

# Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

*Now Incorporating*

# YOUR 64

**Match 1-**  
the first part  
of our type in  
assembler



**Inside  
the 128**

**Full colour  
Action Replay**

**Competition-**  
10 Commodore  
modems must  
be won

**Summer Games II-**  
Game of the month

**WIZARD**  
COMPUTER GAMES

**A NEW EXCITING ADVENTURE GAME** By Tony Crowther  
£1000 Treasure to be won & a free disk with every game

Available for Commodore 64, Spectrum & MSX from October 1985 (to be available for Amstrad)

# WILLIAM WOBBLER



The game is a  
adventure game  
with many things  
to do. It is a  
game of strategy  
and tactics. You  
will be able to  
win the game.

William Wobbler is the latest game from Tony Crowther—a most exciting adventure game. Through the underworld of dark powers in search of golden treasure William struggles against all odds to vanquish foes and reach his goal. A game of skill and excitement.





**Editor:**  
Stuart Cooke

**Assistant Editor:**  
Mark Curry

**Advertisement Manager:**  
Mike Squires

**Advertisement Copy Control:**  
Laura Champion

**Group Editor:**  
Dave Bradshaw

**Group Managing Editor:**  
Wendy Palmer

**Managing Director:**  
Peter Mathison

**Origination:**  
Moore Typetting

**Design:**  
Algor Design

Editorial & Advertisement Office  
No 1 Gifford Square  
London W1R 3AB  
Telephone: 01-837 0026  
Telex: 887188

Your Commodore is a monthly magazine appearing on the Top Profile of each month.

Distributed by Argus Press  
Sales & Distribution Ltd, 11-19  
Paul Street, London EC2A 4LQ  
Owned by Argus Press Limited  
& Sons Ltd, York, Yorkshire,  
York

Subscription rates upon  
application to Your  
Commodore Subscriptions  
Department, Informa Ltd, Times  
House, 118 The Strand,  
London WC2R 0BB, 1985.  
188

The contents of this publication  
including all articles, designs,  
plans, drawings and programs,  
and all copyright and other  
intellectual property rights  
therein belong to Argus  
Specialist Publications Limited.  
All rights conferred by the Law  
of Copyright and other  
intellectual property rights and  
by virtue of international  
copyright conventions are  
expressly reserved to Argus  
Specialist Publications Limited  
and any reproduction requires  
the prior written consent of the  
Company. © 1985 Argus  
Specialist Publications Limited.

## FEATURES

- **The C128 - in Depth** 10  
We take a look at Commodore's new baby.
- **Language Lab** 14  
This month Pascal comes under our expert's eye.
- **Small Print** 48  
Read our C-128 printer review.
- **The Amiga Unveiled** 68  
Commodore's latest offering scrutinised.
- **Teal Chip** 94  
An in-depth look at this clever chip in the C-16 and Plus/4.

## SERIES

- **Mach 1** 18  
The first part of our type-in macro assembler for the C64.
- **Mastering the C-16** 26  
Continuing our C-16 programming series.
- **Top Draw** 37  
Yet more graphics tricks from Allen Web.
- **Build a Better Basic** 58  
The second part of our Basic extension package.
- **Programming Projects** 76  
How to create artificial landscapes on your 64.
- **Reliable Routines** 92  
Some handy routines for medium resolution plotting.

## REGULARS

- **Data Statements** 4
- **Input/Output** 8
- **Listings** 28
- **Submissions** 29
- **Game of the Month** 42
- **Sense of Adventure** 45
- **Action Replay** 50
- **Teacher's pet** 56
- **Scratch Pad** 95

## GAMES AND UTILITIES

- **Full Speed Ahead** 31  
Speed up loading and saving with this tape turbo for the C64.
- **Split 64** 70  
Two into one does go with this handy routine.
- **In Character** 80  
Design your own character set with this program for the C-16.

## COMPETITION

- **Commodore Here** 49  
We one of 10 Commodore models in this month's competition.

## Our COMMENT

THIS MONTH, WE'RE DELIGHTED TO announce that Your Commodore has now joined forces with Your 64 to bring you the best Commodore magazine around.

Publishers of Your Commodore, Argus Specialist Publications, recently concluded the purchase of four 64 from Sportsman Press, Peter Welham, MP, M.D. said: "This acquisition consolidates our position as one of the leading titles in the popular Commodore market and further confirms our commitment to machine specific titles."

We're pleased to welcome all our new readers to our pages and we can promise that you won't be disappointed, we're really going to have our work cut out over the next few months making sure we bring you only the best in software and hardware reviews, games and utility listings, plus many exciting and absorbing features for you to follow.

We're also very conscious of the fact that a magazine is nothing without its readers and so we'll be waiting anxiously for the postage every day to find out what you really think about our magazine.

Write to us at Your Commodore, Pop 3 Golden Square, London W1R 8AL and tell us what you think are the best (and worst) features of the magazine and give us your ideas on what needs to be included. We promise to read everything you send so get those letters in the post now.

Stay with us over the next few months and we'll bring you the only Commodore magazine worth reading.

Steve

# DATA STATEMENTS



Part of Argussoft's new range

### The sky's the limit

ARGUSOFT'S RECENT SUCCESS, SKYFOX, has now been released on cassette priced at £9.95. The game was originally only available on disk and the cassette version eventually released was the third attempt at maintaining the standard of Skyfox since two earlier conversions were rejected by the manufacturers as being not up to scratch.

Skyfox is only one of a number of new releases from Argussoft. The latest titles are Racing Destruction Set, Football Construction Set, Adventure Construction Set, Music Construction Set, Tennis Cities of Gold and Mail Order Monitor. So far these titles are only available on disk and are all priced at £14.95.

Argussoft can be contacted at Suite 109/105 Alphaite House, Palace St, London W1R 9J5.

### Chatting up

COMPUTER IS DEVELOPING A NEW SERVICE scheduled for a late autumn launch. It is a scrolling, CB style chat facility. The chat line will allow users to hold conversations in real time with other users all over the country.

Contributions are typed into a window at the bottom of the screen while a second window above displays messages from other users. By scrolling back and forth the whole conversation can be reviewed. Users with similar interests can form groups by using individual chat lines which will be provided and those wanting a chat can monitor various conversations before deciding which one to join.

For further information contact: Computer, Merford House, 15-18 Clippens St, London W1P 7DQ.



## Starstruck

STARBUCK FROM MELBOURNE HOUSE IS now available for the Commodore. It was originally released for the Spectrum and Amiga computers.

The game will be priced at £9.95 and is obtainable from Melbourne House, Castle Yard House, Castle Yard, Richmond TW9 6TT.

## Beyond the fringe

BEYOND IS BRANCHING OUT WITH two new games.

The first is the computer version of Superman, a game developed as a joint venture with First Star. Bill Delaney, Beyond's MD said: "It will shoot to number one in every software chart."

Beyond has also formed a partnership with another software house, Nexus. The first game to appear on the new label will be called Nexus and will be a "sophisticated graphic adventure where the player becomes an investigative journalist infiltrating an evil drugs ring in South America.

According to Nexus, the game will offer maximum playability and user friendliness. Instead of having to read pages of documentation before starting, the player will be able to experience action straight away through on screen instructions. Nexus will also feature digitised video images which, it is claimed, will generate realistic animation enabling the player to recognise various characters.

Nexus costs £9.95 and is available from Beyond, Durrant House, 8 Herbel Hill, London EC1R 3EL.



Starstruck from Melbourne House

## Gold standards

US Gold has brought out yet another batch of new releases. The three latest titles are Monster Trivia, Beach Head II and Ghostchaser.

Monster Trivia is a spin off from the enormously successful board game, Trivial Pursuit, but in this version you get killed by a monster if you get enough questions wrong.

Beach Head II is the follow-up to Beach Head and features multi-screen play, complex strategy, animation and high speed arcade action.

Ghostchaser takes Harry around Fozzport Island where he must blast large ghosts which materialise at random.

All the new titles are for the C64 on cassette or disk and cost £9.95 and £14.95 respectively. You can get in touch with US Gold at Unit 18, The Parkway Industrial Centre, Henegate St, Birmingham B7 4LJ.

## Vox pop

POP STAR, PEARCEL SHARKEY, HAS recently become the owner of the first production model of Supersoft's digital sampler for the C64.

We visited the Supersoft stand at the Commodore Computer Show and were very impressed with the sound reproduction quality of the new product.

The sampler is called Microvox and offers eight different sampling rates, up to a maximum of 48000 which gives a 100Hz band width. Samples can be played forward or backward and there is full editing and looping with a high resolution stop.

A 3000 note sequencer is included as part of the Microvox software with real time recording and step time editing.

Microvox costs £295.95 and is available through music shops, computer stores or direct from Supersoft at Birmingham House, Canning Rd, Washdonsa, Middlesbrough HA3 7TB. A disc drive is essential.



Pearl Sharkey sits up and takes notes



Oh joy!

### Jumping the gun

C-16 and Plus II owners can now pick up a bargain in joysticks with Italian Electronics' new packages.

The GunShot I joystick is now being sold complete with an adaptor for C16/II. The packs are available from Dions, Lasky and other computer retailers.

Everything the user needs to connect the joystick to his computer is included plus concise instructions and a 12 month guarantee slip.

For more information contact: Italian Electronics, 200 Broad St, Hendon, London HA8 1BH.

### Eye contact

SPECTACLE WEARERS WHO USE YOURS or television screens as a major part of their everyday lives can now try a new way of protecting themselves from eyestrain.

Salvars has come up with Quasar, a tough anti-reflection coating which, claim the makers, virtually eliminates lens surface reflections especially those caused by strong projected images. The coating can increase light transmission to almost 100 per cent.

The result for the user is that extra visual sharpness is noticeable during long periods of exposure, thus reducing visual fatigue.

For more information contact: Salvars, Northbridge Rd, Brixham, Devon TN4 1LN.



"It never has to feel like this again!"



## Become a boffin

A NEW HOME COMPUTER TEACHING course has recently been released by Fearless Software. It is claimed that the course will take you from scratch to complete computer literacy in 12 months.

Peter Ellis, who formed the company to produce and market the Home Tutor said: "It is for children, housewives and the retired. It is also for those seeking new jobs or looking for promotion with their present employer."

The course is available on tape or disk and includes, demonstrates, corrects and tests the students. There is an examination at the end and those who pass receive a certificate.

The course starts by explaining what a computer is and goes on to teach basic and machine code, covering such areas as information storage and communications. There are 12 parts to collect over the 12 month period.

Home Tutor costs £12.50 per month including tapes, teaching notes, a carrying case and ring binder. For disks the cost is £2 extra and postage is £1.15. Quarterly charges are £30 for tapes and £35.50 for disks.

For more information contact: Fearless Software, Implan, 50 Earlsbourne Terrace, London W2.

## Errata

A NUMBER OF PEOPLE SEEM TO BE having problems entering the Sketch Pad program which appeared in the September issue of Your Commodore. There are no errors in the program but some of the codes that our printer interface uses seem to be causing the problems.

In line 1490 the [255] is actually the character code for pi (π). This character is to be found on the key next to the

RESTORE key. When entering the program you should type in the pi and not [255].

Another line that is causing problems is 628. It appears that in some issues of the magazine this has not reproduced very well. Line 628 should start with P% = 256 THEN

A number of people also seem to be having problems finding the I character on their keyboard (this is the up arrow). It to be found on the key next to the RESTORE key. The printer that we use does not print the downstroke of the arrow.



British Telecom and Program Express sign up

## Telecom deal

BRITISH TELECOM AND PROGRAM EXPRESS have joined forces to make the lives of software retailers and buyers less fraught with frustration.

Program Express is the firm which launched the Electronic Distributor for Software Machine in July 1988 which works on a "Central Computer Network/In-store Satellite Terminal" principle. The retailer can download a unit of software from the in-store terminal onto a blank tape. The machine records all relevant details about the sale and at the same time new titles can be added to the hard disk terminal while old ones are deleted.

Under the new agreement British Telecom will finance the operation and Program Express's three directors, Gilmore Kennedy, Bruce Neville and Gwyn Roberts will run the company autonomously.



The in-store terminal

**Agony Aunt Tony Crowther**

answers more of your  
programming questions.

## INPUT

I have been told on many occasions that it is possible to make all of the keys on the Commodore repeat, as on the Spectrum.

This seems to be a very handy facility as you could enter long strings of the same character without having to press the key for each letter.

However, I do have one slight problem, someone can I find manual or how to do this. Would it be possible to provide me with the necessary commands or program to make the Commodore keys repeat?

Mike Adkins  
Tasmania

## OUTPUT

Yes, it is possible to make all of the keys on the Commodore repeat, it is very easy to do. All you have to do is POKE into a few memory locations. The locations are as follows:

Location	POKE	Result
658	255	all keys repeat
658	0	no keys repeat
658	128	just the cursor repeat
651	0-255	repeat speed
651	0-255	delay before repeat

The keyboard scan is done by the hardware (80C) interrupt, if we change the clock rate of timer 4, we can make the rate of the interrupt speed up or slow down. Try:

POKE 1625, number (0 to 255)

A number of zero to 99 slows many interrupts a second and causes the Basic run speed to slow down. This could be used to your advantage when debugging a Basic program as everything will slow down.

## INPUT

I write to congratulate you on the quality of your magazine. I find it to be the best of the British Commodore mags available in Australia. However I feel that the atrocious reproduction of graphics symbols speak a. I would suggest that you employ a system like the American magazine 'Amp', which permits rapid loading and clear understanding of listings.

P Robinson,  
Melbourne,  
Australia.

# INPUT

If you're not sure what system Alton uses but we nevertheless accept that the listings need improving. Therefore we will be using a micrographic M99100 in future, which replaces all graphic characters with a short description.

## INPUT

I have a problem with my 64 that I cannot get an answer to. After about two hours my computer crashes, putting random graphics on the screen and eventually the whole screen is covered in flashing graphics if I turn it off for a couple of hours the problem goes away but it returns more frequently afterwards.

L. Sims  
South Shields  
Type 8 West

## OUTPUT

The problem is that your 64 is overheating. I suggest that you take it back to the dealer and see if he can replace it.

## INPUT

In your March issue you did a feature on Games creators and this article impressed me so much that I wish now to buy one. The one I am interested in is the Quill by Giliboth, could you please tell me their address.  
P. Bazzalo  
Pisa  
Italia

Giliboth are 21-20 Hawthorn Road,  
Ferry,  
S. Coast.  
CA1 8D.

