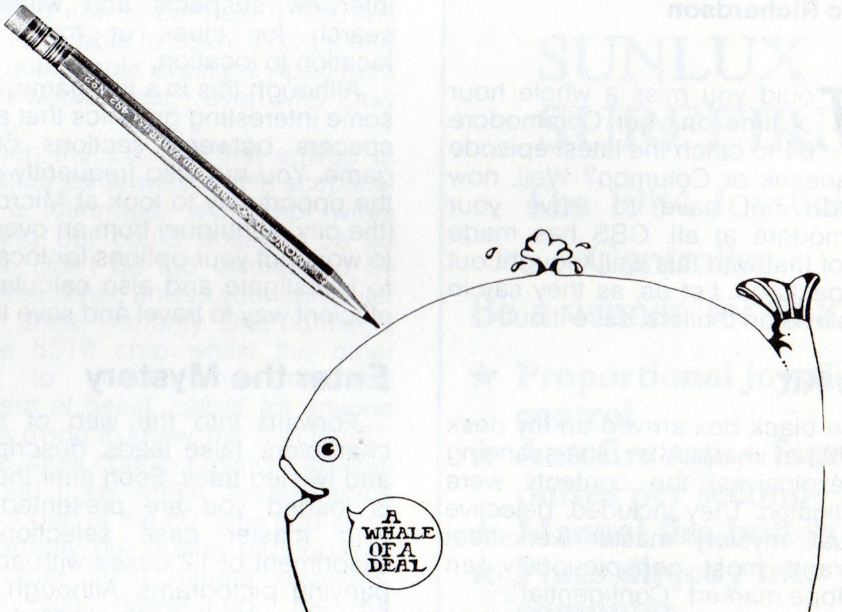
Tech sketch Micro Illustrator

You don't have to be a three year old, an artist or even an engineer to use one. The Micro Illustrator tech sketch is now available. A light pen along with some great software turns your home computer screen into a video canvas.

To start with, you are confronted with the menu. This consists of 16 different options ranging from storage and retrieval to a selection of lines, rays, circles or frames. We can also choose any one of 12 different brush sizes and eventually end up choosing any one combination of the 256 available colours. It's just like dipping a brush into a palette of paint.

Now we have our brush and paint we hit the space bar and we're onto our electronic canvas. A small cross shows where we will draw next. The light pen is fairly accurate with adjustable centering and varying inertia. Once your picture is down on screen, so to speak, with the touch of a space bar, you are back to the menu. You see the storage square and with the touch of your magic wand your picture is saved.

For those who are not too handy with the light pen there is an erase facility to remove your last ghastly mistake or any part of your picture. Using the Micro-Illustrator you can do basically two things – draw and paint. You use them to select your menu options and to create amazing graphics. The lightpen plugs into joyport two of the Commodore 64, and the Micro-Illustrator loads from disk. Our review copy came from OziSoft, the Australian distributors of the package, and retails for $89.95. □



In the readers defence

by Ric Richardson

Not an article about computers

This article has been written "in the readers' defence" because I feel the great majority of people out there who read this magazine are, like myself, not computer naturals.

People who see computers as complex, and involved. People who seem resigned to merely playing games or using word processors. Well, I sympathise with your case, but I would like to encourage you to think about two points that may change your attitude.

To illustrate this point, please think about this: how many people do you know that cannot do minor repairs on their cars? How many people you know wait for the NRMA because they don't know how to clutch start their vehicle.

Being realistic, you will find that "most people" have acquired these skills. Yet, probably none of them spent hours studying or practising these things. No, they, like most normal people, have picked these skills up by keeping their eyes open. Often, the problem merely requires the use of a little common sense.

The mentality behind this can be used to advantage with computers, as well as defunct batteries. Instead of ignoring technical articles you feel are out of your league, have a look at them, explore your USER manual.

You don't have to make a university course out of it. Just every now and then, read a section. The first program I managed, which was only recently, was a combination of the sections on pages 27 and 32 in the 64 manual to make a simple times table project for my little brother to use.

It reads:

```
1Ø A=
2Ø B=Ø
3Ø C=Ø
4Ø PRINT A"*"B"="C
5Ø B=B+1
6Ø C=C+A
7Ø IF B<13 THEN 4Ø
```

All I had to do was teach him how to put the value for multiplication in line 1Ø. If you don't know what is going on here, then I guarantee it will take you less than 15 minutes to find out.

The feeling I had from that initial achievement was much the same as the first time I worked out how to dry the distributor in my car without having to ring my father. This is called self-reliance. The point I am making is you don't have to be an expert but you can get to a stage where you know what is going on with your computer.

It's a real buzz to sit and play your own game. No matter how simple it is, the self-satisfaction is great.

The second point is an extension of the first. I would like to illustrate this aspect as well.

Fifteen years ago, at my parents' prompting, I started learning classical guitar. I used to spend 2Ø minutes a day practising, but by the time I was 12 my interest was waning.

Then I discovered rock music. Suddenly I had to learn a completely new way to play. Strangely, I started to learn faster and enjoy music more than I ever had before.

"Why is this so?" as Prof Miller would say. Well every rock song has a full glass and a half of dairy cream milk. No! On looking back I can see quite a few factors.

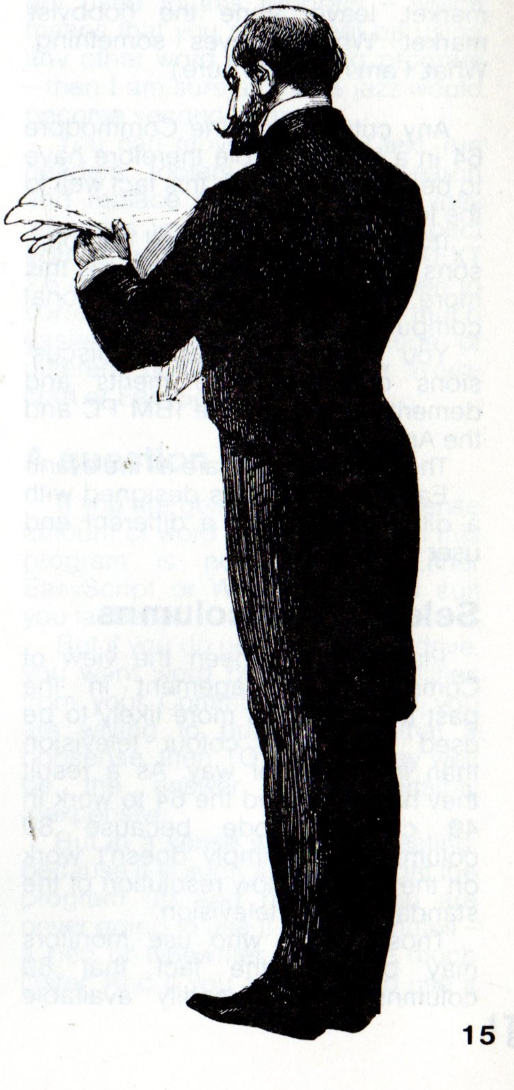
For one, I was very familiar with the music which made it easier to pick up mistakes. The value of the progress I was making was always vivid in my mind. Plus I was learning to play the kind of music I enjoy and love listening to. Also when I practised, it wasn't a chore. As for the harder things to learn, I just spent a little more time each day working out the problems involved.

In retrospect the same attitude toward computers can help you grasp greater understanding and get more enjoyment out of the time you spend.

I will give you an example. You may like to make up projects like these:

make up a BASIC game for yourself.
a personalised system for entering tax deductible items.
a program that makes a task at your place of employment a little easier.

The possibilities are endless, even at this comparatively low level of understanding. To keep interested, though, you must mix a bit of work with pleasure. You can do this by starting a file of some project you have underway, and every time you play a game or use a word processor, make a point of making a little progress with the project. □



The Commodore 64 as a business machine—giving the TOTL picture

by Gareth Powell

Let us start off by accepting that the Commodore 64 was never intended to be a business machine. The fact that it is used by many owners in that role is a tribute to its tremendous inbuilt versatility. (Worth noting here that the Commodore 64 is the largest selling business machine in New Zealand. That is, it is the largest selling personal computer in the business market, leave alone the hobbyist market. Which proves something. What, I am not quite sure.)

Any criticisms of the Commodore 64 in a business role therefore have to be considered with this fact well to the forefront.

It has been said that all comparisons are odious, and nowhere is this more apparent than in the personal computer field.

You hear totally fruitless discussions comparing the merits and demerits of the 64, the IBM PC and the Apple Macintosh.

These discussions are all irrelevant.

Each machine was designed with a different purpose, a different end user in view.

Selecting 4Ø columns

Plainly, it has been the view of Commodore management in the past that the 64 is more likely to be used through a colour television than in any other way. As a result they have confined the 64 to work in 4Ø column mode because 8Ø column mode simply doesn't work on the relatively low resolution of the standard colour television.

Those of us who use monitors may bemoan the fact that 8Ø columns is not readily available (there are conversions, but they do not work satisfactorily with all programs in all situations. Or even the majority of programs in a majority of situations).

Accepting the limitations of 4Ø columns – a purposeful design feature which has taken into account the vast majority of Commodore 64 users – can the Commodore 64 be used easily as a business machine?

The answer is yes, no and maybe.

Big four

In business the major uses of a personal computer – in an arbitary order of importance – are word processing, spreadsheets, databases and accounting.

How does the 64 shape up in each of these areas?

Let us start with word processing.

There are three packages which have been tested and used by us at considerable length.

First of all there is EasyScript, an English product which works well – transforming the Commodore 64 into an acceptable word processing unit when it is joined to a suitable printer and disk drive. Notice the key word is acceptable.

Because of the inherent limitations of speed which appear to be built into the disk drive, there are times when you are sitting twiddling your thumbs and waiting for something to happen.

But, in truth, when you are working with a word processor it is rare that you find yourself without some papers to sort out, some fact to ascertain, some phrase to ponder over. Which means that the slowness of the disk drive does not intrude as much as one might think.

Writing a book

I was recently commissioned to write the major part of a book concentrating on the history of Australia over the last twenty years. It will have been published by the time you read this, and as we are talking about computers and not books I will not mention the title. I won't even mention the publisher, Angus and Robertson.

The book is divided into 64 chapters, each 1,75Ø-2,25Ø words long. I wrote 12 of those chapters on a Commodore with EasyScript and, in truth, I missed certain facilities that I find on a dedicated word processor like my Wang 5. But the Wang cost over $1Ø,ØØØ – and the difference in performance, believe me, is in no way reflected in the price.

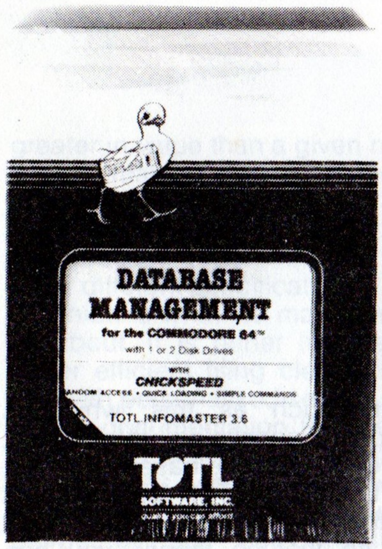
If I had to do all my writing on a Commodore 64 it would not worry me one iota to have to stay with EasyScript despite its "Fawlty Towers" manual – and it is fair to say that I probably churn out more words a day than most journalists in Australia.

In the July issue of "Australian Commodore Review" Andrew Farrell reviewed at length Wordcraft, which he seemed to think an excellent program but lacking the speed of EasyScript in certain areas of operation.

But it is worth noting that its price is under $1ØØ which makes it bargain basement word processing, indeed.

Tape driven

What about the vast majority of Commodore 64 owners who don't have a disk drive? Are they to be left out in the cold?



Not at all, but there are limitations.

Before we look at the word processing package from TOTL – available on either disk or cassette – let us reflect on the inherent problem of using a tape drive for word processing.

It is simply this – speed of access.

The letters, articles, notes that you write are stored on the tape in exactly the same way that they would be stored onto a disk. After all, a tape and a disk work on exactly the same principle.

Getting it out

Where the difference lies is when you want to get that information out again.

With a disk drive the head can go almost immediately to the correct spot and load the information from the disk into the machine.

With a cassette the information is stored sequentially and you have to work your way carefully through the tape to find the right piece of information – which slows down the operation more than somewhat. It also means that you have to keep a written record of each document on each tape and its relative position on the tape.

If you forget to do this you can get round it by asking the program to look for a fictional file name, "Andrew Farrell" for example, when the program will go through the tape in sequential order showing each title on the screen while looking for the mythical "Andrew Farrell" file.

It really is much easier to keep a careful note.

And that is the difference, the only difference worth noting, between using a disk drive and a tape cassette machine. If you are willing to put up with some inconvenience and some lack of speed you are going to save a very substantial amount of money.

How it works

When you load TOTL's word processing program (throughout this review I am referring to the cassette based progam which is marketed as TOTL.TEXT 2.6) from cassette you realise right from the start the inherent problem with tape drives because it takes nearly ten minutes to load.

So you have to plan around it.

If you are going to have a word processing session you start your tape loading and then nip off and have a cup of tea until the machine is ready. This may sound facetious. It is not meant to be.

If you have inherent time limitations built into a situation where you are sacrificing speed for a major saving in money, then you must program yourself to use the spare time instead of sitting around like a wet lettuce waiting for something to happen while times goes by.

Using cassette based programs means that the user must be as carefully programmed as the machine.

TOTL.TEXT 2.6

Once you have loaded TOTL.TEXT from cassette, then all the facilties of the program are as available to you as if you were using a disk. For TOTL have not shortened or simplified the program in any way to get it on a cassette. Which is why it takes a while to load.

It has most of the facilities of a standard word processing package with one or two unfortunate omissions.

Let us get the bad news out of the way first.

Because of the way that the program is written it does not recognise carriage returns as starting a new paragraph. Nor does it recognise spaces inserted through the use of the space bar.

If you want to add extra spaces you have to use "$sp" which tells the computer to insert an extra space. To start a new line you type "$nl" and to start a new paragraph you type in "$np".

Which you may think is too clumsy for words – and I would agree.

To give you an example from the book of instructions:

"$ti Sample Textfile $ $lm 1Ø $rm 1Ø $rj $tb 45 TOTL Software $nl $tb 45 1983 $sk Dear TOTL.TEXT User: $sk $pr You have successfully LOADed . . ."

All of this is necessary to produce an end result which comes out:

```
     Sample Textfile

          TOTL Software
              1983
Dear TOTL.TEXT User:

  You have successfully LOADed . .
."
```

If that little lot doesn't frighten you, you're a better man than I am, Gunga Din.

Having said that, there is an absolute consistency in the codes used throughout and once you have got used to this program – which means that you should not be using any other word processing program – then I am sure all that $ jazz would become second nature.

