commodore
c16-plus/4 HORIZONS

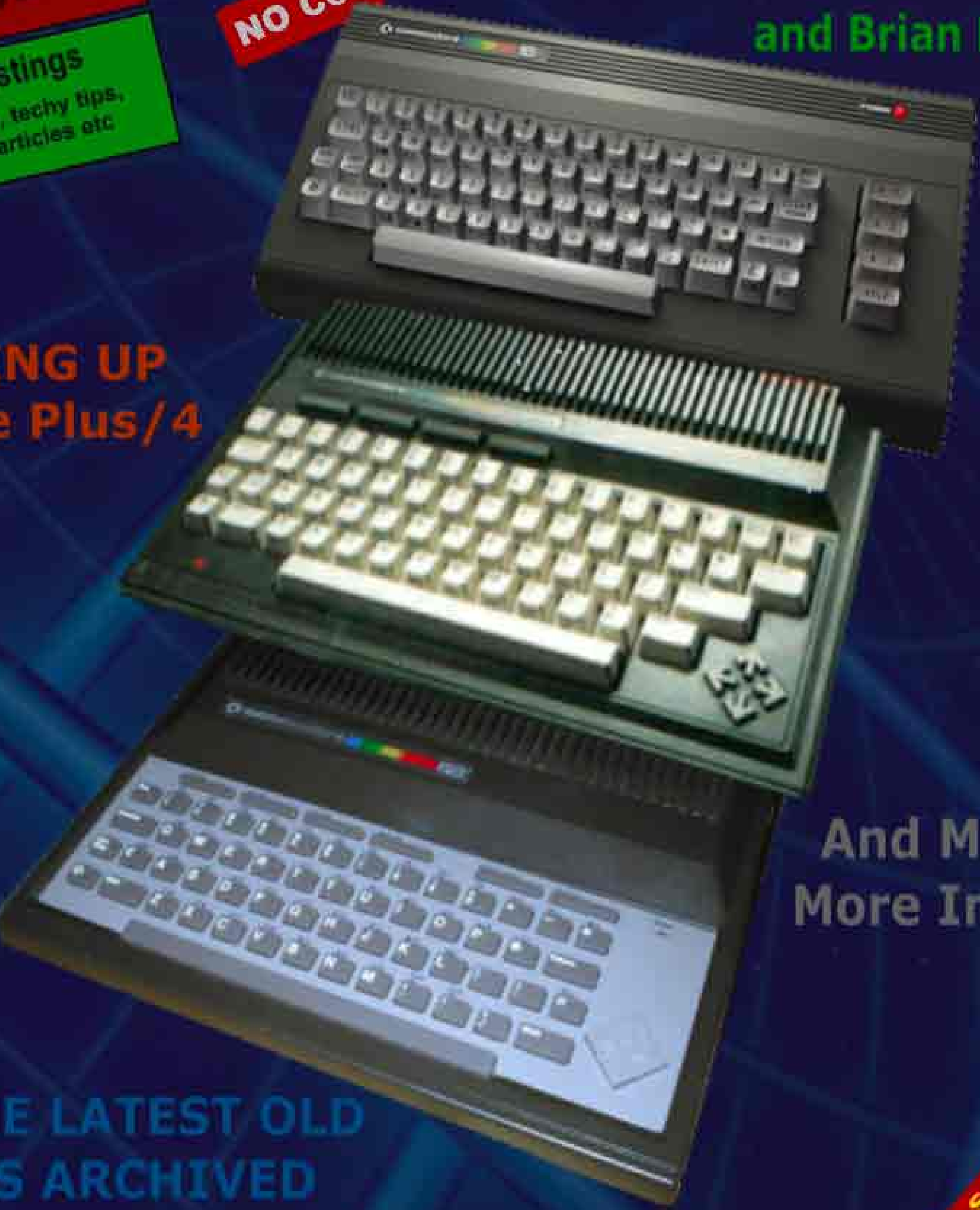
The C16 & Plus/4 Exposed
We took a deep look inside on what both machines has to offer

The Game Reviews
All 14 C16+4 games from our past issues closely scrutinised

Program Listings
Readers listings, techy tips, programming articles etc

NO C64

C16 PROGRAMMING with David Lawrence and Brian Lloyd



ADDING UP On The Plus/4

And Much More Inside

ALL THE LATEST OLD NEWS ARCHIVED

special retrospective issue

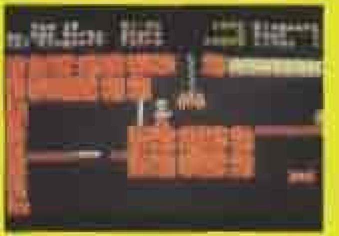
ANIROG

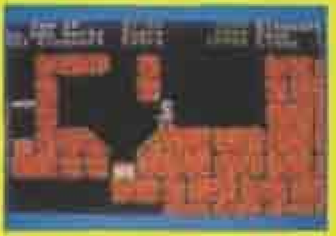
C16/PLUS 4

TOM THUMB

LATEST TECHNIQUE OF DATA COMPRESSION USED TO GIVE YOU SO MUCH FOR SO LITTLE PURE MAGIC?

£6.95





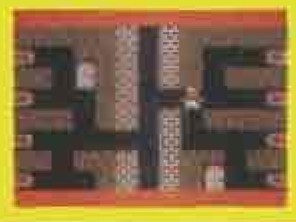
C16/PLUS 4

178 SCREENS OF ARCADE ADVENTURE

GHOST TOWN

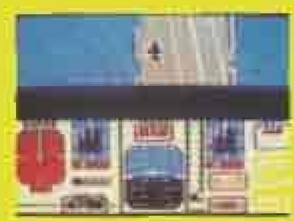
£5.95

A HIGH RES GRAPHIC ADVENTURE C16/PLUS 4



JUMP JET

Flight Simulator & Combat Action



PLUS 4
OR
C16+16K
£9 95

ATLANTIS

C16/PLUS 4

£6.95

FAVOURITE FOUR FOR ALL THE FAMILY

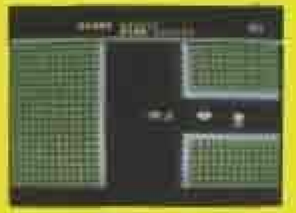
C16/PLUS 4

FLIGHT PATH 737

MOON BUGGY

£7.95

LAS VEGAS



ZODIAC

SKRAMBLE £6 95

PETCH £6 95

CATACOMBS £6 95

MINIPEDES £6 95

STAR COMMANDER
£6 95

3D TIME TREK
£6 95

TRIVIA 1400 QUESTIONS ON SIX DIFFERENT TOPICS. ALL FOR ONLY £7.95

OUT ON A LIMB

48K ONLY ARCADE
ADVENTURE LOADS
IN 3 SECTIONS
ONLY £6 95

16K RAM PACKS AT ONLY £25

64K RAM PACKS AT ONLY £50

TRADE ENQUIRIES Anirog Software Ltd Unit 10 Victoria Industrial Park Victoria Road
Dartford Kent (0322) 92513/8
MAIL ORDER Payment by Cheque, P.O., Visa or Access

Contents

GAMES

HARDWARE

UTILITIES

PROGRAMMING

REGULARS

Editorial

It's been a long 23 years since the last issue of Horizons made its way onto newsagents shelves, and since that time the evolution of home computing has grown immeasurably. The classic computer age of the 1980'a has long since gone since the new evolution of video game consoles from the Sega Master System upto the Nintendo Wii. More powerful PC's has since replaced the more humble microcomputers, and the vast world wide internet has since replaced the more niche market compunet all within the last 15-20 years.

But technology was never one to hold back, just as much as enthusiasts wasn't one's to condemn their machines to 8-bit heaven. So now thanks to modern internet and the more powerful PC's of today, we can again enjoy the delights of the golden age of computing through the use of emulators. Also it's thanks to all the dedicated sceners who over the years has contributed in someway by digitally preserving large software libraries, and especially the many consumer magazines which materially holds a much shorter lifespan.

Commodore Horizons was a popular microcomputer magazine published by Scot Press Ltd which devoted itself to the popular Commodore computers, mainly the C64 and Vic 20. It didn't take too long until the C16 and Plus/4 made it's way into the fold. The magazine ran from December 83 up until March 86. Duncan Scot is the longest running contributor and the only person to work on all 27 issues. This free special edition takes all the bits and bobs which appeared for the 264 series and reformatted for this dedicated C16/+4 retrospective edition which I hope you will enjoy.

Editors Christopher Jenkins, Brendon Gore **Assistant Editor** Christopher Jenkins **Digital/PDF Editor** Barry Flowers (Sixteen Plus) **Managing Editors** Brendon Gore, Duncan Scot **Production Editor** Barbora Hajek **Editorial Secretary** Geraldine Smyth **Publishing Directors** Duncan Scot, Jenny Ireland **Administration** Theresa Lacy **Group Advertisement Manager** David Lake **Advertisement Managers** Simon Langston, Jeremy Kite

Mail Merge

Send your letters on software, hardware, computing or life in general to Mail Merge, Commodore Horizons, 12/13 Little Newport St., London WC2H 7PP.

16K Vic ≠ C16, OK?

I OWN A Vic 20 and 16K expansion pack. I notice that the C16 has only 12K of RAM free to the user — does that mean that it might be possible to convert C16 software to run on the Vic + 16K, or vice versa?

G J Thomas
Ammanford
Dyfed

VIC 20 Basic programs will run on the C16, but it would be difficult to convert C16 programs to run on the Vic since the C16's Basic has several more complex commands. Commercial programs couldn't be converted in a practical way — the memory maps of the machines are totally different.

The C16 expanded

I AM VERY interested in writing computer games. I own a C16 but I am very dissappointed about it's memory and graphics. I find the graphics unrecognisable — they just seem like a blur to me. Plase could you tell me if I can get an expansion pack for the C16?

If not then I am thinking of buying a different computer; please could you also give me adice as to which one I should buy, noting what the graphics and memory are like.

Mark Shattock
Bristol

MCT of Group House, Fisher's Lane, Norwich (0603-666966) and Stonechip of Unit 9, Brook Trading Estate, Deadbrook Lane, Aldershot (0252-333361), both make expansion packs for the C16. The machine's graphics are as good as anything on the market — a good programming book would help you to make the best of them.

Plus/4 plea

I HAVE recently bought a Commodore Plus/4 computer with which I am very satisfied. However it is difficult to find books covering this machine or software.

At the moment I do not have a disc drive, but would be pleased if you could advise me what books and software are available for the Plus 4 and also where they can be obtained.

B Hunt
Dunstable
Beds

THIS MONTH'S Plus/4 software roundup should be of help; as for books, try our *C16-Plus/4 Companion* by Brian Lloyd, available mail order £5.95 + 50p p&p.

C16 on the shelf

THERE ARE several points I would like to raise about the Commodore C16.

The user guide mentions the monitor but no real instructions on how to use it. I was wondering if your magazine has given any detailed descriptions on how to use this command.

Are there any software companies planning to release a version of Pascal which will run on this machine when expanded to 64K?

The third point is, is your magazine planning in the near future to give a listing , of a basic compiler? Do any of the software houses supply one?

Finally, could you supply details of any company which would supply dust covers for the following pieces of Commodore equipment: C16, 1531 Datasette, 1541 Disc Drive, MPS 803 Printer.

John R. Willoughby
Coventry

BRIAN LLOYD'S book *The C16-Plus/4 Companion*, published by ourselves at £5.95 plus 50p p&p, contains a short introduction to using the monitor. A complete explanation of machine code would fill a whole book, and indeed there are many such books available, though none dedicated to the C16.

There's no chance of any company producing Pascal for a 16K machine; the number of people with 64K expansion memories must be minute, so it wouldn't be financially viable. Nor have we heard of a Basic compiler, though that may come.

For dust covers and such like, try Commodore's own Vicsoft service, at 1 Hunter's Road, Weldon North Industrial Estate, Corby, Northants, 0536-205555.

1531 to Vic 20

MY BROTHER has a C16. If I buy myself a cheap Vic 20 on its own, is there a cable available that would connect between the cable of his datasette and the Vic 20, so we could share it between the computers?

Andrew Banks
Wivenhoe
Essex

SOMEONE must manufacture a cable enabling the C16's 1531 cassette unit to be used with a Vic 20, or must be able to supply the parts to make a connector. Any suggestions, anyone?

Plus/4, basically

HAVING just purchased a +4 over the Christmas period, and not being worried about the software shortage, there are a few interesting points which I would like answers to, the main one being "What is Commodore 3.5 Basic" and "why has it not been used before" and why software houses have been slow in writing programs for what seems to be a very advanced basic?

For a person who is trying to master most dialects of Basic, could your recommend a good book on the subject?.

I have just bought my first copy of the Commodore Horizons magazine, and was so impressed I have put in an order for the next twelve months.

Frank Hargreaves
Ellesmere Port

Basic 3.5 was developed in response to criticisms of the Commodore 64's Basic, which included no specific commands for sound and graphics. On the C128, yet another dialect is used — the more advanced Basic V7.0.

Software houses of course write in machine code rather than Basic, because it's a lot faster and more efficient, so the quality of a machine's Basic rarely affects them. To learn more about the C16 and Plus/4's Basic, try *The C16-Plus/4*, by Brian Lloyd, and *The Working C16* by David Lawrence, both publicised by ourselves; and look out for *The Plus/4 Programmers' Reference Guide*, soon to be published in the States by Scott, Foresman & Co. No further details on this one yet, but we expect a UK edition will arrive — eventually.

COMPUTER
FINK
COME BUY! I'VE WRITTEN A FAST AND ORIGINAL GAME!
I CALL IT 'PURSUE'..
333
YOU CONTROL A LITTLE YELLOW THING AROUND A...
WHADDAYA MEAN ITS ALREADY BEEN INVENTED???
N. MURRAY!

Unwrapping the new systems

THERE WAS little new hardware to see at the Las Vegas Consumer Electronics Show, apart from Commodore's 264 and V364 micros.

Eight-bit machines based around the 7501 processor, the two micros are meant to complement the CBM 64 rather than to replace it.

With full 64K memories, typewriter-style keyboards and optional built-in software, the micros were described at the January show as applications computers rather than introductory or games machines.

The V364 additionally features a voice synthesiser with a 250 word vocabulary.

Aiming at the small business user, the most impressive features of the new machines are the screen window facilities and the optional built-in software.

The "3-plus-1" package will be available either as a cartridge for the CBM 64 and the new 264 and V364, or built-in to the 264 range. It consists of a word processor, electronic spreadsheet, file management program and business graphics facility.

Additionally, the machines feature four separate cursor keys, eight programmable function keys, and a HELP function.

The eight-bit 264 — with 7501 processor

Although most CBM 64 and Vic 20 peripherals should be fully compatible with the 264 series, most software will not be. The 264 series uses an extended Basic with over 75 commands.

The voice synthesis module will also be available as an add-on for the Commodore 64. Voice speed can be adjusted to slow, normal or fast, and there will be additions to the range of vocabulary and types of voice, both on disk and on cartridge.

The 264 series, with its accompanying software, is expected to be available in this country in mid-Summer.

No details of price are yet available, but Commodore will have to consider the challenge posed by, among others, the new £399 Sinclair QL, before deciding what would be the appropriate price range.

THE SHAPE of Commodore's new home computer range is beginning to emerge as the UK launch draws closer.

The bottom-end model with a rubber keyboard has been dropped while another machine, the 16, will come in the 64's familiar housing.

This machine, with 16K of RAM, runs Basic 3.5, as does the 264 which offers 64K — the Vic and 64 use Basic 2. The 264 comes in the newly designed casing shown in our March issue.

Final details on UK pricing and availability are expected in May.

Commodore's two new business micros — one IBM compatible and the other running a Unix operating system — are not expected to be available in the UK until the beginning of next year. Both machines were on show at the recent Hanover Fair, the largest computer show in Europe.

The long-rumoured Unix machine, running the Coherent operating system and based on the Z8000 chip, is aimed at the market for distributed processing. The Hanover model only supports two terminals, but "in future stages eight or more terminals can be connected".

As well as jumping on the Unix bandwagon, the CBM Z8000 joins the rat race — it features a mouse connection.

IBM compatibility is the other obvious market trend at the moment, and Commodore has a foothold there now with its PC. Based on the Intel 8008 chip, this is a portable machine resulting from a licensing agreement with Canadian company Bytec-Comtern. Commodore adds that "a desktop version of this newcomer with a larger screen" is expected later.

Also on show at Hanover were two new printers for the Vic and 64 — the DPS 1101 daisywheel and the seven colour dot matrix MCS 801.

Tynesoft promises C16 games

SOFTWARE for the new Commodore 16 and Plus/4, impossible to obtain at the time of writing, may soon be available from at least one major software company. Tynesoft has plans for a Christmas release of a starter pack of four C16 games, Shootit, Climbit, Munchit and Hoppit, at £14.95.

The four games are variants of well-known arcade games, and have been produced in cooperation with Commodore UK.

Tynesoft spokesman Colin Courtney said "The C16 is doing well locally, and we have plans for several more games. Despite the small memory we've been able to design North Sea Oil Rig Disaster, which has eight scrolling screens, and we're now working on Olympiad". Oil Rig Disaster features several rigs.

Anirog back in the game

ANIROG'S new titles for Commodore machines include games for the Vic 20, 64 and C16/Plus-4.

Slap Shot is a two player ice hockey simulation for the 64, incorporating software speech and Turbo loading. Cost is £8.95 on cassette and £10.95 on disk.

Minipedes costs £6.95, and contains versions for the C16-Plus/4, CBM 64 and Vic 20 + 16K. It's a version of the classic Centipede.

3-D Time Trek, *Zodiac* and *Star Commander* cost £6.95 for the C16-Plus/4. The games are, respectively, a version of Star Trek featuring long and short range sensors, Navigation Computer and Damage Sensors; a supernatural maze game with 400 corridors and legions of monsters; and a space shoot-'em-up.

Reviews in due course. Contact Anirog at Victoria Industrial Park, Victoria Road, Dartford, Kent, 0322-92513.

One size fits all

GAIL WELLINGTON, head of Commodore's European software acquisition team, has announced plans to produce software on a single disk which will work on all four current Commodore micros; the 64, C16, Plus/4 and C128.

The different versions of the program will have loading routine headers which will enable the micro to load the correct version from the disk. The software retailer will therefore only have to stock one type of disk to cover all the Commodore machines, cutting down on overheads and saving shelf space. The forthcoming C128 is already attracting software support. Audiogenic's Micro Swift Spreadsheet (reviewed this issue) has been converted for the C128, and should be available as soon as the machines are in the shops.

The C128 version has a worksheet size of 64 columns by 999 rows, and takes advantage of the 128's larger memory and built-in 80-column display. The pop-up menu system and range of functions and formats remains the same as the 64 version. Other software in preparation for the C128 includes Thorn EMI's Perfect series and Precision's Superscript — a follow-up to Easyscript — and Superbase.

New CBM micros debut at Show

Commodore Plus/4 — new Basic, new style, built-in software

COMMODORE'S two new home micros, the Plus/4 (formerly the 264) and the Commodore 16, made their UK debuts at the Fifth International Commodore Show in London on 7th-9th of June.

Both machines operate on a new dialect of Basic, version 3.5, which means that C16 software can be run on the Plus/4. Vic 20 and Commodore 64 programs will not be compatible.

The Commodore 16 comes in a Vic-style metallic-grey case. Designed for the first-time user, it has 75 programming commands including full graphics plotting and program editing. There are four cursor keys and a "help" key, which highlights programming errors.

The C16 has two joystick ports and 121 colours (15 colours at 8 luminace levels, plus black). The display is 40 columns by 25 lines. There are two sound generators, a cartridge port, and a new type of cassette unit port to connect the matching C1531 cassette deck.

The C16 offers 12K user RAM with 32K ROM, and will be sold in the form of a starter pack with the computer, cassette deck, Introduction to Basic part 1, and "several" games for £129.99.

The Commodore Plus/4 "Sits between the CBM64 and the V364 in terms of facilities. At the moment there are no definite plans for the future of the V364."

The Plus/4's major selling point is its four built-in software packages, which are stored in ROM and accessed through the function keys. There's a word processor, spreadsheet, database and business graphics package — the same range as you'll find on Sinclair's QL, which costs £400.

The four packages can share and exchange information through the Plus/4's screen windowing facility, which allows two packages to work simultaneously on the screen.

Further innovations include a new style case similar to the forthcoming Japanese MSX machines, with four cursor keys arranged in a diamond shape, an escape key and a reset button.

The Plus/4 is compatible with the 1541 disk drive, but is being marketed as a "package" with the SFS481 fast disk drive and the MPS802 dot matrix or DPS1101 daisy-wheel printer as an ideal small business system.

The Plus/4 will cost £249, and like the C16 should be available in September.

New peripherals include the MCS801 seven-colour dot-matrix printer, at £399, the MPS802 60 cps dot matrix at £345, and the DPS1101 daisy-wheel at £399. All are compatible with the Vic 20 and 64 as well as the new micros.

The 1542 disk drive will replace the 1541 at the same price, £229, and the new cassette deck, the 1531, for the C16 and Plus/4 only, will cost £44.95.

The Commodore 16 — for the first-time user

New progs aimed at Plus/4, C16

COMMODORE has announced a range of cassette and disk programs for the C16 and Plus/4, covering the games, education and utilities markets.

Start Programming with Zortek and the Krytobytes is a sequel to the popular **Zortek and the Microchips**. It will be available on disk or cassette. **Hide and Seek, Number Builder** and **Number Chaser** are educational games intended to develop memory, shape recognition and arithmetic skills. **Jack Attack, Viduzzles, Pirate Adventure, Atomic Mission** and **Strange Odyssey** are cartridge based games, most of which are priced at £11.99. **Mayhem, Harbour Attack, Stellar Wars/Blitz, Crazy Golf** and **Fire Ant** are cassette games costing £5.99.

There are six Infocom disk-based adventure games planned for the Plus/4 only; **Zork /1/2/3, Starcross, Deadline** and **Suspend**.

Further in the future are **Introduction to Basic Parts 2 and 3, Games Designer**, and for the Plus/4 only **Magic Desk 2**, a cartridge-based icon-driven productivity program, and **Tony Hart's Art Master**, an adaptation of the CBM 64 program.

We'll be reviewing all these products as soon as they're available. Meanwhile contact Commodore at 1 Hunter's Road, Weldon North Industrial Estate, Corby, Northants.

Plus/4 and C16 go down

THE PRICE of the Plus/4 has been cut from £299 to £149 in a bid to boost sales. Commodore's marketing manager, David Gerrard, argued that the new price will allow the machine to compete with the Sinclair Spectrum and Acorn Electron, both of which have recently been cut in price.

Although Commodore announced they had no intention to cut the price of the C16, three major retail chains have taken the decision out of their hands. Dixon's and Curry's have cut the price of the C16 Starter Pack, which includes the micro, data recorder, manual and four software packages, to £79.99, and Smith's has cut the price to £129, from the original price of £139.

The price cuts are thought to be a reaction to poor sales after a reasonable performance over the Christmas period. Ironically, while complaints of software shortages have been heard from many C16 buyers, it is only now, when it seems that the retail chains may be trying to dispose of their stocks, that C16 software is coming out in considerable quantities.

Commodore confirmed that there are no plans to delete either the C16 or the Plus/4, but with the C128 on course for launch in the Summer speculation is that the machines will be sacrificed on the altar of software compatibility.

Plus/4 revival?

IN THE wake of high sales for low-priced Plus/4 and C16 packages over the Christmas period, more software companies are planning releases for the neglected machines.

ACE, the air combat emulator from Cascade, is to be upgraded; the C16 version will now be accompanied by a full 64K Plus/4 game, incorporating the speech, ground and air targets, refuelling routine and two player option of the 64 version.

Novagen's popular Mercenary is set for conversion to the Plus/4; and for the first time US Gold is going into the C16 market with a conversion of the chart-topping Beach Head.

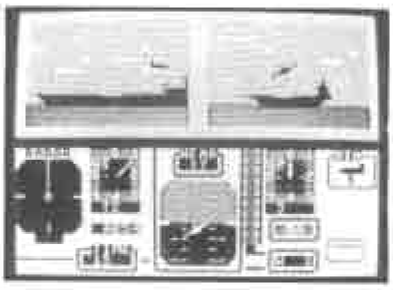
We'll be reviewing all the games as soon as copies are available.

Anirog takes off

ANIROG'S latest game is **Jump Jet**, which will be available in versions for the 64, expanded Vic and C16.

Jump Jet is a complex Harrier flight simulator incorporating a number of options including hovering and landing practice, radar and rangefinder interception and air-to-air combat with enemy aircraft.

Author Vaughan Dow, a Harrier pilot himself, is the author of Flight Path 737 for Anirog, and spend a year developing Jump Jet. The 64 version of the game will cost £9.95, and the disk version £11.95. The game incorporates software speech created using Anirog's Voice Master hardware.



More details on the Vic and C16 versions will be released soon. For more details contact Anirog, Unit 10, Victoria Industrial Park, Victoria Road, Dartford, Kent, 0322-92513.

Plus/4 patterns

DISAPPOINTED by the lack of software for the Plus/4, industrial desinger Ken Blake has written his own graphics program, **Pattern Designer**. In conjunction with an Oldham school, Ken wrote the program in order to make better use of the Plus/4's powerful graphics capabilities. The menu-driven package includes such facilities as free-hand drawing, joining points, polygons, fill, block, clear, screen flip, mirror, zoom, rotate, screen swap and copy, disk load and save, and full use of the Plus/4's 121 colours.

The program is almost totally joystick controlled, and Ken is making it available at £10. If you're interested, contact him at 25 Cornhill Crescent, North Shields, Tyne and Wear NE29 8LN.

More from Supersoft

SUPERSOFT'S latest batch of releases includes utility programs for the 64 and the C16-Plus/4.

Word Perfect is one of the first word processors to appear for the C16-Plus/4. It's a machine code program allowing up to two pages of A4 text to be entered, and displaying text on the screen exactly as it will be printed out. Maximum line length is 160 characters, and facilities include insert/delete, copy line, tab setting, margins and centering. Word Perfect costs £12.95 on tape or £14.95 on disk, and is also available for the CBM 64, at £17.95 and £19.95.

Music Master, the popular 64 composition package, has also been adapted for the C16-Plus/4.

One or both voices can be played using the QWERTY keyboard, and there are ten pre-programmed backing rhythms to accompany your compositions, as well as extensive editing facilities. Music Master C16 is available on tape or disk, at £12.95 and £14.95.

Instant Recall is a filing system for the 64, which allows you to store up to 30,000 characters of information. Particularly well-suited to holding club or hobby records, Instant Recall can be used with a suitable printer to create address labels or reports. Price is £17.95 tape, £19.95 disk.

Next month we'll take a look in more depth at some of these programs; meanwhile contact Supersoft at Winchester House, Canning Road, Wealdstone, Harrow, Middsx, 01-861 1166.

Solar's C16 strategy

SOLAR Software has launched a series of budget-priced games for the C16-Plus/4, and claims to have sold 15,000 units within a fortnight of launch. Solar's Brenda Shay explained that the games, all priced at £29.99, are conversions of successful Vic 20 titles by programmer John Shay. "John has written a routine which enables him to convert the Vic programs directly on to the C16. All the games are retitled to avoid confusion with the Vic originals." Titles include **Mission Mars, Suicide Run, Knock Out, Cruncher, Alien Attack,** and **Quick Draw.**

Solar Software also plans to produce a full-price C16 game in the near future, as yet untitled. The budget titles are being distributed by Centresoft, Leisuresoft, Lazer, and a range of smaller distributors, although Solar has not yet been able to conclude a deal with the chain stores.

"We're not touching the 64 at the moment", commented Brenda Shay — "our last two games, **Monkey Magic** and **Jungle Quest,** didn't sell well despite good reviews and extensive advertising. We've now rereleased them as budget titles, but we feel that the 64 market is glutted; so at the moment we're concentrating on the C16, where there's still a demand for new games, especially at budget prices."

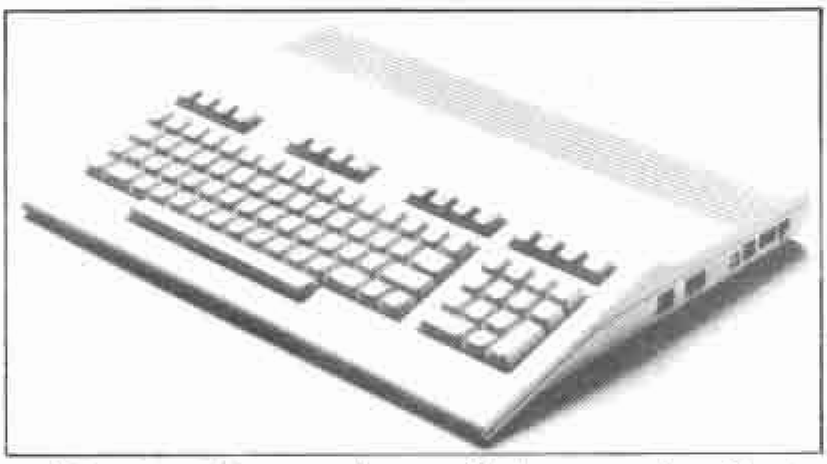
Contact Solar at 77 West Drive, Bury, Lancs, 061-761 1770.

128 delay — C16 to go?

COMMODORE'S C128 has been delayed, and speculation as to the fate of the C16 and Plus/4 has increased. Following the PCW Show, where public reaction to the new machines from Commodore, Atari and Amstrad enabled the marketing men to judge the state of the industry, the C128 launch has been delayed until the end of September so that a cheaper disk drive can be developed.

The new disk drive, the 1570, is a single-head high speed unit, designed to compliment the more expensive double-head 1571. The 1570 will sell at £199, as opposed to the 1571's £299. This means that the C128/1570 combination will sell for £468, £70 more than the price for the Amstrad 6128, which also includes a colour monitor.

Meanwhile, speculation mounts that the C16 and Plus/4 have had their day. At a PCW Show which saw very little in the way of new software for the machines, Commodore's marketing and sales manager Paul Welch was quoted as saying "I don't think there will be any Plus/4's or C16's available after Christmas — by then we will have built out all our remaining component stocks." This confirms doubts over the C16 and Plus/4 prompted by widespread High Street discounting and poor software support, but contradicts Nick Bessey's statement on Thames TV's Database programme that "The C16 and Plus/4 are alive and kicking — we're just concentrating on the C128 this year."