## INPUT

If you're writing a program in machine code that requires a lot of data storage, I understand that there is 8K of RAM underneath the Basic kernel. This would be an ideal place to store my data and machine code routines.

Could you please explain how I access this area of memory as when I try I only get the Basic ROM and not the numbers I have stored beneath it?

## OUTPUT

As you have already found out you can't use the area of memory beneath the Basic kernel where Basic is running as any keys or calls to that area will just go to the kernel. If however you are using machine code you can turn off the Basic kernel and use the memory that sits beneath it quite easily.

First you must switch off the Basic ROM. To do this you must the lower bit of memory location zero. This can be done simply by subtracting one from it. If this action was to take place in Basic your program would crash. However, in machine code it will not affect the program.

If your machine code routine was to start at — in your data was stored at — location 0000 (\$A000), then this small routine would allow you to access it.

```

$C000 DMC 101 ; switch-off Basic ROM
$C000 DMC 10000 ; jump to start of program
$C000 DMC 101 ; back here from your routine

; End here Basic

; back on
$C000 RTS ; return to Basic or end of program

```

# OUTPUT



# COMMODORE SOFTWARE THROUGH THE POST.

(WITHOUT PAYING THROUGH THE NOSE)

Simply send off this page and you'll get:

## SOFT POST MAGAZINE FREE

No more trawling around the shops trying to find the software you want. Let Commodore post it to you. Simply make your selection from our Soft Post magazines which will send you every few months (they're packed full of software information) - but there's no obligation to buy anything, if you don't want to.



## WIDEST CHOICE OF COMMODORE SOFTWARE ANYWHERE

In every magazine there'll be special offers, an enormous selection of great software at amazing prices - and not just games, but educational, business and home use, too. (Sometimes you'll even find peripherals at special prices.)

## PLUS ONE OF THESE PROGRAMS FREE

Tick the box against the software of your choice

COMMODORE 64  
 Robert Carter's Menu  
 Planner (Share)  VS: 80  
 Know your Own ID (Share)  Sargent's Dream (Share)  
 Pascal Moonlit  COMMODORE 16 Plus 4  Design Place (Share)  
 Astronomy (Share)  Harbour Master (Share)  Road Race (Share)  
 Space (Share) (Share)  COMMODORE 16 Plus 4

I would like to receive a regular issue of Soft Post. I own a Commodore (please as applicable):  C64  64+  16 Plus 4

Title (Mr)  (Mrs)  (Miss)  (no title)   
 Mr  Mrs  Miss  no title   
 Surname   
 Address

STD Code            
 Postcode            
 Telephone

## OR, IF YOU INTRODUCE A FRIEND

Fill in your name and address and your friend's below. Now you can choose one item from either list, just tick a box.

VS: 80  COMMODORE 64  
 Programmer's Reference  Programmer's Reference  
 Guide (Book)  Guide (Book)  
 Introduction to BASIC II  Fantasy Plot (Manual)  
 Consultant  COMMODORE 16 Plus 4  
 Jack Attack (Share)  Jack Attack (Share)

I would like to introduce my friend who owns a Commodore (please as applicable):  C64  64+  16 Plus 4

Title (Mr)  (Mrs)  (Miss)  (no title)   
 Mr  Mrs  Miss  no title   
 Surname   
 Address

STD Code            
 Postcode            
 Telephone



ONLY ONE APPLICATION PER HOUSEHOLD.  
 PLEASE ALLOW 28 DAYS FOR DELIVERY.  
 Post to: SOFT POST REQUEST, Commodore Business Machines, Ltd., Dorby, Northamptonshire NN4 7BR.

**Stuart Cooke has spent hours slaving over a red hot Commodore 128 and he's decided that it was definitely worth the effort.**

# THE C128 -IN DEPTH

COMMODORE ARE ALL SET to prove that three into one does go with their new Commodore 128 computer.

The C128 is essentially three computers in one. Inside a case that looks as though it would be quite at home on the flight deck of some futuristic spaceship there is a good old C64. Secondly we have what can only be described as an upgrade of the C64, 128 mode. Hardware-wise this is the same as the C64 with only a few changes. Thirdly there is a machine that has also been around for quite a long time, but not from Commodore. This is a CP/M computer. CP/M has been around for a number of years and is the name given to an operating system that is used on over-280 based business systems to allow a great deal of compatibility between manufacturers. Perhaps one of the most famous business packages that runs under CP/M is the word-processing package Wordstar which is in use in thousands of offices around the world.

## First Impressions

The first thing that you notice about the C128 is that you are not getting some fantastic new machine for your money. What you do get are two very well established machines and one very good upgrade.

The machine itself has been designed to achieve as much compatibility as possible with the C64 and other Commodore products. For this reason many of the expansion connectors will be familiar to owners of other Commodore machines. There are connectors on the machine for two printers, Commodore's own serial peripheral connector, TV interface,

composite video interface, a standard Commodore cassette interface, a user port and a cartridge port. New items obvious on the sides of the machine are a reset switch and a RGB monitor interface.

Even though Commodore has tried to make the C128 as compatible as possible with the C64, the cassette interface is in a silly position. If you plug a Commodore magnet into the cartridge port the casing of the magnet obscures the cassette interface preventing a cassette recorder from being plugged in.

The keyboard bears only a slight resemblance to earlier Commodore machines as there are more keys available. Added keys include a numeric keypad and a large number of function keys.

## In Use

When powering up you are able to decide whether you

wish to boot up the C128 in 40 or 80 column CP/M, 128 or 64 modes. Here we come across the first problem with the C128. The 80 column display can only be viewed on a monitor that is plugged into the RGB interface while the 40 column display can only be viewed on a TV or monitor plugged into the composite video socket if you wish to use both 40 and 80 columns then you will need two different monitors. Commodore has produced a new monitor that will allow you to switch between the two modes of operation. A monochrome composite signal is present on the RGB socket so you can plug a monochrome monitor to view 80 columns if you already own one. Using a monochrome monitor will not cause any problems especially when you realise that the 80 column mode is text only and no graphics are available.

C64 mode offers nothing new, what it does do however

is allow you to use all of the packages that are currently available for the C64 on your C128 computer. This is great news for people who already own C64 and are thinking about upgrading. Over 100 pieces of software were tried on the review machine and all of them worked correctly. The only problem that we found was with programs that used keyboard overlays as the keys are in different positions. This is especially noticeable with Commodore Music Maker package as the keyboard will not fit the new design. Even so it appears that all problems will work with no fault.

An on board 256 micro processor allows you to run CP/M Plus version 3.0. CP/M Plus will give your CP/M programs full access to the 128K of RAM that is built into the machine as standard. Commodore has made a few changes to CP/M Plus, mostly by adding some new com-







Instruction, **RECV/BIND** will allow a number of program lines to be included after a **THEN** statement.

## Graphics

Budding artists are well-served for inside Basic 7.0 by a large number of graphic commands. **GRAPHIC** is used to turn on one of the six different graphics

modes. Because of the total CP/M compatibility is something that will have to be proved. Unfortunately the review machine did not come with any CP/M software so this was something that I was unable to test.

As I have already mentioned, **LOAD** mode offers nothing very new apart from an excellent Basic and an 80 column display. Basic 7.0 can only be described as the Basic that should have been implemented on the C64 as it offers total control of all the facilities offered by the hardware without having to resort to POKEs or machine code.

The new disc drive is also a great improvement on the old **DS4** when used in **CDSB** or **CPM4** mode as it will **LOAD** and **SAVE** programs around three to four times faster — a great relief to anyone who has used a **DS4**. Unfortunately, when used in **C64** mode the drive works at the same slow speed that we all know and hate.

## Basic 7.0

Basic 7.0 contains a large number of commands that are designed to ease the use of sound and graphics together with a large number of 'lookit' commands. For example the **AUTO** command will automatically give line numbers.

**LOAD** and **SAVE** allow you to load and save specific sections of memory. A reminder function helps with the development of long programs. Programs that do not work correctly are easier to debug with the **HELP** command which will show you where your error has occurred or the **TRON** and **TROFF** commands which will display the line number of the line that is about to be executed.

Error trapping is provided by the **TRAP** command. Whenever an error is detected this command will cause the program to jump to the specified program line rather than 'bombing out'. You will then be able to find out what error has occurred and take the appropriate action. This command will make debugging your Basic programs very easy.

Finding out what the controller are doing is also made very easy. **CON** will tell you

which direction a specified joystick is pointing in and whether the fire button is pressed. The position of the paddles and light pen is also easy to find out with the **POS** and **PDN** commands respectively.

Machine code buffs will be pleased to hear that a machine code monitor is included. This will allow you to display, alter and move sections of memory. There is even a machine code assembler and disassembler. The Basic keywords **HEX** and **DEC** will also prove to be very handy as they will allow you to convert decimal numbers to hex and vice-versa.

A number of new structure commands have been added to the standard **FOR/NEXT** loop. These include **DO/LOOP** which will repeat a section of your Basic program UNTIL a specific condition is met or **WHILE** a specific condition is met. **IF/THEN** will only allow one statement after the **THEN**

mode that are available. The six modes are, 40 column text, standard bit-map graphics, standard bit-map with split screen, multi-colour with split screen and 80 column text. As previously mentioned the 80 column screen can only be used with a monitor plugged into the RGB socket. The split screen modes are very interesting as they allow you to use a section of the screen in one of the graphic modes while still retaining a specified segment for text only. This type of feature has been used many times on the C64, especially in graphic adventures, but is now extremely easy to use on the C128.

Drawing lines or shapes on a graphics screen is also very simple. The **COLOR** command allows you to set up the colours that you require for any plotting etc. The **LOCATE** command can be used to position the graphics cursor at any point on the screen, and

**DRAW** will allow you to plot dots or draw lines in the specified colour. Shapes can easily be drawn as the draw command will allow you to string a number of points together by using the word **TO** as in

```
DRAW,100,100 TO 10,100 TO 10,10
```

**BOX** makes it extremely easy to draw rectangular shapes on the screen, all that is necessary is to specify the top right hand and bottom left hand co-ordinates together with the location of the box. You can even specify if you want the box to be filled with a certain colour.

**CIRCLE** is used to draw circles, ellipses and 'other' shapes. This may seem to be a weird statement but when you see how complex the circle command can be you will understand what I mean. The **CIRCLE** command can have up to nine parameters, these are: the colour number, the centre of the circle, the X radius, the Y radius, the starting arc angle, the ending arc angle, the rotation in clockwise degrees and the number of degrees between segments. This does appear very complicated at first glance but playing around with the parameters will soon show you how versatile this command really is. Not all of the parameters need to be used every time that the command is used. The following example is from the manual and will draw a diamond shape on the screen

```
CIRCLE 1,200,40,30,30,20
```

**WIDTH** can be used to set the width of any lines that you are drawing while **SCALE** will allow you to alter the size of your diagrams with ease. Another handy command is **PAINT** which will allow you to fill in any area of the screen with a specified colour.

A very limited form of window is implemented on the C128 through the **WINDOW** command. This allows you to set up a rectangular area on the screen in which all further screen updates will occur. The size of this rectangle can also be set up outside a program by using **ESC T** to set the top left

corner and **ESC B** for the bottom corner.

As well as having commands for producing pictures on the screen, there are also a few that will tell you exactly what is happening on your display. The **BOX** will tell you which graphic mode the C128 is in at the moment. **BOXC** returns the current position of the graphic cursor or the colour of the graphic cursor. You can even find out the window parameters by means of the **WINDOW** command.

## SPRITES

As with the C64 there are eight sprites available for use on the C128. These can be either hi-res, one colour, or multi-colour. However, unlike the C64, you'll never have to perform a single **POKE** to memory as Basic 7.0 supplies all of the commands that you'll ever need. You don't even need a sprite editor as there is one built into Basic.

**SPRIMP** is the command that forms as the C128's sprite editor. On entering the command the screen clears and a sprite grid is displayed on the screen. Facilities available in the editor are: turning on and off multicolour mode, changing the colour, expansion of the sprite in X and Y directions, copying sprites and saving sprites. In fact all of the commands that you are likely to need are implemented. One fairly major omission from the sprite editor is the ability to move the sprite around in the grid. You cannot, for example, rotate a sprite or shift it left by one pixel. No doubt someone will develop a routine to perform these commands.

Another way of defining a sprite is to 'draw' the sprite onto the screen using the more drawing commands. The **SPHAR** command can then read the sprite data into a string variable. Moving the contents of this string into sprite memory is also made very easy by the **SPHSAV** command, this moves the specified string into the sprite not required.

The **SPRITE** command allows you to turn on and off the individual sprites. **SPRITE** also lets you set the sprite colour, whether it passes over or beneath the background,

whether it is hi-res or multi-colour and if it is expanded in either the X or Y directions.

Allowing a sprite around is also made child's play with the **MOVSPR** command. **MOVSPR** can take a number of forms, it can place a sprite at a specified point on the screen, it can be used to move a sprite to a new position relative to its old one, move a sprite a certain distance at a specified angle and, perhaps its most powerful use, it can set a sprite moving in a specified direction at specified speed and keep it moving. As you can probably see, the **MOVSPR** command will be a great boon to anyone who wishes to write a game program. Moving your latest deadly creation across the screen can now be done by one command rather than the numerous lines of code that C64 users are used to.

Looking after your sprites is no longer a problem as the **COLLISION** command will cause a jump to a specified line number when a sprite hits the background or another sprite. The **BUMP** command can then be used to return the values of any sprites that have collided since the last **BUMP** command.

Finding out the specific details of any sprite is also made easy with a number of commands. **SPFCOLOR** returns the colour of a specified sprite. **SPXPOS** will return the X, Y co-ordinates of a sprite, very handy after a **BUMP** invocation, and **SPRITE** will tell you whether a sprite is on, off, expanded etc.

As you can no doubt see from the brief mention of the sprite manipulation commands that are mentioned above, writing any sort of program that uses sprites is now extremely simple. Before very long we should start to see some excellent graphic programs written totally in Basic.