You can move blocks of text, run headers, footers, footnotes, search and replace and all the other jolly attributes you have come to expect from a word processor. TOTL.TEXT 2.6 also appears – and I am open to correction on this – to be much easier to adapt to a multitude of different printers than either Wordcraft or EasyScript.

A question of choice

If you are going to do an immense amount of word processing then this program is not for you. Either EasyScript or Wordcraft would suit you far better.

But if you do not have a disk drive, you want word processing facilities from your Commodore 64 and you are willing to put up with that $ nonsense, then TOTL.TEXT may well be the answer. It is certainly inexpensive.

But in a sense it is self-defeating, because it is such a hassle to get the program up and running you are never going to use it for short notes – a pen or typewriter would be much faster. And if you are going to use it for much larger documents then you probably need a more sophisticated word processing package.

Student use

My guess is that it would be ideal for a student producing term papers and working within a very tight budget. But it would not be acceptable to someone like me who wants semi-professional features built in so that using the system becomes second nature.

Incidentally, those of you who are 8Ø column fanatics will be pleased to know that this package will work with most of the 8Ø column modifications.

Spreadsheets

My only experience with spreadsheets on the Commodore 64 has been with the version put out by the Swedish company Handic, which I reviewed in our historic first issue.

I am used to using spreadsheets on all sorts of different machines and with all sorts of programs, and I found the Handic version was perfectly adequate for the needs of a small to medium sized business, was easy to learn and was extremely well documented.

You can hardly ask for better than that.

It is interesting to note that because of the function keys on the Commodore 64 this spreadsheet is much easier to use than a similar spreadsheet on, say, the Apple.

However, it is very easy to convert an Apple to 8Ø column spreadsheet use. The same is sadly not true of the Commodore 64. And the difference that 8Ø columns makes in this particular type of program is absolutely crucial.

You wouldn't, in all honesty, buy a Commodore 64 for its spreadsheet capabilities.

Commodore have never suggested that you should. They have produced a machine for one market. But such is the strength and versatility of the Commodore 64 that it is being stretched and extended into uses which were not in the designers' minds when it was initiated.

Frequently these adaptations are a great success.

Spreadsheets, because of the inherent limitations of a 4Ø column display, are one of the less successful examples.

Database management

There is a fair amount of rubbish talked about database management programs.

A true relational database program is a thing of awesome power and ability. The classic example is dBase II, which is so amazingly powerful and versatile that literally hundreds of major business programs have been founded on its use. It is a true relational database and as a result it is greedy of memory and an absolute pig to program. dBase II is not for the happy home hacker. You are at an advanced programming level the moment you start to use its inbuilt capabilities.

For the Commodore 64 there are several packages that are called databases. But they are not genuine relational databases within the meaning of the term. They are file and list managers – and none the worse for that. Because a good file and list manager is probably all that you will ever need and is unbelievably easier to use than a true relational database.

TOTL.INFOMASTER 3.6

I've been playing around with TOTL.INFOMASTER 3.6 and it is fascinating to observe that, although on the cover it is referred to as database management, in the text of the instructions the phrase appears "In this manual the words database and file are used interchangeably".

Quite so.

The program is disk based, and I cannot ever see a situation where a program such as this, which is continually accessing information, can ever be cassette based.

TOTL.INFOMASTER locks in to all the other programs in the TOTL group so that, in theory, you can create files of addresses which can be accessed by a form letter from the word processing package and so on. As we will see, in fact this is not so, as another, intermediate, program is apparently needed to effect the transfer.

As the current version stands you can set up to 1ØØ fields on each record and each field can be up to 245 characters long.

Easiest way to think of it is as a card file. You have a filing card which will allow you to write in 1ØØ different pieces of information about a subject provided you keep to less than 245 characters for each entry. Which shows you how much latitude the program gives you, because if you were doing an address list it would be almost impossible to use more than 15 fields.

Mixing files

The records form into a file (just as they would in a filing system) and you are restricted to ten different files per disk. In fact, it is not likely that you would mix different files on the same disk. I find in using this sort of program that I tend to keep different files – names and addresses, magazines in stock, article references and so on – on different disks from the point of view of pure convenience.

The number of records that you write to a disk is only limited by the capacity available on that disk.

Once you have written your records to disk the program will allow you to sort and search with an amazing amount of flexibility.

Sorting

You can specify that it selects records where, for example, a field is greater in value than a given number. Which makes sorting records by date easy.

Or you can sort by key word into alphabetical order. Or to several other different specifications.

Which is what file management is all about. It is rather like having a super efficient filing clerk providing you with personalised service.

The documentation for this package is, sad to say, quite awful. There is no index for the 76 pages of typewritten copy and although the blurbs all say categorically that TOTL.INFOMASTER 3.6 is logically linked to the other programs in the TOTL series, nowhere does it tell you how to integrate a standard letter with any list. My guess is that it simply will not perform this absolutely basic and essential function.

It would appear that in order to create a mailing list you would need to use a different TOTL program called TOTL.LABEL as an intermediate stage. Which strikes me as being daft.

If the information has already been entered through the INFOMASTER why cannot it be easily accessed through the word processing package if, as it is claimed, it "interacts with other TOTL programs".

Again, I am guessing, but I have a feeling that you would have to download a list of names and addresses out of INFOMASTER and into the other program LABEL before you could then match them up to a letter created by the wordprocessing program TOTAL.TEXT.

Having written that and gone back and reviewed the options available, I am certain I am correct.

Which makes this file management system at best cumbersome and at worst (if you do not have the other relevant programs) unworkable in one of the most important areas in which file management programs function.

There are better database programs around, and TOTL are going to have to lift their game if they are going to put forward this program as a serious contender in the file management market area, even accepting its amazingly low cost.

Accounting

Finally we come to accounting.

We have been spoiled in that we have spent a lot of time with the Cashbook for the Commodore, produced by Pittwater Distributors.

It is a program that is simple yet elegant and works like a charm.

Its only limitations are the speed of the disk drive and, as has been pointed out by Bob Drew who is responsible for the program, you can always plan to have something to occupy your time while the disk, like the mills of God, grinds on.

With the Cashbook any Australian business can produce a set of figures at the end of any month from which a trial balance can be struck and the accounts finalised in a matter of hours rather than days. At all times the bank account is completely reconciled and there is even an audit trail built in.

If I were asked to recommend a small business system to anyone using a Commodore 64 I would find it difficult to go past that produced by Pittwater Distributors.

TOTAL.BUSINESS 3.6

TOTL.BUSINESS 3.6 is a different package altogether. Before I make any comments I must explain that it is almost impossible to test adequately any accounting package unless you are working with it very regularly.

It takes time to get used to the way the package works, it takes time to find out if there are any bugs still lurking in the undergrowth and it takes time to see if the program is, in that truly awful piece of computerspeak, user friendly.

Complete program

Within the context of these restrictions the program under review appears to be remarkably complete for the price.

But, and this is an important but, it is an American program that, in the version I tested, had not been modified for use by Australian businesses.

The most obvious example of the problems this creates is the way that Americans insert the date.

We say May 8, 1984 or 8/5/1984.

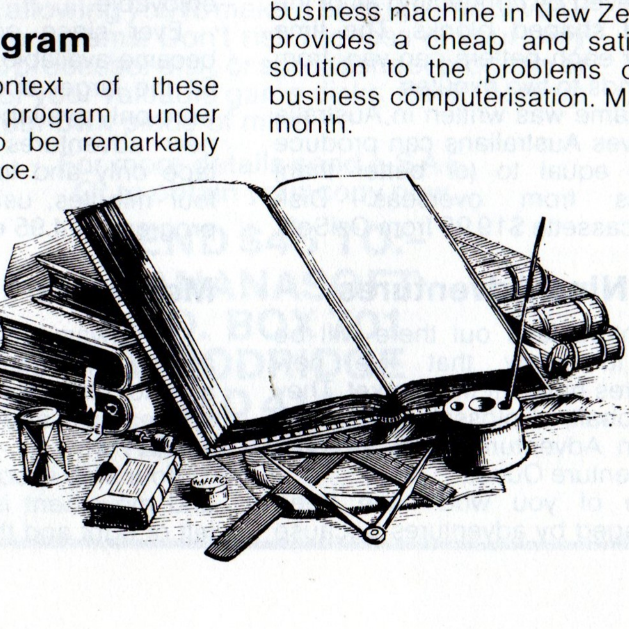
For the same date they write 5/8/1984 which can lead to some very dangerous confusion when you are printing out invoices or statements telling customers that payment is due within 3Ø days of the given date.

This is, as I say, the most obvious example but there are several other subtle indications that this program has been written to follow the General Agreed Accounting Practices, the standard followed by most American accountants but relatively few Australian ones. I would be most interested to hear from any user who has experience with a package with this built-in Americanisation – and how they got around the problems.

Unless there is a simple solution, it would be difficult to wholeheartedly recommend a program that could so easily lead to confusion and customer relation problems. Better by far to stick with an Australian product.

Next month I'll be looking at some more business packages, but there is enough here for you to see that although the Commodore 64 was not designed as a business machine, it can function in that area extremely well.

Perhaps now it is easier to understand why it is so popular as a business machine in New Zealand. It provides a cheap and satisfactory solution to the problems of small business computerisation. More next month. □



HOT STUFF

With the recent influx of new programs for the Commodore 64, keeping informed is becoming a hard task. Purchasing software is no longer an easy choice due to the wide range of programs available.

The following short reviews are aimed at informing you of the latest software, and giving you an idea of the style and content of the games. Certainly we do not claim to mention all of the programs currently available, but we cover some of the more notable programs.

Pingo

No longer must you go to the arcades to play the game Pengo. Pingo, as you may have already have guessed by the subtle difference in their names, is an exact copy of this game.

You are a penguin and must slide blocks of ice in order to destroy your opponent's, the snow penguins. But they are difficult to crush, as each must be hit twice before dying.

Points are awarded for the time in which you take to destroy all of the snow penguins and bonus points are achieved by connecting all of the diamond shaped blocks. The time taken for each pattern can vary from 15 seconds to two minutes.

The game was written in Australia and proves Australians can produce software equal to (or better than) programs from overseas. Disk $24.95, cassette $19.95 from OziSoft.

Level Nine Adventures

Adventure fans out there will be happy to know that five new adventures have hit the market. They are Snoball, Colossal Adventure, Dungeon Adventure, Lords of Time and Adventure Quest.

Those of you who have been discouraged by adventures because they are too short, will be happy to know these are the longest adventures ever available on the Commodore 64 or any other micro for that matter.

Adventure games were first introduced by one game, Colossal Caves, which was originally designed on a mainframe computer. It was said this program could never be available on a micro because of the enormous amount of memory that it required.

But it is now available on the Commodore 64 and is called Colossal Adventure. Those of you who have played the game on a large computer will be pleased to know it is just as large and even claims to be larger!!

Each adventure has over 2ØØ rooms, making all previous adventures look tiny. Unfortunately they are not as intelligent as games such as "Zork" or "Underworld of Kin" but understand enough to make playing enjoyable.

Ever since adventure programs became available for the Commodore 64 the larger and better adventures were only available on disk but these new adventures are available on tape only and load in a surprising four minutes, using a turbo loader program. $24.95 each from OziSoft.

Megawarz

Megawarz is a semi-original arcade type game. You must travel around the screen rescuing lost spacemen and shooting mean aliens.

Your ship is equipped with shields and movement is made by rotating left or right and then thrusting in that direction (much the same as the arcade game, Asteroids).

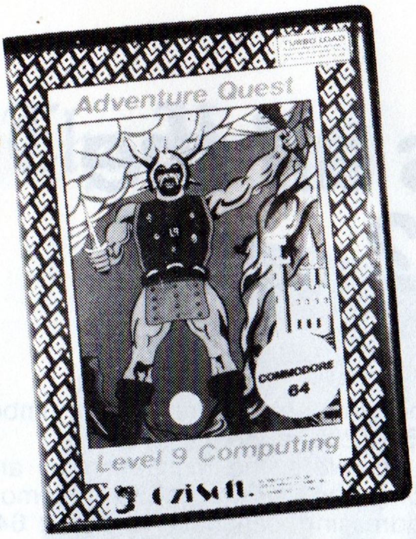
Completing each round will take you to a bonus pattern where you must rescue as many men as you can, within a time limit. Completing this will take you to another planet and start the game again.

The game is adjustable allowing you to select how many men you wish to have and what speed you wish to play at. Quite an addictive game and is presently available only on tape. $19.95 from OziSoft.

Outback

Outback is a modified copy of the arcade game Pooyan but has been changed from a war between pigs and foxes to kangaroos and hunters. You are on the kangaroos' side and must move up and down the side of the screen in order to avoid boomerangs and stones. Shoot the descending hunters because, if they land, they will kill one of your kangaroos. Outback combines several different screens and bonus patterns and is overall a quite enjoyable game.

Sadly though, it is not made in Australia despite its Australian basis. $19.95 from OziSoft.



competition PRO™
5000
Arcade Quality Joystick for video game/computer systems
2 YEAR WARRANTY
For the use with Atari Commodore Sears type systems
Made in U.S.A.

The Competition Pro Joystick

Not only do we review software in this section but also peripherals such as joysticks. One new joystick to arrive on the market is the Competition-Pro. This joystick is the first of its type to sell at a reasonable price.

Most joysticks operate on a system where when the joystick is pushed in a certain direction, pressure is applied to a certain area on the circuit board and this causes a reading to be taken. The Competition Pro however works in a totally different way. When you move the Competition Pro in a direction, it physically closes a switch and registers your movement. This is a much more reliable method of joystick operation and used to only be available with the extremely expensive models like the Wico joysticks.

The Competition Pro also has two fire buttons for left and right handers and sells for approximately $29.95. At this price it must be one of the best value joysticks on the market. □

Where does all the memory go?

by Andrew Farrell

There is a large amount of confusion concerning exactly how much memory the Vic 2Ø has, how much you can add and how much you end up with. Hopefully things will be a little clearer when you've read this.

Most computers on the home computer market use 16 bit memory addressing techniques. Each memory location contains 8 bits which may either be on or off. The largest number we can store in one of these locations is 255 [255(decimal)=11111111(binary)]. By using two of these locations together, the microprocessor can store a much larger number [256 x 256 = 65536].

Therefore, the most memory any computer using 16Bit memory addressing can have is about 64k (65535 bytes). Of course, there are always exceptions as it is possible to get around this problem.