Chain stores such as Dixon's and Lasky's are offering attractive deals on the Plus/4 at the moment; Plus/4, datasette, joystick and software ten-pack for £99, or Plus/4, 1551 fast disk drive, MPS803 printer and business software package for £299.

Corby factory closes

COMMODORE has announced that the Corby assembly plant is to shut down, putting an end to the manufacture of hardware in the UK. From now on, all Commodore products sold in the UK will be manufactured on the continent or in the United States.

The Corby plant, which opened with the help of government grants 18 months ago, consists of marketing, sales and administration headquarters and an assembly line. It is this assembly line which is to close, with the loss of 250 jobs.

The £6m headquarters will remain, with 170 employees, as a centre for sales, administration and servicing.

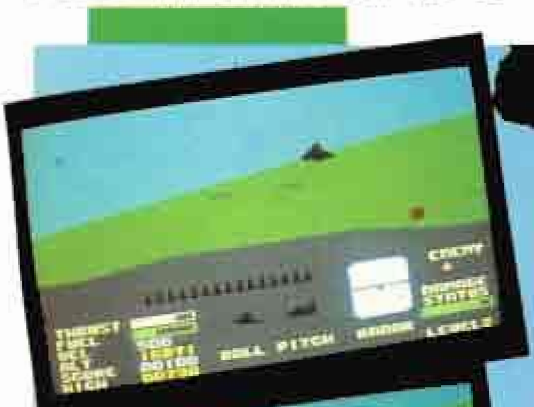
Thomas Rattigan, president of Commodore International, said that to meet the challenge of the next two to three years, Commodore would be concentrating on fewer plants using higher technology. Corby, being essentially just an assembly site, did not fit in with this strategy.

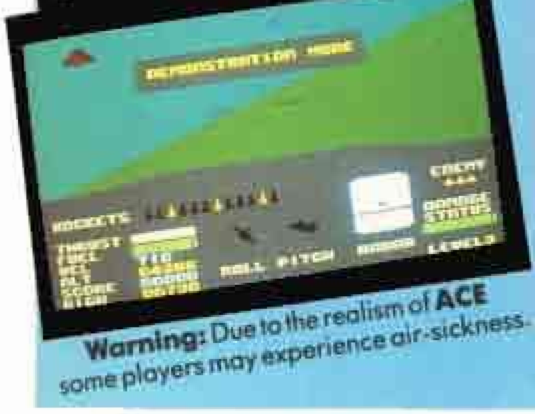
Commodore's $113m loss in the year ending June 1985 has led to widespread speculation as to its ability to carry on in the market. The company had forecast a profit for the next quarter, but has now retracted the claim, at the same time as it announced the closure of the Corby factory and a Californian chip manufacture plant. The company's bank debts now total $192m, though there is no indication yet that the company will file for bankruptcy.

In the UK, the company is to discuss with the Department of Trade and Industry the repayment of grants for the Corby factory. Faith in the Amiga still seems high, but the **Which Computer Show** will be a good indicator of likely response to the machine in the UK. Sales in the States have been "at the low end" of the company's expectations.

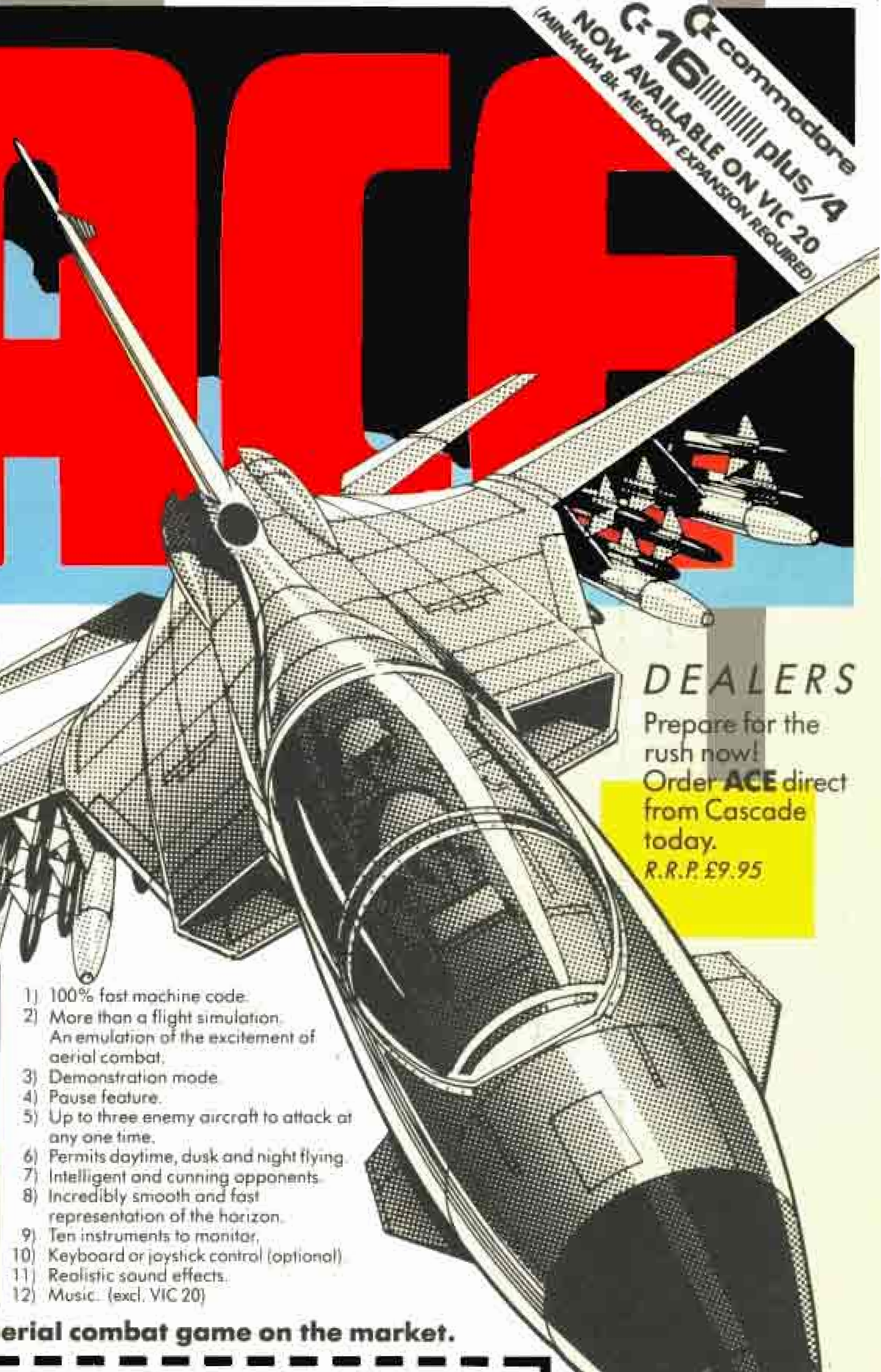
AIR COMBAT EMULATOR

100% MACHINE CODE ACTION
FROM CASCADE GAMES





Warning: Due to the realism of **ACE** some players may experience air-sickness.



commodore C-16 plus/4
NOW AVAILABLE ON VIC 20
(MINIMUM 8K MEMORY EXPANSION REQUIRED)

All of your comrades have been shot down in the face of overwhelming odds. You are the last fighter pilot pitched against multiple waves of enemy fighters. You **must** destroy the enemy — are you good enough to be called an **ACE!**

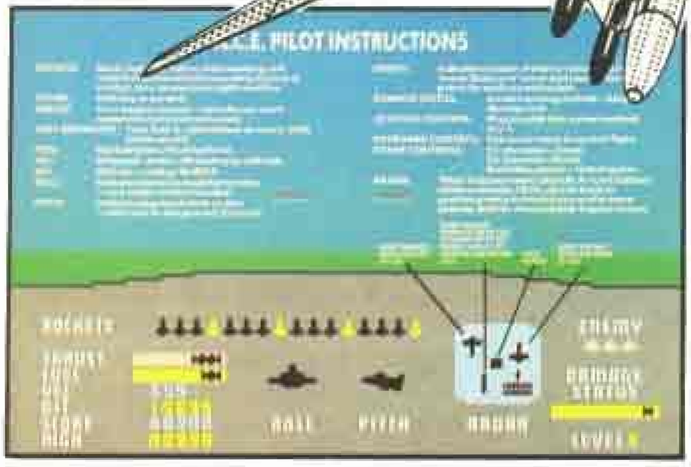
Watch the horizon dip and roll as you pilot your jet fighter. This is what TV personality and World War II fighter pilot, RAYMOND BAXTER had to say about **ACE** . . .

"I've flown Harriers, Tornados, Spitfires and I can assure you that the way this game responds to the controls is very realistic."

DEALERS

Prepare for the rush now! Order **ACE** direct from Cascade today.

R.R.P. £9.95



1) 100% fast machine code.
2) More than a flight simulation. An emulation of the excitement of aerial combat.
3) Demonstration mode.
4) Pause feature.
5) Up to three enemy aircraft to attack at any one time.
6) Permits daytime, dusk and night flying.
7) Intelligent and cunning opponents.
8) Incredibly smooth and fast representation of the horizon.
9) Ten instruments to monitor.
10) Keyboard or joystick control (optional).
11) Realistic sound effects.
12) Music. (excl. VIC 20)

Order ACE now — quite simply the best aerial combat game on the market.

Name ______________________

Address ______________________

____

____

Post Code ______________________

Country ______________________

Postage FREE. Goods will be despatched within 7 days.
I enclose cheque/postal order for:

£ ______ made payable to Cascade Games Ltd or through any Post Office by TRANSCASH (Giro No. 655 6655)

For even faster ordering when charging to ACCESS, BARCLAYCARD and TRUSTCARD VISA use our 24 hour answering service on (0423) 504663

No. ______________________

PLEASE TICK WHICH VERSION YOU REQUIRE

COMMODORE C16 □ COMMODORE PLUS 4 □ VIC 20 □

£9.95
INC. P&P

cascade

CASCADE GAMES LTD.,
1-3 HAYWRA CRESCENT, HARROGATE, NORTH YORKSHIRE, HG1 5BG ENGLAND
TELEPHONE: (0423) 504526 REGISTERED NUMBER: 1755554

Plus/4 — the verdict!

COMMODORE's new 64K machine is now making an entrance to the home micro arena where its predecessor has been so successful. So, how does it compare to the 64?

The first thing that strikes you about the Plus/4 is the size — or rather the lack of it! Instead of that rather cumbersome rounded box inherited from the VIC-20, the Plus/4 features a much smaller and neater MSX-style wedge-shaped design, which is far more visually attractive and places the keyboard closer to the desk top. The rear section of the top provides ribbed ventilation slots and the unit remained cool indefinitely.

The appearance of the white keyboard is rather similar to that on the portable SX64, but the key-tops are comfortably dished. It may not be quite as acceptable to some users as the 64's, as the keys are heavily spring-loaded and rather 'bouncy', but such features are rather a matter of personal taste. In a departure from Commodore tradition the cursor keys are placed away from the main keyboard in a diamond formation at the bottom right corner of the case, which makes cursor control much easier. Four dual-purpose FUNCTION KEYS are placed in a horizontal row at the top left of the keyboard.

Warm start

A power on light and switch are provided, together with a RESET button. The RESET key normally acts in the same way as the power switch, giving a straightforward cold start, but if the RUN STOP key is pressed during RESET a warm start to the machine code monitor is made instead so you can often resurrect programs which 'hang-up'. We feel that this sequence should really have been reversed (ie a warm start by pressing the RESET key alone), as it is rather too easy to accidentally produce an unwanted cold start.

All the I/O connections are placed neatly along the back of the machine. The SERIAL (for disk drives and printers), VIDEO (for monitors), and TV sockets are identical to those on the 64, but the cassette port, user port, and joystick ports are all totally different, using small 8-pin plugs. These socket changes are irritating as the standard C2N cassette recorder and joysticks cannot be plugged in. The power supply is the familiar 'lump' provided with the 64, but with a different connector. An interesting inclusion in the row of sockets is a MEMORY EXPANSION port, promising even greater storage capacity, but there is no separate cartridge port and 64 cartridges are incompatible, as the connection differs. As usual on Commodore machines, the serial I/O does not use the industry standard RS232 voltage levels so an interface will be required for non-Commodore peripherals, and communications.

Memory switching

On power-up "COMMODORE BASIC V3.5 60671 BYTES FREE" appears at the top of the screen. The version number is a reflection of the improvements made to the BASIC since the 64's V2 and the 60671 bytes of memory directly available through BASIC is a whole 21760 bytes (21k!) more than that termed "elephantine" on the 64. Although both machines have 64K of RAM this increase in availability is achieved by some neat automatic memory switching routines (and we can't wait to see if Commodore's TV ad's will now feature a blue whale)!

The Plus/4 retains the full screen editing facilities of the 64 but adds some useful extras, such as automatic insert and flashing characters, and the editor is easier to operate with the new cursor key arrangement. In usual Commodore fashion a wealth of different graphics characters are printed on the fronts of the keys but these are not obviously re-definable.

To keep up with the Joneses a WINDOW facility has been included, although this is rather rudimentary, providing only a single window via ESCape sequences. The top right and bottom left corners of the window must be defined in terms of current cursor position, so including window settings in programs is rather messy. Other ESC facilities include automatic insert, deletion of a whole line, turn off scrolling, reduce screen display (which may be necessary on some TVs), and scroll up and down.

The default settings of the function keys provide one-key entry of "GRAPHIC" (f1), "DLOAD" (f2), "DIRECTORY" (f3), "SCRNCLR" (f4), "DSAVE" (f5), "RUN" (f6), "LIST" (f7) and "HELP" (f8) and in contrast to the situation on the 64 the function keys can easily be redefined using the KEY command. The useful HELP facility assists in debugging BASIC programs. If an error comes to light during execution then pressing HELP displays the appropriate line with the aberrant section highlighted as flashing characters. Entering LIST and then calling for HELP will feature the error in context.

Slow loading

The draft manual suggests that the cassette routines are similar to those of the 64, so we suspect that it may still suffer the same problem of slow loading.

The Plus/4 operates perfectly with the 1541 Disk Drive, but inevitably it suffers from the notoriously slow access time of this unit. The 1541 is to be replaced by the 1542 which promises faster execution times. The commands supplied on the 1541 demo diskette (for the 64) are not required on the Plus/4, as the disk handling commands are all included in the 32K ROM. These disk commands comprise DSAVE, DLOAD, DIRECTORY, BACKUP, COPY, COLLECT, RENAME, SCRATCH and HEADER, with '?DS$' provided to trace any disk handling errors.

BASIC V3.5 is a marked improvement on the tired old BASIC V2, as Commodore has taken to heart the numerous complaints from users and included many of the better features of more modern BASICs. All commands available on the 64 are included together with a number of significant additions. Commands can be abbreviated (and of course you can program the function keys to duplicate any desired sequence).

The extra non graphic and sound commands are as follows. AUTO provides automatic line numbering whilst DELETE removes program lines on request. Although Commodore has not gone the whole hog and provided a completely structured BASIC the addition of DO/LOOP/WHILE /UNTIL/EXIT should go a long way towards appeasing the purists. BASIC 3.5 supports IF . . . THEN . . . ELSE rather than just the simple IF . . . THEN, which makes for tidier programming. Modification to RSTORE now allows a line number to be specified from which to read DATA. Extensive error-trapping routines are included to make total program crashes avoidable. TRAP traps all error conditions except "UNDEF'D STATEMENT ERROR", and will re-route the program to a predefined line. Once an error has been TRAPed the guilty line number can be traced from the variable EL, with the error condition contained in the ER variable. The function ERR$(ER) gives you the error message corresponding to ER.

RESUME allows you to continue program execution after an error has been trapped. GETKEY is a GET (or INKEY$) statement which automatically waits for a key to be pressed instead of falling through. HEX$ converts from decimal to hexadecimal numbers while DEC provides the reverse conversion. The JOY function returns the value of a switch type joystick, but the industry standard 9 way D plug is not used! PRINT USING allows formatted PRINT outputs (such as columns of figures) to be produced with the minimum of effort, and PUDEF allows you to define the characters to be used in PRINT USING. INSTR is a valuable addition to the string handling commands providing searching for one string within another. TRON and TROFF are debugging commands which print the current line number on the screen during program execution, although these commands are often more trouble than they are worth as they upset the screen display.

The graphics commands are a strong point of BASIC V3.5 as they transform graphics programming from an occult art to childs play. The SID and VIC ▶

Software on trial

THE PLUS/4 has been designed as a market leading computer system intended to maintain and, if possible, improve Commodore's prominent market position. An integral part of the system is a software package provided with the machine and held on ROM for near instant use.

Named 3-PLUS-1, the software provides facilities for word-processing, records filing, spreadsheet operations, and creation of simple graphs. The package is not a sophisticated one and does not offer the facilities of most dedicated programs in each of these fields. However, this is the first time that such a package has been made available for a mass-production computer aimed at the home or business user at this price.

So called integrated-software packages are the fashion at the moment in the highly competitive market of business software for machines such as the IBM PC. Commodore has achieved several points in the integration stakes. The 3-PLUS-1 software is integrated into the machine itself; the programs share many of the same commands; data can be transferred from one program to another; facilities from two or more of the programs can be combined to do things like creating mail-shots; and the programs with data can all be 'active' on the machine at one time.

When you switch on 3-PLUS-1 you are started off in the word processor. The working area available to you is 77 characters wide by 99 lines deep. Most of the standard facilities are provided including text entry from the keyboard (obviously), deletion and insertion of text at any point on the work area, copying or moving blocks of text, text formatting through the use of tabulation settings, word-wrapping (to automatically ensure that words are not split at the end of lines), justification by spacing words to completely fill lines, and control of the printed output.

In practical terms the documents which can be produced must be relatively short. Letters can be produced and, when used in combination with the data-base, circulars or mail-shots are possible. It would be over-ambitious to attempt to write books or lengthy articles.

A major restriction on the software is the way that the screen display is used. Only 37 characters are shown across the screen, and this means that a little less than half the width of the document is visible at any one time. This makes reading the document difficult as you must pan through the thing from one side to the other to follow a sentence through. Also it makes it difficult to plan out the format of the final document. A phrase often used to describe some word processors is "what you see is what you get", meaning that the display on the screen can be set to show the document just as it will appear when printed out. No way can 3-PLUS-1 provide this.

On using the word processor I found a number of things not to my taste. The commands are generally rather clumsy in use; for example there is no text insert mode. You must use the INST – DEL key to give you a series of spaces or a special command to insert a blank line and then type in the space provided. Hitting the RETURN key can wipe out a whole line of text if you are not careful. When working with blocks of text, the text is not obviously marked on screen and hence it is easy to make mistakes.

The second program given is a ▶

◀ spreadsheet. This is my least favourite program of the four. The principal of a spreadsheet is to provide a grid into which numbers, formulae, and text can be placed and manipulated. The value of a spreadsheet comes from the ease of setting the thing up or modifying data and, for many practical applications, the size of the grid provided. Commodore has had to cut some corners with this program in terms of the way in which facilities are provided and the size of the spreadsheet grid.

A grid of 50 rows and 17 columns is provided, which is adequate for many uses, but an additional limitation is imposed by the memory allocation (around 7k to allow the other programs space in memory at the same time) which can mean that only half the grid spaces can actually be filled. The screen display is normally set to show 12 rows and 3 columns. A half-height display is also used, 7 rows deep, to permit both word processing and spreadsheet to be shown on the same screen.

Each space, or cell, on the grid may contain a number and may also contain text or a formula. Any formula may contain reference to other cells, either explicitly by the cell row and column numbers or by the text held in the cell. It is here that Commodore has chosen to ignore most of the other spreadsheets by using an unusual method of referring to cells by row/column and using numbers for the column reference. It is more normal to refer to column/row using letters to signify the column. Also, to make it easy to build up the spreadsheet, a function is usually included which copies a single formula into a number of other cells which automatically corrects the formula for each cell. Commodore has not provided such a function.

Data from a single row of the spreadsheet can be used to create a simple block graph and transferred to the word processor. No competition to Lotus 123, but even simple graphs can have their uses.

Graphs are produced in low-resolution to allow printout directly to standard printers. Automatic scaling is used to produce a graph 20 characters high but the scaling cannot accomodate negative values.

Last and in my opinion most useful comes a data-base program. Once again the program is relatively unsophisticated, but is very functional despite this. However the program cannot be used without a disk drive.

Up to 999 records (think of each record as a card in an index file) can be set up to form a file on a disk. Once set up, the records can be modified, sorted, specific data can be searched out, and the records can be used to provide data for use with the word processor.

The most difficult operation is the initial setting up of the data file. You must specify what type of data is required on each record and how much space must be provided for each item of data. This information cannot be changed once set up and so you must be sure of exactly what you want before you start to set up the data. After doing this, the rest is relatively easy. Alphabetical sorting is reasonably fast, as is searching the whole file for a particular item of data.

All the 3-PLUS-1 programs are designed to be used with a disk drive. This obviously will increase the cost of a full system. The data-base is dependent on a disk drive to work at all.

A separate manual is provided for 3-PLUS-1. This provides a tutorial approach to introducing the programs, as well as reference sections. Each function is individually described with the help of examples.

In summary, if you are thinking of getting a PLUS/4 just for the sake of the software, then forget it. You can get much better value elsewhere. As an addition to a potentially good general purpose microcomputer it is of more value. If you want to learn the basics of using these types of archetypal programs, but have no immediate grand plans for their use, then the PLUS/4 system is well worth further investigation. ■

John Cochrane

◀ chips of the 64 have disappeared into obscurity but all that tedious POKEing is a thing of the past, having been replaced by a graphics system comparable to that of the Microsoft Extended Color Basic used by the Tandy CoCo, Dragon and new MSX machines.

Five different GRAPHICs modes can be set up. The default mode 0 provides a standard 25 line by 40 column text display, but selection of one of the hi-res modes reserves 10K of RAM. Mode 1 provides a full-screen 320×200 pixel display with the full range of colours (subject to the restriction of only one foreground colour per character square), whilst mode 3 halves the horizontal resolution but allows three foreground colours in each character space.

Split screen

Modes 2 and 4 are similar to modes 1 and 3, except that the screen is split and retains five lines for text at the bottom. CHAR allows you to print a string of characters anywhere in any mode, providing useful integration of text and hi-res graphics. COLOR assigns any of 16 colours and 8 luminance settings to either the background, any of up to three foregrounds, or the border. BOX allows you to draw filled and empty boxes, which can be rotated about their centre, and the very powerful CIRCLE command allows you to draw anything from a simple circle, through ellipses to a triangle inclined at 45 degrees. DRAW produces a straight line between two points and can be used to produce a whole shape (although not as simply as with the Microsoft DRAW command).

LOCATE places the cursor at any screen coordinates and SCALE allows the screen coordinate scale to be changed. PAINT fills defined parts of the screen with particular colours. SSHAPE and GSHAPE allow blocks of the screen to be picked up and moved, but as the information is stored in a string rather than an array the size of the area taken is much more restricted than with the similar Microsoft GET and PUT commands. The RDOT function provides the X and Y coordinates of the cursor and can also return what is beneath the cursor. RGR returns the current graphics mode, and RLUM the luminance level of a specified colour. All the graphics commands can be specified in absolute terms or relative to the cursor.

Sprite capability

One excellent feature of the 64 which has gone is the sprite capability — although dealing with sprites was so painful that most people never bothered with them anyway! Consideration of the standard of sprite graphics currently being produced on the humble Spectrum leads us to believe that expert games programmers will not find this too much of an obstacle.

Sound on the Plus/4 is a pale shadow compared with the brilliant effects possible on the 64, but once again it is much easier to access and it still puts many other home micros to shame. The sound is routed through the TV speaker, and is controlled internally via the VOLume command which provides seven levels (with a default level of zero — which is why we began to wonder whether the sound channel on our monitor was working!).

White noise

SOUND controls the two voices available, the second of which can produce both tones and white noise. Envelopes are not supported and only the voice, tone, and duration of the note can be specified, although Commodore do give a table of musical note conversions (which regrettably require obscure values).

Although SID and VIC may be dead TED has arrived, in the form of the built in TEDMON which provides a straightforward assembler, disassembler and machine code monitor which is inevitably somewhat limited in its scope, although adequate for dealing with small sections of code. The only feature sorely missed is a trace facility, allowing you to step through your program searching for bugs. However, instant access is definitely a plus point which should lead more users to widen their programming experience into machine code.

Although the Plus/4 is now ready for launch in the UK, Commodore has been dragging its heels somewhat, because of delays in completion of the built-in software packages, so it will be the new year before it is available in volume. There has been much speculation over the market positioning of the Plus/4 and the 64 because they are both in roughly the same price band. Although many people expected that the Plus/4 would be simply a replacement for the 64, it is becoming apparent that this is not true, and that Commodore is aiming at rather different types of user.

Commodore sees the 64 continuing as essentially a games machine, with the Plus/4 appealing more to the "serious" home user, who is interested in programming. The Plus/4 uses the newer 7501 CPU in place of the 6510, but any increase in speed this brings is eaten up by the extensions to BASIC and memory switching, as running the standard PCW Benchmarks on both machines produced effectively the same times. The Plus/4 User Guide was seen only in draft form, but promised to be an improvement on the 64 User Guide, as it contained much of the useful information previously restricted to the Programmers Reference Guide.

Our overall impression of the Plus/4 was that it was well-constructed, more user-oriented, and a marked improvement over the 64 in many other respects. The price of £300 seems rather steep (as production costs cannot be much different from those of the 64), but some allowance must be made for the free software. It is possible that Commodore is deliberately inflating the price to clearly differentiate it from the 64, and leave room for manoeuvre — just remember how much the 64 cost when it was first announced.

Outside competition

Of the outside competition, the QL and Amstrad have much faster BASIC, are technically more advanced, and seem to offer better value-for-money, although perhaps a small user-base in global terms. The MSX machines have a similar turn of speed, offer compatibility, but are rather an unknown quantity. How well the Plus/4 fares against the competition will probably depend as much on brand-loyalty and availability as anything else. 🟩

Keith & Steven Brain

Sweet little 16

AFTER years of rumours about its death, the VIC20 is now officially obsolete, replaced by the Commodore C16.

The C16 comes in a starter pack, similar to the VIC, consisting of dedicated cassette deck, 'Introduction to BASIC', and four software packages, for £139.99. The 16 in the name refers to the total RAM available, though like the 64 the amount of user RAM is considerably less.

It looks just like any other Commodore machine, in the same case as its predecessors, though in a different colour scheme — the case is dark grey, and the keys light grey. The keyboard itself is the usual CBM type of thing, with decent keys, and angled, though the layout is non-standard, even compared to its predecessors. If you've got used to the positions of keys like *, +, HOME, and the all important cursor keys on the 64, then you will have great fun on a C16, as the've all changed round. I find CBM keyboards a little too soft, but typing is a personal thing.

Inside the beast is a smallish circuit board, leaving quite a bit of empty space. The American origins of the machine are immediately apparent by the metal screening all over the circuit board. In the USA, there are certain legal requirements for radio interference, but there are no such laws in the UK, so British micros usually drown out the average radio in the same room. There are few components on it, the main ones being the 7501 processor and the custom chip that does the clever stuff like producing a TV display. The 7501 is just another Commodore version of the illustrious 6502, like the 6510 in the 64.

On the C64, with its 64K of memory, when you switch on you get a message saying something like '38K bytes free'. The remaining 26K is used up by the system, so about 40% is unavailable in BASIC. On the C16, with 16K RAM, you get about 12K free, which isn't bad. However, the moment you select one of the high-res graphics modes, another 10K of memory goes down the tube, leaving you under 2K user RAM, less even than the VIC20. This I feel is the major shortcoming of the C16 — the lack of RAM. On the 16K Spectrum, about 8K is left to the user, which is quite ample for many uses, but designing a 10K screen for a machine that has only 16K is a major oversight by Commodore. It's supposed to be technically possible to expand it up to a Plus/4 and onwards to 128K, but Commodore don't have any plans for doing this themselves yet.

Previous hallmark

It has been the hallmark of previous CBM machines that the BASIC didn't support any of the machine's nice features, such as graphics and sound. It practically made it a necessity to buy one of the many 'extended BASICs', either Simon's BASIC from Commodore themselves, or one of the independent programs. Well, on the C16 Commodore have killed that piece of the market stone dead, and made poor young Simon unemployed, as BASIC version 3.5 is supplied with the machine, which has all the extra commands you would expect on any other machine.

The BASIC sits in 32K of ROM, along with the kernal, which looks after the whole machine, and a machine-code monitor. The BASIC is Microsoft as you'd expect, but in the funny Commodore dialect used before. Programs are edited using the same screen editor as before, but with the addition of a few welcome features such as automatic insert mode. The same PET-type graphics characters are available, and the function keys are initially configured to produce various key sequences. The default commands are a little strange, being mainly concerned with the disk drive, and as most users would not have a disk drive, but the supplied cassette recorder, they seem an odd choice. Anyway, if you don't like them, the KEY command re-defines them to be anything else — no POKEs necessary on the C16. For entering programs there are a few other commands, like AUTO and RENUM, which are again new to CBM owners. Fans of 'structured programming' will not be impressed, its only such features being DO. .WHILE and DO. .UNTIL, with no procedures or long functions.

By pressing the ESC key followed by another, various extra functions can be invoked, the most interesting of which are the 'window' commands. Windows are one of the in things at the moment, and have appeared on the Amstrad and QL, and now the C16. However, you can have only one window at a time, so it's not really as good as the others mentioned. I believe a similar effect can be obtained on the C64 by POKEing anyway. Regrettably there is no WINDOW command, so the process of defining a window has to be done by a tedious sequence of control code printing.

The C16 has five graphics modes. The simplest is Text mode, with the usual 40 by 25 lines of text, with each letter able to be a different colour. This is the only one that doesn't take up the extra 10K of memory. In text mode, there are 15 different colours, each with 8 levels of luminance, and black, effectively giving 121 different shades, all of which can be displayed at once. There are also two high-res modes, one with 4 lines of text at the bottom of the screen, and one without. In high-res the resolution is 320 by 200, which is about average nowadays, though there is a colour restriction — you can only have two colours in one 8x8 pixel character square, the same as the Spectrum. However, you can still have 121 different colours at once. The two remaining modes are 'multicolour' modes, which have a resolution of 160 by 200, but can have up to four different colours per character square, still with the choice of 121 colours.