## Sound

The sound chip that is used in 128 mode is exactly the same as is the C64, the major difference being that there are now a large number of commands available to make control of it easy. There are five commands available for sound manipulation, **SOUND** is used to place quick and easy sound

effects in your program. **SOUND** can have up to eight parameters. These are: the voice number (one to three), the frequency of the note, the duration of the note, whether the sound is to be incremented or decremented while playing, the minimum frequency the note can go to, how big the step-up or down is, the type of waveform to use and the pulse width if you are using a square wave. As you can see, some very interesting effects can be made by using this command. Another command is designed to make the playing of music easy, this is the **PLAY** statement. **PLAY** allows you to set up a string of synthesizer control characters inside quotes. Characters allowed are the musical notes ABCDEFG, characters which tell the synthesizer what type of note is playing and characters to specify the color, octave, envelope volume and filter. Ten predefined envelopes are available ranging from harpsichord to xylophone and you are able to define your own using the **ENVELOPE** command. **TEMPO** defines the speed of the song being played and **VOL** the volume. The command **FINISH** will also let you set up the filter parameters very easily.

Adding a musical accompaniment and sound effects to your program is now very easy. Even a beginner will soon be producing sounds that are the same quality as those that the top programmers have been producing on the C64.

## Verdict

Obviously I have not been able to cover all of the details of the C128 and its Basic, there are many commands that I have not covered. However, from the few that I have mentioned I think it should be fairly obvious that the C128 is a powerful machine, as should I say machines. The fact that it will run all C64 software, and that thousands of business packages are available with CP/M, make the machine a bargain for the beginner, hobbyist and businessman alike.

Commodore seems to have a winner.





**Our resident linguist, David Janita, gives you a breakdown of Pascal packages for the C64.**

# Language Lab

## P·A·S·C·A·L

PASCAL WAS INVENTED BY ONE MAN, Niklaus Wirth of the ETH Technical Institute of Zurich in 1970. It is a compiled language that was designed as an aid to teaching good programming practice.

Because the language is very concise, instructions found it easy to implement on their systems. Software houses also discovered that it was possible to implement Pascal on many home micros, hence the reason for its early appearance on the micro scene.

### Program Body

Unlike Basic, where you have a free hand in program structure, Pascal requires the programmer to "section" his programs. There are three main sections in a Pascal program.

```
PROGRAM - header declarations
BEGIN - Main body
END
```

The first section is the program header. Every Pascal program must start with the reserved word **PROGRAM**, which is followed by the name of the program. This can be optionally followed by I/O declarations, implemented in various ways in different compilers.

The next section is the declarations. There are a number of these, first there is the reserved word **CONST**, used to define a symbolic constant:

```
CONST
  PI=3.14159
  ACQ=23
```

In the example, two constants have been declared. Following the constant declarations, come the more common variable declarations:

```
VAR
  R : REAL;
  MUMWEIGHT,TOTAL : INTEGER;
  MICHIE : BOOLEAN;
  INITIAL : CHAR;
```

The four data types are integer, real, string and boolean. Pascal requires all variables to be declared explicitly, for Basic programmers this may come as a bit of a shock. It's only easy in Basic to declare yet another variable as you need it, but in Pascal this is not the case. The good point in declaring variables is that you need to do some thinking and plan on how many variables you will need in the first place!

The next declaration is probably one of the most powerful features of Pascal - type definition. As you can see from above, there are four data types in Pascal. These are pre-defined data types, and if you wish you can declare more:

```
TYPE
  DAY = (SUNDAY, MONDAY, TUESDAY,
  WEDNESDAY, THURSDAY, FRIDAY,
  SATURDAY);
```

To make this a little clearer imagine the following:

```
TYPE
  INTEGER=(1,2,3,4,.....40965535);
```

In other words, defining your own data type is a convenient means of giving a name to an ordered sequence of known elements. To add the icing to the cake, a SET can be declared which is a collection of objects of the same type:

```
TYPE
  CAPS=SET OF 'A..Z';
```

Now we can have the following in a VAR list:

```
VAR
  LETTER : CAPS;
```

The last two items in the declaration section of a Pascal program are procedures and functions. Basically, a procedure is a sub-program (a sort of sub-routine), which contains other statements and which, functions are means of declaring new operations that Pascal does not have.

So far, the main program body has not been discussed. As you might have already guessed, Pascal programming requires a lot of thought. This is a good thing because people tend to program at the keyboard without thinking first. Thinking about what data types and variables you'll need will encourage thought, and a good program can be the final result.

### Pascal Syntax

The third section in a Pascal program is the main body of the program. Here, the first and last reserved words are **BEGIN** and **END**, with the final **END** followed by a full stop. I say "full" because there are normally more than one on. To under-

stand this better, here is a very simple Pascal program:

```
PROGRAM Greeting;
BEGIN
  WRITELN('Hello Your Commodore
  master!');
END
```

Notice that there is no declaration section. The two statements in our small example are the first line and **WRITELN**, statements are separated by a semi-colon. The exception to this case is the statement preceding and **END**:

```
PROGRAM Greeting;
BEGIN
  WRITELN('Hello');
  WRITELN('How are you today?');
END
```

In the example, the first statement after **BEGIN** ends with a semi-colon because the next line contains another statement. Now look at this example:

```
PROGRAM Count;
VAR
  I : INTEGER;
BEGIN
  WRITELN ('Watch this!');
  FOR I:=1 TO 10 DO
    WRITELN('Ying');
END
```

This simply prints "Watch this!" followed by "Ying" printed 10 times. However, if I wanted to print "Ying" followed by "Yang" on the next line a revised **BEGIN..END** has to be used:

```
PROGRAM YingYang;
VAR
  I : INTEGER;
BEGIN
  WRITELN('Watch this!');
  FOR I := 1 TO 10 DO
    BEGIN
      WRITELN('Ying');
      WRITELN('Yang');
    END;
  WRITELN('That's all folks!');
END
```

The two statements between the second **BEGIN..END** are considered as a compound statement (i.e. treated as one). This example also demonstrates the use of indentation. This is NOT compulsory, but it does make the program easier to read and



follow through logically. The normal rule of thumb is that when nesting starts (more BEGIN) you indent, and as the END is reached on the same column indentation occurs.

```
BEGIN
statements
BEGIN
statements
BEGIN
statements
END
END
END
```

## Procedures and Functions

Tables, one and two like the standard Pascal reserved words as well as pre-defined procedures and functions. There are not many, but they are pretty powerful. Pascal provides the features to define your own procedures and functions made up from existing procedures and functions.

In essence, procedures and functions are mini-programs. They may have their own declarations and program blocks just like the main program. Although similar, there are a couple of differences between the two. Procedures do not have to have a parameter passed to them, although it is possible to pass parameters to and from procedures. Functions on the other hand, must have a parameter and can be used in comparisons while procedures cannot.

Here is an example of a program that uses a procedure. All it does is print the numbers one to 10, 10 times:

```
PROGRAM Test;
VAR
  I : INTEGER;
PROCEDURE Count;
VAR
  J : INTEGER;
BEGIN
  FOR J:=1 TO 10 DO
    WRITE(' ');
  END;
BEGIN
  FOR I:=1 TO 10 DO
    Count
  END;
END
```

Notice that the procedure and the main program use a variable with the same name. This is OK, because variables are local to procedures and functions. Also note that the procedure 'Count' is called from the main program just like any other procedure (no line numbers like Basic).

A function is declared in a similar manner, although its operation is different:

```
PROGRAM Table;
VAR
  I : INTEGER;
FUNCTION Square (N : INTEGER) :
  INTEGER;
```

```
BEGIN
  N:=N*N;
END;
BEGIN
  FOR I:=1 TO 10 DO
    BEGIN
      WRITE(I);
      WRITE(Square(I));
    END;
  END
```

In the example, I have used the function 'Square' directly, but as with normal functions it is possible to pass the result to a variable.

## Control Structures

Commodore Basic is a bit limited as far as control structures are concerned. Pascal on the other hand, offers the programmer some very elegant means of controlling program flow.

One control structure which has already been covered in the examples is the FOR...DO loop, which is similar to the Basic FOR...NEXT:

```
FOR I:=1 TO 10 DO
  WRITE(N);
```

DOWNTO is used to reverse the loop:

```
FOR I:=10 TO 1 DO
  WRITE(N);
```

To include more than one statement within the loop boundaries, a compound statement is used:

```
FOR I:=1 TO 30 DO
  BEGIN
    Statement;
    Statement;
    "
    "
    "
    Statement
  END;
```

One last point about a FOR...DO loop is that unlike Basic, Pascal does not like the index (the variable after the FOR) to be altered within the loop itself.

Probably the major drawback with the FOR...DO type of loop is that you have to specify an end to the loop. That is, a FOR...DO loop must have fixed boundaries - even if passed by variables. Pascal offers a couple more control structures which are more flexible.

The first of these flexible control structures is REPEAT...UNTIL:

```
REPEAT
  A:=A+1;
  WRITE(N);
UNTIL A=10
```

Notice that A would have been initialised before entering the REPEAT loop. Also note that compound statements do not have to be used. Instead, statements are merely separated by the semi-colon. Can you see why this is the case?

You can see from the example that a boolean test is performed after the UNTIL. Any of the boolean operators can be used in this test including '<', '<=' and so on. The most important point worth remembering about the REPEAT...UNTIL loop is that the statement(s) within it will be performed at least once. This is because the test is done at the end of the loop.

The second 'even-flexible' control structure is the WHILE...DO loop which takes the following form:

```
WHILE level<level+1 DO
  BEGIN
    level:=level+1;
    "
  END;
```

Notice that the boolean test is performed before any statements are executed so that if the boolean test is false no statement will be executed. A final point to note is that, unlike the REPEAT...UNTIL loop, multiple statements must be treated as compound statements (i.e. with a BEGIN...END).

The final control structure to be considered is the CASE statement. This is used in situations where the number of alternatives is greater than two. It is best understood by example:

```
CASE month OF
  1:WRITE(N,'January');
  2:WRITE(N,'February');
  3:WRITE(N,'March');
  12:WRITE(N,'December');
END;
```

In the example the variable 'month' has a certain value. Depending on what that value is, perform a different action. That is exactly what the CASE statement does. If 'month' is equal to five, then 'May' will be printed, and so on.

## Oxford Pascal Oxford Computer Systems Disk

Not all compilers produce native machine code, some produce what is known as P-code. The version of Pascal from Oxford Computer Systems does just that. The result is an executable program that doesn't run very fast. However, it should be noted that there are numerous Pascal P-code compilers because they are easy to implement.

Although Oxford Pascal is quite expensive, you do get a lot for your money. First, it follows the Pascal standard almost to the letter. Because of this it

could be used as a serious Pascal training tool.

A major problem with disk-based compilers is the time it takes from writing the program to running it because of this, Oxford Pascal offers two methods of running Pascal programs.

The first method is to use the resident compiler: this is the default option when the compiler is first loaded. In this mode it is possible to write, amend, compile and run a Pascal program without having to access the disk drive. This allows learners to get their feet wet and generally muck about.

The second method of operation is disk mode. Here, a program has to be developed in the standard method, that is, entered, saved to disk, compiled and re-entered if any errors occur. This mode does offer numerous advantages though. Programs can be much larger as compilation is from disk. Other advantages include data file handling, the availability of the full compiler system, external procedures and so on. Object code can also be converted to run on external alone code.

Machine dependent features include a form of peek and poke, colour and sound and some graphic commands. Most notable of these is the window command that allows the screen to be split between the high-resolution graphic screen and the text screen.

#### Pascal 64

##### Oxphos 164

###### Topic

Pascal 64 and Oxford Pascal are so similar in many ways that most of the comments in these reviews apply to both packages.

As with Oxford Pascal, Pascal 64 offers an almost complete definition of Pascal as defined in the User Manual and Report for Work 4. However, it is however, a cassette version and does not produce stand alone code (i.e. does not have a disk mode).

The Pascal source is entered in an editor which is very similar to the standard C64 editor.

There are a few differences and additions. Namely, source code can be entered with indentation, that showing the program structure. Additions to the standard editor include commands such as search and replace, auto line numbering, remember, delete and so on.

The source program can be compiled with or without a listing, which can be directed to the printer. The object code can then be run and/or saved to tape that as stand alone programs cannot be produced, it is necessary to have the compiler in memory when running any object code.

Additions to the standard are similar to what Oxford Pascal offers. But the general purpose DRAW command in Pascal 64 is much faster (see benchmarks). Additional features include sprite handling, the ability to read a light-pen and the joystick

ports.

Both Oxford Pascal and Pascal 64 have common faults. First, both packages are slow (see benchmarks). In fact neither of the two compilers is that much faster than Basic. This is a great pity as one of the advantages of Pascal is its speed.

Other problems include flickering of the high-resolution graphics screen, but the biggest fault is that this screen is divided into 220 by 300 pixels, why not the full 1020?

#### Pascal 64

##### First Software

###### Disk

When I first heard about this package I thought, "Great, a true Pascal compiler." However, after using it I have mixed feelings. It's both very good and really awful.

The package consists of a disk and 70m page poorly written manual/user guide. The software was written by a German company called Data Studios and the rest of the manual and software have suffered through translation!

The compiler is very disk intensive and takes about three stages before a program can be run. The tasks involved in developing a program with this package are: write the source, compile the source, link the object code and save object code to disk. On the good side the compiler produces compact stand-alone code, but don't expect to achieve that in five minutes.

The biggest let-down with this package is the first stage of producing a program. There is no editor and it is necessary to enter the source code using the standard editor. Not only is this a bother (there are no extra editing commands), but it is a disaster. The reason for this is simple. Part of the "beauty" of Pascal is the ability to write programs that are indented. This makes the program look neat, easier to read and identify. Because Pascal 64 requires the source to be entered using the standard '64 editor no indentation is allowed (unless spaces are preceded by a ') which makes the source look ugly!

Another weird thing about the package is that all Pascal keywords must be followed by a space character. This is non standard, and the more experienced programmer will be frustrated by this, what the beginner will pick up a bad habit. Why a space is required, is beyond me.

Although the review so far has been negative, things get better when the source has been entered. Once the code has been compiled and linked a multi-line code file is produced and can be run stand-alone.

The compiler itself is the best of the three packages reviewed, offering more advanced features. These include comprehensive file handling, external procedure support, high-resolution plot,

sprite command etc.

An integer command enables faster execution if only integers are being handled. Another impressive feature is the ability to run a procedure as an interrupt. The constraints on this are rather limited, but it could be useful.

#### Table 1 — Pascal Reserved Words

Here is a list of Pascal reserved words. Note that this list is the required set of words and that some versions may have more — but not less.