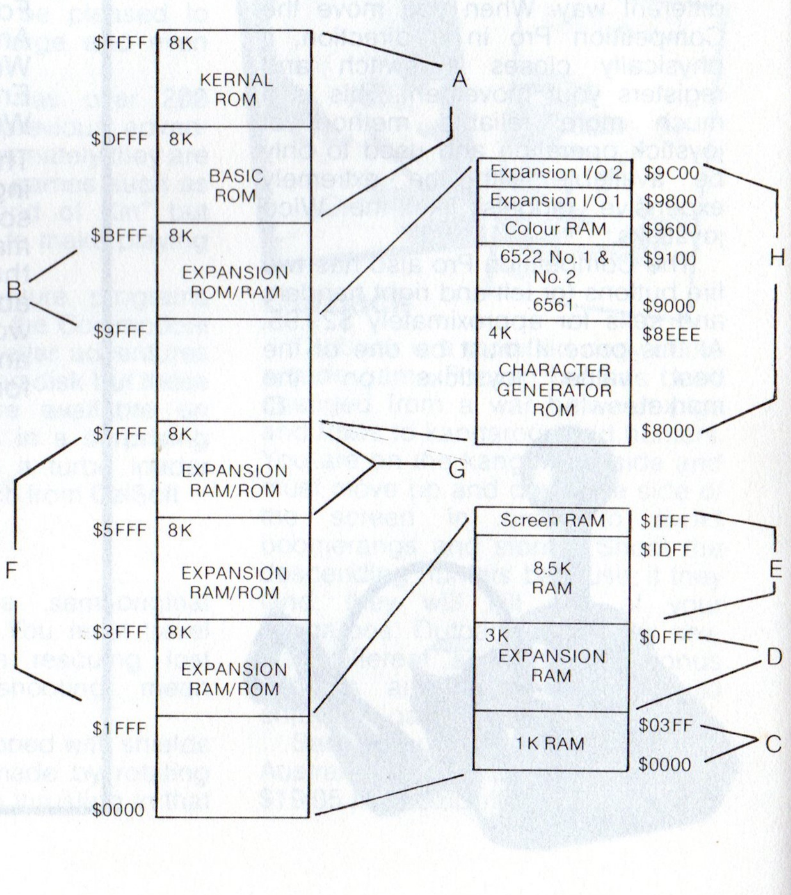
The Vic 2Ø uses the above method to address memory so it can also address a maximum of 65535 bytes. The big difference between the Vic and most other systems is the makers have been very honest about how much memory is actually available for you to use. A 48k Apple, using disk drives and graphics, really only has around 25k free for use by you. A Vic 2Ø, using drives and graphics has around 28k free when fully expanded. Why? How do you work it out? Can you add more?

If you're really desperate to have as much memory as possible you can manage around 38.5k free for machine code and data, 28k of which you can use for your BASIC program and variables.

To understand this better we should have a close look at a memory map of the Vic 2Ø.

A. The top of memory, or the last 16k, contains the Vic ROM's which control the operation of everything which goes on including system initialisation (what happens when you turn the Vic on) and all the standard BASIC operations.

B. This 8k block is used by plug in



games cartridges, the Super Expander and various other ROMs such as Hesmon. When the Vic is first turned on, the power up routine checks to see if there is anything plugged in the top 4k block of this area. If there is, then control is transferred to the machine code routines whose starting address is stored in locations $AØØØ and $AØØ1. An 8k cartridge may be switched to this area and used as RAM for machine code or data.

C. Zero page is here along with the stack. Zero page (the first 256 bytes of memory) contains many interesting locations which are used to control the internal workings of the Vic. This area is used by the ROMs in Section A. The top 5ØØ bytes of this block are used for the cassette buffer (starting from locations 828) and various jump vectors and pointers. The cassette buffer may be used for storing machine code providing the program does not access the cassette player while operating.

D. The Super Expander's 3k and the normal 3k cartridge is located here. If the Vic is expanded beyond 6.5k this area is only available for machine code, character sets, or for the hi-resolution screen area.

E. Available for the user on a normal Vic as well as section C and H. Includes the area (512 bytes) used for screen RAM and all basic variables which work down from the top of available memory (normally $IDFF).

F. Expansion for the Vic usually fits into this area including the Programmer's Aid, Machine Code Monitor and various other ROMs. The 16k cartridge sits in the first two 8k blocks starting from $1FFF. The 8k cartridge may sit in any three of these blocks.

G. Many cartridges which clash when plugged in together do so because they both use the same area of memory. This may be remedied by turning one cartridge off.

H. The Vic Chip, character ROM and various I/O registers are in this block of 8k. The first 4k is ROM only, the second is mostly RAM, however some sections contain no memory at all. □

The Hobbit

The Hobbit, by Melbourne House Software, is based on the novel by J R R Tolkien. It combines great graphics and a large vocabulary with the fun of experiencing the adventures of Bilbo Baggins, the hobbit. It is an exciting adventure game following the storyline of the book. It requires all your skill and cunning to match the wits of the wargs, goblins, elves, spiders and Smaug the dragon. All this along the trail of Lonely Mountain and back again with your treasure.

Although this may sound just like any other adventure game, it is not. With the computer acting as the go between, you use a number of commands that the computer translates and executes. You can also talk to any other character in the game and perform a number of minor tasks, for example shoot, unpack, tie, dig, drop, empty, etc.

The graphics in this program are more than adequate. They show you a good quality picture, a representation of your immediate surroundings. And you work from a map, in the back of the novel, which helps you navigate around Middle Earth.

An added advantage of this game is, having mastered it once, you can try many other different ways to get to the gold and back. With the sophistication and existing popularity of this game I am sure that the stage is set for the sequel, The Lord of the Rings, also written by Tolkien. The game, in conjunction with the novel, is a wonderful adventure for anyone with a taste for magic, fantasy and adventure.

Fastloading cassette, $39.95, distributed by Melbourne House. □

Bat Attack/Blagger

by Adam Rigby

Bat Attack

When I received this game to review and read the back, I instantly thought "Ahah, another Galaxian style game", until I read the words "through a myriad of flying doughnut bars". This, of course, caught my attention.

It took a full ten minutes to load and what is worse, it wasn't worth it. To start off, it was a bad version of Galaxian but when you finish the first frame, the screen clears and these large rotating rings or "doughnut bars" come flying down at you. These are quite hard to dodge because both you and they are so large. You being a sprite as long and wide as you can get without expansion.

These doughnut bars are visually quite effective, differing from the rest of the game which is more or less ordinary. The sound is ordinary and does the job "no frills" style. Lasting appeal is bad to say the least, I was bored after the first few games. The game is hard because of your ship's size and speed, too hard in fact. Our review copy came from OziSoft.

Graphics:	***
Sound:	***
Lasting appeal:	*
Value for money:	*
Originality:	***

Blagger

Blagger is an interesting game where you play the part of Roger the Dodger, master burglar. You must avoid various tricks and traps to collect the golden keys, which then allow you to proceed to the next level. The game has a grand 3Ø levels in all which are full of gags unique to this game.

An example of some of these are: conveyor belts, killer lollies, switches, decaying platforms and various things which will keep you occupied for hours. It took me quite a number of hours and a blistered thumb to master the sixth level.

Because of the difficulty of the game I found the Run/Stop key very useful, as it pauses the game so you can study the level before play. The Run/Stop key is also useful because you have a time counter (a'la Donkey Kong) which, when it runs out, costs you a life. Also good old "look before you leap" prevails in this game.

Good news for cassette owners. Just because the game has 3Ø levels does not mean it is disk based, quite the opposite really – it is not available on disk. Let us hope that Alligata brings out a disk version because 14 minutes is a long time to wait, even for a game this good.

The graphics are quite good but the sound is just the opposite. I found the sound quite irritating and eventually just turned it off. With the almost limitless amount of levels and the necessary thought involved you would find it hard to become bored with the game no matter how hard you try. Well worth the money.

Graphics:	****
Sound:	**
Lasting appeal:	*****
Value for money:	****
Difficulty:	****
Originality:	****

□

Vic-20 games review

Phil Campbell

Name: JETPAC
Format: **Cassette for Vic 2Ø with 8K expansion**
Publisher: **Ultimate Play the Game**
Distributor: **Dolphin Computers, 99 Reserve Rd, Artarmon**

Vic 2Ø fans will doubtless be pleased at the arrival of the latest bundle of goodies from the land of our earliest pale inhabitants. In the past, as you know, the English have sent us nothing but their rejects — convicts, magazine publishers, and others of their ilk. Fortunately, some of their software is not quite so bad. JETPAC, for example, is rather a good game. So good, in fact, that its worst feature is the absolute rubbish printed on the inside of the cassette cover.

The plot

"The 'Acme Interstellar Transport Company' is delivering spaceship kits to various planets in the solar systems throughout the galaxy; and as chief test pilot, all you have to do is assemble the rockets, and thrust on to your next destination". The process of fuelling the rocket and collecting sacks of precious gems is described in similar flowery prose. I vividly recall another pommie program which relied on a similarly styled short story to inform the innocent player of what he should do where, when and why. I was confused for days. This effort is not quite so bad, as the rules become reasonably obvious as the game progresses.

In play

Programs of 8K take quite some time to load, so make sure you pack a picnic lunch. When loading is completed, you are invited to specify whether you intend to play alone or with a friend, and whether or not you wish to use a joystick. The keyboard player (undoubtedly a rare breed) is well catered for, as complete rows of keys are allocated to perform the functions of flying, walking, thrusting, hovering and firing. Unfortunately, it is still necessary to have around seventeen fingers, or be remarkably prestidigitatious. (Neither the publisher nor the editor knows what this word means, but it looks impressive.)

After dispensing with the preliminaries, things begin to happen rather quickly. In fact, you will be destroyed in next to no time. Game designers probably think this is an enticing feature, which generates a strong competitive urge and encourages the player to fight on in the face of insurmountable odds. I think it stinks. With practice, however, (and lots of laser fire), the ranks of the nasty little greeblies thin out a little, and the task of collecting the three bits of rocket and putting them together begins to look a little more attainable.

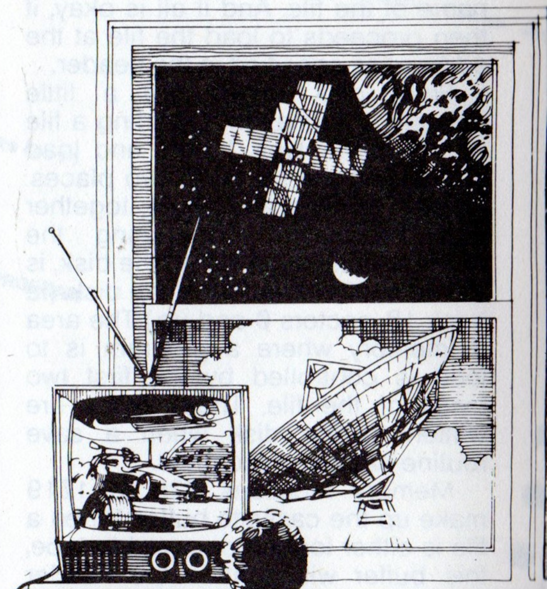
With a bit of effort, it is possible to collect the required fuel pods and even the occasional treasure or two. After the rocket is fuelled, jump in for a flight to the top of the screen and back. You are now on an entirely different planet, although it looks identical to the original. However, the nasty flying things look different, and are quite a bit meaner than the ones you just obliterated. From here on in, it is just a matter of refuelling and taking off, collecting treasures here and there, and surviving for as long as possible.

The verdict

This is good. Graphics and animation are clever, sonics are reasonable, and the challenge is persistant. Loading was reliable, with the added bonus of having two copies of the program on each side of the cassette. Priced at $17.85, JETPAC represents reasonably good value for those who are looking for something interesting to do with their 8K memory expansion cartridges.

Ratings

Graphics	****
Sonics	***
Challenge	***
Value	****
	□



All about file loading

by Tom Quealy

When a program is loaded into memory, via disk or cassette, the computer knows where in memory the program is to be loaded and how long it is. Depending on how a program is firstly saved and then loaded, the user can force a program to load anywhere in memory they desire. Thus, it is possible to have more than one program in memory at any one time.

If we wish to find the starting location of a program, we use the same method as the computer uses, ie we look at the header of a file.

Firstly, tapes

When a file (prg or seq) is stored, the computer places information regarding type, starting address, ending address, and name on the tape. When we load a program the computer, after loading the header, checks the type of file found (prg or seq). If they match, it checks the name of the file. And if all is okay, it then proceeds to load the file at the address as specified in the header.

From disk things are a little different. Information regarding a file (ie the name, type, length and load addresses) are stored in two places. Names, lengths and types, together with information concerning the whereabouts of the file on the disk, is stored on the directory of the disk. (Ie track 18, sectors Ø and up.) The area in memory where a program is to load is controlled by the first two bytes of the file. These bytes are written to the disk when a save routine has been executed.

Memory locations 828 to 1Ø19 make up the cassette buffer. When a file is either loaded or saved to tape, this buffer will contain the header information. The memory is laid out like this:

Location	Contents
28 ($Ø33C)	File type: :1,2,3 – prg :4 – seq
29 ($Ø33D)	Lo Start Address
3Ø ($Ø33E)	Hi Start Address
31 ($Ø33F)	Lo End Address
32 ($Ø34Ø)	Hi End Address
33 ($Ø341)	Start of file Name

By multiplying a high address by 256 and then adding the low address we end up at the location in memory from which a program starts or where a program ends. The reasoning for this calculation is based on the fact that the C64 is an 8 bit computer. (Ie the largest number we can store in one byte is two to the power of eight minus 1. Due to this restriction, the starting and ending addresses are divided into two bytes:- single units (Lo) and blocks of 256 units (Hi).

This may seem a complex way of handling things, (hard luck), but your whole computer runs on the Lo/Hi format even when programming in BASIC.

The first two bytes of a program stored on a disk tell us the start address of where the program should load. Once again this is stored in the Lo/Hi format and was written to the disk when the program was saved.

There are two main ways of loading a program. One is a "straight" load and the other is a load with a secondary address. A straight load, for example LOAD"FRED" will load a program at the location as specified by the pointers to BASIC, by default 2Ø49 ($Ø8Ø1). A load with the use of a secondary address. For example LOAD"FRED",1,1 will load the program from cassette into memory starting from the locations found in the header. Even if a program's header requests it be loaded high up in memory, if the user does a straight load, the program should always load into the area pointed to by the BASIC pointers. (I say "should" as the latest protection routines prevent this.) Disks are much the same except I have yet to see a protection routine that prevents "straight" loading.

If, for example, you wished to load a program, (BASIC or machine code) starting from location 32768 ($8ØØØ), you must change the BASIC pointers to point to 32768 and then do a "straight" load. You may have noticed that 32768 is equal to $8ØØØ in base 16 (hexadecimal). This is the form in which we now must operate. If we split this 4 digit number into two numbers, we get $8Ø and $ØØ. We must now convert these back into decimal, ie. $8Ø=(8*16^1)+(Ø*16^1) and $ØØ=Ø. Therefore the Hi btye is 128 and the Lo is Ø. Now we must change the BASIC pointers. Memory locations 43 and 44 contain the Lo/Hi bytes respectively. Prior to loading, the value Ø must be poked into location 43 and 128 into location 44. (Eg POKE43,Ø:POKE 44,128.) A "straight" load will now load the program into the computer's memory, starting from location 32768.