Missing feature

The main feature missing on the C16 compared with the 64 is sprites. Undoubtedly the best feature of the 64, they made it much easier and faster to write games, as the programmer didn't have to bother about the mechanics of actually putting shapes onto the screen and removing them later, since the hardware did it all. The omission of sprites on the C16 is a sad loss, and it may make it harder to write quality games for it. The Spectrum programs show that attribute problems can be overcome by skillfull software, but it has taken a few years to work it out. How fast it will be worked out on the C16 remains to be seen.

Graphics control

Controlling the graphics from BASIC is easy, with commands like DRAW, LOCATE, BOX, CIRCLE and PAINT. The Circle command doesn't just draw circles, but can produce ovals, arcs, or any kind of polygon. The CHAR command is a sort of PRINT AT command, that works in any graphics mode, and PAINT is a way of filling in areas with solid colours. The speed of the graphics generally, and especially the filling, is positively slothful, and the BASIC itself is a little slower than previous CBM dialects.

The sound facilities of the C16 are not half as good as the 64s, but are more than adequate. There are two voices, one for notes, and one for notes or noise, with overall volume control. It may have more features, but they are not accessable from BASIC, and as the Programmers Reference Guide has not yet appeared, I can't say. As it gets piped through the TV, volume is not a problem.

The built-in machine-code monitor is not earth-shattering but it's better than nothing. It has a one line assembler, disassembler, and various memory read and write modes, though it doesn't have single stepping. It seems to have code save & load commands via the cassette recorder (and not disk), but I can't be sure as I had no instructions at all for the monitor.

The C16 comes with its own dedicated cassette recorder, which seems to be the same as the usual one, except that it has a different plug. The tape format is similar to before, so the market for turbo loaders remains for the C16. It also connects to the notoriously slow 1541 disk drive, but also supports a new disk drive, said to be four times faster. The BASIC has various commands to control the disk drive, including various backup procedures, and a DIRECTORY command, and this latter feature is innovative on a Commodore.

As with Sinclair's QL, the 'joystick' sockets on the C16 are something of a con by Commodore. Instead of the usual Atari-type sockets, it is cursed with two strange looking mini-DIN type sockets, so you can't actually plug anything into them except Commodore's own forthcoming 'super joysticks'. They aren't much different to the normal run-of-the-mill switched ones, so putting obscure sockets on the C16 seems to be just a way of selling more joysticks.

The documentation supplied with the review machine was very much pre-release, consisting of a couple of hundred photocopied sheets. What was there was good, though some of the listings were inaccurate. The C16 will be supplied in a 'starter pack' with four cassettes, but these too were unavailable for review.

The C16 is basically a VIC20 with a better display and more RAM, though when using graphics a similar amount is left to the user. It competes directly with the 48K Spectrum and the Oric. Its advantages include the supplied cassette recorder, good keyboard and better graphics, while its disadvantages are the small amount of RAM and uncertain software support. Conversions from the C64 are going to take time, because of the lack of sprites and minimal user RAM, though VIC conversions will be easier. It can in no way be thought of as a replacement for the 64, though if it had 16K RAM more then it would be a certain blockbuster. As it stands it does not have enough RAM for much programming, though Commodore's marketing muscle will probably ensure that a lot of C16s will find their way into Santas sack this year. ■

Andy Pennell

Wot! No software for the C16?

C16 owners needn't despair. There's lots of software available if you know what to look for. Start with our complete checklist

HEAVY DISCOUNTING on the retail price of the C16 starter pack has led many people to conclude that the machine isn't doing too well in the micro marketplace. But Commodore would have us believe that around 100,000 C16's have been sold, so it would not be true to say that there isn't a large enough user base to justify the production of software by third-party companies.

So, to satisfy those C16 owners who haven't been able to find enough software to keep them happy — and to prove that many software companies are rising to the challenge of providing software — here's a checklist, complete at the time of going to press, of all available C16 software. In future issues we'll be looking at software for the Plus/4, and hardware for both machines.

So what are the best C16 programs on the market so far? Cascade's **ACE** (Air Combat Emulator) must be the best shoot-'em-up. It's not a flight simulator as such — there are no take-off and landing sequences — but otherwise it's every bit as exciting as, say, Digital Integration's **Fighter Pilot** for the 64. Using your altimeter, speed and altitude controls and combat radar, you must home in on enemy aircraft and destroy them with heat-seeking missiles. Keyboard or joystick control are possible, and there are excellent sound effects and music.

For fans of the "cutesy" style of arcade games, there's Commodore's **Jack Attack**. Jack is a harmless creature who must avoid being squashed by hostile balloons, by squashing them first. Moving blocks can be used as weapons, but there's a time limit for completing each screen. With nine levels of difficulty, Jack Attack (no reference is intended to Jack Tramiel, we assume) may not be the greatest game graphically, but it's maddeningly addictive.

For value for money, you can't beat Tynesoft's four-game pack at £14.95. The games, **Climb-It, Shoot-It, Munch-It** and **Hop-It**, and versions of the classics Kong, Invaders, Pac-Man and Frogger, and though they're not the most sophisticated programs on the market they should provide hours of fun. Along with software packaged with the C16 — **X-Zap, Punchy, Mayhem** and **Picture Builder** — the four-pack will be the start of a varied and exciting software collection.

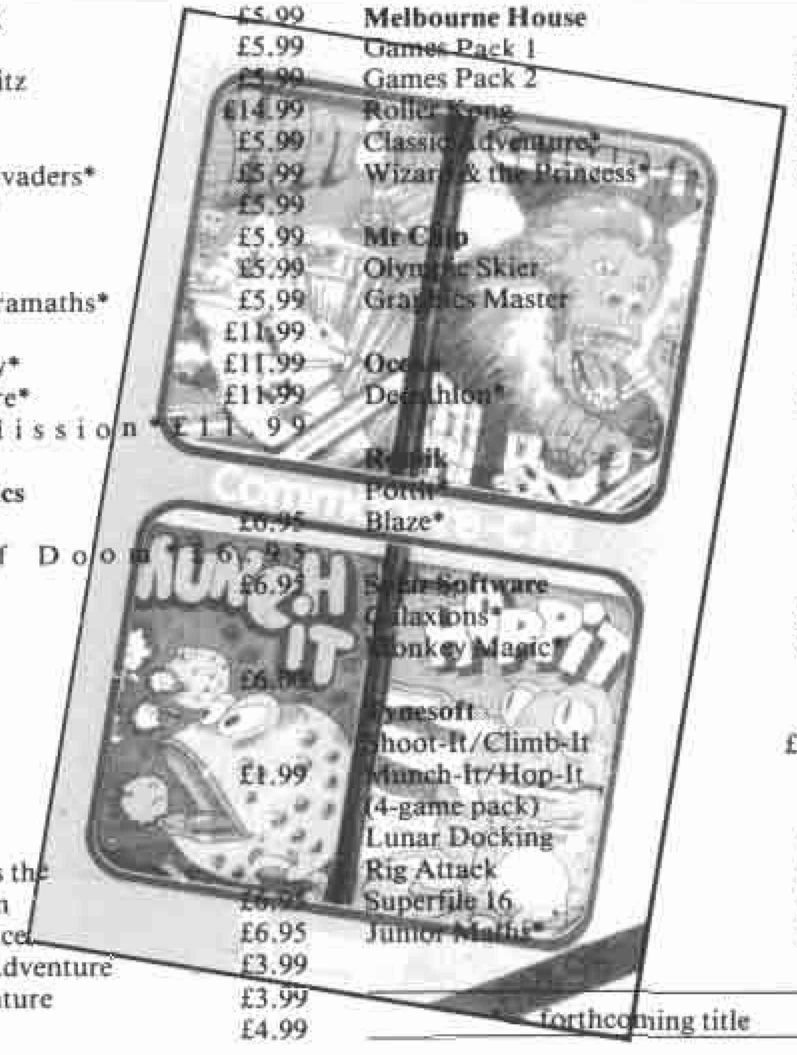
There are already some utility packages available too; Tynesoft's **Superfile 16** is an economical and easy-to-use database which will have hundreds of uses for hobbyists, club secretaries, collectors and small businesses. Up to fifteen fields can be defined for each record, and there are alphabetical and numerical search facilities.

Mr Chip's **Graphic Master** is another useful utility allowing you to redefine the C16's character set, scroll, invert and rotate the characters, then save them to tape or disk. It's a nice package which will be popular with users who chose the C16 for its advanced graphics facilities.

Looking at the list here, you'll see that predictably most of the software so far is in the arcade mould. But there are a few adventures and utilities, and it's likely that if Commodore carries on marketing and supporting the C16, a good software base will be built up. ■

Company/ Title	Price
Anirog	
Skramble	£6.95
Petch	£6.95
Moon Buggy	£6.95
Star Commander	£6.95
3D Time Trek	£6.95
Minipedes	£6.95
Zodiac	£6.95
Flight Path 737	£6.95
Out on a Limb*	£6.95
Bubble Bus	
Hustler	£6.99
Cave Fighter	£6.99
Cascade	
A.C.E.	£9.99
Craig Comms	
Flight 015	£5.95
Creative Sparks	
Tower of Evil	£5.95
CRL	
Berks	£5.95
Commodore	
Mayhem	£5.99
Harbour Attack	£5.99
Crazy Golf	£5.99
Stellar Wars/Blitz	£5.99
Jack Attack	£14.99
UXB*	£5.99
Space Sweep/Invaders*	£5.99
Purple Turtles*	£5.99
Exorcist*	£5.99
Fire Ant*	£5.99
Sandcastles/Paramaths*	£5.99
Viduzzles*	£11.99
Strange Odyssey*	£11.99
Pirate Adventure*	£11.99
Atomic Mission*	£11.99
Gremlin Graphics	
Zargon Wars	£6.95
Petals of Doom*	£6.95
Tycoon Tex*	£6.95
Llamasoft	
Psychedelia	£6.00
Mastertronic	
Jackpot	£1.99
Microdeal	
Cuthbert Enters the Tombs of Doom	£6.95
Cuthbert in Space	£6.95
Williamsburg Adventure	£3.99
Mansion Adventure	£3.99
Arena 3000	£4.99
Melbourne House	
Games Pack 1	£5.95
Games Pack 2	£5.95
Roller Kong	£5.95
Classic Adventure*	£5.95
Wizard & the Princess*	£5.95
Mr Chip	
Olympic Skier	£5.99
Graphics Master	£5.99
Ocean	
Decathlon*	£7.90
Romik	
Pottit*	£6.99
Blaze*	£6.99
Solar Software	
Galaxions*	£6.95
Monkey Magic*	£6.95
Tynesoft	
Shoot-It/Climb-It Munch-It/Hop-It (4-game pack)	£14.95
Lunar Docking	£6.95
Rig Attack	£6.95
Superfile 16	£9.95
Junior Maths*	£9.95

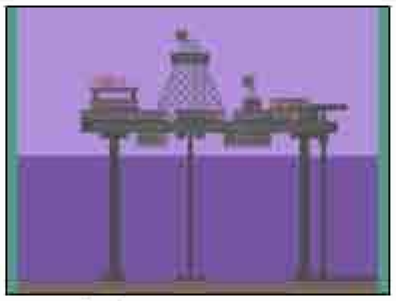
* forthcoming title



Oil be seeing you

This is one of several worthy offerings from Tynesoft, who are doing more than anyone to support the C16. Although the game is very simple, it features excellent use of graphic design and has good basic concept.

You control, using keyboard or joystick, a helicopter which patrols an area containing several oil wells. The border around the screen shows the state of your fuel supply — when it flashes red you must get to a rig and land on a helipad or plunge into the sea.



Your enemies are the submarines which threaten the safety of the oilrigs. They glide beneath the waves firing deadly missiles, and you must stop them by flying low enough to bomb them, while avoiding being destroyed yourself. When the enemy subs are on-screen you can't refuel, and you can't drop bombs while you're descending or you might blow yourself up. All in all it's pretty tricky nailing the enemy subs.

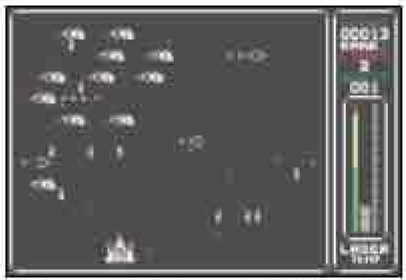
The graphics of the oil rigs are excellently designed — not 3D, not particularly highly detailed, simply well thoughtout. In fact it's doubtful whether you could achieve anything so good on the 64.

In all this program should satisfy most C16 owners, and Tynesoft promises better things to come.

Program: Rig Attack, C16
Supplier: Tynesoft
Price: £6.95
Graphics: ★★★★★★★
Sonics: ★★★★
Gameplay: ★★★★★★

Master blaster

Gremlin's contribution to the growing number of C16 games doesn't have much of the originality and graphic sophistication we've come to expect from the company. It's a very straightforward zapper, based on the arcade favourite Astro Blaster. This was always one of my favourite Invaders variants — the baddies come in waves, either downwards, sideways, or diagonally, and to add to the threat of alien bombs you have a fuel limit and an overheating laser to cope with. Xargon Wars keeps most of these elements, and offers keyboard or joystick control. The graphics are nothing extraordinary, either in design or concept, and some of the aliens seen remarkably slow (though your ship can move pretty quickly from side to side) so the early stages aren't much of a challenge.



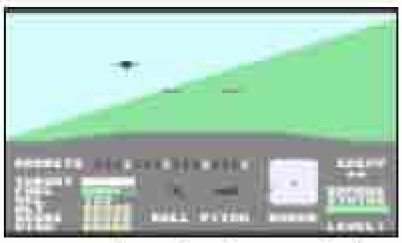
All the usual elements are there; zippy sound effects, high score table, bonuses for completing screens, and so on. It will keep a lot of C16 owners happy while they're waiting for something more spectacular to come along.

Program: Xargon Wars, C16-Plus/4
Supplier: Gremlin Graphics
Price: £6.95
Graphics: ★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★

High flyer

It's a good sign that software this good for the C16 is now starting to turn up. A.C.E. is an air combat emulator which is about as good as anything you can expect to see considering the limited memory of the C16.

Like Digital Integration's Fighter Pilot for the 64, ACE gives you the view from the cockpit of a high-performance fighter aircraft. Your mission is to intercept and destroy enemy planes.



Your radar display gives you the bearing and rough altitude of the enemy aircraft. Your armament consists of normal and heat-seeking missiles, but you must remember that enemies will always retaliate when you attack them, so it's best to make the first shots count.

Fuel, velocity, altitude, pitch, roll and damage indicators help you to monitor your situation, and you can use keyboard or joystick.

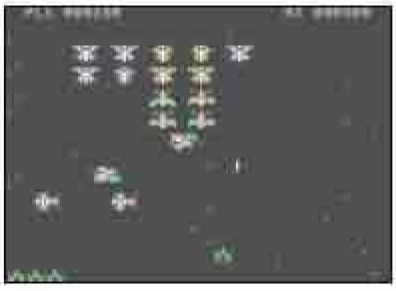
As a bonus there's a stirring musical intro, and the whole thing is remarkably polished and exciting.

Program: A.C.E., C16
Supplier: Cascade
Price: £9.99
Graphics: ★★★★★★
Sonics: ★★★★★★
Gameplay: ★★★★★★★★

Galactic

There aren't many variations possible on the age-old theme of Galaxions, but when it's done well it can still be a very exciting game. Solar's version for the C16 is a conversion of a program for earlier machines, but stands up fairly well.

Compared to the similar Xargon Wars from Gremlin, Galaxions is perhaps inferior graphically but superior sonically. As usual, the baddies stream in from the sides of the screen, whirling and diving, then form a pattern at the top. Your gunship moves left and right at the bottom, and all you have to do is zap the baddies as they dive down at you. The animation is a bit flickery, but to make up for it the sound effects are very good — it just goes to show what can be done with a little imagination, even when you're limited by the C16's sound chip.



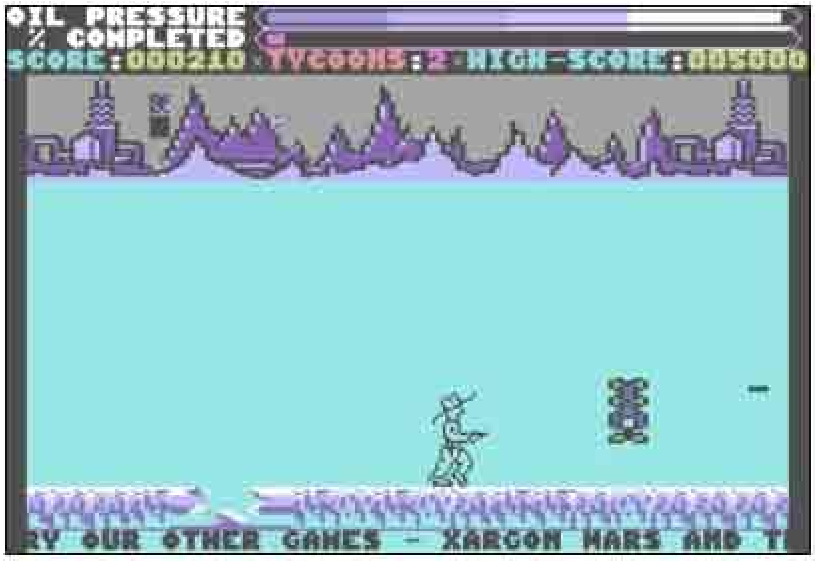
You have the traditional three lives, and there are lots of screens to get through. What more could you want? Well, it would be nice to have something which isn't a conversion, and ideally a totally original idea. Still, I expect most C16 owners are first-time computer users, and may well be happy to play versions of the classics for the moment. I'm looking forward, though, to seeing more originals and fewer oldies on the C16.

Program: Galaxions, C16
Supplier: Solar
Price: £6.95
Graphics: ★★★★★
Sonics: ★★★★★★
Gameplay: ★★★★★

Rootin' tootin'

Tycoon Tex is one of the better C16 efforts so far. Released by Gremlin at the same time as two other C16 programs, Xargon Wars and Petals of Doom, it's also a reasonably original idea, which makes a nice change.

Tex is a gun-toting oil tycoon whose mission is the patrol his pipeline defending it from all sorts of horrors — breaks in the line, bouncy springs, fighter attacks, arrows and bombs.


OIL PRESSURE
% COMPLETED
SCORE:000210 TYCOONS:2 HIGH-SCORE:005000
RY OUR OTHER GAMES - XARGON WARS AND T

You control Tex's speed as he dashes along the pipeline, and make him jump to avoid the breaks. There's a time limit for the completion of each of the three screens — desert, ice and surface — but as you go faster the pressure in the pipeline builds up, so it's more likely to burst. If the oil pressure drops to zero, or if you run out of time, you've lost.

You have six lives and ninety-nine levels to complete, with options including keyboard/joystick, sound on/off, and so on. The background graphics are nice though the foreground characters aren't remarkable, and the sound's OK.

All in all, it's a decent pot-boiler, and I suppose we should be grateful that the basic idea is fairly original.

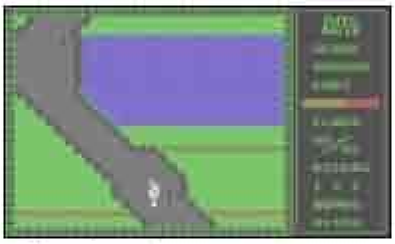
Program: Tycoon Tex, C16
Supplier: Gremlin Graphics
Price: £6.95
Graphics: ★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★

BMX-cellent

Mastertronic's contribution to the C16 software market is a conversion of one of their earliest and most popular games. Despite its simplicity it's still selling well in its other versions, and at the usual £1.99 is extraordinary value for money.

There are five colourful BMX courses, seen from above, through which you have to steer your bike. There are hazardous obstacles to avoid including crash barriers, oil slicks and fences, and you must pick up marker flags as the track scrolls past. Missing any of the flags means that you'll crash at the end of the course.

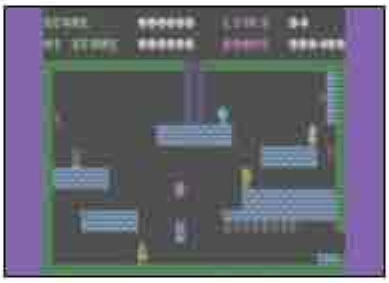
Just to make life more difficult there's an energy readout at the top of the screen, and if you run out of energy before completing the course you'll crash again. Energy pods along the way offer the chance to top up on "fuel", so if you can't manage higher speeds you still have a chance to get through if you're skilful enough. Keyboard or joystick control and an awesome bit of insert artwork complete this fun package from the undisputed kings of the software supermarket.



Program: BMX Racers, C16
Supplier: Mastertronic
Price: £1.99
Graphics: ★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★★

Scramble, egg

This is a 27-screen arcade effort which is probably the best thing of its kind seen so far on the C16. This, of course, isn't saying a great deal, since there's not a great deal of competition. Still, it's all perfectly good fun, as you, playing the accursed Prince Harry, attempt to get through the Dark Tower. Things aren't made easier by the fact that you have been turned into a boiled-egg shaped mutant by the evil Guardian of the tower.



So your little eggy has to collect all the magic jewels hidden around the tower, in order to regain his rightful form. It's all pretty routine stuff, with ladders to climb, platforms to leap onto, and various googlies to avoid. The animation is very smooth but the screen design is a little minimal.

At the end of each game, you have the option of restarting completely, or starting from the last room of the previous game (though you will still lose all your jewels.

The game becomes progressively more difficult as you get further into it, so it should bring some satisfaction into the lives of C16 owners in search of a challenge. Hardly the most exciting prog on the market, however.

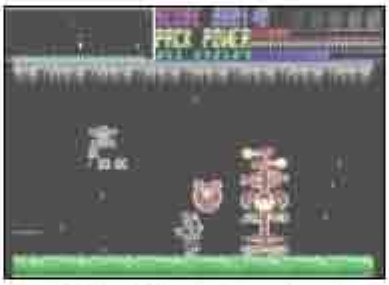
Program: Dark Tower, C16
Supplier: Melbourne House
Price: Tape £6.95
Graphics: ★★★★
Sonics: ★★★★
Gameplay: ★★★★★

Plantlife

Petals of Doom is one of the zappiest games yet for the C16, a combination of Defender and Gardeners' Question Time.

There isn't much logic to the game, but who cares. You are a spaceman, armed with the obligatory jetpack and laser gun, trapped in a series of underground caverns. Each cavern is bounded by a force field, and your only hope of escape is to help the alien plants reach their full growth, at which time the forcefield collapses (don't ask me why). Of course, there are all sorts of fast-moving baddies out to stomp on the flowers and blight your chances of escape.

The graphics are excellent, bearing in mind the limits of the C16, and the sound effects good too. Both joystick and keyboard control are fast and positive. You'll lose a life if overuse of your laser and jetpack, or repeated collision with aliens, depletes your power supply, the status of which is given by a bar graph.

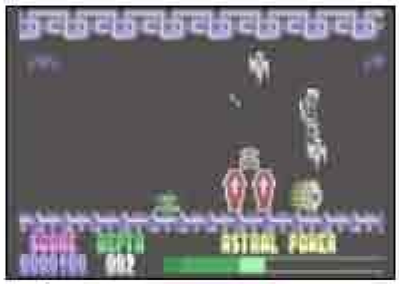


A good, fast shoot'em-up, worthy of inclusion in any C16 owner's software library. Gremlin's forthcoming 64 game **Thing on A Spring** should also be worth looking out for, if it maintains this high quality.

Program: Petals of Doom, C16
Supplier: Gremlin Graphics
Price: £7.95
Graphics: ★★★★★★
Sonics: ★★★★★★
Gameplay: ★★★★★★

Warlocks

One of the first decent arcade adventures for the C16, **Sword of Destiny** manages to pack quite a lot into 16K. As usual with this sort of program, there's a lot of scene-setting dealing with evil warlocks, brave warriors, Abysses of Death, magic, mystic energy and stolen hearts, but you don't want to know that — you just want to know what you get to kill, don't you?



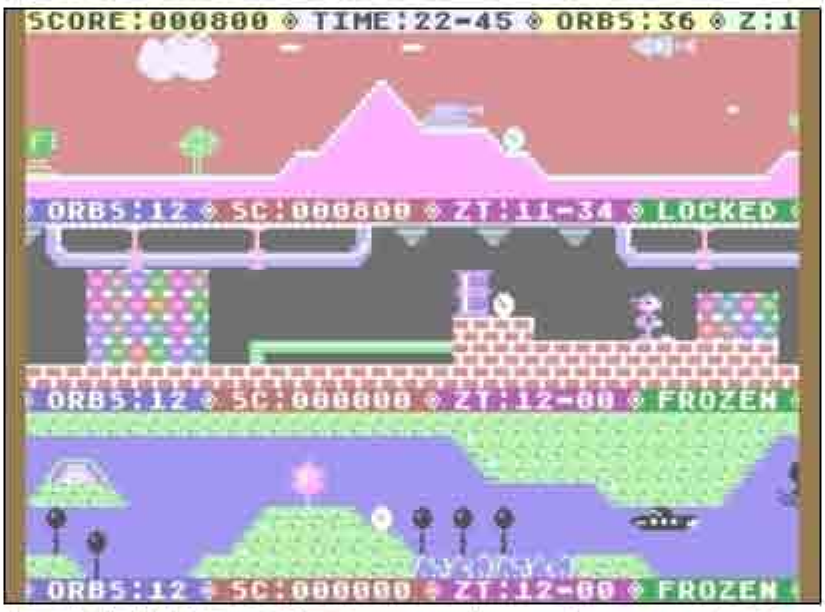
The warrior is depicted by a large, single-colour sprite, and the baddies include ghosts, skulls, serpents and so on. Your magic sword soon makes short work of them, though, adding to your reserves of power. There are many flashing objects to be found, each one of which opens up the way to the next part of the Abyss. If you find yourself stuck at any level, you can take a ride courtesy of a hot air flue, which will bounce you up to another level.

Joystick or keyboard options, and reasonable sound effects and music, complete what is a decent C16 offering. Also available from Gremlin for the C16 is **Dork's Dilemma,** which is playable, but not so sophisticated graphically.

Program: Sword of Destiny, C16
Supplier: Gremlin Graphics
Price: £6.95
Graphics: ★★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★★

Time for the C16

This one's clever — why hasn't anyone thought of it before? Although the main idea is pretty jaded — a sort of **Scramble** game through various stages, avoiding fire from enemy craft and destroying targets along the way — the gimmick is that **Timeslip** gives you three stages on the screen all at once.


SCORE:000800 ◈ TIME:22-45 ◈ ORBS:36 ◈ Z:1
ORBS:12 ◈ SC:000800 ◈ ZT:11-34 ◈ LOCKED
ORBS:12 ◈ SC:000800 ◈ ZT:12-00 ◈ FROZEN
ORBS:12 ◈ SC:000800 ◈ ZT:12-00 ◈ FROZEN

In the top screen, you pilot a fighter over a mountain range: the middle, a jetpacker through tunnels: the bottom, a submarine through underwater caverns. In each scene there are twelve Time Orbs to destroy. You have twenty-four subjective hours to clear all three screens, but if you lose too many lives the three time-zones will slip out of synchronisation. Only by clearing all three and resetting the time readings to nought you can save the universe. You can gain time by hitting fuel drums or barrels.

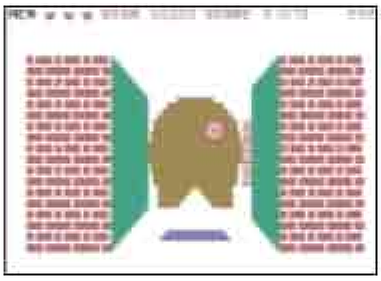
Decent graphics and piercing ZEEP ZEEP sound effects. Good fun for arcade fanatics.

Program: Timeslip, C16
Supplier: English Software
Price: £7.95
Graphics: ★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★★

Hee-hee in 3-D

Is the world ready for another version of PacMan at this stage of the game? Audiogenic obviously think that the answer is yes, since they've now released a version of Supersoft's 3-D Glooper for the C16. Fortunately, the "3-D" element of the title gives away the fact that this isn't the normal bird's eye view maze game. Although the plot is the same — make your way around a maze, avoiding the gloopers while collecting magic squares, seeking out the power packs which enable you to turn the tables and devour the gloopers; the screen display is an impressive perspective graphics affair seen from the viewpoint of the player.

There are 300 dots to collect as you make your way around the maze. Pressing F1 brings up a map showing your current position and that of the nearest glooper. You'll also hear an alarm sound if there are gloopers nearby.



Control is by joystick or keyboard, and though the sound effects are minimal they're quite sufficient.

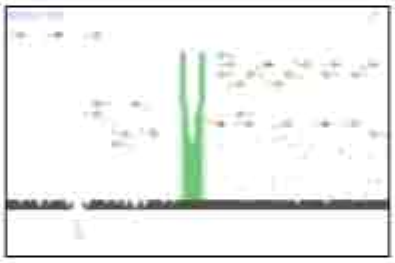
The biggest problem with 3-D Glooper is the packaging. The artwork is so awful, resembling something done in crayon by an 8-year-old, that it could put off potential purchasers. A pity, since this is the first game of its kind for the C16, and it delivers a good deal more than the artwork suggests.

Program: 3-D Glooper, C16
Supplier: Audiogenic/Supersoft
Price: £4.95
Graphics: ★★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★★

Plant life

Audiogenic are understandably reluctant to push the merits of their C16 titles — claiming that they are "as much an advance in software terms as the C16 was over the 64 in hardware". Still, with Commodore on the verge of killing the C16 it's marvellous to see anything coming out for the poor little micro.

Kaktus can claim, if nothing else, the month's most bizarre scenario; a horde of giant wasps is attacking a desert cactus, and only you can stop them. Your little man moves through underground tunnels, surfacing to shoot at the wasps and larger, more dangerous hornets. Additional dangers include moles and wasp droppings (huh?), and on later screens an enraged buzzard.

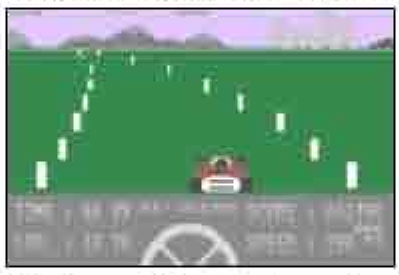


The product of a deranged mind. Approach with caution.