AND, ARRAY, BEGIN, CASE, CONST, DIV, DO, DOWNTO, ELSE, END, FILE, FOR, FUNCTION, GOTO, IF, IN, LABEL, MOD, NIL, NOT, OR, ON, PACKED, PROCEDURE, PROGRAM, RECORD, REPEAT, SET, THEN, TO, TYPE, UNTIL, VAR, WHILE, WITH

#### Table 2 — Standard Functions and Procedures

Every version of Pascal should have the following functions and procedures (with the possible exception of NIM and DISPOSE). All implementations of Pascal on the 64 have more pre-defined procedures which deal with colour and so on. Note that the brackets indicate an argument.

ABS, ARS, PLT, READ, REARM, REARM, REWRITE, WRITE, WRITELN, ARS, ARCTANG, COS, EXP, LN, LOG, SIN, SQRT, SQR, EXP, EXP, EXP, DDD, CHR, ORD, BOA, NID, TRUNC, PACK, UNPACK, NEW, DISPOSE, FREE, NACC

#### Summary

If you intend to learn computer science then I would strongly recommend that you get a Pascal compiler for your 64. Pascal has a small amount of reserved words, yet many of the up and coming 16th generation languages incorporate Pascal-like structures.

As far as recommending one of the three packages I can only suggest this. If you wish to learn Pascal, then Oxford Pascal is by far the best. Even though 'Oxphos' Pascal 64 is very similar, the disk operation within Oxford Pascal is a real improvement. Oxford Pascal is a bit pricy, so Oxphos' Pascal 64 would be a very good second.

For the more experienced programmer I would recommend First Publishing's Pascal 64. Even though the user interface and disk system requirements are off-putting, I say this because the results are quite fast, and it does have a number of advanced features not found in the other packages.

David Janda is prepared to discuss this subject further through electronic mail. He can be contacted on the following

One-To-One 1418800

Faxtel: 97990977



**BORED WITH SHOOT 'EM UP SPACE INVADERS ?**

**then it's time to**

# VENTURE INTO THE UNKNOWN...

Enter the world of computer adventure. The adventures play several weeks to months, providing hours of stimulation and intrigue. You must use your imagination and logic to solve the mystery of each adventure. Top ends are revealed about each Sierra adventure contains over a hundred full-colour computer-generated images which lead you through decisions. You'll receive murder clues, your wit and logic, and perseverance for each Sierra adventure will prove a test to your imagination and logic.

## THE WIZARD AND THE PRINCESS

Become the hearty and intrepid adventurer who rescues the fair princess from the dreadful wizard. It's the fate of Lathander to his lowering castle beyond the great mountains. The most subtle and extraordinary the mystical power of the wizard's life. It begins on 1988 and 1989. Through time. Begin your adventure in the village of Tonia, but later, everything is not as it seems.

## ULYSSES AND THE GOLDEN FLEECE

Rescued by the gods, the Golden Fleece, legendary creature of ancient mythology, has been lost for many years. Every citizen must have attempted its recovery, but none have been successful in achieving it by any single day.

As Ulysses you are entrusted the task of retrieving the Golden Fleece and returning to the king. You set sail from Aegina to Greece and your voyage is under the watch of the sea, that hides your adventures, the wicked Gods, the old sea gods, Poseidon, the King of the Underworld and other deadly enemies. Legend becomes reality and the sea is not as it seems.

Get the perfect logo and then choose if it is to be yours!



## MISSION ASTEROID

A meteorite is about to happen... An asteroid of incredible size is headed for a direct collision with the earth. The problem is you could destroy it... unless something is done quickly.

It's your duty to fly the planet in a space station to allow its course. The time is of the essence. Use your wits and logic and your own logic to solve the problem. Be quick and careful...

Be fast and precise in time to solve the problem. Be quick and careful... the fate of the earth is in your hands now!



Sierra  
**ON-LINE INC**

*All American*  
**ADVENTURES**

All American Adventure Limited,  
Dept. 01, Parkway Industrial Estate,  
Barnes Road,  
Barnes, London W14 9NS.  
Telephone 0181 878 8888.  
Telex 940000

DESK COPY

**£14.95**

This month we begin  
 a new series by Steve  
 Carie. When you've  
 collected all three  
 parts you'll have a full  
 machine code  
 development system.

# MACH 1

THE AIM OF THIS SERIES OF programs is to provide the reader with a set of tools to aid in the production of 6801 machine code programs on a C64 suitably equipped with aTMM disk drive.

There are three separate programs: a Machine Language Monitor/Editor, a Macro Processor and an Assembler. The Monitor is essentially the control program from which the other two may be called upon to process source code written using the Editor.

The Macro Processor is not a full implementation since it does not handle conditional processing. However, it does allow sections of source code to be written as a "macro" with parameters for later inclusion in source code programs.

The Assembler is a two-pass type which produces an executable disk file which may be loaded into memory by using a simple Basic command:

```
LOAD "Filename".A1
```

The programs are disk-based since it was necessary to have the random access and reliability that a disk drive affords.

I will first deal with the Monitor/Editor program and then the Macro Processor and Assembler in turn.

## The Monitor

The four Basic listings, when loaded and RUN in sequence 1-4 will POK the complete Monitor/Editor program into memory. Each listing contains REM statements to show where the various sections of code begin and end. I suggest that you type in and save all four programs before you attempt to generate the code. A word of caution here; check

## Monitor Command Summary

Command	Description
your.acorn.gp	Input processor register values. The R command outputs a similar format allowing free use of the C64 screen editor.
*addr1,addr2,...,addrN	Input N consecutive byte values into memory starting at addr1. The M command outputs a similar format allowing free use of the C64 screen editor.
M	Initialise, performs a warm start.
M addr1,addr2	display contents of memory between addr1 and addr2.
R	display processor register values.
L	exit to Basic command mode.
G addr	restores processor register values and begins execution at addr.
T addr1,addr2,addr3	copy the contents of memory block addr1 - addr2 to a block starting at addr3.
F addr1,addr2,byt	fill the memory block addr1 - addr2 with the value byt.

Note that in the above commands, the value byte may be given in ASCII format by preceding the character by the apostrophe ( ' ). For example, T 3000,4000, A will fill the block 3000-4000 hex with the value 41 hex, the ASCII code for "A".

L filename	loads a file into memory at its original address.
S filename addr1,addr2	save the memory block addr1 - addr2 to a disk file with name filename. Filenames should not contain spaces.

## Monitor Command Summary — Multiple Letter Group

RESET	causes the machine to cold start
DIR	displays current disk directory
CAT	displays current disk directory
DEBR	displays disk drive status
INIT	initialises drive (disk I command)
CDMPACT	validates disk (disk I command)
EDBANK filename,JD format,disk offset,N command	validates disk (disk I command)
COPY filename,copies a file (disk C command)	
filename	
RENAMI filename	renames a file (disk R command)
filename	
SEARCH filename	deletes a file (disk S command)
HIS	sets default I/O mode to hex
DEC	sets default I/O mode to decimal
TTY	defaults output device = PRINTER
YDU	defaults output device = SCREEN
EDITOR	enters editor mode

your typing carefully at no check-out facility is provided.

When all four are ready to be RUN, load and execute them each in turn then give the following Basic direct command:

```
POKE143,POKE144,130:POKE41,130:POKE46,142:SAVE"MONITOR".A,1
```

Having done this, type:

```
POKE141,POKE142,NIM
```

You should now have a copy of the actual program on disk and are ready to test it out.

## Testing the Monitor

Enter the monitor by having the direct Basic command:

```
SYS 1020
```

You should be given the message:

```
COMMANDS: 64 UTILITY  

SERIES.  

MONITOR (EDITOR C.S. V1.1  

K) 985 S.D.C.
```

followed by a " " prompt and a flashing cursor. You are now in Monitor command mode. It is from this that you will command most of the functions of the Monitor/Editor and the other programs.

A summary of commands is given later but for now try the following commands:

Type DEBR followed by return, the disk status is given on screen in a form similar to:

```
DISK DR. 0, OK,0,0
```

where DISK DR. indicates that this is a disk status report. Machine reports are prefixed by MACHINE.

Now type DIR. The directory of the current disk will be displayed on screen in a format similar to that obtained by using the more usual Basic method LOAD "A".



Note that some commands, such as COPY and RENAME automatically produce a disk status report to show whether or not the command has been carried out successfully.

The formats of the disk commands SCRATCH, COPY etc. are the same as are given in the 1541 disk users manual. Note that in all cases, spaces, and not quotes, are required as a r/d. //denotes a r/d. For example, the command to save a block of memory to disk is S. Thus the format is:

```
S filename+addr1:addr2
and run
S "filename" addr1:addr2
```

Note type 1. This will perform a warm start into the monitor, if you type 2, the machine will reenter BASIC. To get back into the Monitor, simply type the SYS 10300 command again.

One thing to note is that the 6503 BRK vector is now set to jump into the monitor. This means that you may insert BRK instructions into your machine code programs to act as STOP for debugging purposes. To see the effect of this, type the following:

```
POKE128:SYS 2
```

A zero byte is the code for the 6503 BRK instruction. You will be given the entry message as before but this time it will be preceded by:

```
***BRK***
```

indicating that a BRK instruction was encountered during execution of a machine code program.

Any other words typed in are assumed to be "internal" commands (i.e. disk files which all sit in memory). The programs MACRO and ASSEMBLER are examples of these. When you want to call the Macro Processor for example, assuming you called the program MACRO, you simply type MACRO plus a couple of filenames and the Macro processor will be loaded and run. An external command is assumed to begin at address 1000 hex. Therefore, if you want to write a program to act as an external command, it should have a start address of

## Editor Commands

### Command

A

### Description

Auto line numbering (this is a useful feature since it does away with the need to type in line numbers. Two formats are valid. 1 A - start 10, increment 10 2 A10:20 - start 10, increment 20 Format 2 may of course use other numbers.

D

Delete lines (there are two

methods of line deletion: 1 The usual Basic method, type a line number

2 The D command. An example would be D20-80 which would delete lines 20 through to 80. Remember lines this simply renumbers the lines of a source code program. The format is the same as for the A command. Use program lines - the RUN/STOP key may be used to terminate the listing. If a line number is supplied, listing will commence from that line (e.g. L200 will begin listing at line 200).

M

Merge file - the format is Mfilename. The file named will be merged onto the end of the file currently in memory.

S

Scratch (delete) a file - this is similar to the Basic command DEL.

I

Initialise - reenter editor deleting any file currently in memory.

-

Exit editor - there are three formats for this command.

1\*

- : save file and exit.

2\*

- : save file without save.

3\*

filename : save file with a new name and exit.

program. Field two is the SYMBOL or LABEL field, field three is the INSTRUCTION or MACHINECODE field and field four is the OPERAND field.

The functions of these fields will become clearer as we go on to explain the Macro Processor and Assembler.

A typical Editor screen layout is shown below. The top line is for guidance only and is not an screen.

```
FIELD 1 FIELD 2 FIELD 3 FIELD 4
1000 LABEL 1000 DELAY
1005      LDA     $40
1008      ASL     A
9000.....
```

If you're wondering how you are going to keep all the field entries in the right places then read on. When in edit mode, the function key F1 is used as a TAB key. When pressed, it will advance the cursor from its present position to the start of the next field. For example, suppose the line 1000 in the above example was being typed in. The line number would be typed in and the F1 key pressed. The cursor would be advanced to the start of field 2 - the L of LABEL1. Pressing F1 again would advance the cursor to the start of field 3 - the 1 of 1000 and so on.

This feature coupled with an auto line numbering facility makes for easy code entry.

Other facilities include line renumbering, block line delete, source code file merging and single line delete. A program is entered in a similar fashion to a Basic program with the line numbers. These numbers are for editing purposes only and do not affect the final object code generated by the assembler. All the normal Commodore screen editing facilities are supported by this editor.

## Entering and Leaving the Editor

While in Monitor command mode, enter the command EDITDE. The message EDITDE ENTER FILENAME will appear. You should enter the name of the file you wish to edit. If the file exists on the current disk the editor will read it in and you will be able to work on it. If however the file does not exist on the current disk, the message

1000 hex. At the end of execution, a JMP \$1200 instruction will cause a re-entry to the monitor.

## The Editor

The Editor is provided so as to allow preparation of source code in the correct format for the Macro Processor and Assembler. The basic idea behind the Editor screen format is that source code symbols, operands and mnemonics all have defined areas in a source line. These areas are known as FIELDS and both the Macro Processor and

Assembler expect to find their source code arranged in these fields.

On a normal C64 screen, there are 25 rows of 40 characters. The editor arranges the screen such that there are four fields of 10 characters each numbered one to four starting at the left side. Since the normal screen editor will accept up to 80 characters (i.e. two lines of text, the last field - four - may extend for up to a maximum of 58 characters. Field one is only used by the Editor to accept and display line numbers. It is not actually stored in the final source

NEW FILE will be printed. In both cases, a flashing cursor (or prompt) will signify Editor command mode.

Before going on to explain the Editor commands, a word or two about leaving it. The name entered to the filename, as stored by the editor. At the end of the editing session the user may simply enter the command "E" and the file is saved back to disk. An auto disk validate is carried out.

This is done to prevent problems associated with the "541" 8" bug. In some cases it may take a long time to validate a disk especially if it is

getting dull. Some of you may find this a bit of a nuisance but I personally prefer it to a corrupted disk. If you want to do away with this facility, refer to listing areas which contain details on removing it. See Editor Commands.

When you have the Macro Processor and the Assembler complete, I will give some programming examples to let you get used to the operation of the Editor and its formats. Next month I will give listings of the Macro Processor and a description of Macros and Macro processing.