BASIC pointers can be modified to permit the user to save any piece of the computer's memory to tape or disk. This involves the modification of the end of BASIC pointers as well as the start. Supposing for example you wish to save what is currently on your screen as a file. Firstly we must find where in memory the screen sits. It starts at location 1Ø24 ($Ø4ØØ) and ends at 2Ø23 ($Ø7E7). Firstly we must set the start of BASIC pointers:

POKE 43,Ø : POKE 44,4

Now we must set the end of BASIC pointer. This is found at locations 45 and 46 in the same Lo/Hi format. Place $E7 (or 224 in decimal) into location 45 and, in the Hi byte, (location 46) we place $Ø7 (or simply 7 in decimal).

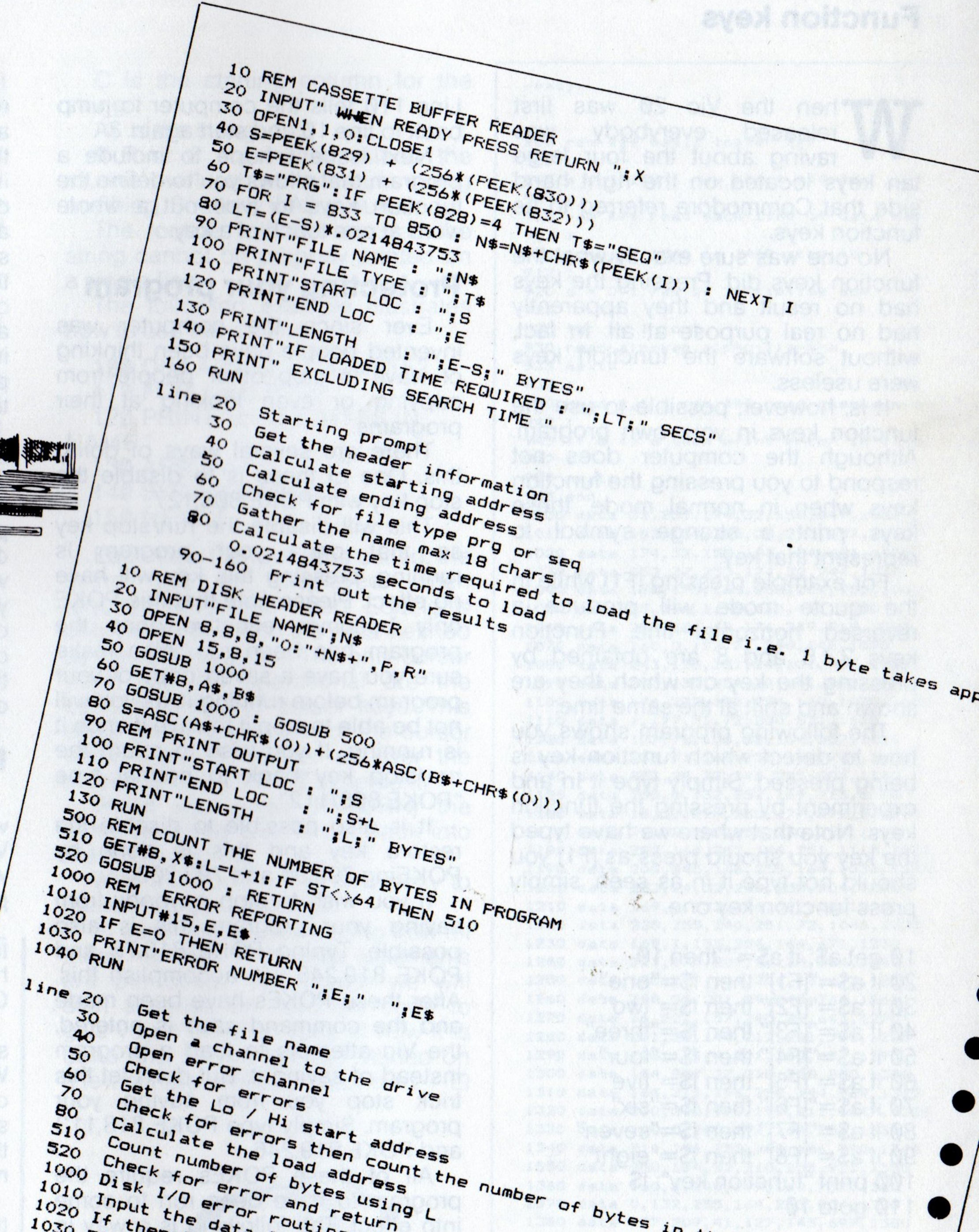
POKE 45,224:POKE 46,7

Now type SAVE"SCREEN",1 or SAVE"SCREEN",8 and press <RETURN>. Note no secondary addresses are required.

Now turn your computer off and on again and clear the screen. Type LOAD"SCREEN",1,1 or LOAD" SCREEN",8,1 and press <RETURN>. When loading is complete type POKE 53281,Ø <RETURN>. This will blacken out the background color, revealing the same display as you had whilst saving the program earlier.

This method of memory dumping can be used to dump any of the computer's memory, regardless of its contents. This may save a lot of time as far as graphics are concerned, hires, lores, sprites, all of these simply occupy a piece of memory. Once you create them you can save them.

The following program allows the user to read the header information of a file and find out where it was SAVED from. □



```
10 REM CASSETTE BUFFER READER
20 INPUT" WHEN READY PRESS RETURN ";X
30 OPEN1,1,0:CLOSE1
40 S=PEEK(829) + (256*(PEEK(830)))
50 E=PEEK(831) + (256*(PEEK(832)))
60 T$="PRG":IF PEEK(828)=4 THEN T$="SEQ"
70 FOR I = 833 TO 850 : N$=N$+CHR$(PEEK(I)) : NEXT I
80 LT=(E-S)*.0214843753
90 PRINT"FILE NAME : ";N$
100 PRINT"FILE TYPE : ";T$
110 PRINT"START LOC : ";S
120 PRINT"END LOC   : ";E
130 PRINT"LENGTH    : ";E-S;" BYTES"
140 PRINT"IF LOADED TIME REQUIRED : ";LT;" SECS"
150 PRINT" EXCLUDING SEARCH TIME "
160 RUN
line 20   Starting prompt
     30   Get the header information
     40   Calculate starting address
     50   Calculate ending address
     60   Check for file type prg or seq
     70   Gather the name max 18 chars
     80   Calculate the time required to load the file i.e. 1 byte takes app
          0.0214843753 seconds to load
  90-160  Print out the results
10 REM DISK HEADER READER
20 INPUT"FILE NAME";N$
30 OPEN 8,8,8,"0:"+N$+",P,R"
40 OPEN 15,8,15
50 GOSUB 1000
60 GET#8,A$,B$
70 GOSUB 1000 : GOSUB 500
80 S=ASC(A$+CHR$(0))+(256*ASC(B$+CHR$(0)))
90 REM PRINT OUTPUT
100 PRINT"START LOC : ";S
110 PRINT"END LOC   : ";S+L
120 PRINT"LENGTH    : ";L;" BYTES"
130 RUN
500 REM COUNT THE NUMBER OF BYTES IN PROGRAM
510 GET#8,X$:L=L+1:IF ST<>64 THEN 510
520 GOSUB 1000 : RETURN
1000 REM  ERROR REPORTING
1010 INPUT#15,E,E$
1020 IF E=0 THEN RETURN
1030 PRINT"ERROR NUMBER ";E;" ";E$
1040 RUN
line 20   Get the file name
     30   Open a channel to the drive
     40   Open error channel
     50   Check for errors
     60   Get the Lo / Hi start address
     70   Check for errors then count the number of bytes in the file.
     80   Calculate the load address
    510   Count number of bytes using L
    520   Check for error and return
   1000   Disk I/O error routine
   1010   Input the data from error channel
   1020   If there's no error then return
   1030   Else report the error
```

Vic 2Ø hints

by Sean Mcsullea

Function keys

When the Vic 2Ø was first released everybody was raving about the four large tan keys located on the right hand side that Commodore referred to as function keys.

No one was sure exactly what the function keys did. Pressing the keys had no result and they apparently had no real purpose at all. In fact, without software the function keys were useless.

It is, however, possible to use the function keys in your own program. Although the computer does not respond to you pressing the function keys when in normal mode, these keys print a strange symbol to represent that key.

For example pressing [F1] while in the quote mode will produce a reversed horizontal line. Function keys 2,4,6 and 8 are obtained by pressing the key on which they are shown and shift at the same time.

The following program shows you how to detect which function key is being pressed. Simply type it in and experiment by pressing the function keys. Note that where we have typed the key you should press as [F1] you should not type it in as seen, simply press function key one.

```
1Ø get a$: if a$=""then 1Ø
2Ø if a$="[F1]" then f$="one"
3Ø if a$="[F2]" then f$="two"
4Ø if a$="[F3]" then f$="three"
5Ø if a$="[F4]" then f$="four"
6Ø if a$="[F5]" then f$="five"
7Ø if a$="[F6]" then f$="six"
8Ø if a$="[F7]" then f$="seven"
9Ø if a$="[F8]" then f$="eight"
1ØØ print "function key"; f$
11Ø goto 1Ø
```

Here is a brief explanation of the program.

Line 1Ø waits for a key to be pressed before proceeding to line 2Ø

Lines 2Ø-9Ø set f$ to the appropriate value depending on which key has been pressed. Line 1ØØ prints the function key pressed. Line 11Ø tells the computer to jump back to line 1Ø and start again.

Next issue I hope to include a program that allows you to define the function keys to type out a whole word at one touch of a key.

Protecting your program

Ever since the computer was invented people have been thinking of ways to stop other people from copying or even looking at their programs.

There are several ways of doing this. One of these is to disable the stop key with POKE 8Ø8,127.

This will disable the run/stop key so that once your program is running, pressing this key will have no effect. Please note that this POKE only becomes effective once the program has been run. Also make sure you have a saved copy of your program before running it, as you will not be able to stop it to save it once it is running. If you wish to return the run/stop key back to normal type "POKE 8Ø8/112".

It is also possible to disable the restore key and this is done by POKEing 792,9Ø and 793,2Ø3.

If you wish to stop anybody from saving your program this is also possible. Typing POKE 818,73 and POKE 819,245 will accomplish this. After these POKEs have been made and the command save is entered, the Vic attempts to load a program instead of saving it. But don't let this trick stop you from saving your program. Simply type POKE 818,113 and POKE 819,246.

All of these POKEs require the program to have been run to come into effect. The following is a way in which you can conceal your listing so that no one can list through your program. All your programming secrets remain yours alone.

Make sure you have a spare line at the start of your program (hint, you can even use line Ø) and make that line read '1Ø rem""'. Do not press return at this stage but press control and nine at the same time (pressing these keys at the same time puts you into reverse mode). Now move the cursor back on place and press shift and inst/del at the same time. This should insert one space in front of the second quotation mark. Having done this, press shift and m together and insert another space by pressing shift and inst/del together. Now press control and one at the same time, then press return.

Let me explain what you have just done. Whenever the computer lists your program, the first line will make it turn white so that the rest of your program isn't visible. This is, of course, assuming that the screen is white at the time. If you wish to list your program simply change the colour of the screen to black or any other dark colour. This is done by typing POKE 36879,a where a is the colour you wish your screen to be.

Faster loading

Those of you with extra memory will agree with me when I say that the Vic isn't the fastest thing on earth when it comes to loading long programs.

What would you say if I told you that you can load your program in half the time? Impossible, you say? Certainly not.

When the Vic saves a program, it saves it twice, one after the other. When loading, it loads the first and checks it against the second. If the second program is any different, then the computer responds with the message "load error".

In order to halve your loading time, you must first add the following lines at the start of your program and don't forget to give them line numbers.

```
poke46,peek(832):poke48,peek(832):
poke5Ø,peek(832):poke45,peek(831)
poke47,peek(831):poke49,peek(831)
```

▷

Screen Input Routine

by Paul Reagan

Have you ever been frustrated by the limitations of the input statement in BASIC? Problems such as:

All the cursor keys are still operational – there is nothing like clearing the screen in the middle of an INPUT.

Getting input from within a border drawn on the screen.

All these problems can be solved using BASIC, but it is a messy business. The routine that follows solves the problem quite neatly.

Type in the loader program and save it.

The loader currently puts the program at location 49152, but if that section of safe memory is getting a bit crowded, change the value of P at line 9Ø5 in the loader to point to wherever you want to put the program.

Once you have typed in the loader, it may be activated by RUN 9ØØ.

The calling sequence for the routine is:

SYS (AD),R,C,A$

Where AD is the starting address you have chosen

R is the row number where you want the string to be printed.

C is the starting column for the string

A$ is the string.

If R or C are invalid values, the routine will abort without doing anything. Similarly if A$ is null.

The routine will also abort if the string cannot be properly printed on a single line.

The following example illustrates how to use the routine.

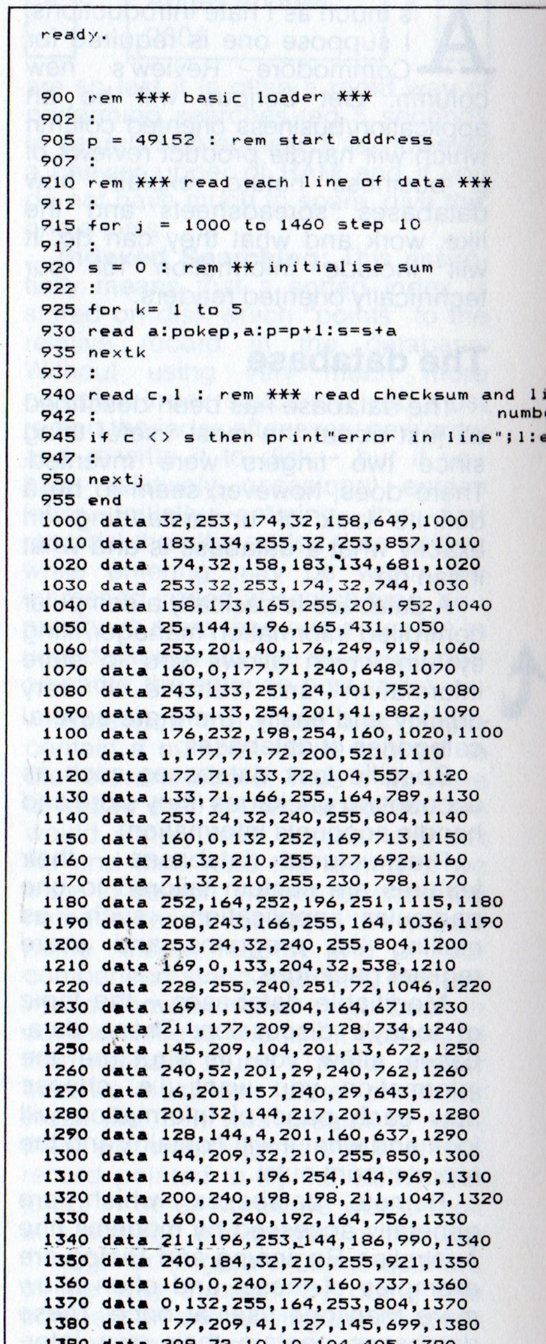
```
1ØØ DIM AR$(2Ø) : AD – 49152
11Ø FOR J = Ø TO 2Ø
12Ø PRINT "(clr,d,d,d)TYPE IN
NAME"
13Ø AR$(J) = "     "
14Ø SYS(AD),3,1Ø,A$(J)
15Ø NEXT J
```

Each time the routine is called, the string will be displayed when required in reverse video. Editing is only possible within the area defined by the reverse video. The only cursor keys that are operational are the LEFT and RIGHT cursor keys. This eliminates the problem of the cursor being able to wander all over the screen with the INPUT statement.