Program: Kaktus, C16
Supplier: Audiogenic
Price: £4.50
Graphics: ★★★
Sonics: ★★★★
Gameplay: ★★★★

Into gear

Good games for the C16 are few and far between. But Mastertronic's **Formula 1 Simulator** is the exception that proves the rule. Using the by now familiar cockpit point of view, this particular variation on the classic **Pole Position** features smoothly — scrolling, varied backdrops (countryside, cities, etcetera), other cars, and a fiendishly devious circuit, marked out by parallel rows of 'roadside markers'. Sound effects are few, but passing another car elicits a nice "EE-ow!", and crashing (which terminates your go) is suitably cataclysmic.



The cars are nicely drawn, large, colourful, and rather cartoon like, with overall realism enhanced by the presence of a steering wheel on the lower half of the screen, which also features hi score, speed, number of laps completed, and so on. The better you hold your car on course (no easy matter on the numerous tight corners), the faster you will go. An additional level of control is provided in the provision of Hi and Lo gears, which can be toggled using the joystick fire button.

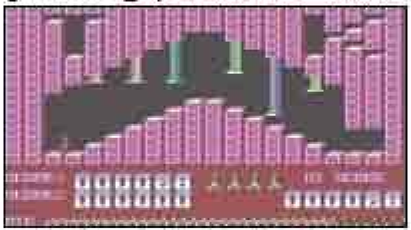
I have seen worse racing games on more sophisticated machines, at five times the price. If you have a C16, you *must* have this game.

Program: Formula 1 Simulator
Supplier: Mastertronic
Price: £1.99
Graphics: ★★★★★★★★
Sonics: ★★★★★★★
Gameplay: ★★★★★★★★

Jumpers

Above average ladder-and-platforms stuff from Mastertronic, **More Adventures of Big Mac the Mad Maintenance Man**, to give it its full title, looks pretty good for a C16 game.

There are fifteen screens to get through, in each of which your time is limited by the air level shown at the bottom of the screen. There are stompers, lasers, elevators, pitfalls and more to negotiate, though fortunately the game includes an all-too-rare feature; each time you restart, you can move to any level you have reached before by repeatedly pressing the V key — so you don't have to waste time getting through screens you've already mastered. Good idea.



The usual LEFT-RIGHT-JUMP business, but particularly well done.

Program: Big Mac, C16
Supplier: Mastertronic
Price: £1.99
Graphics: ★★★★★
Sonics: ★★★★★
Gameplay: ★★★★★★

Top Five C16 Games

1	Gremlins	Adventure International £9.95
2	BMX Racers	Mastertronic £1.99
3	Blagger	Alligata £6.95
4	Zargon Wars	Gremlin £6.95
5	Flight Path 737	Anirog £6.95

Bubblers

1	Vegas Jackpot	Mastertronic £1.99
2	Wizard & the Princess	Melbourne £5.95
3	Classic Adventure	Melbourne £5.95

All data compiled by RAM/C from a panel of specialist dealers. Sales period — 4 weeks prior to June 12th 1985

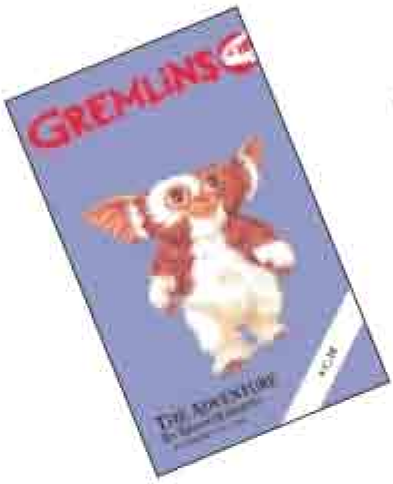
Top Five C16 Games

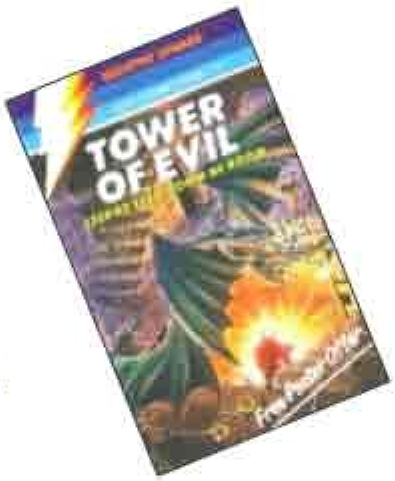
1	(1)	Gremlins	Adventure International £9.95
2	(—)	Dork's Dilemma	Gremlin £6.95
3	(2)	BMX Racers	Mastertronic £1.99
4	(—)	Dark Tower	Melbourne House £5.95
5	(3)	Blagger	Alligata £6.95

Bubblers

1	Tower of Evil	Thorn EMI £5.95
2	Olympiad	Tynesoft £6.95
3	Flight Path 737	Anirog £6.95

All data compiled by RAM/C from a panel of specialist dealers. Sales period — 4 weeks prior to July 5th 1985





Top Five C16 Games

1	(—)	Airwolf	Elite £6.95
2	(3)	BMX Racers	Mastertronic £1.99
3	(2)	Dork's Dilemma	Gremlin £6.95
4	(4)	Dark Tower	Melbourne House £5.95
5	(—)	Tower of Evil	Thorn EMI £5.95

Bubblers

1	Gremlins	Adventure International £9.95
2	Spectipede	Mastertronic £1.99
3	Vegas Jackpot	Mastertronic £1.99

All data compiled by RAM/C from a panel of specialist dealers. Sales period — 4 weeks prior to August 16th 1985

Top Five C16 Games

1	(—)	Formula 1 Simulator	Mastertronic	£1.09
2	(—)	Daley Thomson's Star Events	Ocean	£7.95
3	(2)	BMX Racers	Mastertronic	£1.09
4	(—)	Spectipede	Mastertronic	£1.09
5	(—)	Airwolf	Elite	£7.95

Bubblers

1	Vegas Jackpot	Mastertronic	£1.99
2	Cruncher	Mastertronic	£1.99
3	Suicide Run	Mastertronic	£1.99

Data compiled by Gallup from a panel of specialist and chain stores, for the month ending September 7th.

Top Five C16 Games

1	Formula One Simulator	Mastertronic	£1.99
2	Daley Thompson's Star Events	Ocean	£7.95
3	BMX Racers	Mastertronic	£1.99
4	Squirm	Mastertronic	£1.99
5	Spectipede	Mastertronic	£1.99

Bubblers

1	Air Wolf	Elite	£6.95
2	Quick Draw	Solar Software	£2.99
3	World Cup	Elite	£6.95

Data compiled by Gallup from a panel of specialist and chain stores, for the month ending September 7th.

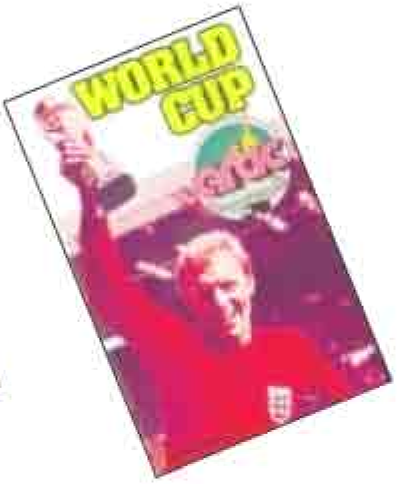
Top Five C16 Games

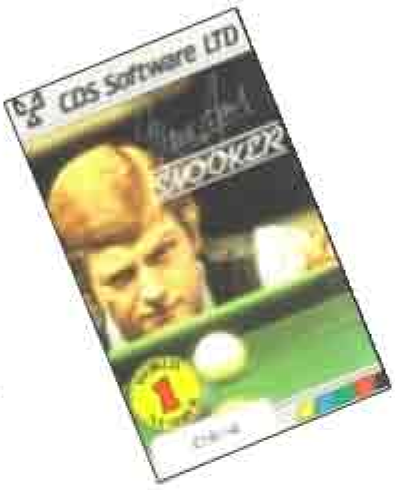
1	(1)	Formula One Simulator	Mastertronic	£1.99
2	(4)	Squirm	Mastertronic	£1.99
3	(3)	BMX Racers	Mastertronic	£1.99
4	(2)	Daley Thompsons Star Event	Ocean	£6.95
5	(5)	Spectipede	Mastertronic	£1.99

Bubblers

1	Vegas Jackpot	Mastertronic	£1.99
2	Air Wolf	Elite	£6.95
3	Steve Davis Snooker	CDS	£6.95

Data compiled by Gallup from a panel of specialists and chain stores, for the month ending November 7th





Top Five C16 Games

1	(1)	Formula One Simulator	Mastertronic	£1.99
2	(3)	BMX Racers	Mastertronic	£1.99
3	(4)	Squirm	Mastertronic	£1.99
4	(2)	Daley Thompson's Star Event	Ocean	£6.95
5	(5)	Spectipede	Mastertronic	£1.99

Bubblers

1	Vegas Jackpot	Mastertronic	£1.99
2	Tutti Frutti	Mastertronic	£1.99
2	Air Wolf	Elite	£6.95

Data compiled by Gallup from a panel of specialist and chain stores, for the month ending November 22nd.

Top Ten C16 Games

1	(1)	Formula One Simulator	Mastertronic	£1.99
2	(—)	Tutti Frutti	Mastertronic	£1.99
3	(3)	BMX Racers	Mastertronic	£1.99
4	(2)	Squirm	Mastertronic	£1.99
5	(—)	Rockman	Mastertronic	£1.99
6	(B)	Vegas Jackpot	Mastertronic	£1.99
7	(4)	Daley Thompsons Star Events	Ocean	£6.95
8	(—)	C16 Classics	Gremlin Graphics	
9	(5)	Spectipede	Mastertronic	£1.99
10	(B)	Air Wolf	Elite	£6.95

Data compiled by Gallup from a panel of specialist and chain stores, for the month ending December 31st.

Charting your progress

***David Lawrence** shows how to create a variety of charts and graphs*

THE C16 is particularly good at displaying information in ways which are more immediately understandable than lists of facts and figures. This article (together with next month's) consists of three programs which will create graphs of one kind or another, both in low and high resolution.

The programs are: Graph which creates a very effective and powerful line graph in high resolution, Piechart which will take a limited amount of data and display it in the form of a circle split into multi-colour segments and Graph II which allows you to create a stunning 3-D barchart.

One of the problems of working within 16K of memory is that high resolution programs will always eat into what is available, leaving very little scope for the manipulation of data in applications programs. In the first **Graph** program, we examine one way round this limitation by creating a program which does not have to devote memory to placing prompts on the screen, storing data in arrays or placing data for storage on to disk or tape. Techniques introduced during the course of the **Graph** program include the flexible use of DATA statements and the use of logical conditions as variables.

Modules 2.1.1 and 2.1.2 hold the data for the graph. Don't be concerned if at this stage you find it difficult to see the relevance of all the figures. Their use will become plain once you have entered and run the program once or twice, and experimented with making some changes.

Commentary

Lines 3030-3050: These lines contain the name which the user wishes to give the graph as a whole, and the labels to be attached to the vertical and horizontal axes. Note that in each case the phrase before the comma in DATA statement is there solely for the convenience of the user and will be ignored by the program — the comma is essential to separate the first phrase from the important information which follows it.

Lines 3060-3070: The two axes of the graph will be divided up into units for ease of reading. The axes will always be the same length, but the user can specify how many units each will be split up into.

Line 3080: If, for example, the graph was meant to be recording tons of wheat produced by a country over a number of years, the user might wish to make each unit on the vertical axis record a unit of 1,000 tons. Rather than make the user divide up the actual figure into units of 1,000 before entering it, the figure in this line allows the units to be specified so that every figure can be entered in full.

Lines 4030-4040: These lines hold the DATA on which the graph will be based. In this example, these figures will produce a smooth, bell-shaped curve. Note that there is provision for only one figure for each unit on the horizontal axis, starting at position one, though the whole of the horizontal axis does not have to be used. As many DATA items as desired may be included on each separate line, within the limit of two screen lines for a program line.

Lines 4050-4090: These lines allow more data to be entered, though they are not needed in the test graph presented here. You can use as many data statements as memory will allow, but the information must be terminated by a DATA statement containing the word END. This is the signal to the program that it has reached the end of the DATA to be used for the graph — even if more DATA statements follow.

Module 2.1.3 draws the framework onto which the eventual graph will be placed, complete with the units on the axes and the various labels specified.

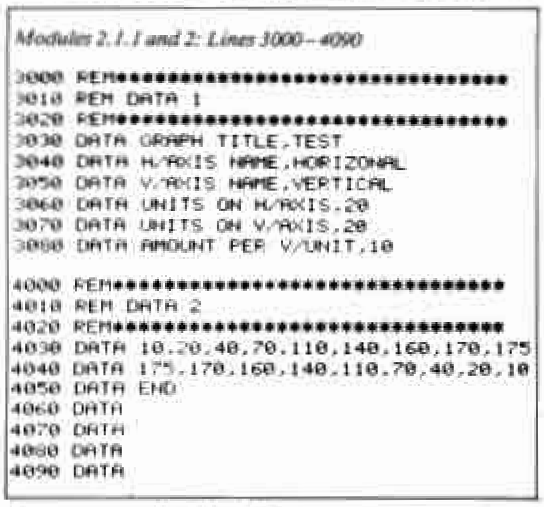
Commentary

Lines 1050-1070: The two axes are drawn, a line from near the top left hand corner of the screen to near the bottom left hand corner, continuing at a right-angle across the bottom of the screen to near the bottom right-hand corner.

Line 1080: The C16's data pointer is set to point to the first item of DATA following the start of the module at 6000. This RESTORE will prevent an OUT OF DATA error being generated if the program is started with GOTO. Using RUN sets the pointer to the first item of DATA anyway.

Lines 1090-1110: The labels for the graph as a whole, the horizontal axis and the vertical axis are read from the DATA statements and printed on the screen. In the case of the label for the vertical axis, a loop is used to print the label character by character down the left hand side of the screen.

Note that in each case there are two READ instructions. The first one picks up the phrase *before* the comma in the DATA statement. This is then immediately discarded by READing another string into the same variable, T$.



Modules 2.1.1 and 2: Lines 3000–4090

```
3000 REM***********************************
3010 REM DATA 1
3020 REM***********************************
3030 DATA GRAPH TITLE,TEST
3040 DATA H/AXIS NAME,HORIZONAL
3050 DATA V/AXIS NAME,VERTICAL
3060 DATA UNITS ON H/AXIS,20
3070 DATA UNITS ON V/AXIS,20
3080 DATA AMOUNT PER V/UNIT,10

4000 REM***********************************
4010 REM DATA 2
4020 REM***********************************
4030 DATA 10,20,40,70,110,140,160,170,175
4040 DATA 175,170,160,140,110,70,40,20,10
4050 DATA END
4060 DATA
4070 DATA
4080 DATA
4090 DATA
```

Line 1120: The number of divisions to be placed on the vertical axis (NV) is read from the next DATA statement. The length of the vertical axis (160 pixels) is then divided by this number to arrive at the length of each division in pixels (LV).

Lines 1130-1150: This loop draws small marks on the vertical axis to record the divisions specified by the user. One thing ▶

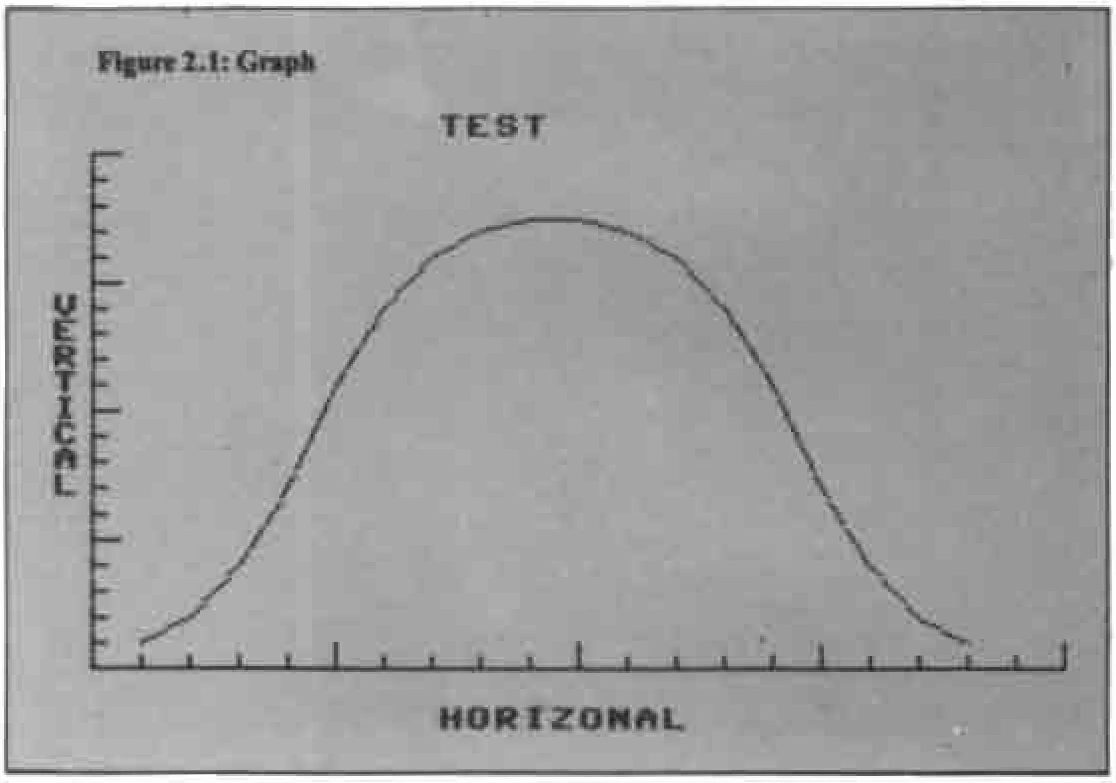


Figure 2.1: Graph

◀ to make note of here is the expression 4*(I/5 = INT(I/5)), which has the effect of making the mark for each unit bigger than the rest. To understand the expression, you need to know something about the way in which logical conditions are created by the C16.

When the C16's BASIC Interpreter, the massive machine code program which runs BASIC for you, comes across a condition following an IF, something like A>B, A=B or A<=B it needs to determine whether that condition is true or false before deciding whether to carry out the action specified by the IF statement. Thus:

IF A>B THEN GOSUB 1000

will be acted upon if A>B is *true* (eg A=10 and B=9) and ignored if it is false (eg A=9 and B=10). In order to make its decision, the condition is evaluated according to the current values of A and B (or whatever variables have been specified) and the condition is given a *value* which is minus one if the condition is true and zero if the condition is false. It is the *value* rather than the condition itself that matters, a fact which you can demonstrate for yourself by conducting the following small test. Enter, in direct mode:

A=3[RETURN]
IF A THEN PRINT "TRUE" *[RETURN]*

The result will be that TRUE is printed on the screen since the value following the IF statement is not zero (any non-zero value, positive or negative, will produce the same effect). Now try:

A=0[RETURN]
IF A THEN PRINT "TRUE" [RETURN]

Module 2.1.4: Lines 2000–2200

```
2000 REM**********************************
2010 REM DRAW GRAPH
2020 REM**********************************
2030 TRAP 2180
2040 RESTORE 4000
2050 COLUMN=1
2060 READ T1$,T2$
2070 DO UNTIL T2$="END"
2080 READ T3$
2090 X1=20+LH*COLUMN
2100 Y1=180-INT(VAL(T1$)/UNIT*LV)
2110 COLUMN=COLUMN+1
2120 X2=20+LH*COLUMN
2130 Y2=180-INT(VAL(T2$)/UNIT*LV)
2140 DRAW,X1,Y1 TO X2,Y2
2150 T1$=T2$ : T2$=T3$
2160 LOOP
2170 GETKEY A$
2180 GRAPHIC 0,1
2190 LIST 3000-
2200 END
```

This time, nothing will be printed on the screen. At the moment, however, we are not so much interested in the way IF works but in the way conditions are evaluated, so try the following:

A=1[RETURN]
B=1[RETURN]
PRINT A=B[RETURN]

What you should see is −1, the value of a true condition. Now try:

A=1[RETURN]
B=2[RETURN]
PRINT A=B[RETURN]

The result now will be 0, since the condition is not true. At first this may seem interesting but moderately irrelevant. In fact, this ability to extract a value from a logical condition is of great value in programming, as line 1140 illustrates.

What line 1140 does is to draw a series of lines at right angles to the vertical axis to mark the divisions specified by the user. The length of these lines is normally four pixels. However, whenever the value of the loop variable I is exactly divisible by five (ie every fifth mark) the condition (I/5 = INT(I/5)) will be true and will take on the value minus one rather than zero. In other words, including −4*(I/5 = INT(I/5)) in the line which specifies the length of the mark to be DRAWN, allows every fifth mark to be doubled in length without the use of a complex IF. .THEN. .ELSE statement.

Module 2.1.3: Lines 1000–1200

```
1000 REM*********************************
1010 REM DRAW GRID
1020 REM*********************************
1030 COLOR 0,1 : COLOR 1,8,4
1040 GRAPHIC 1,1
1050 LOCATE 20,20
1060 DRAW TO 20,180
1070 DRAW TO 300,180
1080 RESTORE 3000
1090 READ T$,T$ : CHAR,15,1,T$
1100 READ T$,T$ : CHAR,15,24,T$
1110 READ T$,T$ : FOR I=1 TO LEN(T$) : CHAR,0
,7+I,MID$(T$,I,1) : NEXT
1120 READ T$,NV : LV=INT(160/NV)
1130 FOR I=1 TO NV
1140 DRAW,20,180-I*LV TO 24-4*(I/5=INT(I/5)),
180-I*LV
1150 NEXT
1160 READ T$,NH : LH=INT(280/NH)
1170 FOR I=1 TO NH
1180 DRAW,20+I*LH,180 TO 20+I*LH,176+4*(I/5=
INT(I/5))
1190 NEXT
1200 READ T$,UNIT
```

Note that to *add* four to the length of the mark, we have to *take away* four times the value of the condition, since its value when true is *minus* one — taking away a minus number is equivalent to adding a positive one.

Lines 1160-1190: Exactly the same process is carried out for the horizontal axis. Note that here, the small marks are drawn upwards from the axis and so move from position 180 to position 176 down the screen. To increase the length of the mark, the 176 has to be reduced to 172, and this is done by *adding* four times the value of the true condition.

Line 1200: The number of units represented by each division on the vertical axis is read from the DATA statement.

Testing

To test this part of the program, all that is necessary is to RUN the program so far. You should see the graph grid drawn on the screen with TEST at the top, VERTICAL down the left-hand side and HORIZONTAL along the bottom. The two axes should be neatly divided up into 20 units each.

Hint: At the end of this test you will be left in high resolution mode. There is no need to use a function key to produce GRAPHIC and add 0 or to try to type in GRAPHIC0 blind to get back to the text screen. Simply press *any letter key* (not a number), followed by RETURN. This will generate a syntax error which automatically returns you to low resolution. This simple technique will save you a great deal of time in the future.

Module 2.1.4 draws the graph itself, using the information specified in the DATA2 module (Module 2.1.2).

Commentary

Lines 2030 and 2170-2190: Pressing STOP at any time during the main drawing sequence, or pressing any key once the graph has been drawn, returns the C16 to low resolution and lists out the data on which the graph is based. This makes examining and changing the graph extremely easy.

Line 2050: COLUMN will be used to record the position of an item of data along the horizontal axis.

Line 2060: Two items of DATA are picked up. The reason that two are READ before the main loop is that the graph can only be begun if there is a place to draw *from* and a place to draw *to*.

Line 2070: On each pass through the loop, a new line will only be drawn if there is a valid co-ordinate for it to be drawn to rather than the END indicator.

Lines 2110-2130: These lines define the two points between which two points on the graph will be drawn. The X, or horizontal, co-ordinates are calculated by multiplying COLUMN (the number of units the graph has progressed along the horizontal axis) by the length in pixels of the horizontal units (LH). The constant 20 is the distance that the start of the horizontal axis is inset from the left of the screen. ▶

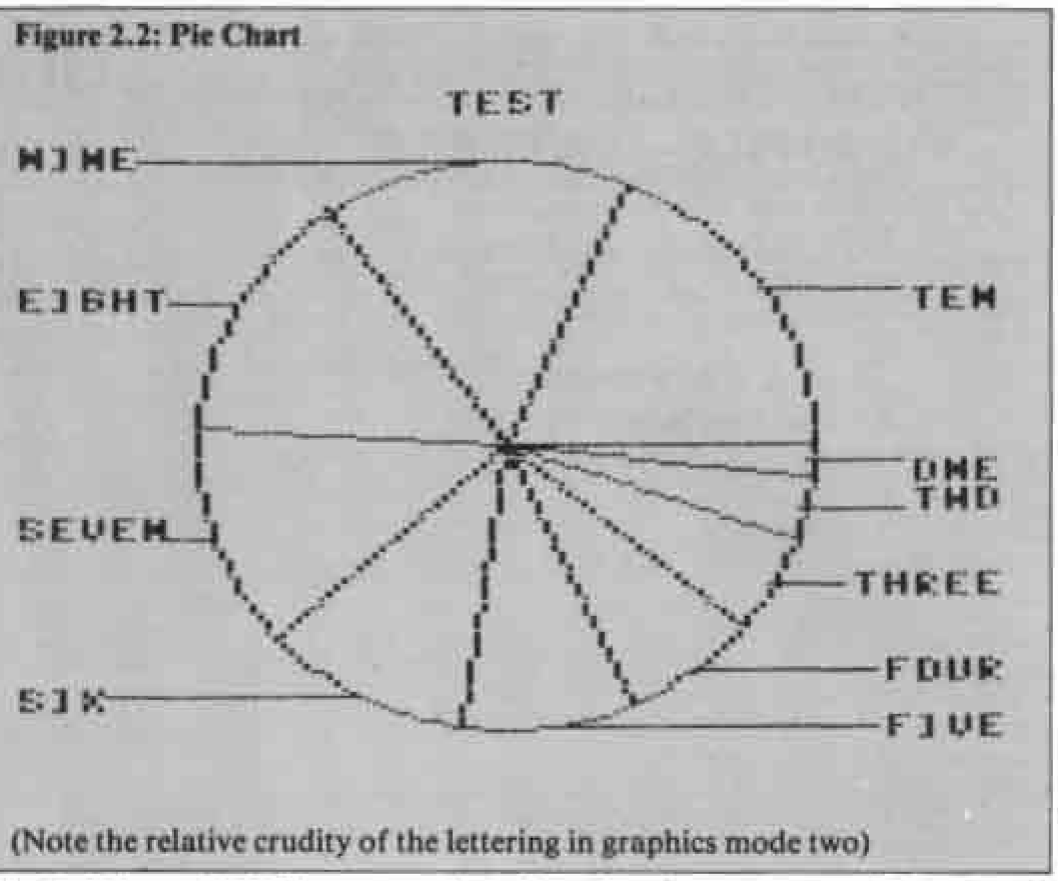
◀ The Y, or vertical, co-ordinates are slightly more complex. Starting from the basis of the bottom end of the axis, which is 180 pixels down the screen, the value of the DATA item is first divided by UNIT. Thus if the DATA item were 1,000,000 and the user had specified that the vertical axis was to be divided into units of 100,000, (UNIT = 100,000), then the result would be 1,000,000/100,000 or 10 units. Having arrived at the number of units, this is then multiplied by the length in pixels of the vertical units (LV). The figure arrived at is subtracted from 180 since the screen is numbered from zero at the top.

Line 2150: Each time a single line is drawn on the graph, its end point becomes the start point for the next line. If T3$ contains END, however, the following line will terminate the execution of the loop.

Testing

RUN the finished program and you should see a smooth, bell shaped curve drawn. When drawing is finished, press any key and you should see the DATA modules listed onto the screen so that you can alter them at will.

Figure 2.2: Pie Chart



(Note the relative crudity of the lettering in graphics mode two)

One very useful way of presenting small amounts of data is the pie chart technique, where a circle is broken up into segments representing the proportion of a total handed over to the different parts which make it up. In the program which follows we shall draw on what we have already learned about the mathematics of circles and the flexible use of DATA statements in the last program.

As with the previous high resolution graph, the figures on which the present graph will be based are contained in DATA statements, this time in module 2.2.1. Note, however, that in the program as listed, the two arrays which will be used to hold the name of each item and its value are not dimensioned, so you are limited to 10 items.

Frankly, a pie chart with more than 10 items is of little value, because it becomes too crowded to really take in the information. Even so, you can if you wish include a dimensioning statement at the beginning of the program. If you do include such a statement and increase the number of items, you may find that you have to drop some of the module headings, since the program runs right up against the limits of memory.

Module 2.2.1: Lines 4000–4080

```
4000 REM********************************
4010 REM DATA FOR CHART
4020 REM********************************
4030 DATA TITLE,TEST
4040 DATA NUMBER OF ITEMS,10
4050 DATA NAMES,ONE,TWO,THREE,FOUR,FIVE,SIX,S
EVEN,EIGHT,NINE,TEN
4060 DATA
4070 DATA QUANTITIES,1,2,3,4,5,6,7,8,9,10
4080 DATA
```

Module 2.2.2: Lines 5000–5110

```
5000 REM********************************
5010 REM PROCESS DATA
5020 REM********************************
5030 RESTORE 4000
5040 READ T$,NAME$
5050 READ T$,ITEMS
5060 READ T$ : FOR I=0 TO ITEMS-1 : READ NAME
$(I) : NEXT
5070 RESTORE 4070
5080 SUM=0 : READ T$ : FOR I=0 TO ITEMS-1 :
READ T : SUM=SUM+T : NEXT
5090 RESTORE 4070
5100 READ T$ : FOR I=0 TO ITEMS-2 : READ T :
A(I+1)=(T/SUM)*360+A(I) : NEXT I
5110 RETURN
```

Module 2.2.2 processes the data for the chart. The information contained in the DATA module is read into the variables NAME$ and ITEMS, and the arrays NAME$ and A.

Commentary

Lines 5080-5100: The value of the items to be charted are first added together to discover the total that the circle will represent. The DATA pointer is then RESTOREd to the beginning of the quantity figures and each quantity is translated into a second figure which, when divided into 360, would give the same result as the original quantity divided into the total. For instance, if the total were 100, and the quantity for one item were 25, this would be translated into 90, or 25% of 360. These new figures will later be used to determine how large a chunk of the pie chart will be given to each item.