## Monitor Part 1

```

150 DATA 78,79,84,78,82,32,88,88,7
3,84,78,88,82,78,48,83
170 DATA 48,38,86,48,48,50,13,10,7
0,87,11,32,48,87,88,83
180 DATA 32,83,48,88,48,87,48,13,1
0,13,10,0,13,13,10,42
190 DATA 42,32,88,88,88,88,75,32,6
9,78,84,82,83,32,42,42
200 DATA 13,10,0,82,78,75,48,13,10
,0,13,10,78,88,87,32
210 DATA 70,73,78,88,13,10,13,10,0
,13,10,88,88,73,84,78
220 DATA 82,13,10,0,13,10,42,42,88
,88,88,78,82,42,42,13
230 DATA 10,0,13,10,77,88,87,78,73
,78,88,32,88,32,0,88
240 DATA 73,83,78,32,88,82,32,88,3
2,0,13,10,32,32,32,32
250 DATA 80,87,32,32,83,82,32,88,6
7,32,88,82,32,88,88,32
260 DATA 83,80,13,10,88,43,32,0,13
,10,88,78,84,83,82
284 REM *****
****
285 REM ** INPUT EVALUATION SUBRT.
**
286 REM *****
****
270 DATA 32,78,73,78,83,78,88,77,6
3,32,0,238,84,48,88,255,32
290 DATA 138,173,78,247,183,201,48
,144,18,201,71,176,14,201,88,178
295 DATA 3,41,16,86,201,85,144,3,2
33,88,88,78,78,178,32,121
300 DATA 0,32,183,131,72,32,116,0,
32,24,130,144,80,104,10,10
310 DATA 10,10,133,2,32,121,0,32,1
81,131,6,2,133,2,78,116
320 DATA 0,104,133,2,86,32,206,131
,188,2,133,20,32,121,0,32
330 DATA 24,130,178,1,86,32,206,13
1,188,20,133,21,188,2,133,20
340 DATA 86,32,27,130,144,1,36,201
,88,144,8,201,71,178,2,86
350 DATA 86,24,38,201,48,144,250,2
01,88,178,248,88,88,32,13,177
360 DATA 144,84,96,183,0,133,80,1
83,21,32,121,0,201,38,208,8
370 DATA 32,116,0,133,20,88,173,17
7,2,84,0,78,245,131,32,121
380 DATA 0,78,173,131
1000 FORS=33280 TO 33876
1010 READA,POKES,A
1020 NEXT
1030 PRINT"FINISHED"

```

## Monitor Part 1

```

5 REM ** JUMP TABLE **
6 REM *****
10 DATA 78,114,137,76,152,135,76,1
31,135,76,188,134,76,138,132,76
20 DATA 158,132,76,84,132,76,51,13
2,78,17,132,76,35,132,76,45
30 DATA 132,76,103,135,76,18,135,7
6,18,141
34 REM *****
****
35 REM ** COMMAND & MESSAGE TABLES
**
36 REM *****
****
40 DATA 77,0,82,0,71,0,84,0,70,0,7
6,0,83,0,43,0,42,0,73,0,88,0
50 DATA 82,88,83,89,84,0,88,73,82,
0,87,85,84,0,83,87
60 DATA 82,85,84,87,72,0,70,78,82,
77,85,84,0,73,78,73
70 DATA 84,0,87,79,80,83,0,82,83,7
8,85,77,83,0,84,89
80 DATA 80,88,0,87,78,77,80,86,87,
84,0,88,89,82,82,0
90 DATA 72,89,88,0,88,89,87,0,88,8
8,73,84,78,82,0,86
100 DATA 88,86,0,84,84,88,0,256,83
,133,45,133,114,134,50,134
110 DATA 248,133,6,130,3,130,208,1
33,171,133,114,137,158,131,258,252
120 DATA 183,138,183,136,57,135,80
,136,101,135,83,135,86,135,48,136
130 DATA 88,136,9,133,223,135,226,
135,91,139,150,136,156,136,13,10
140 DATA 87,79,77,77,73,88,73,82,6
9,32,54,52,32,85,84,73
150 DATA 78,73,84,88,32,83,88,82,7
3,68,83,46,13,10,77,78

```

## Monitor Part 2

```

5 REM ** OUTPUT SUBROUTINES **
6 REM *****
70 DATA 100,32,76,210,255,168,171,7
8,210,255,168,13,32,210,255,168
90 DATA 10,76,210,255,168,147,76,2
10,255,72,74,74,74,32,128
110 DATA 132,32,210,255,104,41,15,3
2,129,132,76,210,255,201,10,144
120 DATA 3,105,54,56,3,48,52,173,17
4,2,240,8,152,32,103,132
130 DATA 128,76,103,132,152,76,208,
189,78,173,174,2,240,4,104,76
140 DATA 108,132,104,174,253,132,25
4,170,168,0,32,203,183,164,254,188
150 DATA 253,26,163,6,160,5,162,15,
140,32,208,143,174,2,142,33,208,55
160 REM *****
170 REM ** ERROR HANDLING **
180 REM *****
90 DATA 138,18,3,108,2,3,72,32,180
,138,168,88,180,131
100 DATA 32,30,171,104,10,170,168,1
28,133,157,168,38,163,133,34,189
110 DATA 35,163,133,35,32,204,253,
188,0,133,19,32,220,170,180,0
120 DATA 177,34,72,41,127,32,71,17
1,200,104,16,244,32,122,165,32
130 DATA 18,130,168,2,3,32,18,130,
183,111,160,131,32,30,171,162
140 DATA 15,32,188,253,32,228,255,
32,210,255,201,13,208,248,162,0
150 DATA 32,188,255,32,18,130,108,
2,3
160 REM *****
170 REM ** MONITOR O.S. SUBRT.
180 REM *****
190 DATA 183,122,160,131,32,30,171
190 DATA 172,168,2,174,167,2,32,12
,150,162,0,32,83,132,189,168
190 DATA 2,32,15,130,232,224,5,208
,242,32,18,130,108,2,3,32
170 DATA 21,130,168,20,133,251,165
,21,133,252,32,253,174,32,21,130
180 DATA 32,18,130,168,62,32,210,2
55,163,42,32,210,255,32,84,132
190 DATA 184,252,165,251,32,12,130
,160,0,32,88,132,177,251,32,15
200 DATA 130,200,152,8,174,243,32,
225,255,240,22,152,24,101,251,133
210 DATA 251,165,252,105,0,133,252
,187,21,144,157,165,251,157,20,144
220 DATA 151,32,18,130,108,2,3,32,
21,130,168,20,133,251,165,21
230 DATA 133,252,160,0,132,88,32,2
23,174,32,21,130,164,90,165,20
240 DATA 145,251,200,132,90,152,8,
144,225,108,2,3,32,21,130,165
250 DATA 20,168,21,174,167,2,142,1
68,2,162,0,134,253,32,233,174
260 DATA 32,21,130,165,20,165,253,
157,168,2,232,134,253,224,5,144
270 DATA 234,108,2,3,32,21,130,165
,20,168,21,133,251,134,252,32
280 DATA 253,174,32,21,130,165,20,
165,21,133,253,134,254,32,253,174
290 DATA 32,21,130,160,0,165,20,14
5,251,230,251,208,2,230,252,165
300 DATA 251,157,253,208,240,165,2
52,157,254,208,234,108,2,3,32,21
310 DATA 130,168,20,168,21,133,251
,134,252,32,253,174,32,21,130,165
320 DATA 20,168,21,133,253,134,254
,32,253,174,32,21,130,160,0,177
330 DATA 251,145,20,230,251,208,2,
230,252,230,20,208,2,230,21,165
340 DATA 251,157,253,208,234,165,2
52,157,254,208,228,108,2,3,32,121
350 DATA 0,240,13,32,21,130,165,20
,166,21,141,167,2,142,168,2
360 DATA 174,173,2,154,173,168,2,7
2,173,167,2,72,173,168,2,72
370 DATA 173,170,2,174,171,2,172,1
72,2,64,169,169,162,227,141,0
380 DATA 3,142,1,3,163,131,162,164
,141,2,3,142,3,3,174,173
390 DATA 2,124,163,128,133,157,76,
123,227
400 REM *****
1000 FORS=33876 TO 34482
1010 READA=POKES,A
1020 NEXT
1030 PRINT"FINISHED"

```

Monitor Part 3

```

5 REM *****
***
6 REM 140 SUBROUTINES
7 REM *****
**
10 DATA 32,121,0,162,0,184,122,132
,93,165,0,2,240,15,201,32
20 DATA 240,11,200,232,224,30,144,
241,162,23,108,0,3,132,122,224
30 DATA 0,208,5,162,8,108,0,3,164,
53,56,32,5,132,134,2
40 DATA 152,170,168,58,202,167,0,2
,202,165,24,157,0,2,134,84
50 DATA 162,2,232,232,134,2,165,24
,132,50,162,2,132,51,184,2
60 DATA 162,0,145,20,56,162,15,162
,8,160,15,32,165,255,162,0
70 DATA 32,162,255,76,132,255,162,
15,76,165,255,162,15,32,201,255
80 DATA 165,24,160,2,32,30,171,162
,0,76,201,255,162,83,44,162
90 DATA 76,44,162,67,44,162,82,132
,54,32,232,134,32,40,132,76
100 DATA 8,132,141,70,2,162,0,141,
71,2,162,1,132,2,162,70
110 DATA 132,54,76,40,132,162,86,4
,4,162,73,32,72,132,162,2,3
120 DATA 32,2,132,132,72,152,170,1
04,162,2,76,162,255,162,1,162
130 DATA 8,160,2,76,162,255,32,32,
130,32,122,132,162,0,32,212
140 DATA 255,171,2,162,4,108,0,2,1
08,2,3,32,33,130,32,122
150 DATA 132,162,32,32,255,174,32,
21,130,162,20,162,21,72,132,72
160 DATA 32,252,174,32,21,130,162,
20,184,21,164,132,21,104,132,20
170 DATA 162,1,43,254,132,1,162,20
,32,212,252,8,162,1,8,1
180 DATA 132,1,40,144,5,162,24,102
,0,3,32,162,252,208,246,162
190 DATA 2,3,162,252,44,162,0,141,
174,2,162,51,160,131,32,30
200 DATA 171,102,2,3,32,207,252,72
,32,162,252,41,54,208,2,104
210 DATA 26,104,162,0,32,162,252,1
62,2,32,192,252,102,2,3,162
220 DATA 2,162,8,160,2,32,162,252,
32,32,130,24,32,132,252,176
230 DATA 11,32,162,252,208,0,56,16
2,2,76,162,252,162,4,108,0
240 DATA 3,32,12,132,32,204,252,32
,131,132,32,241,132,32,241,132
250 DATA 32,241,132,32,241,132,32,
241,132,32,241,132,160,0,32,241
260 DATA 132,48,231,153,0,2,200,20
1,0,208,242,32,204,252,32,180
270 DATA 136,162,0,160,2,32,30,171
,32,18,132,32,204,252,32,131
280 DATA 136,32,222,252,240,140,16
0,0,76,81,136,173,172,2,202,1
290 DATA 26,162,200,76,201,252,162
,2,76,132,252,162,0,142,172,2
300 DATA 32,201,252,162,20,32,122
,252,96,32,136,132,102,2,3
310 DATA 162,202,141,172,2,162
311 REM *****
****
312 REM THIS IS THE DEVICE NUMBER
FOR
313 REM THE PRINTER
314 REM ***
315 DATA 4
316 REM ***
317 REM CHANGE TO WHATEVER YOU NEED
D
318 REM *****
****
319 DATA 162,1,32,162,252,162,0,32
,162,252
320 DATA 162,4,141,147,2,32,162,25
2,162,200,32,201,252,102,2,3
330 DATA 32,132,132,162,32,132,231
,162,48,132,252,162,2,162,251,160
340 DATA 0,32,162,252,162,1,162,8,
160,0,32,162,252,32,132,252
350 DATA 147,10,72,162,184,32,132,
252,104,76,2,132,162,2,132,162
360 DATA 162,184,32,132,252,32,207
,252,132,87,32,162,252,208,102,32
370 DATA 207,252,132,88,32,162,252
,208,96,164,162,132,202,224,132,12
3
380 DATA 32,207,252,72,32,162,252,
170,104,224,0,208,75,164,162,132
390 DATA 80,176,62,153,0,2,170,210
,4,230,162,208,227,162,0,32
400 DATA 162,252,162,87,162,88,32,
202,162,162,32,32,210,252,160,0
410 DATA 162,0,2,240,8,32,210,252,
202,202,242,32,12,130,162,0
420 DATA 32,152,252,32,252,252,240
,16,32,222,252,201,32,208,5,32
430 DATA 222,252,240,251,160,2,208
,164,162,0,32,132,252,162,184,32
440 DATA 162,252,76,2,132

```

## Monitor Part 3

```

444 REM *****
***
446 REM ENTRY POINT/MAIN LOOP
448 REM *****
***
448 DATA 188,150,182,137,141,82,3,
142,23,3,185
450 DATA 188,162,132,141,0,3,142,1
,1,183,219,162,137,141,2,3
460 DATA 142,3,3,234,234,234,76,18
8,137,104,141,172,2,104,141,171
470 DATA 2,104,141,170,2,104,141,1
63,2,104,106,255,141,167,2,104
480 DATA 105,255,141,168,2,168,28
160,131,32,30,171,76,114,137,32
490 DATA 182,132,168,1,141,171,2,1
68,139,162,0,134,51,134,55,133
500 DATA 52,133,56,162,206,160,130
,32,30,171,188,142,173,2,32,182
510 DATA 132,163,0,133,157,32,36,1
35,162,52,32,210,255,174,173,2
520 DATA 154,32,56,162,134,122,132
,123,32,115,0,201,62,240,240,201
530 DATA 0,240,219,166,122,134,93
,160,0,122,2,185,42,130,201,255
540 DATA 240,65,201,0,210,65,221,0
,2,208,4,232,200,208,236,185
550 DATA 42,130,240,3,200,208,240
,200,166,122,230,2,76,8,188,188
560 DATA 0,2,32,18,177,144,6,186,4
2,130,76,18,138,166,2,10
570 DATA 134,122,170,188,162,130,1
33,20,183,153,130,133,21,32,18,136
580 DATA 108,20,0,165,93,133,127,1
62,2,133,123,32,18,136,32,32
590 DATA 130,32,122,135,24,168,0,3
2,213,255,178,6,32,36,136,76
600 DATA 0,144,168,4,108,0,3
1000 FOR S=34453 TO 35413
1010 READ A:POKEA,A
1020 NEXT
1030 PRINT"FINISHED"