When you press (RETURN) the entire reversed area is put back into the original string.

With this routine it is very easy to make changes to strings.

In case you are wondering how all those numbers in the data statements are produced, next month we will start a series on machine code to give you a bit of an idea of how it is done and maybe get you started on a few simple utilities of your own. □

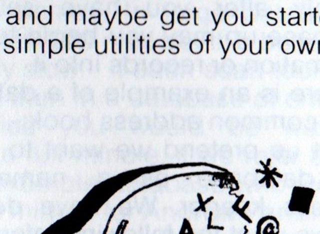


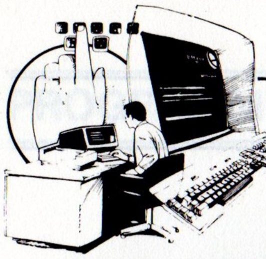
```
ready.

900 rem *** basic loader ***
902 :
905 p = 49152 : rem start address
907 :
910 rem *** read each line of data ***
912 :
915 for j = 1000 to 1460 step 10
917 :
920 s = 0 : rem ** initialise sum
922 :
925 for k= 1 to 5
930 read a:pokep,a:p=p+1:s=s+a
935 nextk
937 :
940 read c,l : rem *** read checksum and li
                                     numbe
945 if c <> s then print"error in line";l:e
947 :
950 nextj
955 end
1000 data 32,253,174,32,158,649,1000
1010 data 183,134,255,32,253,857,1010
1020 data 174,32,158,183,134,681,1020
1030 data 253,32,253,174,32,744,1030
1040 data 158,173,165,255,201,952,1040
1050 data 25,144,1,96,165,431,1050
1060 data 253,201,40,176,249,919,1060
1070 data 160,0,177,71,240,648,1070
1080 data 243,133,251,24,101,752,1080
1090 data 253,133,254,201,41,882,1090
1100 data 176,232,198,254,160,1020,1100
1110 data 1,177,71,72,200,521,1110
1120 data 177,71,133,72,104,557,1120
1130 data 133,71,166,255,164,789,1130
1140 data 253,24,32,240,255,804,1140
1150 data 160,0,132,252,169,713,1150
1160 data 18,32,210,255,177,692,1160
1170 data 71,32,210,255,230,798,1170
1180 data 252,164,252,196,251,1115,1180
1190 data 208,243,166,255,164,1036,1190
1200 data 253,24,32,240,255,804,1200
1210 data 169,0,133,204,32,538,1210
1220 data 228,255,240,251,72,1046,1220
1230 data 169,1,133,204,164,671,1230
1240 data 211,177,209,9,128,734,1240
1250 data 145,209,104,201,13,672,1250
1260 data 240,52,201,29,240,762,1260
1270 data 16,201,157,240,29,643,1270
1280 data 201,32,144,217,201,795,1280
1290 data 128,144,4,201,160,637,1290
1300 data 144,209,32,210,255,850,1300
1310 data 164,211,196,254,144,969,1310
1320 data 200,240,198,198,211,1047,1320
1330 data 160,0,240,192,164,756,1330
1340 data 211,196,253,144,186,990,1340
1350 data 240,184,32,210,255,921,1350
1360 data 160,0,240,177,160,737,1360
1370 data 0,132,255,164,253,804,1370
1380 data 177,209,41,127,145,699,1380
1390 data 209,72,10,10,104,405,1390
1400 data 8,41,63,201,32,345,1400
1410 data 176,2,9,64,40,291,1410
1420 data 144,2,9,128,164,447,1420
1430 data 255,145,71,230,255,956,1430
1440 data 164,253,196,254,240,1107,1440
1450 data 4,230,253,208,214,909,1450
1460 data 96,0,255,255,0,606,1460

ready.
```

Save your program before doing anything else. These two lines reset the basic pointers so that next time you load your program, you can stop it anywhere over halfway, and run it without a hitch.

The above mentioned pokes can also come in handy for restoring old programs that are constantly getting load errors. Load the program, type in the pokes and you will find most of the time your program is restored to health.

I hope that I have give you enough to work on until the next issue, but till then I am still keen to hear from you if you have any questions or information you would like to share with us. □





GET SERIOUS

by Dirk Williams

As much as I hate introductions, I suppose one is required for Commodore Review's new column. Get Serious will be an application/business oriented column which will handle product reviews of a business nature, explain how databases, spreadsheets and the like, work and what they can do. It will include information for our technically oriented readers.

The database

The database has been described as just about the most useful thing since two fingers were invented. There does, however, seem to be a definite lack of understanding on exactly what a database is and what it can do.

A database is basically a computer controlled information manager/filing system which allows you to store information and retrieve it very quickly and easily. There are several categories of databases:

Specific task databases such as accounting packages (they store and handle accounts information),

Prestructured databases – their features are custom tailored to one particular application, such as mailing list programs and share register packages,

Modifiable databases – the topic of today's discussion. These databases allow you to structure the information you want (i.e. choose how each page of information will look and what it will contain), and the reports (print outs).

Remote databases, which are generally accessed by modems (the Australian Beginning and Dialog are examples of these) and often allow more than one user at once. These databases contain their own information which the user can have access to and perform searches (analogues to research) on. These are a powerful and, I believe, comparatively neglected resource which I will be discussing next month.

As mentioned, the type of database we are dealing with in this article is the modifiable database. The modifiable database can be seen as a super efficient filing system which has a page of information for each entry. These pages are referred to as records. A page may contain any number and arrangement of pieces of information (like name or suburb) up to the limit of the modifiable database program you are using.

These pieces of information are called fields. When you set up your database, you generally must tell it how long each field will be (i.e. the maximum number of characters you can type into a field), and may additionally give it a character type. By character type, we mean such things as alphanumeric (you can enter numbers and letters in this field), numeric (numbers only, such as you may use on a post code field), logical (this type is usually a simple yes/no, on/off field, e.g. married).

The hard part for most newcomers to the database is the set up stage. This is when we tell the program how our records will load, what they will contain and often it is necessary to set up reports (e.g labels). Setting up your database need only be done once for each database unless you need to modify the structure of an existing one. It is therefore very wise to sit down with a pen and paper and work out exactly what you want to store in each record and what kinds of report/printout you want.

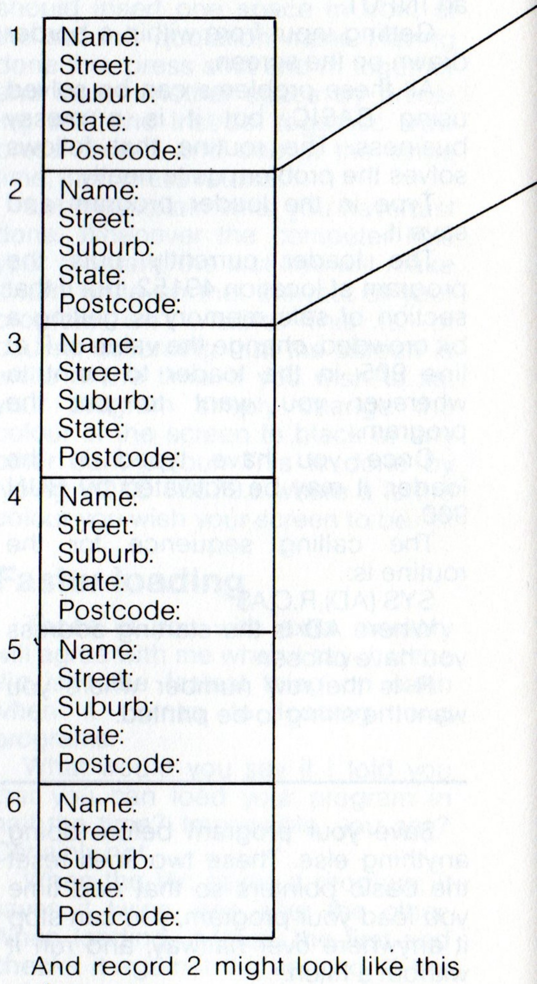
Only after you have set your database up may you begin to enter information or records into it.

Here is an example of a database – the common address book:

Let us pretend we want to define our database as a name and address keeper. We have decided that we want the following information for each entry:

Name maximum length 3Ø characters, alphanumeric (because we want to be able to type letters in this field.)
Street maximum length 3Ø characters, alphanumeric
State maximum 1Ø characters, alphanumeric
Post code maximum 4 characters, numeric

So our database would appear something like this if we had a few stored on disk or tape:



1 Name:
Street:
Suburb:
State:
Postcode:

2 Name:
Street:
Suburb:
State:
Postcode:

3 Name:
Street:
Suburb:
State:
Postcode:

4 Name:
Street:
Suburb:
State:
Postcode:

5 Name:
Street:
Suburb:
State:
Postcode:

6 Name:
Street:
Suburb:
State:
Postcode:

And record 2 might look like this on the screen.

LAST Name:
Street:
Suburb:
State:
Postcode:

Notice how our record screen can contain writing (prompts). However, we must type these on the screen when we are defining our database.

We do not have space this month

to show exactly how you do the setting up (which buttons to press and when) with a real live database program (we will tell you how it is done with Superbase 64 next month) – but this article should explain enough to get you started. Anyway, back to our discussion.

The real power of the database, once you have set it up, is you can forget where each record is (e.g. you would not have to remember that Andrew Farrell was Record 2 in our above example), rather, you can perform searches. The result of a search may be just one record. So I might have typed in Andrew Farrell in the name field in search mode and found Record 2, and unless there was another Andrew Farrell (be thankful) on our database, that is all you would find. Alternatively, I might have entered NSW as the state in search mode and I would have founnd all the people in NSW on our database.

Name: Andrew Farrell RECORD 2
Street: 24 Rubber-
appliance Rd
Suburb: Latex
State: N.S.W.
Postcode: 2000

Search mode, sometimes called report mode, will often have options such as reports (a label is a report) that cause certain things to be printed in certain places on our printer, and sorts (eg Sort on Name) which sorts the recalled records before it prints them. If we did, for way of a change, want to print a label for all people in NSW sorted on Name we would enter NSW as the state in search mode, select label type report/printout (presumably we would have set one up by now) and select sorting on the Name: Field of our database.

Thinking readers will have noted that our list would be sorted on the first names then surnames (eg Andrew Farrell would be considered an "A") – so we have found a shortcoming of our database. If we have no need to ever produce an alphabetically sorted list then our database is still OK – otherwise we would need to have included a SURNAME: field when we set our database up.

Setting up reports (printer forms, etc) is much the same as setting up our database. We simply decide for each form what fields we want printed and when we want them printed.

Please note that when you use a database program (like Superbase 64) to create a database, you are not limited to that database alone. If you wanted a database of Names and Addresses and a database of Records and their locations you would need to have two Superbase 64 programs. Rather you would set up the two databases on two different disks and after loading Superbase 64 you would put the disk of database of your choice in the disk drive and use it.

I hope I have made the world of databases a little easier to understand. Next month, as explained before, we will be showing you exactly how to create a database using the Superbase 64 program – we will also go into the abilities and use of remote/commercial databases. If you are technically minded then read on.

On the technical side

If you are writing a database program of some sort you would have come across the problem of accessing the stored information. These are several methods commonly employed to facilitate rapid recall, each with its own good and bad points. Here is a brief summary of how they work (a future issue will contain a sample database program).

Scan Searching: fine for small databases and requiring no memory overhead, basically means a record by record search of the file for eligible entries. In large databases (more than 1ØØ records) this becomes painfully slow – if each scan took Ø.5 second then in a database of only 1Ø Ø entries you would be waiting almost a full minute, if we refer back to our example database who would waste a minute for Andrew Farrell?

Memory Table Searching: This method largely depends on how much memory your computer has and how many records you have. It essentially means that for each record a string is stored in memory and that string is called the search key. Because in memory searches are so fast it is often a rapid way of performing searches (eg 5 seconds to search 5ØØ entries). It is, however a chewer upper of RAM and if you do not have much to spare, give this one a miss.

5	Andrew Farrell 2083	1	5
3	Glady's Peters 2694	2	1
1	Garry Baby 2147	3	3
2	Sara Smith 3012	4	2
4	Agnes Andrew 2001	5	4

Indexed Searching: This essentially means that a sorted index is stored on disk which "points" to the relative record in the database. Without using very much more complex index structures, you must re/sort the index after every new entry and re-write it to disk – but if you have relatively occasional entries (after initially entering the first records) then the added time waiting while entering may be worth the increased speed at search time. You may have more than one index file in your database (eg one may be sorted on name, the other on post code).

Note how the index files simply contain a number which "points" to the relevant record. Thus if we were to recall the list alphabetically we would simply read our name index file and then read the record. If we were to search, we would enter the name to search for, recall the middle name index record and if our comparison string was less, then we would know we have to go back in our index file to find it, if it was equal then we have found it, if it is greater then we must look further on in our index file. If it was greater or less, then we would jump (look at the record pointed to by 1.5 of the first jump, in other words we would look at a position one-quarter of the way through or (if greater) three-quarters of the way through. By doing this comparison and halving the distance jumped each time, we will find our record in the least time. This is called "binary searching".

Hash tables: use "magic" numbers which are calculated from the name key to directly point to the record in question. We will have to devote a whole article to these, as they are quite sophisticated. •□

Program Directory

by Geoff Rayner

Ever forgotten where you wrote the counter value when you saved a program on a cassette? Well, I have experienced this and have spent quite a bit of time trying to locate the errant program. So I thought it was about time I wrote a program to store this information at the start of every tape.

When you use a new blank cassette you load this directory program into the computer and save it again at the start of the new tape. You will notice in the data statements that there is "dummy" data. This is to take up space so when you save another program, noting the values of the counter at the start and the end of the program, you can then go back and reload the directory program. And you can change the data statement to read the new program name and counter values.

This "dummy" data must be used so when we resave this directory we do not overwrite another program. Note: Insert blanks where the program name does not take up all allowed space in the data statement.