Testing

Enter the following lines from what will eventually be the control module, and then RUN the program:

```
1040 GOSUB 5000
1110 END
```

If all is well, then nothing should happen visibly — only if there is an error of some kind will you see anything. If you wish, however, you can print out the contents of the variables and arrays named in ▶

◀ the module just to reassure yourself.

Module 2.2.3 sets up the graphics mode and associated colours, and draws an 80*80 circle in the centre of the screen, together with the name for the chart.

Commentary

Line 2030: The graphic mode we are going to use is multicolour mode, which will allow us to have three different colours (plus the background) on the screen at the same time. The advantage to this is that it makes the chart easier to read. But the more colours it is possible to have on the screen at any one time, the more complicated it is for the C16 to record the colour of each individual pixel, and the more memory is needed.

Since the graphics screen already eats up enough of the memory, when we go into multicolour mode the operating system does a trade-off between colour and detail. Once in mode three the smallest unit we shall be able to actually *do* anything with will be one pixel down by *two* pixels across. The screen, as far as the C16 is concerned, will no longer be 320 pixels across but 160, thus saving memory which can be devoted to remembering the greater range of colours.

From now on, therefore, whenever you look at the X co-ordinate (across the screen) of anything in this program, remember that you need to double it before it is comparable with the same co-ordinates in previous high resolution programs in the book which use normal high resolution.

Line 2040: The little expression in the CHAR statement simply ensures that no matter how long the title given to the chart, it should always look as if it has been printed roughly in the middle of the top line.

Module 2.2.3: Lines 2000–2070

```
2000 REM*******************************
2010 REM DRAW FRAMEWORK
2020 REM*******************************
2030 GRAPHIC 3,1
2040 COLOR 0,1 : COLOR 1,2 : COLOR 2,4,4 :
COLOR 3,6,4
2050 CHAR ,20-LEN(NAME$)/2,0,NAME$
2060 CIRCLE ,80,100,40,80
2070 RETURN
```

Line 2060: An illustration of the point made earlier about dimensions in multicolour mode. To make sense of the position and dimensions of this circle, multiply the first and third parameters by two.

Testing

Add the following lines and then RUN the program:

1060 GOSUB 2000
1080 GETKEY A$
1090 GRAPHIC 0

The result should be nothing more exciting than the title given to the chart and a white circle. Press any key except RUN/STOP to get back to the normal screen.

Module 2.2.4 draws in the segments into which the chart will be divided, colours them and attaches the labels specified in the DATA module. In order to understand what is going on, you will need to be able to remember the simple mathematics of a circle.

Commentary

Lines 3030-3060: A series of lines are drawn from the centre of the circle to the circumference, dividing the circle up into the segments for the chart. The figures used are those calculated in Module 2.2.2.

Lines 3080-3100: The angles at which the segments start are calculated again, but four degrees are added to each to clear the lines drawn by the previous loop. A position is then calculated on the basis of this angle which is just inside the circumference of the circle. The PAINT command is then used to colour the wedge shaped segment in which the current point falls.

Line 3090 produces a cycle of the three foreground colours specified in Module 2.2.3, except for the last segment, which is left as background colour. This is because it ensures that the final segment cannot be the same colour as the first. Since they are next to each other this would make the chart harder to read.

Lines 3110-3120: These two lines calculate an angle half way between the start and finish point of the current segment. It is possible that in moving from the start point to the finish point we will cross the 360-0 boundary on the circle, producing a nonsense figure — adding 180 rectifies this.

Lines 3130-3150: The figures needed for defining a point in relation to the radius. The reason that the SIN and COS values are stored in the variables TX and TY on this occasion is that they are going to be used more than once in a line and there would not be room to spell them out in full.

Line 3160: You will recognise the use of a logical condition here. Its effect is to make DX either zero or 159 according to whether the point defined by TX and TY is to the left or the right of the centre of the circle.

Lines 3170-3180: A line is drawn from the circumference of the circle to the edge of the screen on the left or the right as defined by DX. At the end of the line, or rather over it, is printed the label for the segment to which the line points. The print position for labels on the right hand side is moved to the left so that they do not run off the end of the screen, again using a logical condition.

Testing

Add the following line and RUN the program:

1070 GOSUB 3000

You should see a display like the one at the beginning of the section for this program, except that it will be in colour rather than plain black and white.

Module 2.2.5 is the control module. ▶

Module 2.2.4: Lines 3000–3200

```
3000 REM*******************************
3010 REM INSERT SEGMENTS
3020 REM*******************************
3030 FOR I=0 TO ITEMS-1
3040 R=A(I)/180*π
3050 DRAW ,80,100 TO 80+40*COS(R),100+80*SIN(
R)
3060 NEXT I
3070 FOR I=0 TO ITEMS-1
3080 R=(A(I)+4)/180*π
3090 CC=I-3*INT(I/3)+1  : IF I=ITEMS-1 THEN C
C=0
3100 PRINT CC,80+32*COS(R),100+72*SIN(R),1
3110 TA=(A(I)+A(I+1+ITEMS*(I=ITEMS)))/2
3120 IF A(I+1)<A(I) THEN TA=TA+180
3130 R=TA/180*π
3140 TX=80+40*COS(R)
3150 TY=100+80*SIN(R)
3160 DX=ABS(159*(TX>80))
3170 DRAW ,TX,TY TO DX,TY
3180 CHAR 1,DX/4+(LEN(NA$(I))-1)*(DX=159),TY/
8,NA$(I)
3190 NEXT I
3200 RETURN
```

◀ Most of the lines for this have already been entered, but make sure that you have all the lines listed in the module or you will miss one or two refinements.

Module 2.2.5: Lines 1000–1130

```
1000 REM********************************
1010 REM CONTROL
1020 REM********************************
1030 TRAP 1120
1040 GOSUB 5000
1050 TRAP 1090
1060 GOSUB 2000
1070 GOSUB 3000
1080 GETKEY A$
1090 GRAPHIC 0
1100 SCNCLR : LIST 4000-4999
1110 END
1120 PRINT "PROBABLE▽INVALID▽DATA▽FORMAT"
1130 FOR I=1 TO 3000 : NEXT : RESUME 1100
```

Commentary

Lines 1030 and 1120-1130: A brief error message to indicate that there is probably an error in the layout of the DATA module — that is not infallible as a test, since some mistakes will not generate errors detectable by TRAP.

Testing

Change one of the figures under the QUANTITIES heading to a letter, then RUN the program. You should see the program error message and then the DATA module listed out. Correct the deliberate mistake and RUN the program again. This time, pressing RUN/STOP (or any key once the chart is drawn) should list out the DATA module.

Having looked at two different means of presenting data in high resolution mode, it is wise to remind ourselves of the immense flexibility which Commodore's excellent low resolution graphics set provides. Using the low resolution gaphics characters not only provides the user with ready made effects which it would be extremely difficult to provide in high resolution, it allows the full range of colours to be employed and, not least, frees the memory which would otherwise be tied up in the high resolution screen.

In the final program we shall create a three dimensional bar chart whose display is, I think, one of the best demonstrations of just how impressive the C16's low resolution can be — in fact the kind of display that will have you calling in the family to show them just how clever you really are.

Module 2.3.1 looks after the initialisation. It is a straightforward module to declare a small number of variables and arrays.

Commentary

Line 2050: The array HH will be used to store the data for the graph. Since the figures in the DIM statement must be counted from zero, what is provided here is space for three sets of seven data items.

Line 2060: The string R$ will be used in the data file module and explanation can wait until that module.

Line 2070: This string is made up of three colour control codes, ie characters which, if printed, change the print colour. The three colours, in order, represent yellow, green and purple. Later on, the string will be used to provide flexible changes from one colour to another.

Line 2080: This INPUT is designed to allow a later module to call up a set of data for a graph from disk (or tape).

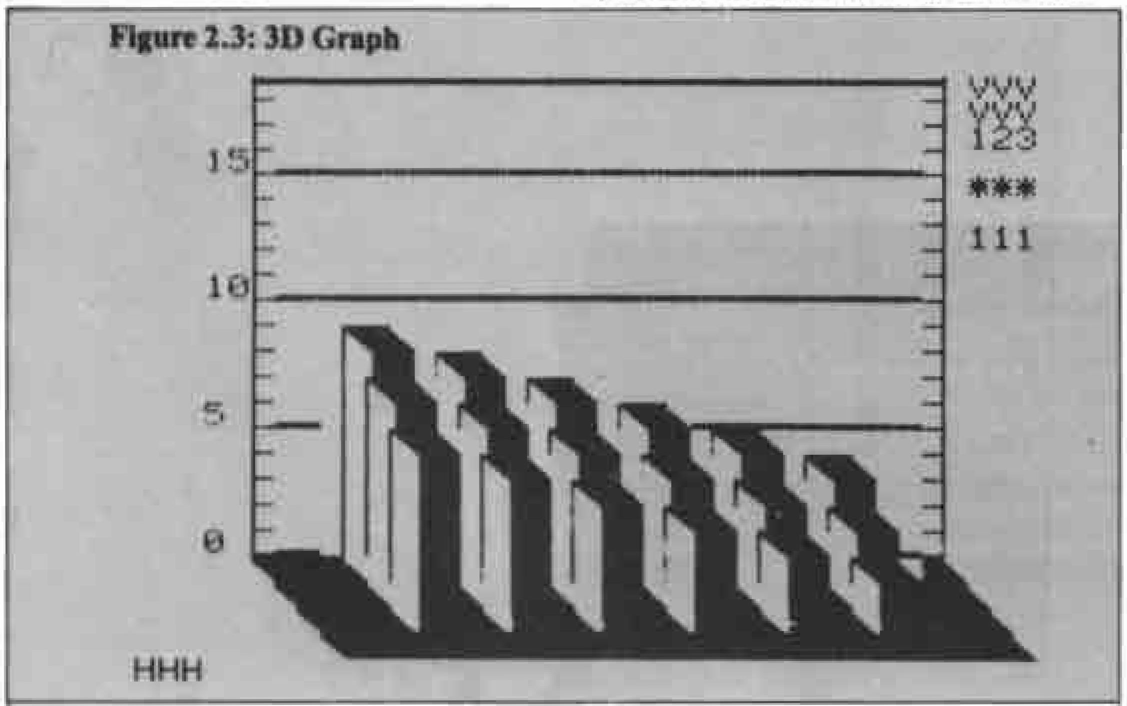
Module 2.3.2 (which will be printed next month) accepts the data. Since we are working in low resolution, there is no need for us to adopt the ruse of using DATA statements to store changing information. Most programs which work on useful information will make provision for the user to input such information while the program is running — such programs are known as interactive. In the case of the present program, all the information can be gathered at one time, so what we see is a module which requests information, uses that information to ask for more and performs checks for mistakes in entry.

Commentary

Lines 3050-3060: As with the line graph program, earlier, each single unit on the vertical axis can represent any value specified by the user. Note that because we are working in low resolution we do not have the same flexibility as the previous program in the size of the vertical units. The only practical size for each unit is one character square's height, and the format of the graph will allow 19 units on the vertical axis.

Lines 3070-3080: The graph will allow, as mentioned previously, three sets of six items to be presented. They will be displayed in the form of three lines of up to six solid bars. Each line of bars will be called a bank, and the individual bars will be called columns■

Figure 2.3: 3D Graph



Module 2.3.1: Lines 2000–2090

```
2000 REM********************************
2010 REM INITIALISE
2020 REM********************************
2030 COLOR 0,11 : COLOR 4,10 : SCNCLR
2040 CHAR,16,1,"[RVS ON][RED]GRAPH" : PRINT
2050 DIM HH(2,6)
2060 R$=CHR$(13)
2070 CO$="[YEL][GRN][PUR]"
2080 INPUT "[BLK][CD]LOAD▽FROM▽DISK▽(Y/N):";Q
$
2090 RETURN
```

This article is based upon an extract from *The Working C16* by David Lawrence. It is published by Sunshine Books and costs £6.95.

Graphs and the C16

***David Lawrence** continues looking at programming Commodore's new micro*

Lines 3110-3130: Since there can be up to three banks, there need to be up to three names for the horizintal axis, eg sale price, cost, profit. Note that the names have to be put into the array NV$ (*N*ame of *V*ertical) from element zero. This array is not dimensioned in the initialisation routine because it will only ever have three elements. Simply mentioning the array name in the program, automatically sets up the array with 10 elements (0-9).

Lines 3150-3230: The user is requested to input, for each bank, the data for the number of columns specified. The number input is checked to ensure that it will not move the graph above the 19 unit mark. The purpose of the loop at 3165-3210 is to ensure that no figure is entered which would require a graph taller than 19 character squares on the screen.

Testing

Enter the following lines (eventually some will become part of the control module for the program):

1030 GOSUB 2000
1040 GOSUB 3000
1090 SCNCLR:END

Now RUN the program you have entered so far and answer its questions as follows:

For the number represented by each vertical unit: 1
Columns: 1
Banks: 1
Name for horizontal axis: HORIZONTAL
Name for vertical axis 1: VERTICAL 1
Bank 1, value 1: 10

After the entry of the last figure, the screen will clear and the program stop with the message READY. Now enter:

?,UV,ND,NB,NH$,NV$(0),HH(0,1)

and the result should be:

1 1 1 HORIZONTAL VERTICAL 1 10

You can, if you wish, RUN the program again and try entering false values. You should not be able to enter any column value which is more than 19 times the value of a single unit on the vertical axis.

Like the line graph, Module 2.3.3 needs a framework to be drawn before the information it presents becomes meaningful. The task is accomplished by the present module, which uses loop variables and the CHAR command to position low resolution characters at the correct positions on the screen.

Commentary

Lines 4050-4070: This loop prints a line of 30 inverse brown spaces to provide the base on which the graph will stand. The graphic character is obtained on the * key.

Line 4080: A line across the top of the screen, consisting of the graphic character on the P key 30 times.

Lines 4100-4110: The vertical lines at each end of the graph, marked for each unit. Graphic characters are on L and P keys.

Lines 4130-4150: Four lines across the graph representing units of five on the vertical axis. **Line 4160:** The horizontal axis label.

Lines 4170-4220: These loops print the name(s) for the vertical axis down the right hand side of the screen, together with the value which each unit represents. Note the use of MID$ to extract a single character from CO$ and print it, thus changing the colour of what is printed. The colour of each title will correspond to the colour of one of the banks. Using CO$ in this way allows us to specify any sequence of colours easily — simply alter the control characters in the string.

Line 4230: This line marks the left hand vertical axis using only cursor control characters rather than a loop and CHAR.

Testing

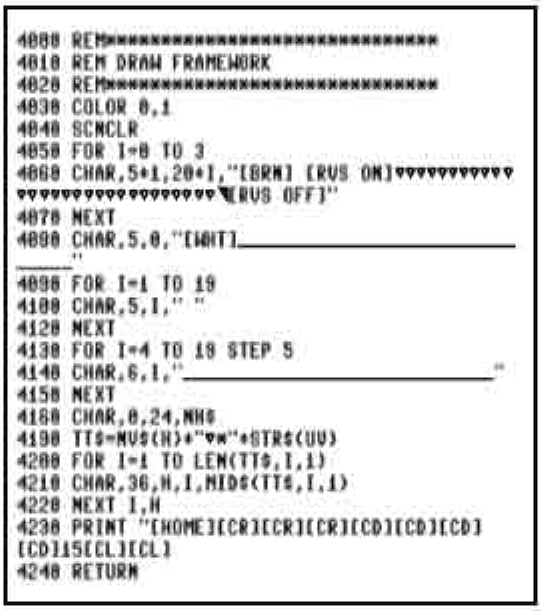
Enter one more line:

1050 GOSUB 4000

and now RUN the program. Specify the unit value, one column and three banks. Give the three names for the vertical axis. When asked to specify the value of the columns, you need only press RETURN. You should then see the graph framework displayed, with the horizontal axis name at the bottom and the vertical axis names printed at the top right.

Module 2.3.4 draws the graph. This is a module which is, frankly, easier to understand once it has been entered and you can see its effect. The module is based not so much on broad principles but on the results of experimentation to see which combinations of characters, printed in which positions, provided the desired effect.

```
3000 REM*******************************
3010 REM ACCEPT DATA
3020 REM*******************************
3030 SCNCLR
3040 CHAR,16,1,"[RVS ON][RED]GRAPH" : PRINT
3050 PRINT "[CD][BLK]THERE▽ARE▽19▽UNITS▽VERTI
CALLY."
3060 INPUT "[CD][BLU]NUMBER▽REPRESENTED▽BY▽EA
CH▽UNIT:";UV
3070 INPUT "[CD]COLUMNS▽(1-6):";ND
3080 INPUT "[CD]BANKS▽(1-3):";NB
3090 PRINT "[CD]▽*****************************
***********▽"
3100 INPUT "[CD][RED]NAME▽FOR▽HORIZONTAL▽AXIS
";NH$
3110 FOR I=0 TO NB-1
3120 PRINT "[GRN]NAME▽FOR▽VERTICAL▽AXIS";I+1;
:INPUT NV$(I)
3130 NEXT I
3140 SCNCLR
3150 FOR I=0 TO NB-1
3160 FOR J=1 TO ND
3165 T=20*UV : DO UNTIL T/UV<=19
3170 PRINT "[CD]INPUT▽BANK";I+1;"▽VALUE";J;":
";
3180 INPUT T
3200 IF INT(T/UV)>19 THEN PRINT "[CD]VALUETOO
▽HIGH."
3210 LOOP
3220 HH(I,J)=T
3230 NEXT J,I
3240 RETURN
```



```
4000 REM********************************
4010 REM DRAW FRAMEWORK
4020 REM********************************
4030 COLOR 0,1
4040 SCNCLR
4050 FOR I=0 TO 3
4060 CHAR,5+I,20+I,"[BRN] [RVS ON]♥♥♥♥♥♥♥♥♥♥♥♥
♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥♥◥[RVS OFF]"
4070 NEXT
4080 CHAR,5,0,"[WHT]________________________________
________"
4090 FOR I=1 TO 19
4100 CHAR,5,I," "
4120 NEXT
4130 FOR I=4 TO 19 STEP 5
4140 CHAR,6,I,"______________________________"
4150 NEXT
4160 CHAR,0,24,NH$
4190 TT$=NV$(H)+"♥↑"+STR$(UU)
4200 FOR I=1 TO LEN(TT$,I,1)
4210 CHAR,36,H,I,MID$(TT$,I,1)
4220 NEXT I,H
4230 PRINT "[HOME][CR][CR][CR][CD][CD][CD]
[CD]15[CL][CL]
4240 RETURN
```

Commentary

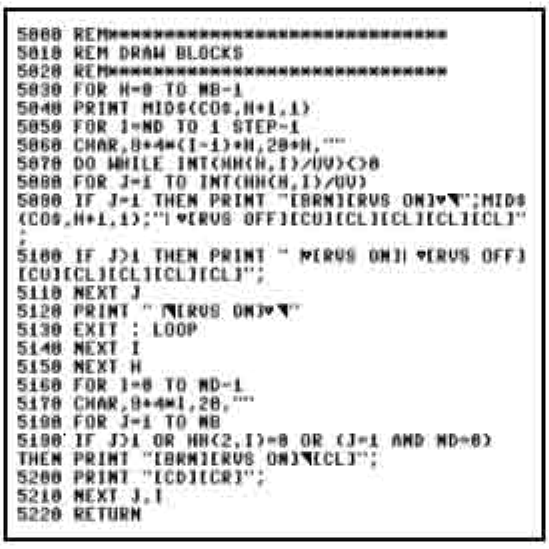
Lines 5030 and **5150:** This loop creates the number of banks specified. **Lines 5050** and **5140:** The second loop will create the number of columns specified.

Line 5060: The start position for drawing each separate column. The position will start at the far right of the number of columns and move four spaces to the left for each new column. In addition, when one bank is finished, the next bank of columns will be printed one space down and to the right of the last.

Lines 5070 and **5130:** These lines create a dummy loop to default around the section which draws a column if the value for that column is zero — the loop is a dummy one because it is never executed more than once in any circumstances.

Lines 5080-5110: The loop which draws the main part of the 3-D column. The loop variable is set to move from one to the height of the column. On the first pass through the loop, the base of the column is created — a character from CO$ is used to move from the brown of the base to the colour of the current bank. On subsequent passes, different characters are used to represent the side and front of the column as it is built up.

Note that the upward movement of the printing is created by the use of cursor control characters rather than the use of CHAR to position the printing. To use CHAR would mean that we would always have to know where we were on the screen.



```
5000 REM********************************
5010 REM DRAW BLOCKS
5020 REM********************************
5030 FOR H=0 TO NB-1
5040 PRINT MID$(CO$,H+1,1)
5050 FOR I=ND TO 1 STEP-1
5060 CHAR,8+4*(I-1)+H,20+H,""
5070 DO WHILE INT(HH(H,I)/UV)<>0
5080 FOR J=1 TO INT(HH(H,I)/UV)
5090 IF J=1 THEN PRINT "[BRN][RVS ON]♥◥";MID$
(CO$,H+1,1);"| ♥[RVS OFF][CU][CL][CL][CL][CL]"
;
5100 IF J>1 THEN PRINT " ♥[RVS ON]| ♥[RVS OFF]
[CU][CL][CL][CL][CL]";
5110 NEXT J
5120 PRINT " ◥[RVS ON]♥◥"
5130 EXIT : LOOP
5140 NEXT I
5150 NEXT H
5160 FOR I=0 TO ND-1
5170 CHAR,8+4*I,20,""
5180 FOR J=1 TO NB
5190 IF J>1 OR HH(2,I)=0 OR (J=1 AND ND=0)
THEN PRINT "[BRN][RVS ON]◥[CL]";
5200 PRINT "[CD][CR]";
5210 NEXT J,I
5220 RETURN
```

Using the cursor controls we only have to know where the column starts, then move relative to that until the top of the column is reached.

Line 5120: The top of the column is added. **Lines 5160-5190:** When the columns are finished there are some untidy edges left at the bottom, these lines neaten them up.

Testing

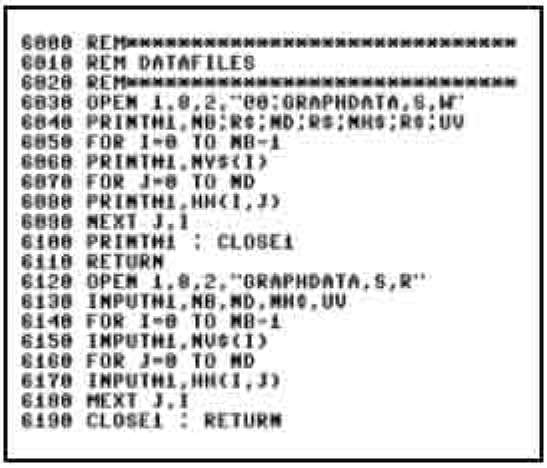
Enter a new line:

1060 GOSUB 5000

and RUN the program. Specify a unit value of one, three columns and three banks. The names for the axes are unimportant, so make your own choice. When prompted for the values of the columns, enter the following:

6,12,18,6,12,18,6,12,18

You should now see the three banks and three columns clearly displayed, with the tops of each of the three columns appearing to make a smooth surface from the rear bank to the front. Note that in reading the values for the three banks, you must assume that the top of the front-most bar continues back and up to the rear-most position. In the example on the screen, what you have is three columns, with the three bars in each column representing the same value, though the front bar is physically lowest on the screen. This is necessary to preserve the illusion of three-dimensionality.



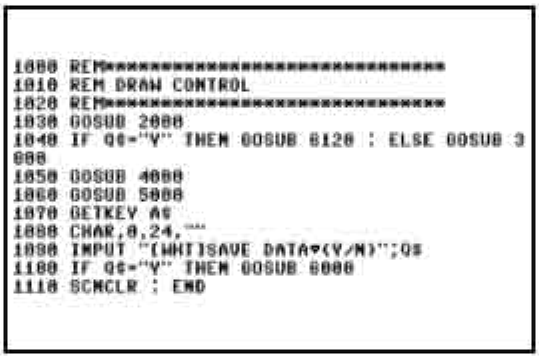
```
6000 REM********************************
6010 REM DATAFILES
6020 REM********************************
6030 OPEN 1,8,2,"@0:GRAPHDATA,S,W"
6040 PRINT#1,NB;R$;ND;R$;NH$;R$;UV
6050 FOR I=0 TO NB-1
6060 PRINT#1,NV$(I)
6070 FOR J=0 TO ND
6080 PRINT#1,HH(I,J)
6090 NEXT J,I
6100 PRINT#1 : CLOSE1
6110 RETURN
6120 OPEN 1,8,2,"GRAPHDATA,S,R"
6130 INPUT#1,NB,ND,NH$,UV
6140 FOR I=0 TO NB-1
6150 INPUT#1,NV$(I)
6160 FOR J=0 TO ND
6170 INPUT#1,HH(I,J)
6180 NEXT J,I
6190 CLOSE1 : RETURN
```

Experiment with the program to see how it deals with differing data values. You will find that it will only really work for data where a bank is never higher than the one behind it.

Module 2.3.5 stores data on disk or tape.

Commentary

Line 6040: One thing to note about the PRINT # statement is the presence of a number of R$s in the line. You may remember that in the first module of the program, R$ was set equal to CHR$(13), which is the RETURN character which signifies the end of an item to be printed. When printing several items to a file from a single PRINT # statement, unless R$ is included in between the items being printed, all the items will be run together. The actual variables printed to the file are simply some of the important values which were input in the second module of the program.



```
1000 REM********************************
1010 REM DRAW CONTROL
1020 REM********************************
1030 GOSUB 2000
1040 IF Q$="Y" THEN GOSUB 6120 : ELSE GOSUB 3
000
1050 GOSUB 4000
1060 GOSUB 5000
1070 GETKEY A$
1080 CHAR,0,24,""
1090 INPUT "[WHT]SAVE DATA♥(Y/N)";Q$
1100 IF Q$="Y" THEN GOSUB 6000
1110 SCNCLR : END
```

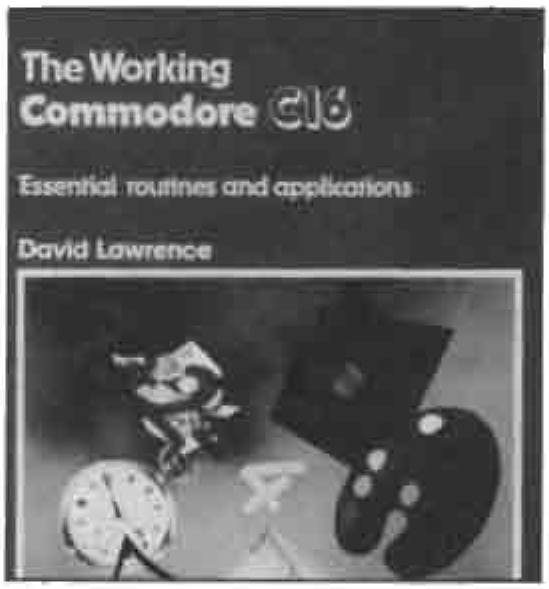
Line 6050: In the commentary on the last line it was stated that R$ (or some other variable equal to CHR$(13)) had to be included to separate the items. So why is it not done in the case of these two loops, which print the contents of two arrays to the file on disk? The answer is that whenever a PRINT or PRINT # statement ends without punctuation, the C16 automatically follows the last item printed with a RETURN character — that is why items are printed on separate lines of the screen if the preceding item does not have a comma or semi-colon on the end.

Lines 6080-6110: Note that we do not have to use the R$ separator when INPUTting data. It is in the nature of INPUT and INPUT # that they do not recognise that they have received an item of data until either RETURN is pressed or a RETURN character read from the disk.

Module 2.3.6 is the control module. Again, you have already entered many of the lines of this module in building up the test procedures on the program. All that remains is to ensure that the module is completed by checking against the listing.

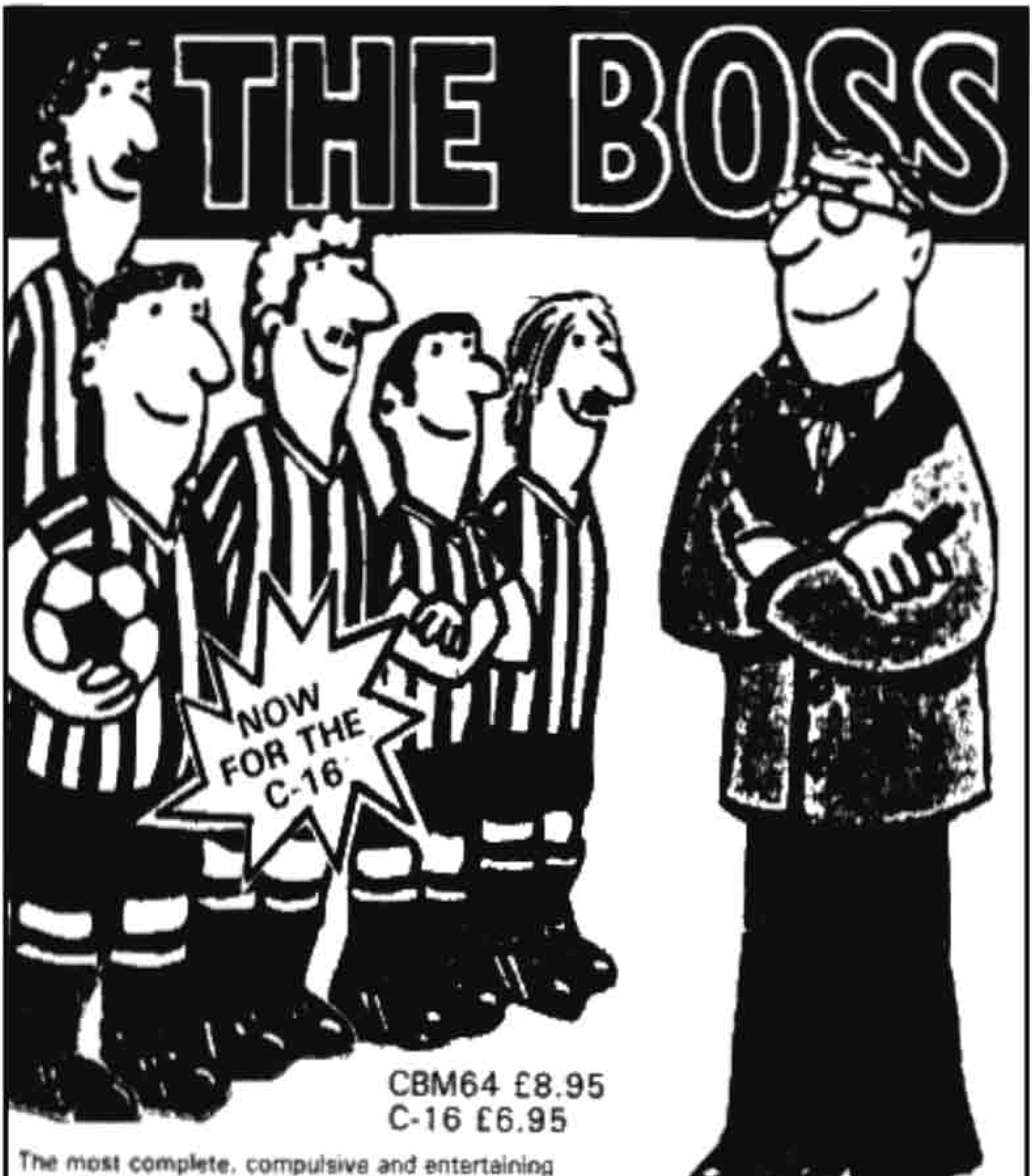
Testing

Simply RUN the program. You should now be able to input the data for a graph. When the graph has been displayed, pressing any key will result in the prompt for saving the data. If you respond Y, then RUN the program again and answer Y when asked if you wish to load from disk. You should now see the same graph displayed. ■



This article is based upon an extract from *The Working C16* by David Lawrence. It is published by Sunshine Books and costs £6.95.