```

## Monitor Part 4

```

5 REM *****
***
6 REM EDITOR SUBROUTINES 1
7 REM *****
***
10 DATA 162,10,160,4,185,0,2,232,2
99,153,251,1,185,251,1,208
20 DATA 243,153,253,1,188,123,169
,258,133,122,56,120,163,116,162,138
30 DATA 141,4,3,142,5,3,168,240,16
2,138,141,2,3,142,3,3
40 DATA 162,255,133,125,169,104,16
2,140,141,20,3,142,21,3,169,219
50 DATA 162,141,141,0,3,142,1,3,88
,26,120,162,124,162,165,141
60 DATA 4,3,142,5,3,168,32,133,128
,168,219,162,137,141,2,3
70 DATA 142,3,3,168,48,162,234,141
,29,3,142,21,3,169,198,162
80 DATA 132,141,0,3,142,1,3,88,56
84 REM *****
***
85 REM ** EDITOR MAIN LOOP
86 REM *****
***
88 DATA 108,2,3,32,36,136,32
90 DATA 18,136,32,86,165,134,122,1
32,123,32,115,0,170,240,237,182
100 DATA 255,134,58,144,68,201,42
,208,3,76,252,139,201,76,208,3
110 DATA 78,88,141,201,82,208,3,76
,241,140,201,73,208,5,32,130
120 DATA 138,78,51,138,201,68,208
,3,76,168,140,201,83,208,5,32
130 DATA 80,140,108,2,3,201,77,208
,3,76,178,141,201,65,208,6
140 DATA 32,228,141,108,2,3,108,0
,3,32,107,168,32,123,0,208
150 DATA 3,76,230,140,76,168,184,3
2,136,136,162,73,160,131,32,30
160 DATA 171,32,80,140,168,0,141,1
74,2,162,152,160,131,32,30,171
170 DATA 32,56,165,134,122,132,123
,32,115,0,170,208,3,76,109,188
180 DATA 160,0,186,170,131,153,178
,2,200,201,255,208,245,32,33,130
190 DATA 185,183,201,20,144,5,162
,23,168,0,3,160,0,162,3,177
200 DATA 187,153,180,2,200,232,156
,183,144,245,142,176,2,165,160,160
210 DATA 2,133,167,132,188,202,202
,202,134,163,32,162,130,32,143,138
220 DATA 183,0,32,213,255,144,13,3
2,18,130,165,58,160,131,32,30

```

## Algorithm Part 4

```

230 DATA 171,76,101,139,92,19,130,
32,229,139,32,33,165,108,2,9
240 DATA 169,256,160,1,175,43,32,5
1,165,165,34,24,215,105,2,139
250 DATA 45,165,35,105,0,139,46,96
,160,1,177,43,209,3,76,66
260 DATA 140,32,229,139,32,115,0,2
40,13,201,33,240,45,32,33,130
270 DATA 32,132,132,76,99,140,32,1
82,136,173,176,2,162,177,160,2
280 DATA 32,189,256,166,45,164,46,
169,43,32,216,233,147,3,76,219
290 DATA 135,32,183,253,240,3,76,2
10,136
291 REM *****
***
292 REM TO REMOVE AUTO-VALIDATE, R
EPLACE LINE 295 WITH THIS
293 REM 295 DATA 234,234,234,234,2
34
294 REM *****
***
295 DATA 169,96,32,79,135
296 REM *****
****
300 DATA 32,130,139,169,1,141,174,
2,32,18,130,76,114,137
304 REM *****
****
305 REM ** EDITOR SUBROUTINES 2
306 REM *****
****
310 DATA 169,0,169,145,43,200,146,
43,165,43,24,106,2,139,45,165,44,1
06,0,133
320 DATA 46,76,99,166,166,137,201,
64,209,3,76,45,234,201,4,209
330 DATA 249,165,159,209,246,66,32
,240,255,152,56,233,10,176,252,73
340 DATA 255,105,1,170,236,160,0,2
02,240,3,163,29,153,118,2,200
350 DATA 76,139,140,132,199,162,96
,160,256,136,208,253,202,209,248,7
6
360 DATA 49,224,32,115,0,32,21,130
,32,15,166,144,61,165,95,72
370 DATA 165,96,72,163,45,32,256,1
74,32,21,130,32,18,166,144,42
380 DATA 160,1,177,95,170,136,177,
95,168,104,139,96,104,133,95,152
390 DATA 160,0,146,96,200,136,146,
96,200,177,95,139,20,200,177,95
400 DATA 133,21,169,0,141,0,2,76,1
64,164,76,227,168,32,247,140
410 DATA 108,2,3,32,115,0,240,20,3
2,21,130,165,20,133,251,165
420 DATA 21,133,252,32,253,174,32,
21,130,76,28,141,169,10,162,0
430 DATA 133,251,133,20,134,252,13
4,21,165,43,166,44,133,253,134,254
440 DATA 160,1,177,253,208,4,32,18
,130,56,160,3,165,259,145,253
450 DATA 136,165,251,146,253,136,1
77,253,170,136,177,259,133,253,134
,254
460 DATA 165,251,24,101,20,130,251
,165,252,101,21,133,252,80,209,162
470 DATA 15,108,0,3,32,115,0,240,3
,32,21,130,32,15,166,76
480 DATA 110,141,165,43,166,44,133
,55,134,96,160,1,177,95,208,6
490 DATA 32,18,130,108,2,3,160,2,1
77,95,170,200,177,95,32,205
500 DATA 189,66,32,240,255,160,10,
24,32,240,255,160,4,177,95,240
510 DATA 8,32,210,256,200,208,248,
32,18,130,160,1,177,95,170,136
520 DATA 177,95,133,95,134,56,32,2
25,253,240,197,76,110,141,32,115
530 DATA 0,32,33,130,169,1,162,8,1
60,0,32,186,253,165,45,56
540 DATA 233,2,170,165,46,233,0,16
8,169,0,24,32,213,253,176,3
550 DATA 76,216,139,108,0,3,169,94
,160,131,32,30,171,109,2,9
560 DATA 32,115,0,208,13,169,10,16
2,0,133,251,134,252,133,253,76
570 DATA 11,142,32,21,130,32,253,1
74,165,20,133,251,165,21,133,252
580 DATA 32,21,130,165,20,133,253,
165,21,162,162,141,2,3,142,3
590 DATA 3,173,0,2,240,46,165,251,
165,252,32,95,142,169,11,133
600 DATA 159,160,0,169,29,153,113,
2,200,152,11,144,246,169,0,2
610 DATA 157,119,2,209,16,247,24,1
65,251,101,253,133,251,144,2,230
620 DATA 252,76,240,136,169,240,16
2,139,141,2,3,142,3,1,108,2
630 DATA 3,134,93,133,20,162,147,5
0,32,73,169,96,223,169,32,133
640 DATA 160,32,166,162,162,0,169,
0,1,157,0,2,240,3,232,208
650 DATA 245,96
1000 FOR B=35444032640
1010 READ:POKEB,A
1020 NEXT
1030 PRINT"FINISHED"

```



This month Joe Nicholson continues our C-16 series with an explanation of programmable characters.

# PROGRAMMING THE C16

## Programmable characters

IT IS POSSIBLE TO USE CUSTOM DESIGNED character sets on the C-16, even though Basic 5.5 does not have supporting commands.

To understand how this is done, one must first examine how the normal character set is stored. Data for the character shapes is stored in the ROM from address \$D000 to address \$D7FF (\$D200 to \$D2FF) hold the data for character set 1 (upper case letters and graphics). Locations \$D400 to \$D4FF (\$D420 to \$D42F) hold the data for character set 2 (upper and lower case letters). Each of the two character sets takes up 1K of memory. Characters with screen codes between 128 and 255 are reversed images of codes 0 to 127 and are therefore not stored in memory.

The characters are stored in the order shown in the screen display codes in Appendix I of the C-16 User Manual. For set 1, the first character is therefore "0". This is stored in eight bytes, one byte per pixel line (eight dots) of the screen display. Each byte of the character "0" contains the eight bits needed for each row of the character, viewed in binary form (one for on, zero for off). The left-most bit of the row is the "0th" bit, the second to the left is the "1st" bit and so on to the "7th" bit on the far right. The "0" sign is therefore stored as shown in Figure 1.

For a character set to be created in RAM, space must first be made available in which to put the character set. Assuming the high-resolution screen will not be used in conjunction with programmable characters, the top 1K of RAM (\$5100-\$5200) is the most convenient. This is done by entering the "highest address used by Basic" pointer (\$5-06) and the "bottom of string storage" pointer (\$5-52) down 1K from the top of RAM (see last month's article: The Memory map and hardware/monitor machine code). Type:

```
POKE $5100,$5200-1024
```

As "CLK" is used this should be done at the beginning of the program.

Assuming you don't want to redefine all 128 characters of the new character set, you need first to move one of the ROM

character sets down into the 1K block. This can be done easily by entering the MONITOR and typing:

```
T D000 D1FF $C00 — for set 1, or
T D400 D7FF $C00 — for set 2.
```

Then enter "X" to leave the Monitor.

To move the character set down into a Basic program is more difficult. A FOR-NEXT loop takes over 12 seconds, so I've written a short machine code routine which does the task almost instantly. The program is completely relocatable, i.e. it will work wherever it is stored in memory. The Start, End and Length values may be altered as desired for different applications. Figure 2 shows an assembler listing of the routine using the C-16

Assembler published in the June edition of Four Commodore. It is positioned in a free space below Basic at \$400 hex, 1536 decimal (see last month's "Where to store machine code"). To execute the routine from Basic, type: SYS 1536.

To make your character set the current one, you must first disable the Shift+Commodore Key with POKEV (\$C00-0), and then set the "Character data base address" pointer (\$4-20) as desired. Sixteen to seven of the high byte of the upper six bits of the high byte of the character set address. This enables the character set to start at any multiple of 16. We are using the address \$C7000 to \$C8000, so the number entered is \$C7000 (decimal).

To specify that the character set will be

```
START: 10000 1$H$FT MEMORY
10010 0R0 M000
10015 1
10020 1:START READING FROM
10030 LDA #0
10040 STA $00
10050 LDA #1000
10060 STA $01
10070 1
10080 1:START WRITING TO
10090 LDY #0
10100 STA $02
10110 LDA #3C
10120 STA $03
10130 1
10140 1:NUMBER OF BLOCKS TO
10150 1:BE MOVED (4 FOR 1K)
10160 LDX #4
10170 1
10180 1:PERFORM MOVE
10190 1:1 LDY #0
10200 1:2 LDR ($00),Y
10210 STA ($02),Y
10220 INY
10230 BNC R:1:2
10240 JNC $01
10250 JNC $03
10260 DEY
10270 BVE R:1:1
10280 RTS
>> OK.
```

Figure 1.

Address	Hex	Binary	Image
hex			
\$C000	\$1C	00111100	****
\$C001	\$66	01100110	****
\$C002	\$66	01101110	****
\$C003	\$66	01101110	****
\$C004	\$60	01100000	**
\$C005	\$60	01100000	**
\$C006	\$1C	00111100	****
\$C007	\$60	00000000	

Figure 2. Shift memory routine.





accessed from RAM as opposed to ROM. Bit two of address 60200 must be set. As it is important that the other bits at that address should remain unchanged, a line like this should be used:

```
POKE 60200,PEEK(60200) AND 255
```

To specify "character set in RAM". To get back to the normal ROM character set,

the following POKEs should be entered:  
 POKE 60200,255  
 POKE 60200,PEEK(60200) OR 4

Whenever an error is encountered, the ROM/RAM select bit is reset back to ROM, creating havoc on the screen if the "Character data base address" pointer is not pointing to the ROM character set.

This means that editing should always be done in normal (ROM) character mode.

Also, it is a good idea to put the "get back to the normal character set" commands as the destination of a TRAP command to stop this happening (see page 141 of the User Manual). Remember, however, that the TRAP command must come after the CLR command used when lowering the top of RAM.

## Entering programmable characters

The address of the character in RAM can be found as follows:

Address = Base address + (screen code \* 8)

The eight bytes for each character can be read into memory by a simple FOR-NEXT loop, with the numbers stored in DATA statements in line 150-170. It then calls this routine. The routine shifts the ROM upper case character set down into the new RAM area.

Line 180 shifts the top of memory down 16 for the new character set.

Line 190 PEEKs into memory the machine code routine \$M07-\$M0E held in the DATA statements in line 150-170. It then calls this routine. The routine shifts the ROM upper case character set down into the new RAM area.

Line 200 \$BASE in the Programmable character data stored in lines 10000-10080 into the start of the new character set (first character to be defined is '0', then 'A', then 'M' etc.).

Line 210 - PRINT CHR\$(0) - disables the ability to change character sets with the Shift+Comma key. As only one character set has been redefined, this disables the ability to change to a garbage character set.

Line 220 jumps to TRAP mode to Line 250 and jumps to the Demonstration (line 1000).

Line 250-260 The TRAP routine.

Line 270 prints the error and the line number on which the error occurred.

Line 280 line changes the \$BASE address, so point to the ROM character set, then prints "character set to be taken from ROM", then FNVD.

Line 1000 defines the colours and clears the screen.

Lines 1010-1030 print the border.

Line 1040 prints your base.

Line 1050 prints the scores.

Lines 1060-1070 move aliens right.

Lines 1080-1090 move aliens right.

Line 1100 performs this spectacular feat of imagination all over again.

Lines 10000-10080 the programmable characters.

Lines 10090-10090 the six programmable characters needed for the alien.

Line 10000 the border character.