E.G. Line 1ØØ

Reads 1ØØ DATAAAAAAAAAAAAAAAAAAA,ØØØ,ØØØ,BBBBBBBBBBBBBBBBBB,ØØØ,ØØØ

Change to 1ØØ DATA SPACE ALIEN ,ØØ2,Ø28,BBBBBBBBBBBBBBBBBB,Ø3Ø,ØØØ

Now when we are looking for a certain program just load the directory and find the counter value of the desired program then reset counter and fast forward to your program.

I hope this will help you, as it has certainly saved me some time. □

DIRECTORY LISTING

```
10     REM DIRECTORY (C) GEOFF RAYNER
20     PRINT" CLR " :LO=0 :V=21 :W=0
30     POKE 36879,93 : POKE 36878,15 : C=38399
40     PRINT" (rvs on)(23 spaces)(rvs off)"
50     PRINT" (21 csr down)(rvs on)(21 spaces)(rvs off)(25 csr up)"
60     FOR Y = 1 TO 2 :FOR X= 1 TO 22
70     POKE C+X,CO : CO=CO+1 : IF CO>8 THEN CO=0
80     IF CO=5 THEN CO=CO+1
90     POKE 36876,(X*2+130):FOR Z=1 to 20 :NEXT Z
100    NEXT X
110    C=38883 :D=8163 : NEXT Y
120    POKE 36876,0 :PRINT"(blk)"
130    A$="    **DIRECTORY**  ": B$="(20 spaces)-------------      "
140    FOR X=1 TO 18 :PRINT RIGHT$(A$,X)
150    PRINT"(white)(csr up)":PRINT MID$(B$,W+X,21)
160    PRINT"(3 csr up)(blk)"
170    POKE 36876,200:FOR Q =1 TO 70:NEXT Q:POKE 36876,0
180    NEXT X:PRINT
200    FOR KT=1 TO 30:LO=LO+1
210    READ N$,BG$,FI$
220    IF LO>5 THEN LO=1:GOSUB 1000
230    PRINT"(blk)"
240    PRINT TAB(2)N$:PRINT TAB(4);"(white)"BG$;"-"FI$
250    POKE 36875,230:FOR B=1 TO 30:NEXT B :POKE 36875,0
260    NEXT KT
300    FOR F=1 TO 7-LO :PRINT"(22 spaces)":NEXT F
310    PRINT"(blue) RUN AGAIN (Y,N)"
320    GET Q$: IF Q$="" THEN320
330    IF Q$="Y" THEN RESTORE : GOTO20
340    GOTO 9000
1000   PRINT
1010   PRINT"   (rvs on)(yellow) HIT ANY KEY (rvs off)(blk)(csr up)"
1020   FOR HU=1 TO 100:NEXT HU
1030   PRINT"    HIT ANY KEY     (csr up)"
1040   FOR HU=1 TO 100:NEXT HU
1050   GET Q$:IF Q$="" THEN 1010
1060   PRINT"(17 csr up)": RETURN
2000   REM DATA FOR PROGRAMS
2010   DATA AAAAAAAAAAAAAAAAAAA,000,000,BBBBBBBBBBBBBBBBBB,000,000
2020   DATA CCCCCCCCCCCCCCCCCC,000,000,DDDDDDDDDDDDDDDDDD,000,000
###Continue this until you have enough space for each program on the tape
    e.g 30 for one side of a C60 tape

9000   POKE 36879,27:PRINT"(blue)(clr)(2 csr down)(rvs on) RESET COUNTER BEFORE
         SEARCH"
9999   END
```

Starting with disk drives— Part III

Paul Blair

To provide a better understanding of the housekeeping functions in a 1541 disk drive, it is necessary to understand some of the things that go on in Track 18. Before starting, I will assume you know a little about the structure of 8-bit bytes. If you don't, read pages 215-219 of the Programmer's Reference Guide (PRG) first. As far as possible I will use decimal notation in these notes.

To start with, let's look at the 'front page', Track 18 Sector Ø. If you didn't know already, each sector (or block) on a 1541 disk can store 256 bytes. The usual convention is that the first two bytes on any sector hold the 'chain bytes' to the sector that follows the one being looked at. It happens (by the design of the disk operating system, DOS) that Track 18 Sector 1 follows Track 18 Sector Ø, so the first two bytes on Track 18 Sector Ø will be '18' and '1'. That eliminates 2 bytes – let's go on to the other 254.

A picture would probably help, so let me put down in tabular form the sort of thing you will find in the first 144 bytes of Track 18 Sector Ø. All values are in decimal.

Byte #	Ø	1	2	3	4	5	6	7
Ø	18	Ø	65	Ø	21	255	255	31
8	21	255	255	31	21	255	255	31
16	21	255	255	31	21	255	255	31
..								
..								
64	21	255	255	31	21	255	255	31
72	17	252	255	7	19	255	255	7
8Ø	19	255	255	7	19	255	255	7
..								
..								
96	19	255	255	7	19	255	255	7
1Ø4	18	255	255	3	18	255	255	3
112	18	255	255	3	18	255	255	3
12Ø	18	255	255	3	17	255	255	1
128	17	255	255	1	17	255	255	1
136	17	255	255	1	17	255	255	1

That odd assortment of figures is the Block Allocation Map (BAM), which is a critical part of disk management. The BAM is where DOS stores information about which sectors are in use (so can't be used right now for any fresh storage) and which are not (so are available for use).

When a disk is NEWed, there are 664 sectors available to the user for storage. So how can information about 664 sectors be stored in 142 (remember the chain bytes!) locations in the BAM? The answer is that the information is encoded in a cunning way.

Any 8-bit byte in the Commodore 64 can store any number from Ø to 255, 256 numbers in all. The eight bits that make a byte take on a special meaning in the BAM. Starting from byte 4 (see above), each group of 4 bytes 'remembers' the contents of a track. Bytes 4-7 look after Track 1, 8-11 do the same for Track 2, and so on. Bytes 72-75 look after Track 18. See if you can work out which group looks after Track 26 (no prizes!) The bytes are used like this.

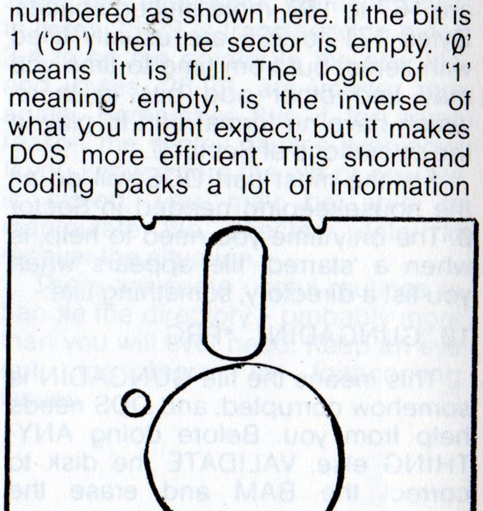
The first byte in any group of 4 is the big picture – how many sectors are free in the particular track. In this case, Track 1 has 21 (i.e., all) sectors free. The BAM above reflects a freshly NEWed disk, so all blocks (except in Track 18) are empty. As the disk has been NEWed, some of Track 18 is being used. This is reflected in bytes 72 and 73.

Notice too, that the number of sectors decreases as we read from Track 1 (21 sectors) to Track 35 (17). Commodore has taken advantage of the longer tracks at the outer edge of the disk to increase the number of sectors in them.

Now the detailed picture. The next three bytes in each group of four mark precisely which sectors are in use. If we stretch out the 3 bytes as 24 bits, we have this:-

```
           Byte 5=255
        x x x x x x x x
        1 1 1 1 1 1 1 1

Sctr    7 6 5 4 3 2 1 Ø


              Byte 6=255
           x  x  x  x  x  x  x  x
           1  1  1  1  1  1  1  1

          15    13    11    9
             14    12    1Ø    8


                    Byte 7=31
                 x x x x x x. x x
                 Ø Ø Ø 1 1 1 1 1

                 spare   19   17
                       2Ø   18   16
```

Each bit represents 1 sector, numbered as shown here. If the bit is 1 ('on') then the sector is empty. 'Ø' means it is full. The logic of '1' meaning empty, is the inverse of what you might expect, but it makes DOS more efficient. This shorthand coding packs a lot of information



into a small space. Just for fun, what would be in byte 5 if sectors 2, 3 and 4 were full?

The next part of Track 18 Sector Ø holds the disk name (which can be no longer than 16 characters). If you use a shorter name, DOS puts CHR$(16Ø) in the unused locations. We now have:-

Byte #	Ø	1	2	3	4	5	6	7
144	T	E	S	T	D	I	S	K
152	1	9	8	4	+	+	+	+

For simplicity, I have shown the disk name in plain English, but in fact it would be stored in ASCII (see Appendix C of PRG if this is a new one to you), and I have used '+' to show the locations where CHR$(16Ø) will be used to 'pad out' the name to 16 characters.

The next two bytes (16Ø and 161) are always CHR$(16Ø) or $AØ if you are into hexadecimal notation. By way of explanation, CHR$(16Ø) is usually termed 'shifted space' – not that it's gone anywhere, but it is the character you generate if you hold down the SHIFT key while pressing the SPACE bar.

Bytes 162 and 163 hold the disk ID that you selected when NEWing the disk. Byte 164 is another shifted space, while 165 and 166 hold the DOS version, which is 2A in the case of a 1541 drive. You see this at the right-hand end of the reversed 'header' line every time you read the disk directory. The last bytes used are 167 to 16Ø, more shifted spaces, Bytes 171 to 255 are usually filled with zeros, but from time to time you may find other odd bits in there, which have no formal part to play in the operations of Sector Ø.

For the most part DOS will do all the housekeeping needed in Sector Ø. The only time you need to help, is when a 'starred' file appears when you list a directory, something like:-

```
1Ø "GUNGADIN"  *PRG
```

This means the file GUNGADIN is somehow corrupted, and DOS needs help from you. Before doing ANYTHING else, VALIDATE the disk to correct the BAM and erase the corrupted file.

DOS also takes care of other matters. When you read or write a disk, DOS reads the BAM into its own memory. Each subsequent time you use the disk, DOS checks the ID again. If the ID has not changed from last time, DOS doesn't reread the BAM. It is for this reason that you are warned (and warned and warned . . .) against using duplicate ID's. If you use a mainly empty disk, then use a nearly full disk with the same ID. the DOS will remember the original BAM, and write to the second disk based on the free sector pattern of the first. The nearly full disk will probably not have free sectors where BAM thinks it does – disaster.

You may interfere with Track 18 Sector Ø to some extent. The most common feature is to change the disk name. This is done by using an advanced disk command to write a new name to bytes 144 to 159. There are utilities about that automate the process for you.

You may also alter the disk ID, but be warned. The ID is also written in many other places on the disk where you can't see it (or, for that matter, alter it). If you do change the ID in Sector O, it will be only a cosmetic change. DOS will ignore this change when deciding the true ID of your disk, with possible disaster as discussed above. If you are in any doubt about the true ID of a disk try this little routine, which reads the true disk ID stored in 1541 disk RAM.

```
1ØØ OPEN 15,8,15,"I"
11Ø FOR X = Ø TO 1
12Ø PRINT#15,"M-R";CHR$(22+X);
    CHR$(Ø)
13Ø GET#15,A$: A$ = A$ + CHR$(Ø
    )
14Ø IF X = Ø THEN PRINT "TRUE ID
    = ";
15Ø PRINT A$;
16Ø NEXT : PRINT : CLOSE 15
```

Track 18 – Directory

We now move on to the actual directory entries. Storage of these starts on Track 18 Sector 1. Commodore drives use an optimal search technique to save time during disk access. The selection algorithm used means the sequence of sectors that follow Sector 1 goes something like; Sectors 4, 7, 1Ø, 13, 16, 19, 2, 5, 8 . . . etc. In this way, the DOS doesn't have to wait a full disk rotation to find the next sector it wants. Quite nifty.

Sectors again hold 256 bytes, 254 when the link bytes are subtracted. Eight file entries are stored in each sector. The first entry uses 3Ø bytes, and the next seven each use 32. As before, a picture would help.

Byte#	Ø	1	2	3	4	5	6	7
Ø	18	4	13Ø	17	Ø	T	E	S
8	T	F	I	L	E	1	9	8
16	4	+	+	+	+	Ø	Ø	Ø
24	Ø	Ø	Ø	Ø	Ø	Ø	5	Ø
32	Ø	Ø	129	19	Ø	T	E	S
4Ø	T	D	A	T	A	1	9	8
48	4	A	+	+	+	Ø	Ø	Ø
56	Ø	Ø	Ø	Ø	Ø	Ø	1Ø	Ø
64	Ø	Ø	132	22	5	R	E	L
72	A	T	I	V	E	F	I	L
8Ø	E	+	+	+	+	12	14	1ØØ
88	Ø	Ø	Ø	Ø	Ø	Ø	6Ø	Ø

Let's stop there. The pattern of each entry repeats itself in blocks as shown between the dotted lines, so let's look at that pattern. Please note, my description of byte numbers will differ slightly from that found in Commodore documentation (eg, page 56 of the 1541 Manual) for reasons of clarity.

Skip over the first two bytes in each panel in the table. I take each of these panels as being a directory entry block, but Commodore don't seem to think the first two bytes are! Apart from the link bytes before the first entry, the two bytes (32, 33, 64, 65, and so on) are unused.

The next byte in each entry is the file type, which are numbered Ø to 4:-

File#	Type	Abrvn
Ø	Deleted	DEL
1	Sequential	SEQ
2	Program	PRG
3	User	USR
4	Relative	REL

When the file is being written to disk, the DOS uses values Ø-4 for internal management. When the file is closed properly, the file# has 128 added to it to indicate a successful 'save'. Byte 2 shows a program file, byte 34 a sequential file and byte 66 a relative file. If any file is SCRATCHed, the type byte should revert to Ø (DEL) but it may not always, despite this being the intent of DOS. Provided the byte contains less than 128, DOS seems to ignore the error and take the file as being erased.

The next pattern pair (bytes 3, 4, 35, 36, etc) tell DOS where the actual file storage starts. The first sector of TESTFILE1984 is stored at Track 17 Sector Ø, TESTDATA1984 at Track 19 Sector Ø, and RELATIVEFILE at Track 22 Sector 5. Using this information, DOS can go to the file that has been chosen. From there, it reads the first 2 bytes of the first sector to find the second sector, and so on.

Bytes 5-2Ø, 37-52, 69-84 and so on are the filenames given to the specific files. As with the name of the disk on Track 18 Sector Ø, the names are 'padded' out to 16 characters with CHR$(16Ø). SCRATCH does not erase these characters.

For most files, there is nothing more until bytes 3Ø-31, 62-63 etc. These bytes store the number of blocks or sectors used to store the file. The bytes are in low byte-high byte order, so byte 3Ø holds the low byte and 31 the high byte. The total number of blocks is thus (byte 31 * 256 + byte 3Ø), or the correct byte numbers for the file in question.