THE BOSS



CBM64 £8.95
C-16 £6.95

The most complete, compulsive and entertaining football management game ever!
● 4 divisions ● FA Cup ● European Cup ● European Cup Winners' Cup ● Promotion/Relegation ● Transfers ● Suspensions ● Injuries ● Substitutes ● Name team ● Team selection ● Choose/change team formation ● Weekly league tables ● Reserve squad

Also on disk — The Boss (CBM 64) £9.95, Tim Love's Cricket (CBM 64) £9.95, The Boss + Tim Love's Cricket (CBM 64) £14.95, The Boss (C16) £9.95

VISA

PEAKSOFT

Check with your retailer.
phone your Access/Visa number or just note your name, address, machine and game requried on back of cheque or P.O.
48 QUEEN STREET, BALDERTON,
NEWARK, NOTTS.
Tel. 0636 705230 (24 hours answering)

Plus/4 adding up?

Although one major selling-point of the Plus/4 is its ROM software, most owners want to supplement the 3+1 suite with programs on disk or tape. ***John Cochrane*** *looks at some of the few products available*

SOFTWARE of any sort for the Commodore C16 and Plus/4 machines is thin on the ground. Utility software is almost non-existent, which makes the recent crop of programs for these machines particularly pleasing. The value for money offered by the C16, and even more so by the Plus/4 if you are prepared to shop around a little, is sadly let down by a lack of commercial software, so any new software is very welcome.

Accounts

Company Pac 1.2.3 by Impex Software is for the Plus/4 only, and a disc drive and printer are also required. The package is intended for keeping accounts, stock control, and the printing of invoices and credit notes. Commodore think highly enough of Company Pac 1.2.3 to set up a special all-in-one deal around it consisting of a Plus/4, a 1541 disc drive, an MPS-801 printer, and the Company Pac 1.2.3 software for £449. Alternatively Company Pac 1.2.3 is available by itself for £48.95. The printer is a very important part of the system as all reports are sent to it whilst using the program.

The program is intended primarily for a small trading business, one which buys and sells manufactured goods rather than a manufacturer or a supplier of services. Thus shopkeepers, for example, are more likely to be able to make good use of the package than say estate agents. The main record keeping system is designed around the buying and selling of goods.

Company Pac 1.2.3 records the purchases and sales made by a company in the form of a sales ledger, a purchase ledger, and a nominal ledger. All of these require data in the style of an invoice, although a real invoice need not necessarily exist. The sales ledger records data concerning stock which has been sold, the purchase ledger records data about stock which has been bought. The nominal ledger combines data from the other two, along with business expenses other than stock purchase, and uses this data to calculate profit and loss sums and to set up a series of company accounts. A separate disc is required for each of these ledgers, the program reminds you which disc to use when.

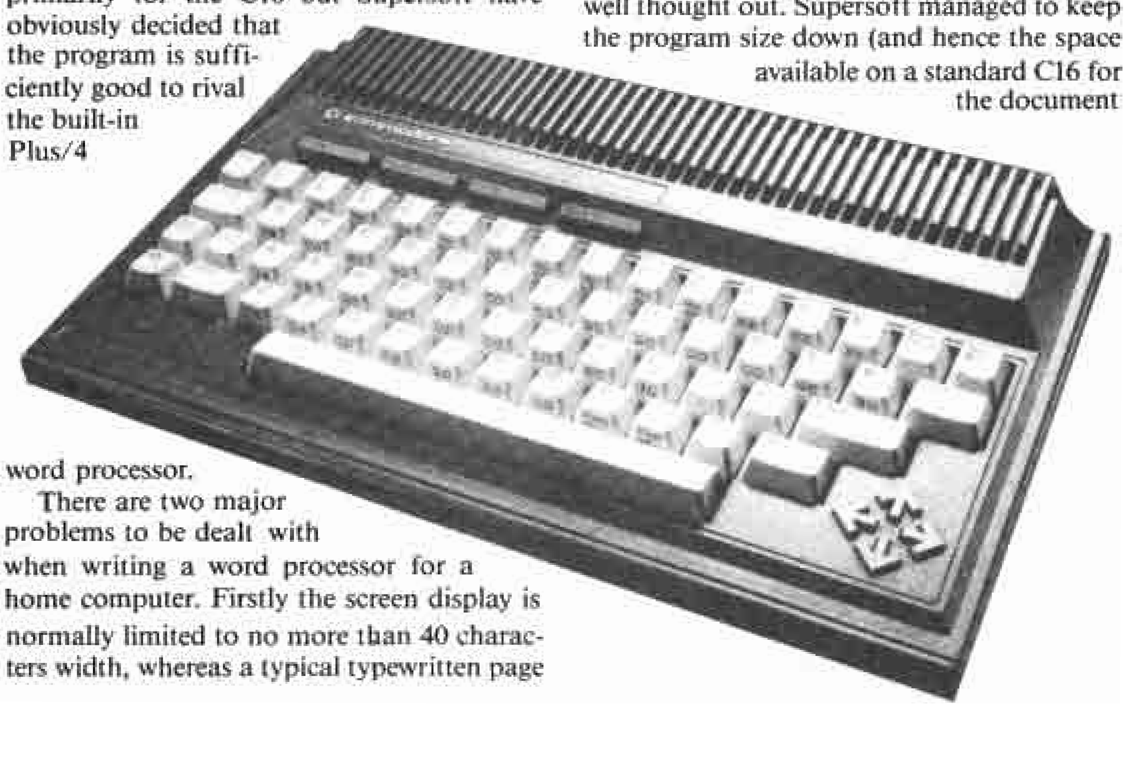
Stock control, or more correctly stock recording, uses a standard form of reach item of stock which notes not only the current stock level but also details of the normal supplier, cost price, selling price, and the current location of the stock. When items are bought or sold these forms are updated to keep a track of current stock levels.

Impex has done a very good job of producing a package which works within the limitations of the computer hardware yet which remains effective and simple to use. The documentation is organised as a tutorial, guiding the user through the program step by step giving tips for getting the best out of each of the facilities in turn. Full-blown accounting and stock control programs developed to run on the more expensive business micros often try to be all things to all men, they end up being powerful but rather cumbersome to use. Company Pac 1.2.3 has necessary restrictions on aspects such as the numbers which can be used for invoices and the way in which data must be presented to the system, but these restrictions are refreshingly explained in full rather than hidden in the usual mound of cross-talk.

Wordprocessing

The software is well thought out and well written. Screen displays are clear and uncrowded. It does not take long before you are using the package confidently and without needing to keep the manual by your side.

Word Perfect is a word processor produced for both the C16 and the Plus/4 by Supersoft. In truth Word Perfect has been developed primarily for the C16 but Supersoft have obviously decided that the program is sufficiently good to rival the built-in Plus/4 word processor.



There are two major problems to be dealt with when writing a word processor for a home computer. Firstly the screen display is normally limited to no more than 40 characters width, whereas a typical typewritten page will be 60 to 70 characters wide. This makes it very difficult to show the document which is being processed in a meaningful way. The second problem is the lack of available memory. Text uses up memory at an alarming rate and this means that the program itself must be kept as small as possible. You either end up with a word processor which is almost useless because it is too simple, or one which is almost useless because it will only process a very small document. There are of course exceptions to this trend, in the form of well thought out and useful processors, but these tend to be rare beasts.

Document

To be honest I must say that I am no more than lukewarm about Word Perfect (Just as I am best lukewarm about the 3-plus-1 word processor of the Plus/4). The program has some limitations which I was disappointed to find in a "modern" word processor. Text seems to be held as single lines of characters (much as is done in 3-plus-1). Although some clever programming allows you to ignore this fact most of the time, some of the text-editing functions work in rather strange ways. The maximum number of lines available is 254, so Plus/4 owners or C16 users with expanded memories will not be able to use the full capacities of their machines.

Word Perfect scores well in two areas: it is available now and it is, though simple, quite well thought out. Supersoft managed to keep the program size down (and hence the space available on a standard C16 for the document up) by making the program "dumb". Thus, for example, if you insert text into the middle of an existing document you will normally have to create blank lines for the new text, enter the text, and then reformat the paragraph concerned to get rid of unwanted spaces. On a more intelligent word processor most of this would be done automatically. Although wordwrap is included, the text cannot be justified to the right to keep it tidy.

Filing

When it comes to actually using Word Perfect you can define the width of the printed document up to 160 characters and the page length as required. You can set tabs, move either margin, centre text, and change the colours used on screen. Blocks of text can be erased, moved, or copied although the blocks must be complete lines of text and some of the processes require rather a lot of keypushing. There is even a simple search command to help you locate a particular piece of text and some vertical-scroll functions which can be used for moving columns of text. When the document has been typed in and set up in the desired format it can be saved to tape or disc, verified, and printed out. The print functions allow for non-Commodore printers and some attempt is made to give features such as text underlining and alternative typefaces, but this is only suitable for some printers.

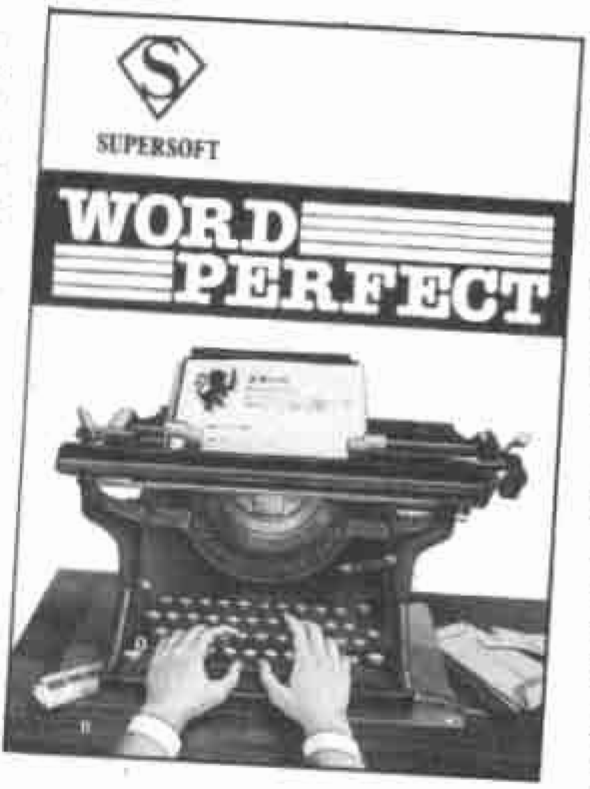
Overall I found Word Perfect quite straightforward to use and fully capable of preparing most simple documents. On the C16 you will be limited to around 2.5 printed pages of A4 Text. On the Plus/4 the available document space increases to around 5 pages of A4 text which is 2.3 times the space provided by the built-in 3-plus-1 word processor can give.

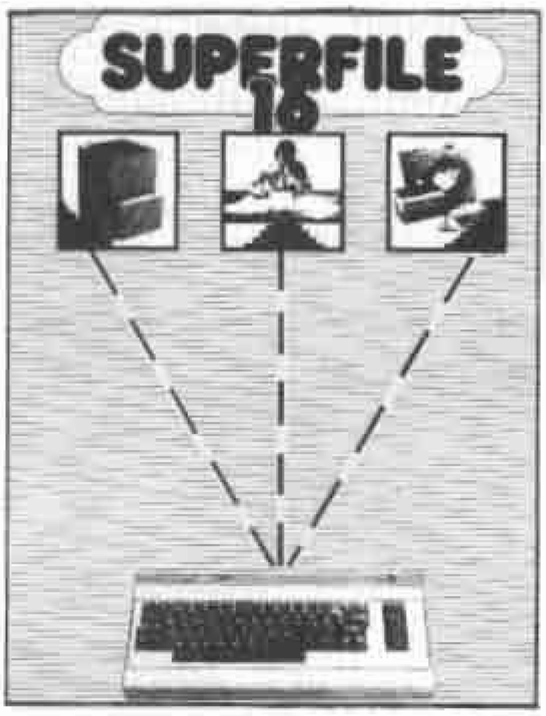
Superfile 16 from Tynesoft is aimed squarely at the C16 using the tape recorder. The program is a simple filing system for keeping records such as names and addresses or details about collections of objects. You can only have one file in memory at a time, and hence you will not be able to cross-reference information between files. What you can do is set up a file which you can sort into ascending order or which you can search for some particular feature.

Menu-driven

Somewhat surprisingly these days the program has been written in Basic, so there is the possibility of modifying the thing to use disc. Also rather surprising is a limit of 30 on the number of records which can be held in memory, memory extensions are no good to you here! Despite the Basic the program is quite simple to use and reasonably fast.

Superfile 16 is menu-driven, that is you have to select options from a series of menus. This makes the program very easy to use. Unfortunately the program, like many others, is let down by its documentation. What there is of the documentation is good, it's clear and to the point. The problem is that the manual is simply incomplete. Nowhere is the use of a printer mentioned, or the setting of screen colours, or the maximum number of records which can be set up. Trial and error indicates that although you can print recorded data but you have to print the lot, and a Commodore printer is assumed.





Music Master affords a little light relief after the rigours of accounts and word processing. The C16 and Plus/4 were criticised at first for having limited sound capabilities when compared with the C64. Much of this criticism was, in my opinion, misguided as few programs and fewer programmers ever use the C64 sound capabilities to the full because of the difficulties of programming the thing. The two voices provided on the C16 and Plus/4 are good enough for most requirements yet simple to use.

Supersoft takes things somewhat further than most with Music Maker, and shows very effectively what can be done. With the software loaded (from tape or disc) you can use the keyboard as a simple piano, making music to your heart's content, or you can use the memory to record and edit your tunes for replaying at a later date. Either of the voices can be used when playing from the keyboard, or both together to produce a richer sound. Voice 2 can additionally be used to provide a background or rhythm while you play on voice 1. Ten built-in rhythms are provided or you can write your own.

The length of tune which can be held in memory is limited to 425 notes per voice. I have not yet found a way of increasing this for use on the Plus/4. How long such a tune will last depends on the tempo selected. Tunes, or disgusting patterns of noise depending on your skill, can be saved to tape or disc. You can also, with the aid of some program listings printed in the manual, transfer tunes developed using Music Master to programs of your own.

I am no musician so I am not able to comment seriously on the technical capabilities of Music Maker, other than to say that you need to have a pretty good imagination to make the thing sound like anything but a cheap electric organ. What I can say is that I found the program easy to use and fun to play. The manual is well written and complete with plenty of helpful comments on making practical use of the program. ■

SOFTWARE: Company Pac 1.2.3
SUPPLIER: Impex Software Ltd., Metro House, Second Way, Middlesex HA9 0TY. Tel: 01-900 0999.
PRICE: £98 Disk.

SOFTWARE: Word Perfect
SUPPLIER: Supersoft, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex. HA3 7SJ. Tel: 01-861 1166.
PRICE: £14.95 Disk, £12.95 Tape.

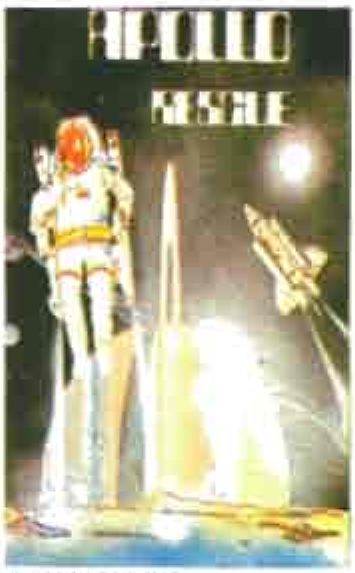
SOFTWARE: Music Master
SUPPLIER: Supersoft, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex. HA3 7SJ. Tel: 01-861 1166.
PRICE: £14.95, £12.95 Tape.

SOFTWARE: Superfile 16
SUPPLIER: Tynesoft, Addison Industrial Estate, Blaydon, Tyne & Wear. NE21 4ZE.
PRICE: £9.95 Tape.

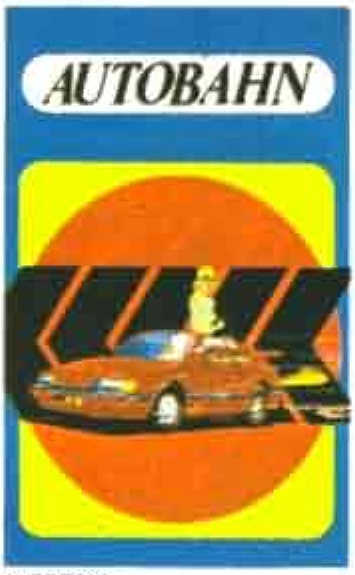
To all C16/Plus 4 and Electron Users

Tynesoft's January big name and value for money releases – watch out for our new titles on your machine every month in 1986.

FOR THE C16/PLUS 4, STARTER PACK 2



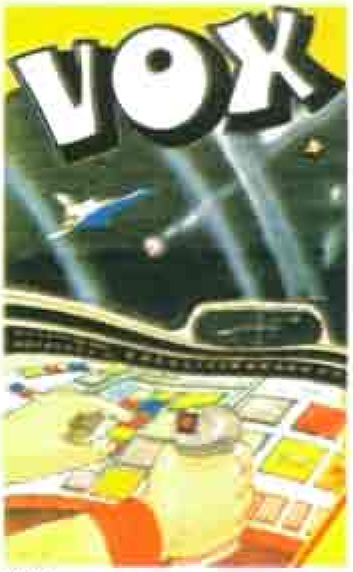
APOLLO RESCUE
Apollo Rescue Mission is a two part game challenging and full of action Your mission is to retrieve and reassemble sections of the ill fated Apollo 16 scattered over the lunar surface. Assembly of rocket must be done in the correct order to enable you to complete your task and blast off to a new universe. Beware of those dangerous aliens, that have to be eliminated with your laser. The second part of the mission is to retrieve Satellites floating in space with the use of a special arm to be returned to earth for servicing



AUTOBAHN
A fast Arcade game where your driving skills are tested to the full. Your brief is to take out a new prototype car on a stretch of the German Autobahn and put it through its paces.
You have to complete 4 challenging tests before your job is finished

WATER GRANDPRIX
You are in the driving seat of a powerful speed boat entered in the Annual Water Grand Prix
Skill is needed to navigate your boat along the twisting river. Avoiding the river banks and passing your fellow competitors along the way to victory

R.R.P. **£14.95**



VOX
You have to defend the 8 Cities of the planet 'VOX' from the aliens invasion force. You must destroy the alien craft by moving a sight over the screen and firing guided missiles to the sight's location. There are 40 levels. Each level consists of a set number of alien craft but an unlimited number of Ammo Saucers and Stunt Planes. To complete a level you must destroy all the aliens apart from Ammo Saucers and Stunt Planes

FORTHCOMING RELEASES:

POGO PETE · MOUNT VESUVIUS · WINTER OLYMPICS · GUNSLINGER · JET SET WILLY

FOR THE BBC/ELECTRON

IAN BOTHAM'S TEST MATCH

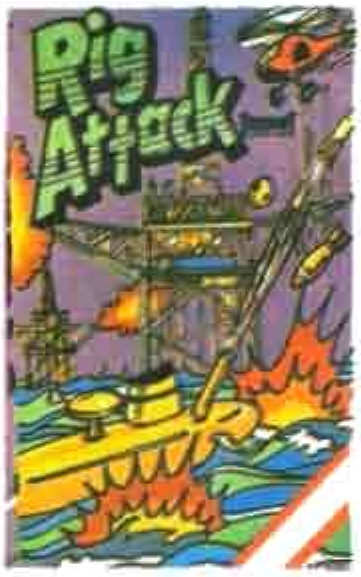
We think the best sports simulation yet on the BBC Electron and special features include
Four Bowling Options. Four Batting Options. One Player Mode. Limited Over game (16 overs). One Day game (32 overs). Test Match (Two full innings). Demonstration Mode

R.R.P. **£7.95**

RIG ATTACK
One of our value for money tapes at a R.R.P. of £4.95. Rig Attack is an action packed game demanding skill and quick reflexes to guide your helicopter from rig to rig without coming under attack from enemy submarines. The submarine is equipped with the latest missiles, your copter carries short range bombs that need to be dropped at low altitude to reach that darting submarine in the waters below

Other leading titles for the Electron in the near future
JET SET WILLY
NINJA CHALLENGE

R.R.P. **£4.95**



TYNESOFT
COMPUTER SOFTWARE

To make sure you find out about all our exciting new releases and register on our mailing list fill in the Registration Form below, making sure to state your machine and send it to:

Tynesoft Computer Software, Addison Industrial Estate, Blaydon, Tyne & Wear NE21 4TE.

Title (Tick) Mr Mrs Miss

Initials

Surname

Address

BBC/Electron

C16/Plus 4

Post code

Telephone no. (including STD code)

COMMODORE SOFTWARE FILE

Flight Master

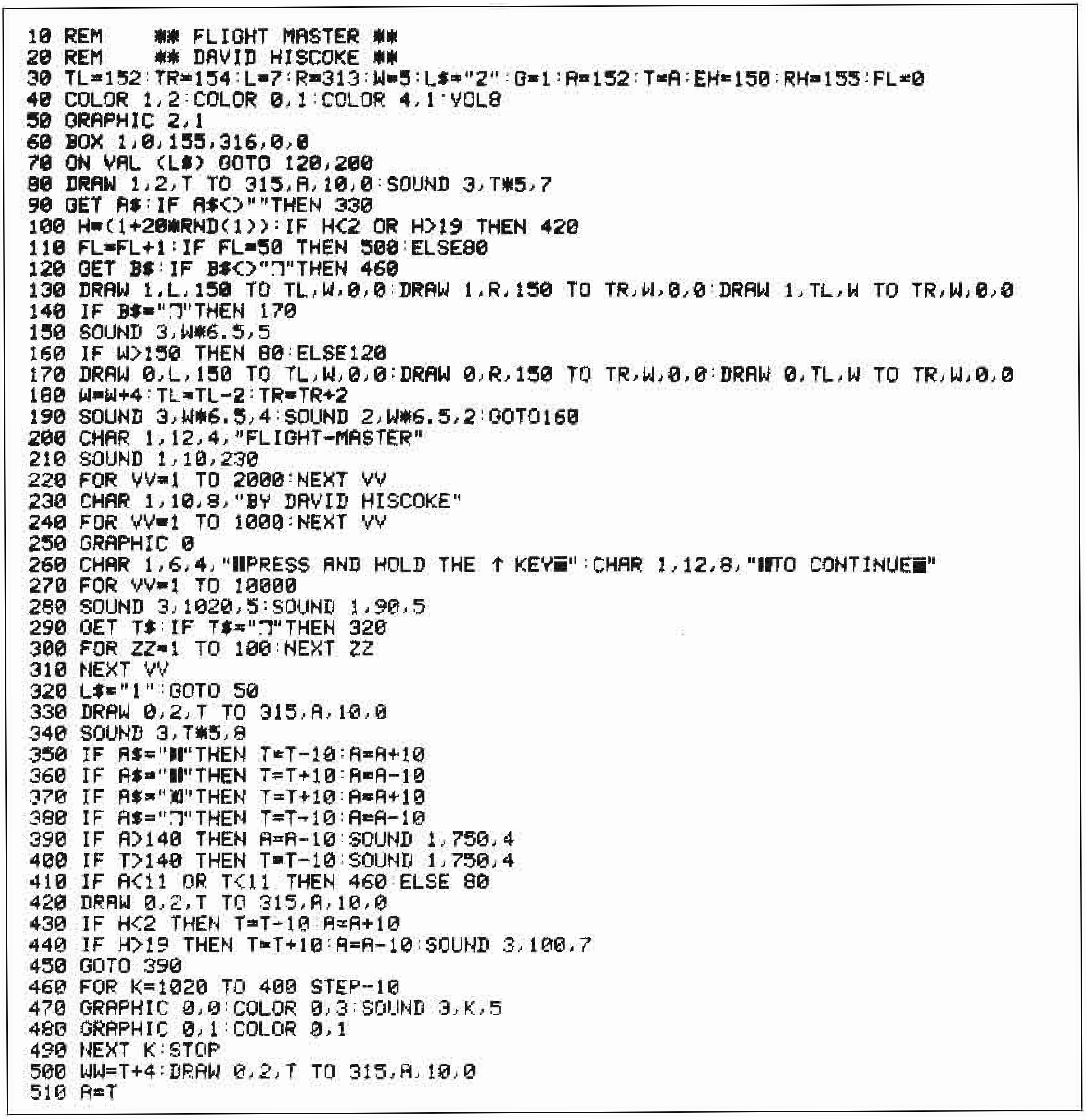
Our first readers program for the C16 comes from David Hiscoke of London

FLIGHT MASTER isn't so much a flight simulator, it's more an interactive movie animation.

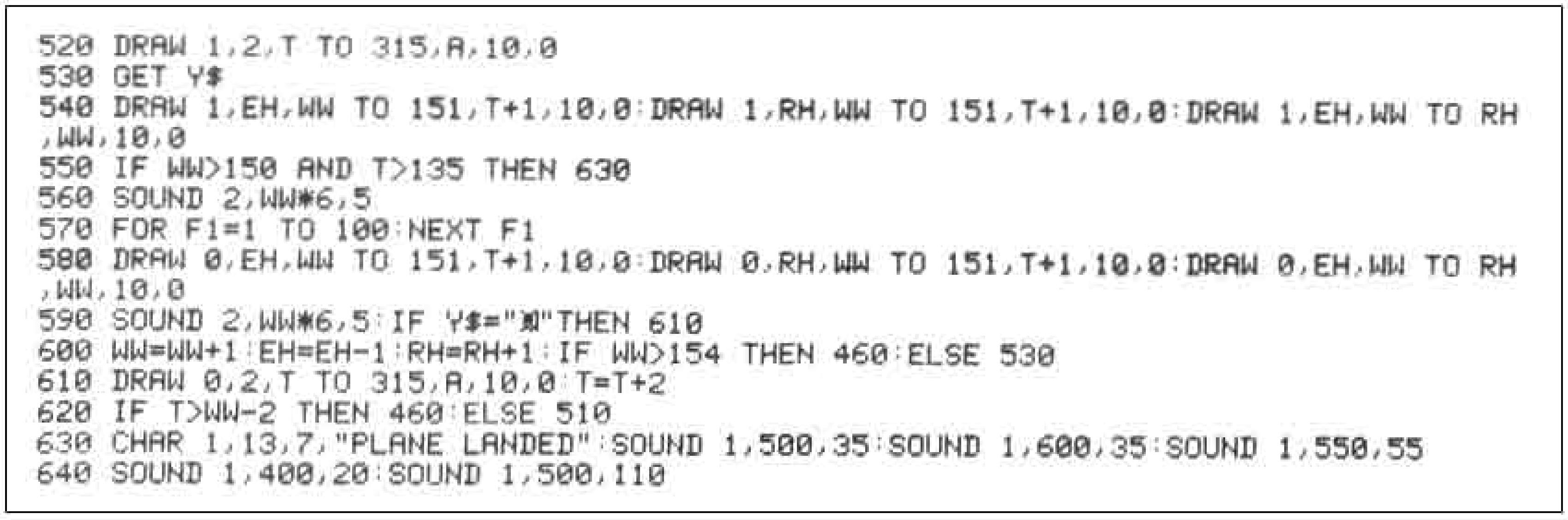
After the title screen you see the runway displayed, and you take off by pressing the up cursor key. If you release it too soon or press the wrong key you will crash.

Once in the air, you must keep the horizon level and in the middle of the screen by using the four cursor keys. If you can keep level for long enough you will see the landing runway, and you must then land using the down cursor key.