Line 10070-10080 the base.

```

10 REM PROGRAMMABLE GRAPHICS DEMO
100 POKE255,59:POKE52,59:CLR
110 FORA=1535TO1570:READ:POKEA,B:HEXT:5
  YS1535
120 DATA100,0,153,200,100,200,133,200,10
  5,0,133,210
130 DATA100,00,133,211,102,4,100,0,177,2
  00,145,210
140 DATA200,200,249,230,200,230,211,202,
  200,240,20
150 RESTORE10000:FORA=0TO7:READB:POKE15
  300+A,B:HEXT
160 PRINTCHR$(A):POKE65200,60:POKE65200,
  PEEK15320+4*0251
170 TRAP250:GOTO1000
180 PRINTERRCHR$(0),CLR
190 POKE65200,200:POKE65200,PEEK15320+10
  041END
10000 COLOR,1:COLOR4,1:COLOR1,0,0:SCHELR
10100 PRINT"FFFFFFFFFFFFFFFFFFFFFFFFFFFF
  FFFFFFFFFF":
10200 FORA=1TO32:PRINT"0"TAB(35)"0"||HEXT
10300 PRINT"FFFFFFFFFFFFFFFFFFFFFFFFFFFF
  FFFFFFFFFF"
10400 COLOR1,2,0:CHAR,10,31,"001"
10500 COLOR1,2,4:CHAR,8,34,"0015423
  00000000"
11000 FORB=3TO12:FORA=1TO8:COLOR1,A+1,4
11100 CHAR,B,1A*3+1,"000 000 000 000
  000"
11200 CHAR,B,1A*3+1,"000 000 000 000
  000"
11300 FORC=1TO100:HEXT:HEXT:HEXT
11400 FORB=12TO32STEP-1:FORA=1TO8:COLOR1,A
  +1,4
12100 CHAR,B,1A*3+1,"000 000 000 000
  000"
12200 CHAR,B,1A*3+1,"000 000 000 000 0
  00"
12300 FORC=1TO100:HEXT:HEXT:HEXT
12400 80TO1000
100000 DATA32,00,152,153,191,255,255,253
100100 DATA0,0,0,155,231,231,255,01
100200 DATA4,0,2,249,253,255,255,03
100300 DATA253,252,255,255,121,03,1,1
100400 DATA100,24,255,00,231,255,190,150
100500 DATA191,03,255,255,150,252,130,130
100600 DATA255,130,191,101,105,100,120,35
  5
100700 DATA0,0,0,3,10,31,03,355
100800 DATA04,34,120,250,255,255,255,255
100900 DATA0,0,0,192,240,240,252,255

```

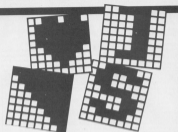
Figure 3. Programmable graphics demo

Listings will be much easier to enter with our new system.

COMMODORE LISTINGS ARE RATHER well known for the horrible little black blobs that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to find on the Commodore keyboard.

For this reason your Commodore started to precede any control character with a REM statement on the previous line that explained exactly what the black blob was meant to be. Unfortunately the graphics characters were not documented and these will cause some confusion. For this reason we are starting to use a new method for marking the control and graphic characters in our listings.

In future all control and graphic commands will be replaced by mnemonic within square brackets. This mnemonic is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.



# LISTINGS

Any character that is accessed by pressing shift and a letter will be printed as [< LETTER>]

[< A>  
[< C>

shift & A  
shift & C

Any character that is accessed by pressing the Commodore key and a letter will be printed as [< LETTER>]

[< A>  
[< C>

Commodore & A  
Commodore & C

[< F>

Any control key will be printed out as a number. For example [001]. Control codes are accessed by pressing the CTRL and a letter at the same time ([001] is CTRL & A, [002] is CTRL & B etc.). See the manual for more information about control codes.

[001]  
[006]

Commodore & F

CTRL & A  
CTRL & Z

Mnemonic	Symbol	what to press
[LEFT]		left/right
[RIGHT]		left/right
[UP]		shift left/right
[DOWN]		shift & up/down
[F1]		up/down
[F2]		F1
[F3]		shift & F1
[F4]		shift & F2

Mnemonic	Symbol	what to press
[F5]		F5
[F6]		shift & F5
[F7]		F7
[F8]		shift & F7
[CLEAR]		shift & CLR /HOME
[HOME]		CLR/HOME
[BROWN]		CTRL & B
[YELLOW]		CTRL & B

Mnemonic	Symbol	what to press
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8



Your

Submissions

# COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together!

If you have written an exhilarating game or an invaluable utility on your Commodore micro, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be documented and type-written and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original; if it is not chosen for

publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below — you never know, you might even get paid for airing your views! All submissions should be sent to:

The Editor  
Your Commodore  
Angus Specialist Publications Limited  
No. 1 Golden Square  
London W1R 3AB

PLEASE COMPLETE IN BLOCK CAPITALS

Your Name \_\_\_\_\_

Program Name \_\_\_\_\_

Computer/memory size it runs on \_\_\_\_\_

Amount of memory program occupies \_\_\_\_\_

Other computer/memory size which your program runs on without conversion or use \_\_\_\_\_

Does your game need or use joystick? Yes No

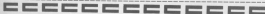
Have you sent your game to another magazine? Yes No

Is it original or a variation on a theme? \_\_\_\_\_

Your Address \_\_\_\_\_

Telephone Number \_\_\_\_\_

Times to contact you \_\_\_\_\_



# Evesham Micros

## THE UTILITY SPECIALISTS

### 3M Floppy Disk

3M Floppy Disk  
5 1/4" Diskettes  
5 1/4" Diskettes  
5 1/4" Diskettes



### HARDWARE SPECIALS

Hardware Specials  
Hardware Specials  
Hardware Specials

### THE NEW GENERATION OF BACKUP NETWORKS HAS ARRIVED

The New Generation of Backup Networks has arrived...  
The New Generation of Backup Networks has arrived...  
The New Generation of Backup Networks has arrived...

ONLY £29.95

### ALIGNMENT PROBLEMS?

1981 PHYSICAL EXAM



1981 Physical Exam  
1981 Physical Exam  
1981 Physical Exam

### DISC DIRECTOR V3.0

Disc Director V3.0  
Disc Director V3.0  
Disc Director V3.0

## Quickdisc+ PLUS 10MB

Quickdisc+ Plus 10MB  
Quickdisc+ Plus 10MB  
Quickdisc+ Plus 10MB

Evesham Micros  
Evesham Micros  
Evesham Micros

# 128

128  
128  
128



128  
128  
128

## Superbase 64

Superbase 64  
Superbase 64  
Superbase 64

Superbase 64  
Superbase 64  
Superbase 64

ALL FOR ONLY £949.00

## BOOKS

Books  
Books  
Books

## Super TYPE

Super Type  
Super Type  
Super Type

## Superbase 64

Superbase 64  
Superbase 64  
Superbase 64

Superbase 64  
Superbase 64  
Superbase 64

NOW ONLY £949.00

## Taproot

Taproot  
Taproot  
Taproot

## ASSEMBLER MONITOR 64

Assembler Monitor 64  
Assembler Monitor 64  
Assembler Monitor 64

OUR PRICE (D&H) £249.00

(OH D&H) ONLY £199.00

## VIZASTAR 64

Vizastar 64  
Vizastar 64  
Vizastar 64

OUR PRICE (D&H) £949.00 (OH D&H) £799.00

## VIZAWRITE 64

Vizawrite 64  
Vizawrite 64  
Vizawrite 64

NAME IT TERMINOLOGY	CALL US TODAY!	OH D&H £199.00
BOOKS £49.00	£49.00	£49.00
DISC DIRECTOR V3.0	£249.00	£199.00
ASSEMBLER MONITOR 64	£249.00	£199.00
QUICKDISC+ PLUS 10MB	£249.00	£199.00
ASSEMBLER MONITOR 64	£249.00	£199.00
QUICKDISC+ PLUS 10MB	£249.00	£199.00
ASSEMBLER MONITOR 64	£249.00	£199.00
QUICKDISC+ PLUS 10MB	£249.00	£199.00

Price includes VAT and 10MB diskette...  
Price includes VAT and 10MB diskette...  
Price includes VAT and 10MB diskette...



Calco Software  
Calco Software  
Calco Software

Give your data  
recorder the speed  
of a disk drive with  
this program from  
Nick Hampshire.

# FULL SPEED AHEAD

THE MOST TICKLE AND frustrating operation on the C64 is using the cassette deck to LOAD or SAVE a program. The operating system tape routines function OK but are so cumbersome and slow that the thought of loading a disk drive soon becomes a matter of very high priority in every user's mind. However, while waiting for your disk drive, it is not lost. With ingenious pieces of software, it is possible to speed up the tape routines to give a loading speed equal to that of a disk drive.

Virtually all C64 software currently being marketed uses some form of fast loader. These fast loaders are given names like Turbo (this was the first fast loader available), Fastload, Flash load, etc. The origin of these fast loader routines is rather obscure since many of the software houses use the same loader routines. In this article we give the source code for two fast loaders and their associated SAVE routines. These have been used on several software products of Zita Software Ltd, under the name of ZF1load and ZF2load.

A fast loader is a routine which replaces the existing LOAD and allows a program or data to be loaded from tape at about 10 times normal speed thus making a tape as fast as a disk drive. A fast loader simply changes the format of the pulse sequence stored onto the tape in order to allow a far greater density of information storage per inch.

In order to create a fast loader two programs are needed. Firstly, a fast loader program, which is a fairly short machine code routine loaded at the beginning of a LOAD operation and auto run to LOAD the rest of the program and/or data stored in fast loader format. The second program is a routine to SAVE a program in fast loader format, the fast SAVE.

The first major design problem to be overcome is the storage of each bit on the tape. Each bit is stored as a pulse which goes through a high-low transition (see figure 1). The length of the total pulse decides whether the bit is a one — a long pulse — or a zero — a short pulse. The bit is flagged in the interrupt register on the falling edge of the pulse.

The loader is a machine code program which runs with the interrupts disabled. It sets a timer between the two lengths, and when the timer runs out the interrupt register is checked to see if the pulse came in or not. If the falling edge of the pulse generates an interrupt before the timer runs out then the pulse was a zero, otherwise it was a one. The bits are then rotated into a byte storage until eight bits have been read, thereby loading a full byte.

Before any bytes can be read and stored, the loader must be in sync with the bits on the tape. A string of zero bits with a single one bit at every byte interval achieves this. The routine then tries to align itself by recognizing the value of the byte.

An example of a loader byte for alignment would be the value 16, hex 80 or, in binary, 80000000. A series of these bytes is written as the header. Only when this byte has been read and recognized can the actual program be read without risk of alignment errors.

The program is stored in different ways depending on how much is desired. The simplest way of formatting the file is to first SAVE the two byte load address followed by the actual file. The final byte following the end of the file is a checksum calculated by the save routine and also during loading. If the two values are the same, the LOAD was successful. The routine for this

form of fast loader is given in Program one.

Another type of LOAD, which uses the same save but is slower, is the interrupt loader. This method has the advantage of LOADING with the screen on and a foreground program running while the main program is loaded. Loaders of this type are Novoload and Microload. The difference is that an interrupt is created when a pulse is read by the tape recorder, and the timer is checked to find out whether the pulse was zero or a one. The advantage of this is done in the background allowing a foreground program to play music, run a clock, etc. The foreground program must check at regular intervals to see if the loader has flagged for the end of load. The background LOAD in Program two has only a foreground program which is waiting for the end of LOAD flag to be set.

## Fast Tape Routines — Making Them Work

Putting the theory into practice to create the fast loader routine is not difficult. The actual timing for the SAVE routine was not calculated from any theoretical formula but just by trial and error. The only guidelines were that the short pulses should be slightly shorter than half the long pulse, as the waveforms of the pulse is skewed due by the cassette hardware. The timing value used by the loader is just shorter than the time required before the long pulse reaches its falling edge.

There are two program listings for the C64 in this article, one for each of the two types of LOAD. Each program will SAVE a Basic program in its fast format and automatically put the fast loader routine into the filename where it is stored. When loaded, it will automatically start on the warm

start vector. The routines are initialized by SYS#9102. A Basic program can be fast-loaded by using the SAVE command as normal but with a device number of seven, thus:

```
SAVE "PROGRAM".7
```

In addition the first fast LOAD also makes use of the secondary address to auto run the program, thus:

```
SAVE "PROGRAM".7:1
```

will cause the program to auto run when loaded back. With both routines, when a program has been saved using one of these fast loader SAVE routines it is unnecessary to LOAD anything before the program; it will LOAD directly from the LOAD command.

An example of how fast these routines can be is shown by the following timing table. This was based on the first listing to LOAD a 25.6K byte Basic program.

Method 1	: 1 minute
Disk	
	: 1 minute 18 seconds
Method 2	
	: 1 minute 25 seconds
Normal tape	
	: 8 minutes 40 seconds

It should also be noted that the SAVE routines for the fast tape operation are considerably shorter than the normal tape routines. One wonders why Commodore has not included these types of fast tape routines in the new machines.

By loading these into your C64 you will be able to take some of the tedious and frustration out of using a tape system. In addition it will also make your programs look far more professional.

This article is reprinted from one of the 64 Revealed series of books by Nick Hampshire and published by Collins.











A NEW CLASSIC

FROM PAUL WOODS  
AUTHOR OF ENCOUNTER

# YOU CHOOSE THE ACTION IN MERCENARY

Mercenary - a unique combination of flight simulation, adventure and arcade fun. You choose the action.

There is complete freedom of movement in a truly three-dimensional vector-graphic environment. Graphics of exceptional speed create a very realistic experience.

You'll never play the same game twice. Random elements attempt to fool your fate. Your interaction is crucial.

Mercenary presents an absorbing challenge that you will repeat again and again.

## MERCENARY ESCAPE FROM TARG

Fly through the stormy Targ in a world of war. Investigate the complete situation for the player before you in your quest for essential weapons. Search for mysterious substances to complete a part of your mission.

Develop your skills in the complex world of the Targ and Mercenary. An exciting game with many secrets to work independently to solve.

You only have one shot - the one you have in your hands.

Repeat your adventures on you with 1 to 30 levels of difficulty. Upgrade from Targ.

**NOVAGEN**

INITIAL RELEASE  
FOR COMMODORE 64 - ADAM 400, 600 XL, T30 HE  
CASSETTE £9.95 DISK £12.95

NOVAGEN SOFTWARE LTD, SALES DEPT, ALLOT ROAD, BRIMMINGHAM B37 7YU

# SCHIZOFRENIA

or how to be in two minds about paranoia



£7.99

SPECTRUM 48K  
COMMODORE 64



Alphonse T. Nurd needs help. His alter-ego has discovered the meaning of independence. If you can discover the secrets of sub-atomic particle separators and clean lifts, this is the game for you. The puzzle of a lifetime ... or two.

# QUICKSILVA

*The most powerful games in the Universe!*

Available from all good software shops.

In case of difficulty write to:

QUICKSILVA Ltd, Liberty House, 222 Regent Street, London W1R 7DB tel: 01 439 0666



Let some fresh air into your  
graphics with this look  
through windows by Allen  
Webb.

# TOP DRAW

GREETINGS FELLOW 64 OWNERS, HERE are some more graphics advances to add that bit of power to your efforts. First, however, it's grovelling time. Remember part one of this series? You may have noticed a POKE to location 8008 in the demonstration. Much to my embarrassment, I didn't tell you what it does. It holds the number of shifts executed when you roll the roll routines. A value of one will roll the design one pixel, a value of two rolls it two pixels, etc. Values greater than three aren't too helpful, but the use of this register will make your efforts that little bit more interesting.