Relative files use 3 bytes to store information peculiar to the file type. If all the files in the example were REL type, bytes 23,55,87 etc would hold the length of record selected when the file was set up. Byte 87 above indicates a length of 100 bytes per record. REL files also use a sequence of sectors ('side sectors') to assist fast access. Bytes 85 and 86 indicate that the side sectors for this file start at Track 12 Sector 14.

Two other bytes are used for 'save with replace', such as when you use SAVE"@0:MYFILE",8. This syntax has a mixed reputation, but if you do use it, it works like this. DOS firstly writes a new file in some free disk location, remembering the track and sector of the first block it uses in bytes 28-29, 60-61 etc. Then, if the SAVE was successful, the information in bytes 28-29 is transferred to bytes 3-4 to be the pointer to the new file, and bytes 28-29 are made zero again. Remember this if for some reason the syntax causes problems. If you know where the old or new file is stored, you may be able to manipulate the directory entry to recover the situation.

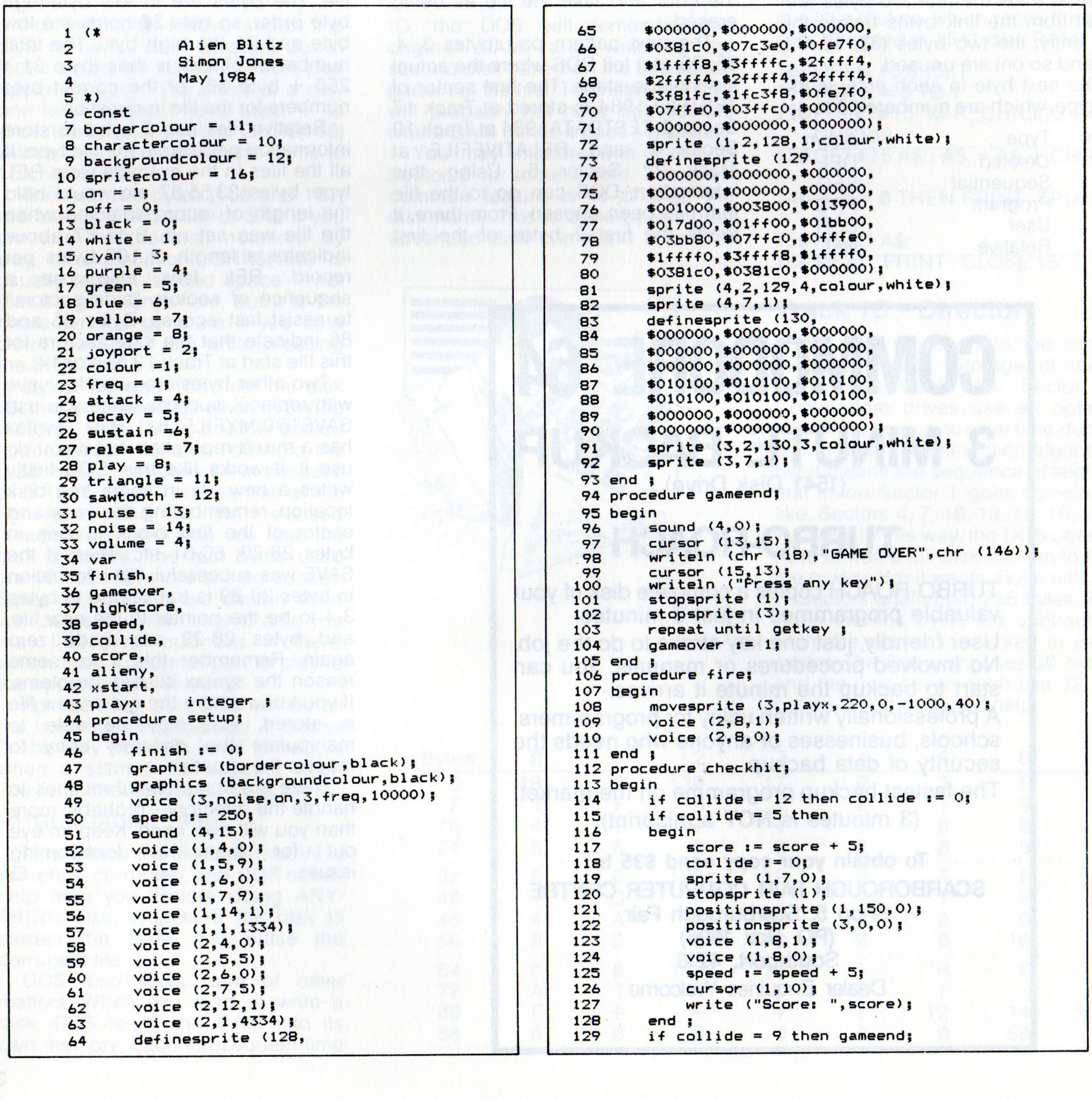
There are some useful routines to handle the directory – probably more than you will ever need. Keep an eye out for them in forthcoming issues. □

Alien Blitz

by Simon Jones

Alien Blitz is a game program for the Commodore 64. It is written using G-Pascal. The program is a good example of the powerful commands built into G-Pascal to control sprites. These features, along with other graphics and sound commands, allow game programs to be easily written. The program also shows how the structure of Pascal programs is very different to BASIC programs.

You control your fighter with the joystick in port two. To fire your laser press the fire button on the joystick. Your mission is to kill the aliens before they can reach the bottom of the screen. The more aliens you kill the faster the surviving aliens move. Should just one alien reach the bottom of the screen, the game ends. □



```
(*
            Alien Blitz
            Simon Jones
            May 1984
*)
const
bordercolour = 11;
charactercolour = 10;
backgroundcolour = 12;
spritecolour = 16;
on = 1;
off = 0;
black = 0;
white = 1;
cyan = 3;
purple = 4;
green = 5;
blue = 6;
yellow = 7;
orange = 8;
joyport = 2;
colour =1;
freq = 1;
attack = 4;
decay = 5;
sustain =6;
release = 7;
play = 8;
triangle = 11;
sawtooth = 12;
pulse = 13;
noise = 14;
volume = 4;
var
finish,
gameover,
highscore,
speed,
collide,
score,
alieny,
xstart,
playx      integer ;
procedure setup;
begin
    finish := 0;
    graphics (bordercolour,black);
    graphics (backgroundcolour,black);
    voice (3,noise,on,3,freq,10000);
    speed := 250;
    sound (4,15);
    voice (1,4,0);
    voice (1,5,9);
    voice (1,6,0);
    voice (1,7,9);
    voice (1,14,1);
    voice (1,1,1334);
    voice (2,4,0);
    voice (2,5,5);
    voice (2,6,0);
    voice (2,7,5);
    voice (2,12,1);
    voice (2,1,4334);
    definesprite (128,
    $000000,$000000,$000000,
    $0381c0,$07c3e0,$0fe7f0,
    $1ffff8,$3ffffc,$2ffff4,
    $2ffff4,$2ffff4,$2ffff4,
    $3f81fc,$1fc3f8,$0fe7f0,
    $07ffe0,$03ffc0,$000000,
    $000000,$000000,$000000);
    sprite (1,2,128,1,colour,white);
    definesprite (129,
    $000000,$000000,$000000,
    $000000,$000000,$000000,
    $001000,$003800,$013900,
    $017d00,$01ff00,$01bb00,
    $03bb80,$07ffc0,$0fffe0,
    $1ffff0,$3ffff8,$1ffff0,
    $0381c0,$0381c0,$000000);
    sprite (4,2,129,4,colour,white);
    sprite (4,7,1);
    definesprite (130,
    $000000,$000000,$000000,
    $000000,$000000,$000000,
    $000000,$000000,$000000,
    $010100,$010100,$010100,
    $010100,$010100,$010100,
    $000000,$000000,$000000,
    $000000,$000000,$000000);
    sprite (3,2,130,3,colour,white);
    sprite (3,7,1);
end ;
procedure gameend;
begin
    sound (4,0);
    cursor (13,15);
    writeln (chr (18),"GAME OVER",chr (146));
    cursor (15,13);
    writeln ("Press any key");
    stopsprite (1);
    stopsprite (3);
    repeat until getkey ;
    gameover := 1;
end ;
procedure fire;
begin
    movesprite (3,playx,220,0,-1000,40);
    voice (2,8,1);
    voice (2,8,0);
end ;
procedure checkit;
begin
    if collide = 12 then collide := 0;
    if collide = 5 then
    begin
        score := score + 5;
        collide := 0;
        sprite (1,7,0);
        stopsprite (1);
        positionsprite (1,150,0);
        positionsprite (3,0,0);
        voice (1,8,1);
        voice (1,8,0);
        speed := speed + 5;
        cursor (1,10);
        write ("Score: ",score);
    end ;
    if collide = 9 then gameend;
```

```
130 end ;
131 procedure moveplayer;
132 begin
133     if joystick (joyport) and 4 then playx := playx - 4;
134     if joystick (joyport) and 8 then playx := playx + 4;
135     if playx < 25 then playx := 25;
136     if playx > 320 then playx := 320;
137     positionsprite (4,playx,220);
138     if joystick (joyport) and 16 then if spritestatus
      (3) = 0 then
139     fire;
140     if spritestatus (3) = 0 then positionsprite (3,0,0);
141 end ;
142 procedure movealien;
143 begin
144     if spritestatus (1) = 0 then
145     begin
146         repeat
147         xstart := random ;
148         until (xstart >25) and (xstart < 320);
149         movesprite (1,xstart,50,0,speed,200);
150     end ;
151     alieny := spritey (1);
152     if alieny > 220 then gameend;
153 end ;
154 procedure titlepage;
155 begin
156     sound (4,0);
157     positionsprite (1,0,0);
158     positionsprite (3,0,0);
159     positionsprite (4,0,0);
160     writeln (chr (147));
161     cursor (5,15);
162     write (chr (18),"ALIEN BLITZ",chr (146));
163     cursor (8,10);
```

```
164     write ("Written By:");
165     cursor (10,22);
166     write ("S. Jones");
167     cursor (20,9);
168     write ("Press any key to start");
169     cursor (14,10);
170     write ("High score: ",highscore);
171     cursor (16,10);
172     write ("Last score: ",score);
173     repeat until getkey ;
174     sound (4,15);
175 end ;
176 procedure alienblitz;
177 begin
178     setup;
179     titlepage;
180     write (chr (147));
181     gameover := 0;
182     score := 0;
183     repeat
184         moveplayer;
185         movealien;
186         collide := spritecollide ;
187         if (collide <> 0) and (collide <> 12)
          then checkhit;
188     until gameover = 1;
189 end ;
190 begin
191     highscore := 0;
192     score := 0;
193     repeat
194         alienblitz;
195         if score > highscore then highscore := score;
196     until finish;
197 end .
```

TECH TIPS

by Andrew Karagiannis

Well, it seems last month went over some of your heads so I'll tone it down a bit. This month we'll start off with a few meaningful queries and then as usual back to the basics with an enlightening look at your collective innards!

?

Dear Andrew,

Recently I acquired (at a fantastic price) a dot matrix printer and RS-232 cartridge for my computer. A friend made up a lead for it but we can't get it to work with our software, HELP!

Yuk! I've never come across anything messier than RS-232 on C-64 and VIC-2Ø computers. Just look in your programmers reference manual to see how (as usual) Commodore has turned an industry standard into a programmers nightmare. Bad dreams aside, assuming your printer and cable are OK, then you probably have a D.N.H. (Device Number Headache). This beastie only affects Commodore owners due to Commodore using a serial IEEE buss (outside Commodores' collective intelligence there is no such thing as IEEE Serial it is really a parallel industry nightmare I mean standard.) Commodore printers use Device No. 4 as standard and most software is written for this. Some software allows you to select the device number (3 for RS-232) but most don't.

What all this means in terms of Life and the Universe is your fantastic price isn't really fantastic at all. However, if your printer has a centronics interface as well (and most do, since serial is normally an option), you can buy commercial interfaces which let the computer see the printer as device No. 4. Two that I know of are the Data-2Ø (for text only) or the Printron (Commodore graphics as well).

?

Dear Andrew,

I am trying to convert my menu driven programs to operate with a light pen I purchased some time ago. As far as I can see, the routines wrote are correct but the light pen doesn't seem to be very accurate. Have I done something wrong or is the light pen faulty or what?

Neither. Your routines seem OK and the fact the light pen works at all would suggest it is fine. The problem lies in the inherent inaccuracy of the lightpen when a TV set is used as a monitor. This is never very accurate because the picture on a TV is not as stable as a monitor and causes the light pen to falsely trigger. Most light pens come with software to overcome this problem. The more elaborate software does statistical deviation and momentum calculation to the values in the light pen registers, to get the correct position. A simple routine you could use would be to read each register 1Ø times using a FOR.. Next Loop then divide by 1Ø and change to an integer value to get the average value. This should give accurate enough values for your application.

?

Dear Andrew,

Whilst in America I purchased a modem for my C-64 computer. I've been told that it is illegal to use it on Australian Telecom lines. Is this true and if so is there any way I can get around it?

America uses the Bell standard for modems, whereas in Australia we use the C.C.I.T standard. Telecom does not give approval to Bell standard modems and you could cop a hefty fine for using one. Believe me, they DO have ways of finding out when incompatible items are put on their lines. The other problem you would have is that since everybody else would be using C.C.I.T standard modems and databases you wouldn't be able to communicate with anyone.

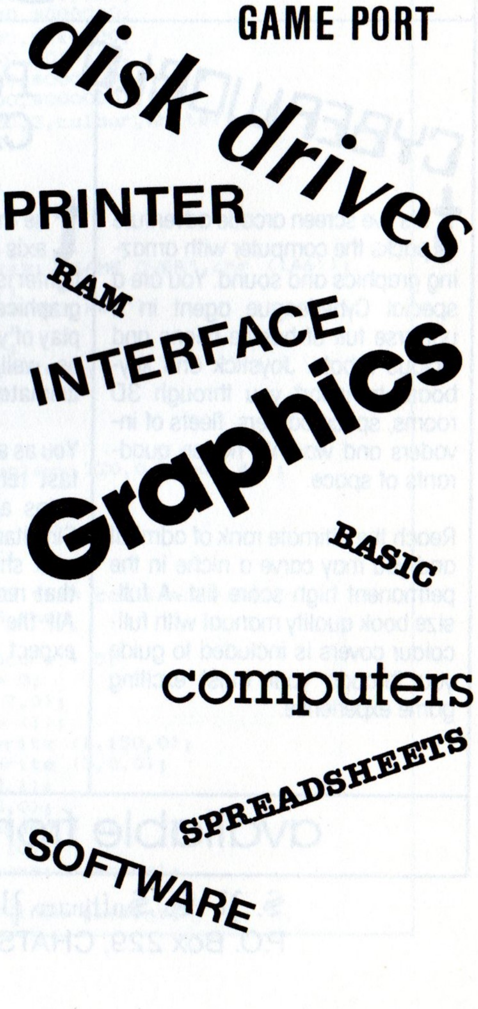
?