The program demonstrates the C16's superior sound and hi-res graphics commands.



```
10 REM    ** FLIGHT MASTER **
20 REM    ** DAVID HISCOKE **
30 TL=152:TR=154:L=7:R=313:W=5:L$="2":G=1:A=152:T=A:EH=150:RH=155:FL=0
40 COLOR 1,2:COLOR 0,1:COLOR 4,1:VOL8
50 GRAPHIC 2,1
60 BOX 1,0,155,316,0,0
70 ON VAL (L$) GOTO 120,200
80 DRAW 1,2,T TO 315,A,10,0:SOUND 3,T*5,7
90 GET A$:IF A$<>""THEN 330
100 H=(1+20*RND(1)):IF H<2 OR H>19 THEN 420
110 FL=FL+1:IF FL=50 THEN 500:ELSE80
120 GET B$:IF B$<>"┐"THEN 460
130 DRAW 1,L,150 TO TL,W,0,0:DRAW 1,R,150 TO TR,W,0,0:DRAW 1,TL,W TO TR,W,0,0
140 IF B$="┐"THEN 170
150 SOUND 3,W*6.5,5
160 IF W>150 THEN 80:ELSE120
170 DRAW 0,L,150 TO TL,W,0,0:DRAW 0,R,150 TO TR,W,0,0:DRAW 0,TL,W TO TR,W,0,0
180 W=W+4:TL=TL-2:TR=TR+2
190 SOUND 3,W*6.5,4:SOUND 2,W*6.5,2:GOTO160
200 CHAR 1,12,4,"FLIGHT-MASTER"
210 SOUND 1,10,230
220 FOR VV=1 TO 2000:NEXT VV
230 CHAR 1,10,8,"BY DAVID HISCOKE"
240 FOR VV=1 TO 1000:NEXT VV
250 GRAPHIC 0
260 CHAR 1,6,4,"▮PRESS AND HOLD THE ↑ KEY▪":CHAR 1,12,8,"▮TO CONTINUE▪"
270 FOR VV=1 TO 10000
280 SOUND 3,1020,5:SOUND 1,90,5
290 GET T$:IF T$="┐"THEN 320
300 FOR ZZ=1 TO 100:NEXT ZZ
310 NEXT VV
320 L$="1":GOTO 50
330 DRAW 0,2,T TO 315,A,10,0
340 SOUND 3,T*5,8
350 IF A$="▮"THEN T=T-10:A=A+10
360 IF A$="▮"THEN T=T+10:A=A-10
370 IF A$="▩"THEN T=T+10:A=A+10
380 IF A$="┐"THEN T=T-10:A=A-10
390 IF A>140 THEN A=A-10:SOUND 1,750,4
400 IF T>140 THEN T=T-10:SOUND 1,750,4
410 IF A<11 OR T<11 THEN 460 ELSE 80
420 DRAW 0,2,T TO 315,A,10,0
430 IF H<2 THEN T=T-10 A=A+10
440 IF H>19 THEN T=T+10:A=A-10:SOUND 3,100,7
450 GOTO 390
460 FOR K=1020 TO 400 STEP-10
470 GRAPHIC 0,0:COLOR 0,3:SOUND 3,K,5
480 GRAPHIC 0,1:COLOR 0,1
490 NEXT K:STOP
500 WW=T+4:DRAW 0,2,T TO 315,A,10,0
510 A=T
```



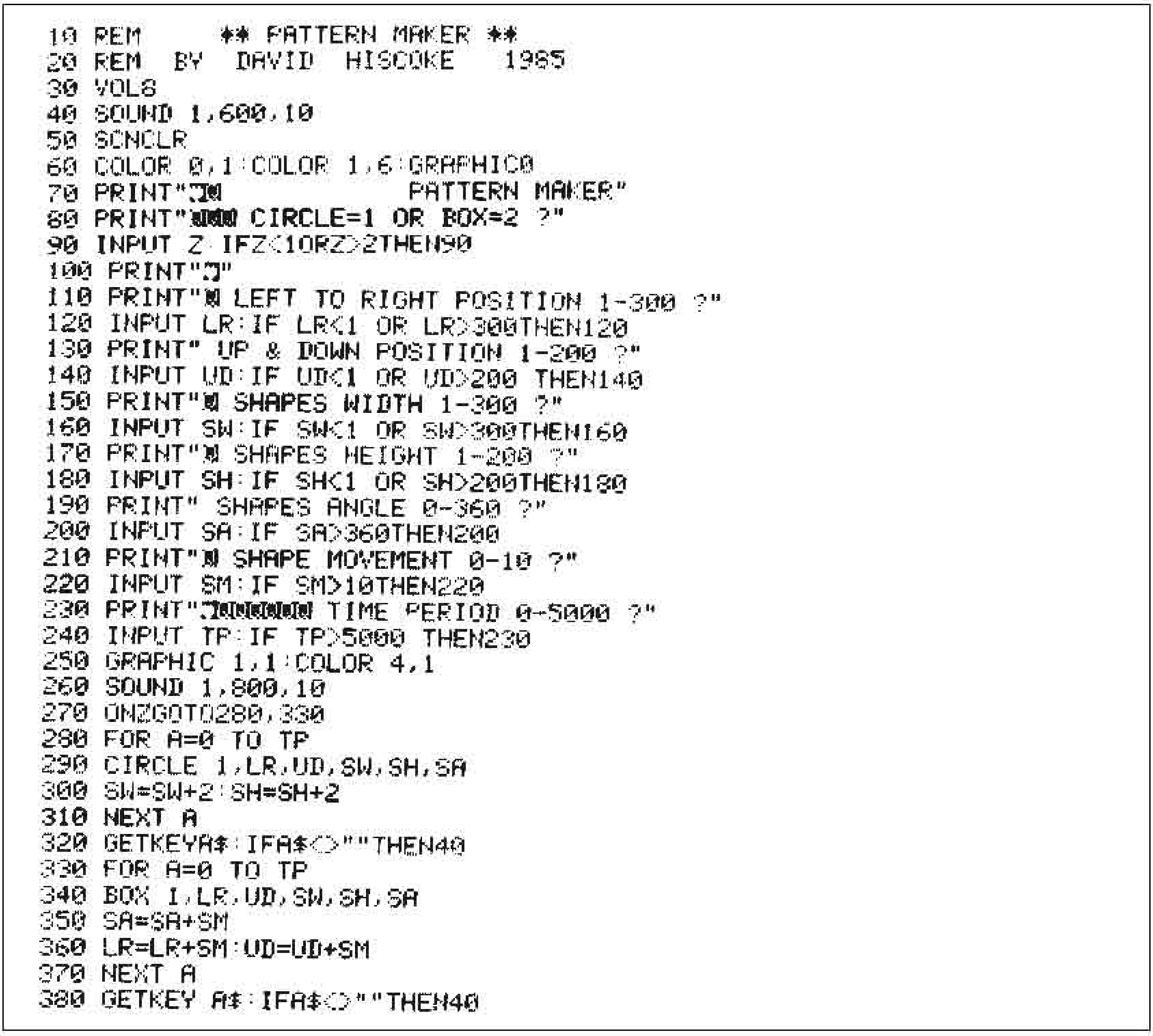
```
520 DRAW 1,2,T TO 315,A,10,0
530 GET Y$
540 DRAW 1,EH,WW TO 151,T+1,10,0:DRAW 1,RH,WW TO 151,T+1,10,0:DRAW 1,EH,WW TO RH
,WW,10,0
550 IF WW>150 AND T>135 THEN 630
560 SOUND 2,WW*6,5
570 FOR F1=1 TO 100:NEXT F1
580 DRAW 0,EH,WW TO 151,T+1,10,0:DRAW 0,RH,WW TO 151,T+1,10,0:DRAW 0,EH,WW TO RH
,WW,10,0
590 SOUND 2,WW*6,5:IF Y$="■"THEN 610
600 WW=WW+1:EH=EH-1:RH=RH+1:IF WW>154 THEN 460:ELSE 530
610 DRAW 0,2,T TO 315,A,10,0:T=T+2
620 IF T>WW-2 THEN 460:ELSE 510
630 CHAR 1,13,7,"PLANE LANDED":SOUND 1,500,35:SOUND 1,600,35:SOUND 1,550,55
640 SOUND 1,400,20:SOUND 1,500,110
```

Patterns

THIS C16 routine comes from David Hiscoke of London

PATTERN MAKER is a hi-res graphics utility. It uses two of the C16's graphics commands, BOX and CIRCLE. On running, enter 1 for CIRCLE or 2 for BOX. Left to right position of the shape is from 1 to 300. 200, Width is from 1 to 300. Height is from 1 to 200. Angle from 0 to 360. Movement is from 1 to 10, and time period from 1 to 5000. On completion, press any key to restart.

As a good example, select options 2,10,10,200,5,0,7,50, to generate a fast-moving display of moving boxes which fills the screen with geometrical patterns.



```
 10 REM      ** PATTERN MAKER **
 20 REM  BY  DAVID  HISCOKE   1985
 30 VOL8
 40 SOUND 1,600,10
 50 SCNCLR
 60 COLOR 0,1:COLOR 1,6:GRAPHIC0
 70 PRINT"■■                PATTERN MAKER"
 80 PRINT"■■■ CIRCLE=1 OR BOX=2 ?"
 90 INPUT Z:IFZ<1ORZ>2THEN90
 100 PRINT"■"
 110 PRINT"■ LEFT TO RIGHT POSITION 1-300 ?"
 120 INPUT LR:IF LR<1 OR LR>300THEN120
 130 PRINT" UP & DOWN POSITION 1-200 ?"
 140 INPUT UD:IF UD<1 OR UD>200 THEN140
 150 PRINT"■ SHAPES WIDTH 1-300 ?"
 160 INPUT SW:IF SW<1 OR SW>300THEN160
 170 PRINT"■ SHAPES HEIGHT 1-200 ?"
 180 INPUT SH:IF SH<1 OR SH>200THEN180
 190 PRINT" SHAPES ANGLE 0-360 ?"
 200 INPUT SA:IF SA>360THEN200
 210 PRINT"■ SHAPE MOVEMENT 0-10 ?"
 220 INPUT SM:IF SM>10THEN220
 230 PRINT"■■■■■■■ TIME PERIOD 0-5000 ?"
 240 INPUT TP:IF TP>5000 THEN230
 250 GRAPHIC 1,1:COLOR 4,1
 260 SOUND 1,800,10
 270 ONZGOTO280,330
 280 FOR A=0 TO TP
 290 CIRCLE 1,LR,UD,SW,SH,SA
 300 SW=SW+2:SH=SH+2
 310 NEXT A
 320 GETKEYA$:IFA$<>""THEN40
 330 FOR A=0 TO TP
 340 BOX 1,LR,UD,SW,SH,SA
 350 SA=SA+SM
 360 LR=LR+SM:UD=UD+SM
 370 NEXT A
 380 GETKEY A$:IFA$<>""THEN40
```

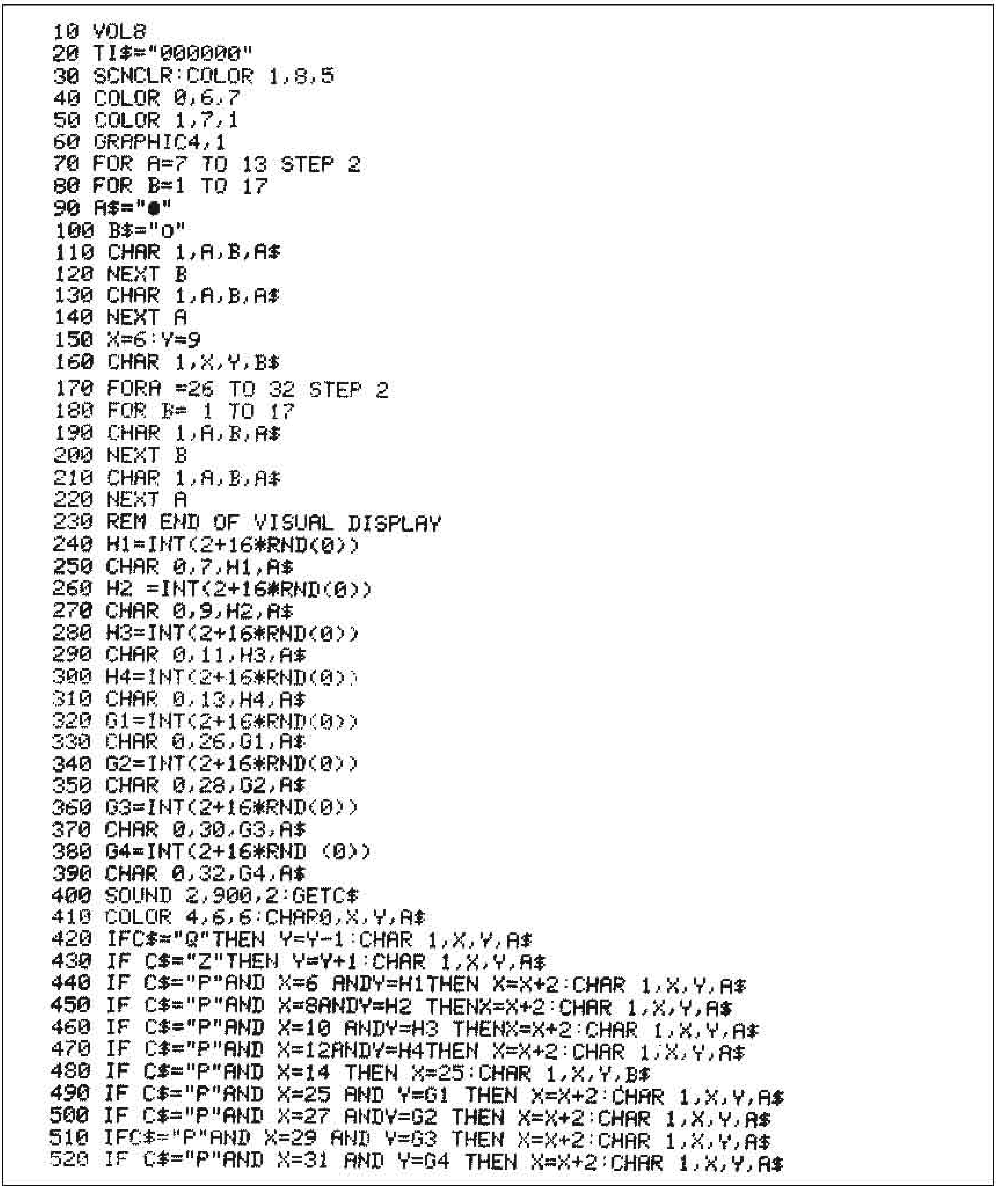
Actionblob

Permjot Valia from London sends in this game for the C16 and Plus/4

THE AIM OF Actionblob is to see how quickly you can move from the left hand side of the screen to the right.

On the way you will encounter lines of enemy blobs who will only allow you through at one point on the line. Using Q=UP, P=RIGHT and Z=DOWN, see if you can beat the clock.

The game uses both the C16's graphics abilities, and simple sound effects, but since there are no POKEs used it's ideal for beginners.



```
10 VOL8
20 TI$="000000"
30 SCNCLR:COLOR 1,8,5
40 COLOR 0,6,7
50 COLOR 1,7,1
60 GRAPHIC4,1
70 FOR A=7 TO 13 STEP 2
80 FOR B=1 TO 17
90 A$="●"
100 B$="O"
110 CHAR 1,A,B,A$
120 NEXT B
130 CHAR 1,A,B,A$
140 NEXT A
150 X=6:Y=9
160 CHAR 1,X,Y,B$
170 FORA =26 TO 32 STEP 2
180 FOR B= 1 TO 17
190 CHAR 1,A,B,A$
200 NEXT B
210 CHAR 1,A,B,A$
220 NEXT A
230 REM END OF VISUAL DISPLAY
240 H1=INT(2+16*RND(0))
250 CHAR 0,7,H1,A$
260 H2 =INT(2+16*RND(0))
270 CHAR 0,9,H2,A$
280 H3=INT(2+16*RND(0))
290 CHAR 0,11,H3,A$
300 H4=INT(2+16*RND(0))
310 CHAR 0,13,H4,A$
320 G1=INT(2+16*RND(0))
330 CHAR 0,26,G1,A$
340 G2=INT(2+16*RND(0))
350 CHAR 0,28,G2,A$
360 G3=INT(2+16*RND(0))
370 CHAR 0,30,G3,A$
380 G4=INT(2+16*RND (0))
390 CHAR 0,32,G4,A$
400 SOUND 2,900,2:GETC$
410 COLOR 4,6,6:CHAR0,X,Y,A$
420 IFC$="Q"THEN Y=Y-1:CHAR 1,X,Y,A$
430 IF C$="Z"THEN Y=Y+1:CHAR 1,X,Y,A$
440 IF C$="P"AND X=6 ANDY=H1THEN X=X+2:CHAR 1,X,Y,A$
450 IF C$="P"AND X=8ANDY=H2 THENX=X+2:CHAR 1,X,Y,A$
460 IF C$="P"AND X=10 ANDY=H3 THENX=X+2:CHAR 1,X,Y,A$
470 IF C$="P"AND X=12ANDY=H4THEN X=X+2:CHAR 1,X,Y,A$
480 IF C$="P"AND X=14 THEN X=25:CHAR 1,X,Y,B$
490 IF C$="P"AND X=25 AND Y=G1 THEN X=X+2:CHAR 1,X,Y,A$
500 IF C$="P"AND X=27 ANDY=G2 THEN X=X+2:CHAR 1,X,Y,A$
510 IFC$="P"AND X=29 AND Y=G3 THEN X=X+2:CHAR 1,X,Y,A$
520 IF C$="P"AND X=31 AND Y=G4 THEN X=X+2:CHAR 1,X,Y,A$
```

```
530 IF TI$="000020"AND X<>33 THEN 560
540 IF TI$="000020"AND X=33 THEN 590
550 IF C$<>" "THEN 400
560 GRAPHICCLR
570 PRINT"SORRY,YOURE OUT OF TIME"
580 END
590 GRAPHICCLR
600 PRINT"     CONGRATULATIONS YOU MADE IT"
610 END
```

Zig-Zag

This game for the C16 comes from D O'Connor of Farnham, Surrey

ZIG-ZAG is a simple game using many useful programming techniques. The player, represented by a star in the top right hand corner of the screen, has to move around the screen filling in the blank spaces. Direction is controlled by keys W,A,Z and X. The = key advances the player a level.

You can stop your movement by hitting the side of a room, but this loses points. You must dodge the blocks which appear throughout the room, or the game ends. Points are even for the number of steps covered before the timer runs out. The screen is drawn using second set graphics; the text in line 270 reads PRESS ANY KEY, and that in line 280 GAME OVER.

```
10 PRINTCHR$(14):PRINT"▓▓▓":X=121:L=L+1:F=0:COLOR0,1:P=79:A=0:VOL8
20 FORT=1TO2:PRINT"▓                                                ▓";:NEXT
30 PRINT"ππππππππππππππππππππππππππππππππππππππ"
40 FORT=1TO20:PRINT"π▓                                              ▓π";:NEXT
50 PRINT"ππππππππππππππππππππππππππππππππππππππ"
60 COLOR1,4:FORT=3TO22:FORQ=1TO2+L:POKE3072+(T*40)+INT(2+37*RND(0)),160:NEXTQ:NE
XTT
70 Y=X:GETA$:IFA$="Z"THENA=-1
80 IFA$="X"THENA=1
90 IFA$="Q"THENA=-40
100 IFA$="A"THENA=40
110 IFA$="="THEN10
120 X=X+A
130 IFPEEK(3072+X)=94THENX=Y:SC=SC-10
140 IFPEEK(3072+X)=160THENSOUND1,500,50:SOUND3,400,50:GOTO280
150 IFPEEK(3072+X)=32THENF=F+1:SC=SC+2:SOUND1,800,5
160 POKE3072+X,42:IFY<>XANDZ=0THENPOKE3072+Y,95
170 IFY<>XANDZ=1THENPOKE3072+Y,105
180 IFZ=1THENZ=0:ELSEZ=1
190 FORT=1TO60-(L*10):NEXT
200 READD:POKE3072+P,D:IFD=32THENP=P-1:RESTORE
210 IFP=-1THENPRINT"▓┌ ╱ ┌▁ │╲╲‾!!!!":GOTO260
220 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓SCORE=";SC"▓"
230 IFF=721-(L*20)THENPRINT"▓ o┐┴┴ ┬╱‾":GOTO10
240 GOTO70
250 DATA231,231,234,234,246,246,97,97,117,117,116,116,101,101,32
260 FORT=1TO1300:GETA$:NEXTT
270 A$="":PRINT"▓┐▁‾♥♥ ♠╱│ ╱‾│":GETC$:IFC$=""THEN270:ELSESC=0:RUN
280 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓SCORE=";SC;"   │♠╲‾ ┌╳‾▁!!!":GOTO260
```

Send us your Commodore programs, enclosing a cassette — and a printout on plain white paper if possible. Each listing must be accompanied by a typewritten introduction describing the program and explaining how it is constructed. We pay £6.00 for each bug-free listing published. We cannot guarantee to return every program submitted, so please keep a copy. If you want us to return your listing you must include a stamped, addressed envelope. If you have any problems with the programs, please write to the appropriate author, Software File, Commodore Horizons, 12-13 Little Newport Street, London WC2R 3LD

Paintbox

£9.95 R.R.P.

THE COMPLETE GRAPHICS SYSTEM FOR PRODUCING FULL COLOUR HIGH RESOLUTION PICTURES!

Paintbox is an advanced graphics creator program, containing all the latest hi-res drawing facilities. With a full selection of brush sizes and colours you can draw direct to the screen, with either joystick or keyboard control, and use the special facilities to create fabulous high-resolution pictures. Paintbox's special facilities include...

- **Fill outlines with a colour**
- **Make frames or boxes**
- **Save pictures to tape or disk**
- **Swap shapes between pictures***
- **Two separate drawing areas***
- **Copy shapes***
- **Draw straight lines**
- **Draw circles**
- **Plot radians**
- **Colour luminance control †**

Now all the facilities of expensive touch-tablet and sketch-pad packages are available in one low cost, high value package - Paintbox!

* 64 version only.

† C16 version only.

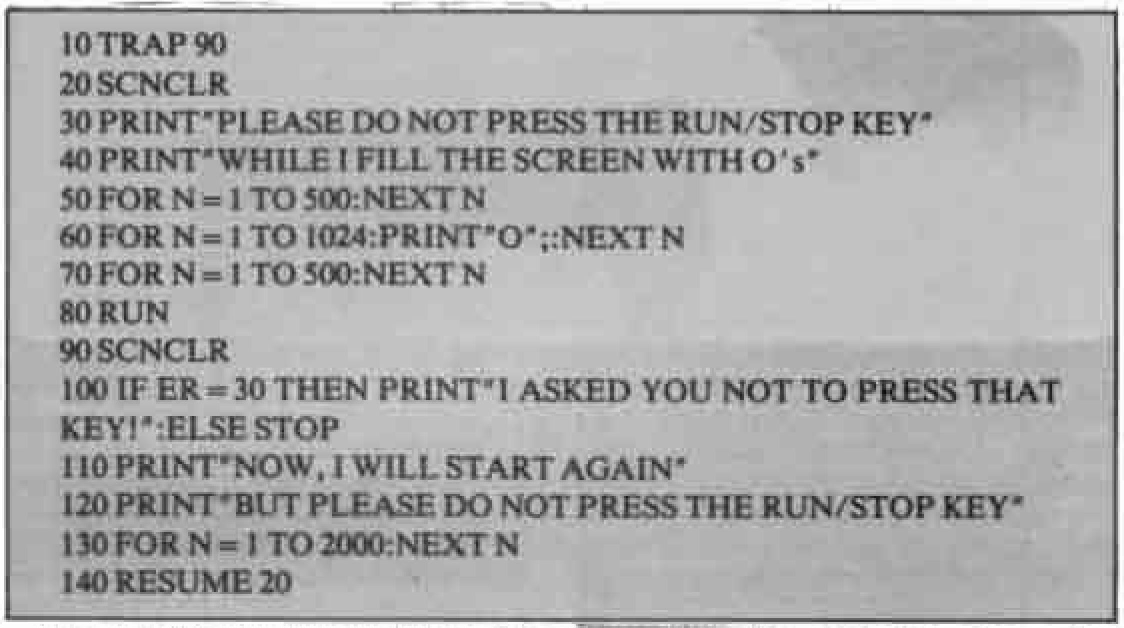
Audiogenic

AUDIOGENIC SOFTWARE LTD. P.O. BOX 88 READING, BERKSHIRE, ENGLAND. 0734 664646

User-friendly programming on the C16 and Plus/4

An extract from ***Brian Lloyd's*** *Commodore 16 — Plus/4 companion*

FOR A COMPUTER program to be really good it must be 'user friendly', which means that whoever uses the program is given full instructions on how to use the program properly, and that whatever that user does, however stupid, the program should carry on working and tell the user what he or she has done wrong. To help you to write such programs your computer has been supplied with error trapping facilities, so that if, for example, you press the RUN/STOP key when you do not want to, then the program will be able to explain what you have done wrong. Figure one is a short program which illustrates this.



```
10 TRAP 90
20 SCNCLR
30 PRINT"PLEASE DO NOT PRESS THE RUN/STOP KEY"
40 PRINT"WHILE I FILL THE SCREEN WITH O's"
50 FOR N=1 TO 500:NEXT N
60 FOR N=1 TO 1024:PRINT"O";:NEXT N
70 FOR N=1 TO 500:NEXT N
80 RUN
90 SCNCLR
100 IF ER=30 THEN PRINT"I ASKED YOU NOT TO PRESS THAT
KEY!":ELSE STOP
110 PRINT"NOW, I WILL START AGAIN"
120 PRINT"BUT PLEASE DO NOT PRESS THE RUN/STOP KEY"
130 FOR N=1 TO 2000:NEXT N
140 RESUME 20
```

Line 10 of this program contains the first of the error trapping commands — TRAP. This command tells the computer which line to jump to when it comes across an error. In this case the TRAP command tells the computer that if there is an error it should go to line 90 and carry on with the program from there.

The rest of the program is straightforward, until you come to line 100. In this line we refer to a variable, ER, which has not been assigned any value by the computer. This variable is a system variable, which is a variable used by the computer. ER contains the number of the last error which was made. Since the error number for pressing the RUN/STOP key (pressing the RUN/STOP key is classed as an error by the computer) is 30, the computer will carry out all the instructions from lines 100 onwards only if the RUN/STOP key had been pressed.

Line 140 has another new command on it. The RESUME command allows the computer to continue with the main program after an error has occurred. RESUME is similar to GOTO except that it also marks the end of the error-trapping routine, so when the computer comes across line 140 it realises that this is the end of the error-trapping subroutine, and then jumps to line 20 to continue with the program from there.

There is another version of the RESUME statement — RESUME NEXT. This statement tells the computer to go to the main program and carry on with it from the NEXT statement after the one which caused the error. For instance, change line 140 to

140 RESUME NEXT

then type RUN. At some point press the RUN/STOP key. You will receive the same message telling you that you shouldn't have pressed that key, and then, after a short pause, the computer will continue with the program from where it was stopped.

There is another system variable which is used for errors. This variable is EL and contains the line number in which the last error occurred. So if you change line 10 to

10 HELLO

and RUN the program you will immediately get an error (obviously). If you now type

PRINT EL

the computer will display the number 10, which is the line in which the error occurred.

We know that the system variable ER contains the error number of the last error that occurred, but a number doesn't tell you much about what has happened. For instance, if you were told that error number 11 has just occurred you wouldn't be much the wiser. Fortunately, there is a function which helps us in this matter — ERR$. If you type

PRINT ERR$(11)

the computer will display the message SYNTAX on the screen. This is because error number 11 is a SYNTAX ERROR. Similarly, if you type

PRINT ERR$(14)

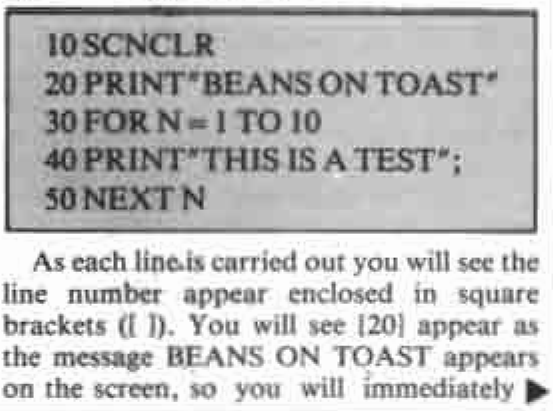
you will see the message ILLEGAL QUANTITY displayed on the screen. You can use the ERR$ to find the message for any error number except those which refer to the disk drives.

The HELP command is extremely useful for when you are trying to find a mistake in a program line. If, for instance, you have a line with four or five commands in it, and you know that there is a mistake on that line but do not know where, you simply have to type HELP and the line with the mistake in it will appear on the screen. The actual command with the mistake will flash so that you can identify it easily. The HELP command will only work after you have received an error message (pressing the key marked HELP has the same effect as typing HELP).

It is very rare when a program of any length or complexity works first time. Finding real errors (the ones which the computer spots as errors and tells you about) are no problem, especially with the HELP command. However, there are often bugs in the program which, although the program works, prevent the program from doing exactly what it is supposed to do.

In order to make the extermination of bugs much easier your computer has been given two commands — TRON and TROFF. The TRON command tells the computer to turn the TRaceON. When this happens the computer will display the line number of the line which is currently being carried out on the screen, so as soon as you see the mistake you can look to see which line is being carried out, and that will be the line with the mistake in it.

For instance, if we wanted the message on line 20 of program two to read SAUSAGE AND MASH instead of BEANS ON TOAST we could use the TRON command to find the line which displays the message BEANS ON TOAST (I know you can see it, but in a program with about 200 lines it would be a bit more difficult to spot, and this is an example). So, type in the program, type TRON and then RUN the program.



```
10 SCNCLR
20 PRINT"BEANS ON TOAST"
30 FOR N=1 TO 10
40 PRINT"THIS IS A TEST";
50 NEXT N
```

As each line is carried out you will see the line number appear enclosed in square brackets ([ ]). You will see [20] appear as the message BEANS ON TOAST appears on the screen, so you will immediately ▶

◀ know that line 20 displays that message. The trace will remain on until you turn it off with the TROFF (TRace OFF) command.

It is very often useful to be able to have a list of numbers or characters which the computer can refer to and use. These numbers or characters could be a list of names and addresses which you want the computer to search through for a specific name, for example. One way of doing this is to store each person's name and address in a string variable, or possibly a string array, and search through each variable when you want to search for a person's name. An easier way, though, would be to have a list of DATA which the computer could look through for the correct name. To set up such a list we use the DATA statement, like this.

1000 DATA "FRED BLOGGS", 123456, 'JOHN BROWN", 123642

Statements

This is just a short list, only two names and telephone numbers, but the list could go on, using several DATA statements. As you can see from the above DATA statement, any characters can be included in the list, and can be enclosed in quotation marks, although this is not essential. Numbers can also be stored in the list, and both numbers and characters can be mixed freely, as you can see.

It is not much use having a list if you cannot do anything with it. What we have to do is READ each piece of data into a variable, the numbers being read into a numeric variable, and the other characters being read into a string variable (numbers can also be read into a string variable if you want, but you cannot perform calculations with them if you do this). Program three is a simple telephone directory which asks you for the person's name and then tells you their telephone number.

```
10 SCNCLR:INPUT"WHAT IS THE PERSON'S NAME";NAME$
20 FOR N=1 TO 5:READ A$,X
30 IF A$=NAME$ THEN PRINT"THAT PERSON'S TELEPHONE
NUMBER IS";X:GOTO 60
40 NEXT N
50 PRINT"SORRY, I DON'T KNOW THAT PERSON'S TELEPHONE
NUMBER"
60 PRINT:PRINT"PRESS ANY KEY"
70 GETKEY A$:RUN
80 DATA FRED BLOGGS, 123456,JOHN BROWN,123642,BILL
SMITH,129832
90 DATA JACK JONES,126452,PETER JOHNSON,327428
```

When you RUN this program the screen will clear and you will be asked for the name of the person whose telephone number you want (choose one from the names in the DATA statements). The computer will then search through the list of DATA and if it finds that person's name it will tell you his telephone number (this is immediately after the name in the list). If the computer does not find the person's name then it will tell you.