Dear Andrew,

I would like to connect a green screen monitor to my C-64 since I'm sick of the quality I'm getting on a TV. I know colour monitors come with a lead for C-64 but can a lead be made up for a green screen?

Yes, a suitable cable can be made up quite easily. You will need a length of co-axial cable, an RCA phone plug and a 5 pin DIN plug. Connect the braid of the coax to the shield of the R.C.A. plug and the inner conductor to the centre lug of the R.C.A. plug. At the other end connect the braid and centre conductor to pins 2 & 4 respectively. This should work with most green screen monitors I know of.

Well it seems those enquiries have taken up all my space this month, so I promise to get back into your innards next month. Until then, Goodbye! □



View from the Hold

Rat's complaint

This Rat realises nobody is going to listen to him, but he feels a strong protest must be made about the quality of manuals that come with programs these days.

The programs themselves can be excellent. The hardware is often of the highest quality.

But the manuals appear to have been written by a macrocephalic troglodyte dyslexic.

Bearing in mind that this Rat has been playing around with computers for a long time, you would think he would be able to understand what the programs' instructions are about.

With any set of instructions, it is odds-on it will require at least five readings before you can even start to understand what the hell they are on about.

Part of the problem may be the fact that the instructions are the last thing written. Somebody devises a piece of software or hardware and leaps around in jubilation.

At the last moment they realise they have to provide some sort of technical manual. As far as I can make out, they give the job to the office boy or to somebody in the packing department.

It is all a bit of a disaster area. I thought the problem lay with this Rat until I investigated. Everybody complains about manuals.

Manufacturers have really got to take their finger out.

They should clearly understand that manuals are the only way in which they can talk directly to the customer.

It does not matter if you have the best product in the history of the human race. If a customer can not get it up and running in a reasonable time, he will be less than pleased. And he will speak badly of you and your product.

He may, if he wants to go too far, become slanderous. This Rat never becomes slanderous.

One of the best advertising media for machines and programs is word-of-mouth advertising. It is free, and when it is positive it can be wonderful. On the other hand if it is negative it can be an utter disaster.

In this Rat's humble opinion the manuals supplied by Commodore are better than competent.

Indeed having read the manuals for a lot of other computers I would go so far as to say that they are in the top few.

It is also true that the manuals that come with any products from Handic are of an excellent quality.

This may be because the company concerned is Swedish, and as they have a struggle to get the manuals together in English, they bother to get it right.

And the English company that produces Easy Writer, although they are, sadly, afflicted with a touch of the Monty Pythons, produce manuals which are literate and relatively easy to read.

As for the rest, it is difficult to say one good word for them.

Manuals should be concise, easy to read and also at the same time be useful as reference works. Most manuals for software and hardware do not live up to these ideals. Which is a pity.

Perhaps we should start a school for manual writers.

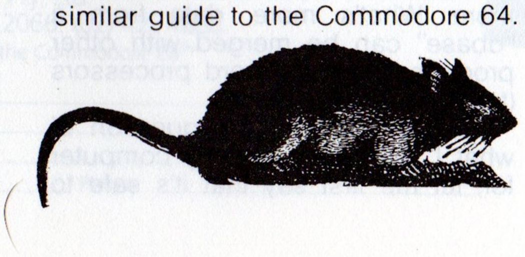
Getting it fixed

A friend of mine who owns an Apple computer has got hold of a book by a company called Dataline on "How to Repair the Apple Computer".

It is a thing of wondrous beauty. Even this Rat can understand it.

Following fairly uncomplicated directions, you can fix most of the simple problems that arise on the Apple.

I have never seen or heard of a similar guide to the Commodore 64.



And yet it would appear to me to be an obvious best selling item.

This Rat does not know one end of a screwdriver from another, but is absolutely convinced that it would be a lot easier to fix small problems on the Commodore 64 himself than to send it back to Commodore headquarters and then wait for it to return.

Andrew "Anything for a Dollar" Farrell has never heard of such a book, and so, with a little help from his friends, he is going to write one which will be published by this magazine, first as a part work serial and then as a full-scale book.

It will, he assures me, be aimed at the absolute beginner. "First get hold of the screwdriver at the handle end with the pointy bit facing away from you."

If there is anyone out there who feels he or she has something useful to contribute, then let us hear from you.

Of course, Commodore owners would only be able to start tampering with their machines once the three month guarantee was out.

And it may be that Commodore would take a fairly dim and dusty view about supplying spare parts for happy home hackers to insert themselves. But these are all problems that can be overcome.

Facial problems

Speaking of Andrew Farrell, I must draw readers' attention to his face.

No doubt it is the sort of face that mothers like.

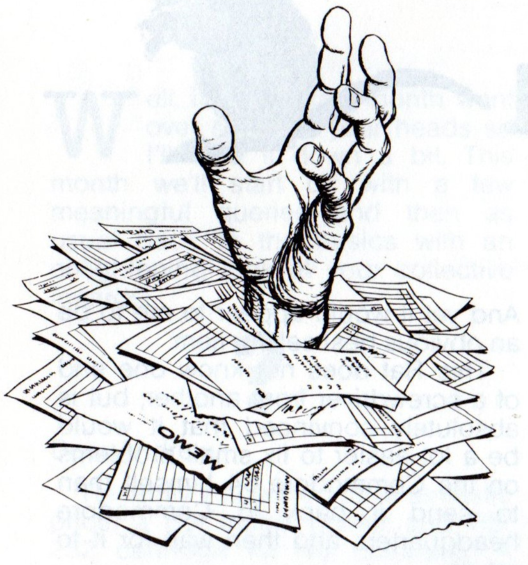
And it is probably the sort of face that fathers trust.

But every time you see it, it is scarred. And the excuses always appear, if I may say so, fairly weak.

Now this Rat is used to people running into trouble and ending up in rat fights. Yet every time I see the editor there is a new scar on his admittedly fairly well-worn visage.

The first time there was a slit across his upper lip which he said he got ice skating.

⇨



VIEW FROM THE HOLD

Ice skating?

If that cut was caused by ice skating, it could only have been by Torvill and Dean skating over his face.

No sooner had that cut healed, and he had once again started to semi-resemble a human being, than we see a slash across the front of his nose.

And how had this been caused, we wondered.

A major auto accident?

A collision in an aeroplane?

Not at all.

According to Andrew his nose received the scar when he was dancing.

Who does he think he is fooling?

Plainly these are marks inflicted on his face by irate readers who feel that the magazine is not being edited to their satisfaction. If this carries on, we shall seriously have to consider getting the lad a full-time bodyguard.

Pascal

In order to impress our readers with our erudition may we mention in passing that Pascal is named after Blaise Pascal, the famous French mathematician? Andrew Farrell still believes that Pascal wrote the language. As Blaise Pascal died in 1763 this would be a bit difficult. □

Wonder if it will do the washing up?

Micros have been on the world market since about 1976 and were adopted rapidly by the business community. As soon as prices started to fall, the home computer revolution began. Pet, Apple and TRS 8Ø computers pioneered this huge industry and these brands, although changed and updated, are still market leaders.

Now enough of this ancient history. I personally have been involved with micros for about four or five years and I am asked the same basic questions over and over again: "What do you do with your Commodore 64" or "Now I have got my Commodore 64, what can it really do?"

OK: for a start, here are the obvious answers.

Word processing is still not getting the recognition it deserves. Did you know, by using the C-64 you can edit a piece of text that's more than 15Ø,ØØØ characters (letters) long, which is about 25,ØØØ – 3Ø,ØØØ words and you can cut and paste sections of it (move or copy pieces from one place to another in the text), store it, get it back, print it, cross reference or search through it, and even get "Micro" to check your spelling.

"What the heck is a database?", and "What can use it for?"

Well, a data base is a "top of the line" electronic filing system. They can be used to store almost any type of information from a simple address book to an accounting system. Applications include: phone books, club records, invoicing, stock control, patient records, price lists and much more. What's more, data from a "dbase" can be merged with other programs such as word processors (to produce form letters).

Before I answer the question of what personally use my computer for, let me first say that it's safe to leave Micro running twentyfour and a half hours per day, (providing it's well ventilated).

Micro can talk to people in plain English and, oddly enough, people can even understand him. Micro can, if I ask him nicely, wake me in the morning and even have a pot of hot coffee ready when I get out of bed. Micro can, if requested, monitor my home security by giving me a verbal warning or by setting off a siren. If security has been broken Micro can tell me exactly where and when it happened. Micro can remind me of things that have to be done. He can also do things for me, like switching on and off my electric blanket and act as a sleep switch for my TV, radio or heater. He can announce the time when asked or volunteer it on the hour.

Micro helps me compose music. Not only that, he can control a whole professional music synthesiser and play back our creations all by himself. He even helps me listen to the radio. He can translate morse code, radio teletype and slow scan television.

Micro helps me keep in touch with other users and the world. He can send programs, word processing files, he can tell me what's on in town, give me the news, the weather forecast, Lotto results and even read The Australian Commodore Review. Micro can send telexes for me (without paying extra thousands of $$ for a telex machine). I can send electronic mail and programs all over the world. We can go to electronic shopping centres. On the way we can stop and chat to other Micro friends (and strangers).

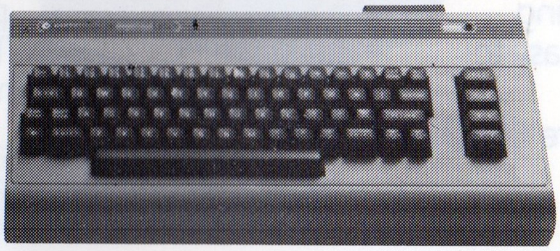
Now that you know what your computer can do for you, for heaven's sake, do it! Well, maybe not all of it, but I urge all Commodore 64 users at least to go out and buy a modem. □

Who's keeping up with Commodore?

The Commodore 64 is designed and priced so that you can pile on the peripherals. Like a disc drive, printer and even a printer plotter. You can own the Commodore 64 AND disc drive and printer for little more than you'd pay for many other computers alone.

With 64K memory, outstanding colour graphics, a music synthesiser to rival the professionals, a fantastic range of games cartridges and full upper and lower case keyboard.

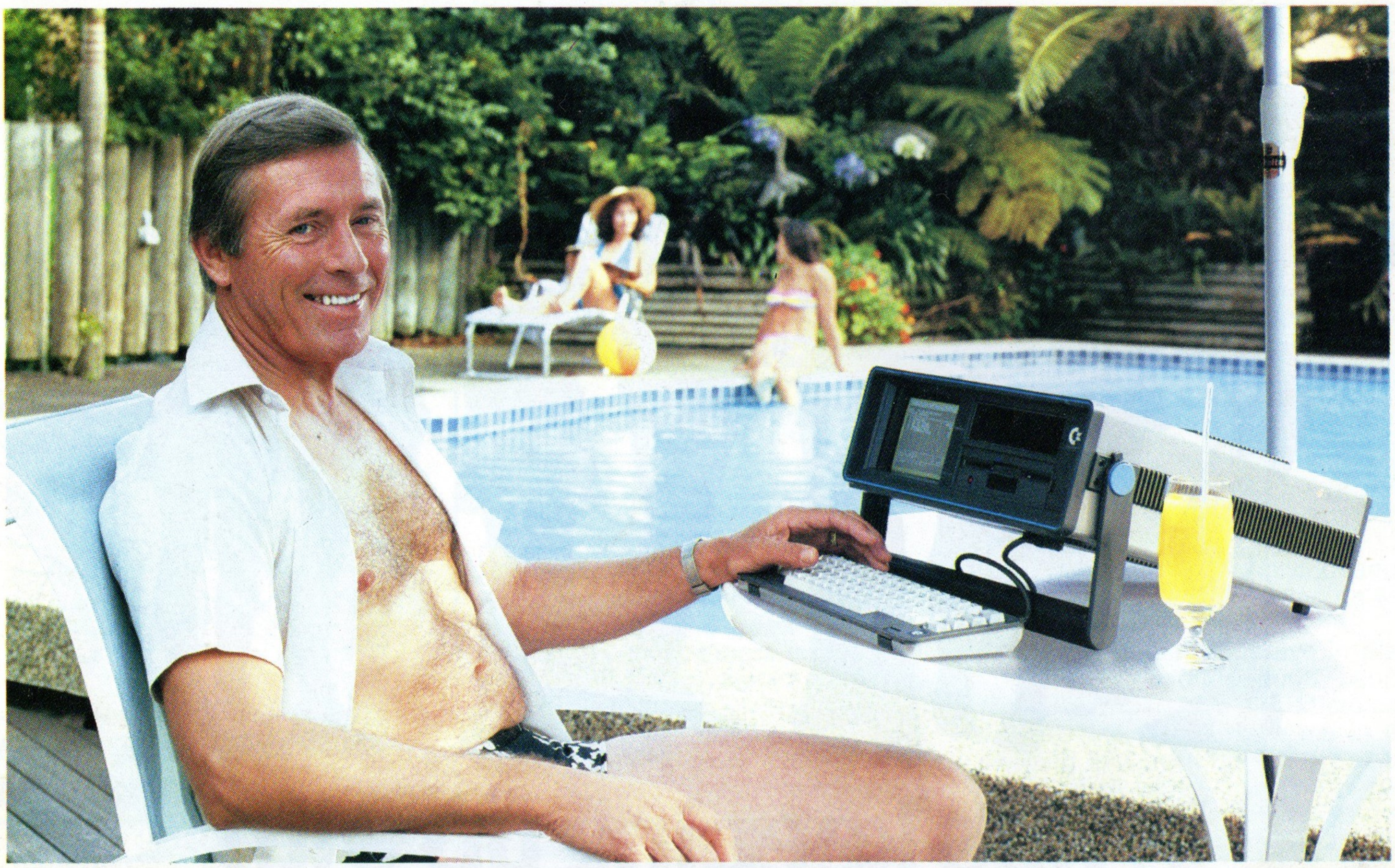
It's outstanding!



commodore
COMPUTER
Keeping up with you.

Commodore Business Machines Pty. Ltd.
5 Orion Road, Lane Cove NSW. 2066. (02) 427 4888.
Please send me more information on the Commodore 64™

Name ____________________
Address ____________________
____
BMS/CC157 Postcode ____________ Phone ____________ CM



Who's keeping up with Commodore?

The Commodore EXECUTIVE 64. A personal, portable computer with outstanding graphics, colour, music and astonishing computing capability, all in an easy-to-carry case.

The Commodore EXECUTIVE 64 is designed for the movers of this world. Designed to give you the power. Power at your fingertips. The power of 64K memory. The power to keep up. In the office. At home. Or in your home away from home.

commodore
COMPUTER

Keeping up with you.

Commodore Business Machines Pty. Ltd.
5 Orion Road, Lane Cove NSW. 2066. (02) 427 4888.
Please send me more information on the Commodore Executive 64™

Name ______________________________
Address ______________________________
____
BMS/CC 152 Postcode ______________ Phone ______________ CM

P. TURLEY Pena