The program works like this

Line 10: Clear the screen. Display the message 'WHAT IS THE PERSON'S NAME' and then wait for a response before assigning that response to the string variable NAME$

Line 20: Start repeating everything between the FOR and NEXT command 5 times, with the value of the variable N starting at 1 and increasing by 1 each time round the loop until it reaches 5. READ the next piece of DATA from the list and assign it to the string variable A$, then READ the next piece of DATA and assign it to the variable X

Line 30: Test to see if the string variable A$ is the same as the string variable NAME$. If it is then display the message 'THAT PERSON'S TELEPHONE NUMBER IS' and then display the value of the variable X before jumping to line 60 and carrying on with the program from there

Line 40: Marks the end of the FOR . . . NEXT loop

Line 50: Display the message 'SORRY, I DON'T KNOW THAT PERSON'S TELEPHONE NUMBER'

Line 60: Display a blank line then display the message 'PRESS ANY KEY'

Line 70: Wait for a key to be pressed and assign the symbol on that key to the string variable A$ before re-starting the program

Line 80: List of DATA

Line 90: List of DATA

Line 20 of the program READs in the next piece of DATA from the list at the end of the program (in fact it READs two pieces of DATA, but only one at a time). Each time a piece of DATA is read in the computer remembers where the next piece of DATA is, so that the next time it comes across a READ statement it knows where to take the DATA from.

Several pieces of DATA can be read in with one READ statement, as you can see from the above program. All that you have to do is tell the computer which variables you want the DATA to be assigned to in the correct order, and separate each variable with a comma. You can READ in as many pieces of DATA as you like, as long as the variable list will fit onto a normal program line, if it doesn't you will need to use two lines and two READ statements.

But what happens when the computer reaches the end of the DATA list? Well, once the computer reaches the end of the list it has no more DATA to READ in, so it gives you an OUT OF DATA ERROR if you keep on trying to READ in more DATA. For instance, if you change line 70 of the above program to

70 GETKEY A$:GOTO 10

and RUN the program for a while you will eventually receive an OUT OF DATA ERROR. This is because the RUN command (which was originally used in line 70) also resets the DATA pointer (the one which the computer uses to remember where it is in a list of DATA) back to the start of the list, whereas a GOTO command does not. To overcome this problem, we use the RESTORE command, which tells the computer to start READing in DATA from the beginning of the list again. If you now change line 70 to

70 GETKEY A$: RESTORE: GOTO 10

then the program will work perfectly.

You may also tell the computer to start taking DATA from a certain line number. For instance, alter line 70 to

70 GETKEY A$: RESTORE 90: GOTO 10

The RESTORE 90 statement tells the computer to set the data pointer to the start of the DATA in line 90. You will find that you will only be able to find the addresses of Jack Jones and Peter Johnson.

You may have thought up to now that when the computer reaches a READ statement it jumps to the line which has the DATA on it and looks through that line for the next piece of DATA. This is not the case, however, and to prove this type TRON and RUN the program. You will see that at no time does computer jump to lines 80 or 90 to look through the DATA list. This is because the computer knows exactly where the next piece of DATA is stored in its memory, so it does not need to bother about line numbers, it goes straight there.

String variables are extremely useful, and there are many ways in which you can chop them up and re-arrange them to suit your needs.

If you are familiar with the commands IF . . . THEN, you will understand lines like

```
130 IF A$ = "YES" OR A$ = "Y"
THEN . . .
```

It would be much easier if we could test to see if the first letter of A$ is a Y because then we could accept answers such as Y, YES, YEAH, or virtually any other version of the word YES starting with a Y. To do this we use the LEFT$ command, like this

```
130 IF LEFT$ (A$,1) = "Y" THEN . . .
```

This particular line tests to see what the first character of the variable A$ is. If you change line 130 to

```
130 IF LEFT$ (A$,2) = "YE" THEN . . .
```

then the computer will test to see what the first two characters of the variable A$ are (the 2 in the LEFT$ command tells the computer you want the first 2 characters, changing it to 3 would mean the first 3 characters).

Variable

Of course, you don't have to use a string variable with the LEFT$ command (or any of the other string handling commands, come to that), you could also use characters enclosed in quotation marks, like this

```
250 IF LEFT$ ("COMPUTER", 4) =
"COMP" THEN . . .
```

Here is a short example program:

```
10 SCNCLR: INPUT "DO YOU LIKE
USING COMPUTERS"; CO$
20 IF LEFT$ (CO$,1) = "Y" THEN
PRINT "I'M GLAD ABOUT THAT!"
30 IF LEFT$ (CO$,1) = "N" THEN
PRINT "OH, I'M NOT TO BLAME!"
```

The RIGHT$ command is very similar to LEFT$, except that LEFT$ tests to see what the first characters of a string are, but the RIGHT$ command looks to see what the last characters of a string are. Try this example

```
10 SCNCLR: INPUT "TYPE IN SOME-
THING PLEASE"; ZZ$
20 PRINT "THE FIRST 2 CHAR-
ACTERS THAT YOU TYPED IN
WERE";
30 PRINT LEFT$ (ZZ$,2): PRINT "THE
LAST 2 CHARACTERS THAT YOU
TYPED IN WERE";
40 PRINT RIGHT$ (ZZ$,2)
```

The 2 in the RIGHT$ command tells the computer that you want the last two characters of the string, and can easily be changed, just like the LEFT$ command.

The MID$ command is used to find out what the middle characters are, rather than the beginning or end ones. Rather than just saying how many characters you want to test, as you do with the LEFT$ and RIGHT$ commands, you also have to say where you want to start from. For example, if you had a line such as this

```
310 A$ = MID$ (B$,4,3)
```

then the computer will assign three characters from the middle of the string variable B$ to the string variable A$, with the first of these characters being the fourth character of the string variable B$.

It is also possible to replace parts of a string variable by using the MID$ command. Here is a short example program which does this;

```
10 SCNCLR
20 A$ = "HELLO HELLO EVERY-
BODY!"
30 PRINT A$
40 MID$ (A$,7,5) = "THERE"
50 PRINT A$
```

If you look at line 40 you will see the MID$ command being used to replace the second HELLO with THERE. This is done simply by telling the computer where in the string variable you want to start (the seventh character in this case) and how many characters you want to replace (in this example we want to replace five characters) and then tell the computer which characters you want to replace the old ones with. As you can see the replacement characters must be enclosed in quotation marks.

The INSTRing command is used to find out if one string is contained IN another STRing. Try this short program

```
10 SCNCLR: A$ = "PETER PIPER
PICKED A PECK OF PICKLED
PEPPERS"
20 PRINT INSTR (A$, "PICK")
```

When you RUN this program the number 13 will appear on the screen, because the P of the letters PICK is the 13th letter of the string variable A$. What you have just told the computer to do is to search through the string variable A$ to see if the letters PICK are contained within it. If these letters are contained in A$ then the computer will tell you exactly where the first letter of PICK appears in the string A$.

If you look at the string A$ you will see that the letters PICK appear twice, once in the word PICKED and once in the word PICKLED. The computer will only find the first occurrence of the characters which you are searching for. In order to make it find the second occurrence of the letters PICK you will have to change line 20 to

```
20 PRINT INSTR (A$, "PICK", 15)
```

When you RUN the program this time the number 30 will appear on the screen. This time the computer has started searching for the letters PICK at the 15th character of the string A$. In other words, the computer will search through the letters CKED A PECK OF PICKLED PEPPERS for the first occurrence of the letters PICK, and find that the P of PICK is the 30th character of the string variable A$.

If the computer cannot find the characters that you are searching for in the string then it will return the number 0.

Brackets

The LEN command is used to find out the LENgth of a string variable, or how many characters it contains. This short program illustrates the use of the command:

```
10 SCNCLR: A$ = "SUPER-
CALIFRAGILISTIC
EXPIALIDOTIOUS!"
20 PRINT "THE STRING VARIABLE
'A$' CONTAINS";
30 PRINT LEN (A$); "CHARACTERS!"
```

As you can see, the string variable that you want to refer to must be enclosed in brackets after the LEN command. ■

Brian Lloyd's article will be continued next month. The Commodore 16 — Plus/4 companion is published by Sunshine Books at £5.95, and is available from bookshops or by mail order

Sounds and strings on the C16

Another extract from the Commodore 16-Plus/4 Companion *by* ***Brian Lloyd*** *— music and more command routines*

It is time we explored your computer's sound capabilities. Your computer has three different 'voices', and you can use any two of these together. This means that you could have one voice playing a tune and another voice playing the rhythm. Two of the voices produce tones and the other produces white noise (this is useful for explosions and gun shots).

However, before we can produce any sound we have to set up the volume level. To do this we use the VOL command, together with a number from 0 to 8 (8 is maximum VOLume, 0 is minimum, or off). It is best to set the VOLume to maximum, so type in

VOL8

Now all we have to do is to choose a voice and a note and decide how long to play it for. For instance, if we wanted to play the note C on voice one for three seconds then we would type the command

SOUND 1,810,180

The number 1 tells the computer that we want to use voice one, the number 810 tells it that we want to play the note C in the third octave, and the number 180 tells the computer that we want the note to last 3 seconds (the length of the note is in sixtieths of a second).

Try changing the 1 of the SOUND command to a 2 and then a 3 to see what the different voices sound like.

As I said before, it is possible to have two voices playing at once. You can either have voices 1 and 2 playing together, or voices 1 and 3. If you type this line you will hear a note being played over the top of white noise

SOUND 1,810,360: SOUND 3,917,360

The SOUND command can be used to play tunes and to create effects.

The computer can continue to carry out other instructions while making a sound, so you may have a tune playing in the background while something else is happening.

Doctor Foster

```
10 SCNCLR:VOL 8:PRINT"DOCTOR FOSTER"
20 FOR N = 1 TO 36:READ NOTE, LENGTH
30 SOUND 1, NOTE,LENGTH
40 NEXT N
50 FOR N = 1 TO 5000:NEXT N
60 RUN
70 DATA 739,60,739,30,834,60,834,30,810,30,834,30,810,30,798,60,
770,30
80 DATA 739,60,739,30,798,30,770,30,739,30,770,90,770,60,770,30,
739,30
90 DATA 739,30,739,30,834,30,810,30,798,30,810,30,834,30,810,30,
854,30
100 DATA 834,30,810,30,798,30,798,30,798,30,880,60,770,30,881,90,
881,60
```

Telephone

```
10 SCNCLR
20 VOL8
30 FOR M = 1 TO 10
40 FOR N = 1 TO 10:SOUND 1,650,1:SOUND 1,700,1:NEXT N
50 FOR N = 1 TO 50:NEXT N: FOR N = 1 TO 10:SOUND 1,650,1:
SOUND 1,700,1:NEXT N
60 FOR N = 1 TO 1000:NEXT N,M
70 FOR N = 1 TO 8:SOUND 1,650,1:SOUND 1,700,1:NEXT N
80 SOUND 1,920,20
90 FOR N = 8 TO 0 STEP - 1:VOL N:FOR M = 1 TO 5:NEXT M,N
```

The note table covers four octaves, but sharps and flats are not shown. The frequency of the note is given for reference only. The value shown for each note are the ones which you should use as the second number after the SOUND statement. For example, to play the note C (third note down) for half a second you would use the command

SOUND 1,173,60

NOTE	FREQUENCY (Hz)	VALUE
A	110	11
B	123.5	122
C	130.8	173
D	146.8	266
E	164.7	349
F	174.5	387
G	195.9	457
A	220.2	520
B	246.9	575
C	261.4	600
D	293.6	647
E	330	689
F	349.6	708
G	392.5	743
A	440.4	774
B	494.9	802
C	522.7	814
D	588.7	838
E	658	858
F	699	868
G	782.2	885
A	880.7	901
B	989.9	915
C	1045	921
D	1177	933
E	1316	943
F	1398	948
G	1575	957

You may play a note of nearly any frequency. If you know the frequency of the note which you require then you can calculate the value to be used in the SOUND statement using this formula

VALUE = 1024 − (110840.45/FREQUENCY)

In using GOTO and GOSUB you cannot use a variable instead of a line number with these commands. To make up for this, your computer has been supplied with the commands ON . . . . GOTO and ON . . . . GOSUB. These commands will GOTO or GOSUB a line depending on the value of a variable. For instance, if the computer encountered this line

120 ON ZZ GOTO 1000, 2000, 3000, 4000

then it would look to see what number the variable ZZ represents and then GOTO one of the following line numbers depending on that value. If the value of ZZ is 1 then the computer will GOTO line 1000. If the value is 2 then the computer will GOTO line 2000, and so on.

The ON . . . . GOSUB command works in

```
10 SCNCLR
20 FOR N=9 TO 13:CHAR 1,17,N,"[CONTROL 5 CONTROL 9]
      ":NEXT
30 DICE=INT(RND(0)*6)+1:IF DICE=7 THEN GOTO 30
40 ON DICE GOSUB 60,70,90,110,130,150
50 FOR N=1 TO 1000:NEXT:RUN
60 CHAR 1,19,11,"*":CHAR 1,17,14," ONE ":RETURN
70 CHAR 1,18,10,"*":CHAR 1,20,12,"*"
80 CHAR 1,17,14," TWO ":RETURN
90 CHAR 1,18,10,"*":CHAR 1,19,11,"*":CHAR 1,20,12,"*"
100 CHAR 1,17,14,"THREE":RETURN
110 CHAR 1,18,10,"* *":CHAR 1,18,12,"* *"
120 CHAR 1,17,14,"FOUR ":RETURN
130 CHAR 1,18,10,"* *":CHAR 1,19,11,"*":CHAR 1,18,12,"* *"
140 CHAR 1,17,14,"FIVE ":RETURN
150 CHAR 1,18,10,"* *":CHAR 1,18,11,"* *":CHAR 1,18,12,"* *"
160 CHAR 1,17,14," SIX ":RETURN
```

exactly the same way as the ON . . . GOTO command, except that it GOes to a SUB-routine which should end with a RETURN statement, as with a normal GOSUB command.

Program Four simulates a die and uses the ON . . . GOSUB command.

Where you see the square brackets in Line 20, this indicates that you should change the text colour to purple and switch reverse on by pressing CONTROL 5 then CONTROL 9.

The program works in this way:

Line 10: Clear the screen

Line 20: Change the text colour to purple and print a solid block 5 lines deep and 5 characters wide, with the first block in the character square 17 across and N characters down

Line 30: Choose a random number between 0 and 1, multiply it by six and round it up before adding one to the result. Assign the final number to the variable DICE. If the value of the variable DICE is seven then carry out this line again (choosing a random number between 1 and 6, then adding 1 and rounding down makes it possible that a six will come up. The computer very rarely chooses the highest number possible when choosing a random number and therefore RND(0) * 6 is almost always less than seven)

Line 40: If the value of the variable DICE is 1 THEN GOTO the subroutine starting at line 60. If the value of DICE is 2 THEN GOTO the subroutine starting at line 70. If the value of DICE is 3 THEN GOTO the subroutine starting at line 90, and so on.

Line 50: Empty FOR . . . NEXT loop — causes a delay before re-running the program

Line 60: Display a single star at the character square 19 across and 11 down then display the message ONE with the 0 in the 17th column across and the 14th row down before returning to the command immediately after the GOSUB command which jumped to this routine

Line 70: Display a single star in the character square 18 across and 10 down then display another star in the character square which is 20 across and 12 down

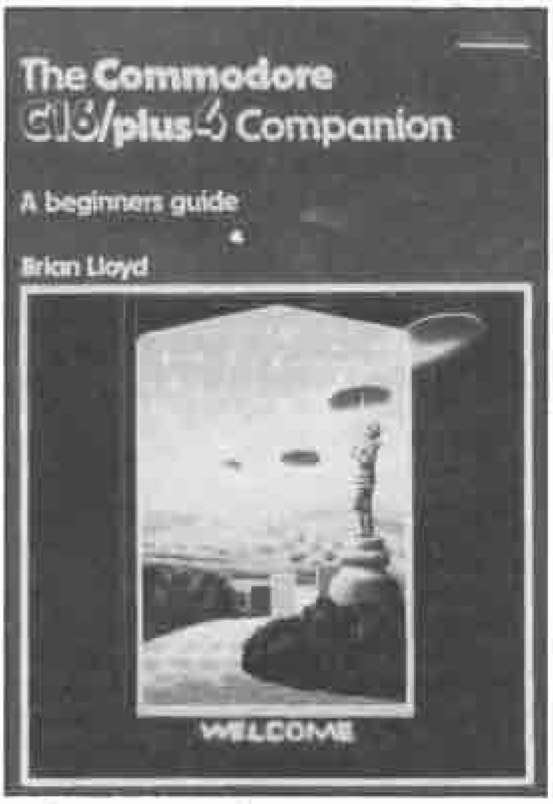
Line 80: Display the message TWO with the T in the 17th column across and 14th row down

All lines after line 80 are similar to 60-80 — they just display various numbers of stars.

If you are typing in a program from a book or a magazine then it can become very boring having to type line numbers, especially if the lines are numbered evenly (e.g 10, 20, 30 and so on). In order to speed up the entry of such programs, and also to make it less boring, your computer has been equipped with an AUTOmatic line number command. If you type in

AUTO 10

and then start typing in a short program you will see that as soon as you have pressed RETURN at the end of the first line, the next line numbers will appear for you AUTOmatically.



The number after the AUTO command tells the computer how much each line number is increased by, so if you type AUTO 50 the computer will number the lines 50, 100, 150 and so on.

Once you come to the end of the program you should press the RETURN key without typing anything. For instance, if you have finished your program at line 1510 and the computer displays the next line number 1520 and waits for you to type in some more of your program you should just press RETURN without typing anything else.

In some circumstances you may need to reset all the variables in the middle of a program. The easiest way to do this is to use the CLR command. The program itself is not stopped or altered in any way.

The CLR command is carried out automatically when you alter a program line, or RUN a program.

It is often useful to be able to find out the CHR$ code for a character. Fortunately for us, your computer has a command which allows us to find out the CHR$ code for any character without having to look it up in a table. This command is ASC.

If you type

PRINT ASC("A")

You will see the number 65 appear on the screen. 65 is the CHR$ code for the A symbol. You can find the CHR$ code for any character like this, all you have to do is enclose the character in quotation marks, and enclose them in brackets, as in the above example.

The VAL command is a function which returns the value of a string variable. For instance, if the string variable BS$ had previously been set to 921 then the command.

Z = ZAL(BS$)

would assign the number 921 to the variable Z.

If there is a combination of letters and numbers in the string variable then one of two things will happen. If the string starts with a number then the value returned by the VAL command will be the value of that number (eg PRINT VAL ("A12") would return the value 0).

STR$ is the opposite to VAL, for this function converts a number to a string. For instance, if you had a line such as this

320 A$ = STR$ (864)

then the string variable A$ would be assigned the characters 864.

The STR$ command will always add a space before the number at the start of the string variable in which the characters are to be stored. For instance, if you entered this program

```
10 SCNCLR: A$ = STR$ (864)
20 PRINT A$: PRINT LEN (A$)
30 A$ = RIGHT (A$,3)
40 PRINT A$: PRINT LEN (A$)
```

You will see that a space has been added before the number when it was stored in the string variable A$. Line 30 effectively removes this space, and when the string is displayed a second time along with the number of characters in that string, you will see that the space has been removed. ■

The *Commodore 16-Plus/4 Companion* by Brian Lloyd is published by Sunshine Books at £6.95, from bookshops or by mail order.

Answer Back

If you need help with a technical problem write to ANSWER BACK, Commodore Horizons, 12/13 Little Newport Street, London WC2H 7PP

Wiring the stick

I HAVE RECENTLY bought a C16 joystick and a 9-pin D plug to use on my CBM 64.

The problem is, I can't find out what coloured wires go where, since the 9-pin D plug has nine coloured wires and the C16 has only 7 colours. Can you help?

Matthew Andrew
East Ham
London

LOOKING at the 64's joystick port, pins 1-5 are on top and pins 6-9 are underneath, in order left to right.

Joysticks use pin 1 (up), pin 2 down, pin 3, (left), pin 4 (right), pin 6 (fire), and pin 8 (ground). Please be careful with pin 7, which carries five volts. Connecting this to the wrong input could cause damage to your computer. As you can see, only six of the nine pins on the D plug need to be connected.

Playing on the C16

I HAVE A C16, and I'm having a problem with a program I'm writing. I need to know how to make a character in the program, controlled by the player, be unable to enter an area in the centre of the screen ten characters square. Can you help?

Fergus Rhodes
Wellington
New Zealand

IF YOU use a simple, unique character such as "*" to represent the player, you can PEEK the screen to find out where it is at any stage of the game. The C16 screen consists of 40 columns by 25 rows, so we will assume that your forbidden window starts at the seventh row and finishes at the 17th, and starts at the 15th column and finishes at the 25th. In terms of screen memory, these are defined by 3367,3377,3767 and 3777. Use routines similar in principle to the following.

```
100 INPUT "move"; R,C:
REM or use joystick or key
input routine
110 GOSUB 500: REM test for
valid inputs
120 IF F=1 THEN F=0:
GOTO 100: REM rejected
inputs
130 POKE P,42: POKE P-1024,
colour: REM screen code for
"*" and colour value
140 rest of program
500 P=3072 + R*40 + C:IF P
(3072 OR P) 4095 THEN F=1:
RETURN
510 FOR J=1 TO 10: FOR
K=0 TO 9
520 IF (3367 + K*40 + J) =
P THEN F = 1:J = 10: K =
9: RETURN
530 NEXT K: NEXT J
540 RETURN
```

You can you see from the above that the character is only moved (line 130) if the move is accepted. Rejected moves are those outside the screen, negative values for R or C, and those in the 10×10 grid in the centre.

Quiz query

I AM writing a quiz game for my C16, which has two players pressing buttons when they think they know the answer. There are going to be about 600 questions in the quiz, but I can't work out a routine to make it pick random questions then not pick them again. Could you help me?

Andrew Roberts
St Helens
Merseyside

I ASSUME that your 600 questions are in the form of data statements in your program. If you don't mind the questions always being asked in the same order, a simple loop will suffice.

```
100 FOR I=1 TO 600:
READ Q$,A$:PRINT "The
next question is" Q$
110 INPUT "Your answer";
R$
120 IF R$=A$ THEN
PRINT "Correct":
C=C+1:NEXT:GOTO 140
130 PRINT "Wrong answer":
NEXT
140 PRINT "You have
answered "C" questions
correctly"
150 END
160 DATA REM each
question, followed by its
answer.
```

The alternative is to pick a question using a number generated using RND(X), from 1 to 600 inclusive. The questions and answers will have to be loaded into an array first. The number picked will then print Question$(I) say, and check the answer. Using a marker array such as A(I), which is set to equal 1 for each value of I that is used, test this when generating a random number to make sure that this number has not been used previously. If so, then discard it and try again. You must also test with a counter to see if you have already used all 600 questions, then exit from the random number loop otherwise you will go around forever!

Use the monitor

I HAVE a C16, and am wondering if there is any way to stop the screen flashing during loading. I would like to be able to produce a loading screen, and also play music while loading. Also, how do you program multicolour UDG's?

Michael Chambers
Ryde
Isle of Wight

THE FULL answers to your questions are too complex to give in great detail. You will need to learn to use the built-in machine code monitor, so get hold of some general C16 programming books, and something on machine code. In general terms, you should copy the relevant cassette loading routine into RAM using the monitor and the Transfer Memory command. The screen blank can then be eliminated from the code using a JMP, then your program can be loaded or saved using the amended routine, with a final jump to the normal routine at the end of the cassette load routine.

To use music while loading, the music routine is patched into the system via the interrupt, using an amended CHRGET routine, and returned to normal after cassette use.

Multicolour UDG's use a lot of system memory, but if you want to find out more get *The Working Commodore C16* by David Lawrence, published by Sunshine Books at £6.95. The advantages and disadvantages of multicolour mode are dealt with on page 61, but bear in mind that once you have combined this with your own UDG's ther will be very little memory left for the rest of the program.

C16 set-up suggestion

I HAVE RECENTLY found a program which allows you to redefine the character set of the Vic 20. Can you tell me how to modify it for the C16?

M Lethrope
Leyland
Lancs

BEFORE you can make any real use of such a program, part or all of the character set in ROM should be transferred to RAM. Because of limitations in memory size, I suggest that the uppercase set (2K) only is transferred.

The simplest and fastest way on the C16 is to use the inbuilt machine language monitor. Type MONITOR and press RETURN, then enter DOOOO D7FF 3800 (RETURN). Exit the monitor by typing X.

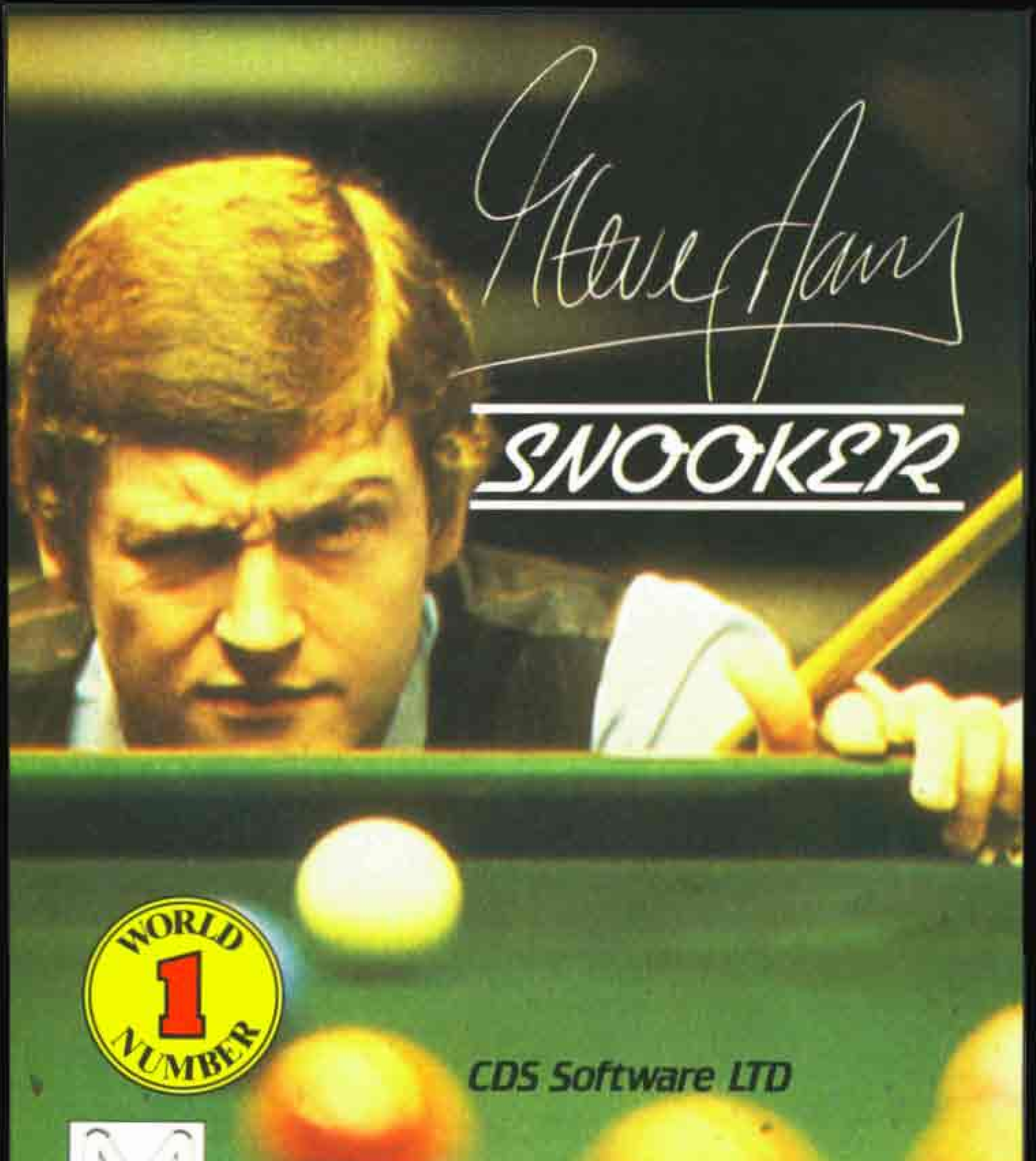
Then use the following program;

```
10 DATA
0,254,124,56,16,16,65,0
20 TRAP 10000
30 FOR A=14336 TO 14343
40 READ B
50 POKE A,B
60 NEXT A
70 C=65298:POKE C,
PEEK(C) and 251
80 POKE 65299,56:POKE
52,55:POKE 56,55

90 REM restoring of the normal
set follows — use END if this
is not required

10000 POKE 65299,208: POKE
65298,PEEK(65298)OR4
10010 SCNCLR
10020 PRINT ERR$(ER),EL
10030 END
```

Recommended reading: *The Working C16* by David Lawrence, published by Sunshine Books at £6.95.



Available for the C16 £7·95
C64/128 Cassette £8·95 Disk £12·95

From all good computer stores

OR DIRECT FROM **CDS Software Ltd**

CDS Software Ltd, Silver House, Silver Street, Doncaster DN1 1HL (0302) 21134 (5 Lines)

USEFUL LINKS

www.commodore16.com

The main site for the Commodore 16 Computer
with games, scans, news and more.

http://plus4world.powweb.com

The main site for the Commodore Plus4 Computer
with games, scans, news and more.

http://c16.c64games.de

Another games resource site for the C16 and plus4

http://yape.homeserver.hu

Home of the YAPE emulator for the Commodore16 and Plus4

http://www.zzap64.co.uk/zzuperstore.html

Many computer game retro magazines available to buy on DVD

C16/+4 YOUTUBE PAGES

http://www.youtube.com/user/fonsisweb

Lots of nice shortplays of many C16/+4 classics uploaded by Fonsis

http://www.youtube.com/user/16plus4equals264

Updated longplays, walkthroughs and shortplays by Sixteen Plus