This month I want to deal with graphics windows. A window is a definable area of the screen which can be manipulated independently of the rest of the screen. The real value of windows is in the generation of menus and text games such as adventures.

As usual, the software is given as a basic loader in listing one. The code is at \$C000 so it will unfortunately clash with the character routines - accept apologies but such is life.

The first problem to resolve is how to define a window. Consider Figure 1. The rectangle represents a window of width W characters and height H units somewhere on the screen. The top left hand corner is fixed by the co-ordinates XC and YC. These four values are all that is necessary to specify the window. The machine code allows you to resize and move up to 16 windows - numbered 0 to 15. Each window is defined in four tables in terms of co-ordinates of the top left hand corner and the height and width. The syntax for specifying a window has the format:

```
W# X# Y# W# H#
```

W# is the window number and the other parameters are as specified earlier. It is important that you set up a window before manipulating it. When the program is set up the window tables contain area and tap action on an on-defined window will, at best, corrupt your Basic program so be warned!!

What can you do with the window? The next command has two forms depending on what you wish to do. A flag decides post action and the syntax. The first form will fill the window with a specified character CH:

```
W#-X#W#Y#H#CH
```



i.e. the flag has a zero value. A non-zero flag simply reverses the contents of the window:

```
W#-X#W#Y#H#F#
```

If you use the fill command with a value of CH equal to 32, the window will be cleared.

Finally, you can scroll the contents of the screen. To maintain compatibility with the normal 64 operation, the scrolling is upwards with the bottom line of the window being filled with blanks. The syntax of this command is:

```
W#-X#W#Y#H#
```

To help you see what these routines will do, I've given two demonstrations. The first shows the manipulation of three windows at once. Since the window scrolls upwards, you read something print out at the bottom line. In my experience, the use of cursor control codes and TAB are both elegant and tiresome. It's far better to use a routine in the 64's Kernel. The following line will do this:

```
POKE 761,Y:POKE 762,X:POKE 763,W:SYS 6152
```

where X and Y are the co-ordinates of the point to which you want to move the cursor. Line 200 in the demo does this.

The second demonstration uses randomly created windows to produce a pattern.

I have included a degree of checking in the routine so that some illegal values (e.g. width 40 or height 25) will be

rejected. Not all possibilities are catered for and it's up to you to ensure that silly values are not generated in your program.

The routines should be of most value to adventure freaks since it is simple to create two or three text windows with independent scrolling.

Now, I want to deal with a feature which the BBC, C/M and Spectrum have in common. The ability to use flashing characters. The simple way to implement this is to use the attribute of each character to hold a flash flag. This is not simple to do on the 64 because there is no spare bit in the video matrix and it isn't easy to use the top four bits in the colour matrix. My solution is to let you select a colour to flash. This colour is kept in location 8008. The routine is called every 50th of a second by the IRQ interrupt. The screen is scanned every 25 interrupt calls, so that the flash rate is about twice a second. The routine scans the colour matrix and inverts every character of the specified colour. This routine is given in listing two and a demonstration in demo three.

Owners with new ROM 64s should take a little care. These machines fill the colour matrix with the current colour each time the screen is cleared. If the current colour is the flash colour, the whole screen will flash. Given three choices here to use the routine.

To turn off the character flash, simply enter:

```
W#-827#
```

That's all for this time, see you again next month. Happy hacking!

### Demonstration 1

```

0 REM DEMONSTRATION 1
1 REM
10 POKES9891,1
20 OS="*****"
30 DATA 0,0,9,10
40 DATA 15,8,9,18
50 DATA 5,3,10,10
60 FOR WN=0TO2: READ X(WN),Y(WN),W
  I(WN),H(WN)
70 NEXT
80 FOR WN=0TO2
90 SYS 12*4096+3,WN,X(WN),Y(WN),W
  I(WN),H(WN)
100 NEXT
110 FOR WN=0TO2
120 DO=ROUND(10*16:IFDO=1THEN120
130 POKES98,DO
140 IFWN=0ORWN=1 THEN GOSUB 150:PR
  INTLEFT(08,WI(WN)):SYS12*4096,WN
150 IFWN=2THENSYS12*4096+6,WN,0,WN
  D(1)*256
160 IF RND(1)>.4,THEN SYS 12*4096+6
  ,WN,1
170 NEXT
180 GOTO 110
190 POKET81,Y(WN)+H(WN)-1:POKET8
  2,X(WN):POKET83,0:SYS98580:RETURN
  
```

### Demonstration 2

```

0 REM DEMONSTRATION 2
1 REM
10 XS=INT(RND(1)*20)+1
20 YI=INT(RND(1)*20)+1
30 YS=INT(RND(1)*12)+1
40 HI=INT(RND(1)*12)+1
50 SYS12*4096+9,1,XS,YS,WI,XI
60 SYS12*4096+6,1,0,RND(1)*128
70 SYS12*4096+6,1,1,GOTO10
  
```

### Demonstration 3

```

10 REM DEMONSTRATION 3
20 REM
30 SYS 62736: REM TURN THEM ON
40 POK 1000,1: REM WRITE TO FLASH
50 PRINT"CLLDR3CYM3TRIS CWITES
  18CYELLOW A CWITESDEMONSTRATIONE
  3 73 OF CWK
  ITESFLASHINGCG NO CHARACTERE"
  
```

### Listing 1

```

0 REM LISTING 1
1 DATA78,3,152,76,114,188,76,133,1
  53,32,298,188,166,20,141,232,3,32,
  210,182
2 DATA32,25,189,24,166,163,105,40,
  133,187,166,164,105,0,133,108,27,1
  66,168
3 DATA108,40,133,176,166,170,105,0
  ,133,177,174,52,133,170,51,189,136
  ,177,187
4 DATA163,163,177,178,148,188,136,
  15,245,208,240,28,24,166,163,105,4
  0,133
5 DATA183,144,2,230,184,24,102,187
  ,126,40,133,187,144,2,230,188,32,6
  ,133,76
6 DATA92,188,178,21,123,130,168,92
  ,145,163,163,1,145,183,138,18,245,
  92,32
7 DATA98,132,162,20,201,10,176,84
  ,141,232,3,32,238,182,178,232,3,18
  2,20,201
8 DATA41,178,43,153,23,189,92,238,
  132,172,232,3,166,20,201,25,178,24
  ,133,168
9 DATA133,32,238,182,178,232,3,166
  ,20,201,41,178,19,163,113,153,32,2
  38,132
  
```

### Listing 2

```

0 REM LISTING 2
1 DATA188,25,143,233,3,120,188,78,
  141,20,3,189,206,141,21,3,88,96,18
  8,0,133
2 DATA251,168,218,139,252,160,0,17
  7,251,41,15,205,232,3,208,17,166,2
  51,133
3 DATA263,66,188,252,233,212,133,2
  54,177,263,73,128,148,263,230,251,
  208,2
4 DATA220,252,188,251,201,232,240,3
  ,78,29,206,165,252,201,219,240,3,
  76,28
5 DATA208,98,206,233,3,208,8,168,2
  5,141,233,3,32,18,206,76,48,234,29
  8
6 FORI=62736TO58831
7 READX:T=T+X
8 POK 1,X: NEXT
9 IF T<>13738THENPRINT"ERROR IN DAT
  A"
10 REM
11 REM FLASHING CHARACTERS ROUTINE
  
```

10 DATA 178, 238, 3, 188, 80, 801, 28, 178  
 4, 153, 153, 133, 98, 163, 155, 165, 168,  
 38, 30

11 DATA 171, 98, 73, 76, 76, 69, 71, 66, 76  
 38, 88, 88, 78, 88, 83, 83, 0, 172, 232, 3,  
 185, 93

12 DATA 133, 141, 83, 133, 185, 103, 183,  
 141, 83, 183, 183, 133, 133, 141, 91, 153,  
 185, 183

13 DATA 133, 141, 93, 133, 98, 38, 263, 17  
 4, 38, 138, 178, 38, 847, 183, 98, 185, 183,  
 274, 103

14 DATA 40, 133, 178, 188, 170, 103, 0, 13  
 3, 177, 98, 84, 165, 163, 105, 40, 133, 183,  
 144, 8

15 DATA 230, 170, 24, 165, 176, 106, 40, 1  
 33, 178, 144, 2, 230, 177, 98, 163, 0, 133,  
 163, 168

16 DATA 1, 133, 164, 174, 90, 193, 240, 16  
 24, 168, 183, 108, 40, 133, 163, 183, 183,  
 103, 0

17 DATA 133, 184, 202, 208, 240, 174, 88,  
 133, 240, 14, 24, 165, 163, 108, 88, 183, 1  
 33, 183

18 DATA 165, 184, 103, 0, 133, 184, 188, 1  
 83, 133, 183, 165, 184, 24, 105, 212, 133,  
 170, 98

19 DATA 10, 2, 10, 10, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 0

20 DATA 0, 0, 5, 5, 0, 0, 0, 0, 0, 0, 0, 0, 32,  
 838, 188, 188, 80, 141, 238, 3, 38, 838, 18  
 2, 165

21 DATA 20, 141, 231, 3, 208, 0, 38, 238, 1  
 88, 188, 80, 141, 838, 3, 38, 210, 188, 38,  
 20, 133

22 DATA 182, 0, 180, 0, 38, 204, 193, 200,  
 204, 90, 183, 240, 3, 76, 188, 183, 84, 165,  
 183, 105

23 DATA 40, 133, 183, 188, 184, 105, 0, 13  
 3, 184, 232, 238, 98, 133, 240, 3, 76, 167,  
 183, 98

24 DATA 173, 234, 3, 208, 5, 173, 233, 3, 1  
 45, 183, 201, 1, 208, 8, 177, 163, 73, 128,  
 148, 183

25 DATA

26 END

27 END WINDOW GENERATOR C84

28 END A + M ASSOCIATES 1985

29 END

30 FOR I = 18152 TO 18632

31 READ X: T = T + X

32 POKE I, X

33 NEXT

34 IF T<=5680 THEN PRINT "ERROR IN  
 DATA"

## FAST LOADERS TO DISK

COMMODORE 64  
54K DISK DRIVE

## TAPE FAST LOADERS

## MEGA-TRANSFER DISK — \$168

A highly collection of routines which will effectively transfer an arbitrary sized tape-format loadingspace to disk. Several purposes require to transfer Micro, Pac, Human, Flash, Hyper, 1000 systems and others. Archive users of individual routines for unusual loaders, including coordination inter-leaf systems, the latest sports simulations and top arcade and adventure games. General purpose routine for those without 11.5 disk loaders which need Diskus 11-leaf transfer. And transfer routines for programs requested using E.S.O. software and Hypertape-84. Includes a program identifier. All transfer to systems for user knowledge required. MEGA-TRANSFER is the finest transfer utility available for fast loading programs. Don't waste for this.

## MEGA-UTILITY DISK — \$110

1. **40 PRG/LOAD** Disk Turbo Loader. Just this one program is worth of your disk and your programs will load at over 700K/Sec. Uses normal and 50-Kbaits is required and you DON'T have to load separately. Features 200 bytes format speed 188 seconds - with ALPHALOAD will 30 seconds.
2. **FAST COPY** A high class single disk whole disk system. Requires what disk in a time over 1 minute. No table utility conventional disk but not if handy program.
3. **A SILENT TAPES** Transfer any single disk program to tape (fast load). No memory conflict. No transfer programs of 200 books and more.

## DISKUS I — \$110

The Premier tape to disk transfer utility for your 50K load tape-collection. Multiple file-format and archive programs are all optional for 50 user knowledge required. Diskus I has provision for program cycles which no other utility can handle. Shipped on disk.

DISKUS I is a highly efficient transfer utility for your 50K load tape-collection. Multiple file-format and archive programs are all optional for 50 user knowledge required. Diskus I has provision for program cycles which no other utility can handle. Shipped on disk.

## 11 TAPE SPECIAL !!

REPO BACKUP 80/240 (40K) Diskus. Control your tape loading with 1000-2000000. Backup and archive programs are optional and use the user knowledge. Includes MEGA and archive more programs than any competing utility.

DISKUS I is a highly efficient transfer utility for your 50K load tape-collection. Multiple file-format and archive programs are all optional for 50 user knowledge required. Diskus I has provision for program cycles which no other utility can handle. Shipped on disk.

SPECIAL NUMBER DISKUS WITH PROGRAMS JUST \$110.00. Includes 80/240 (40K) Program backup utility 80/240 (40K).

All programs for the Commodore 64.  
 Send SAE for full details or cheque/P.O. for fast dispatch to:

DISOFT  
 2 OAKMOOR AVENUE  
 BLACKPOOL FY5 8EE

U.K. Postage included. Europe please add 70p. Overseas add a £10 for airmail.

# COMPETITION

Want to get your C64 on line?  
Do you feel the need to  
communicate with other 64  
owners? Well you could win  
one of 10 Commodore  
modems that we are giving  
away in this month's  
competition.

## The rules

ENTRIES WILL NOT BE ACCEPTED FROM employees of Argus Specialist Publications, Commodore Business Machines and Alabaster Pannone and Sons. This restriction also applies to employee's families and agents of the companies.

The how to enter section forms part of the rules. The editor's decision is final and no correspondence will be entered into.

## How to enter

There are a few differences between the two pictures. Study them carefully and circle the differences on the picture attached to the coupon. Fill in the coupon clearly and seal the picture and coupon in an envelope. Write clearly the number of differences you found on the back of the envelope.

Post your entry to:  
Modern Competition, Four Commodore,  
No 1 Golden Square, London W1R 6AB.  
The closing date is first post on Thursday, 31st November 1985. You may enter as many times as you like but each entry must be on an official coupon - not a copy - and sealed in a separate envelope.

Important: Please follow closely the guidelines on entering - incomplete coupons and entries with no numbers on the back cannot be considered. If you are a winner, the coupon will act as a label for your prize so clear writing is essential.



"Darling, it's been on that phone all day"



"Darling, it's been on that phone all day"

## ENTRY FORM

Name .....

Address .....

.....

.....

..... Post Code .....

Age ..... Occupation .....

Please complete the following tie breaker:

I would use my Commodore modem to .....

.....

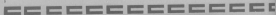
.....

.....

.....

.....

.....



THE COUNTDOWN HAS BEGUN

# CRITICAL MASS

From DURELL



R.R.P. £29.95

Countdown 64

Spectrum

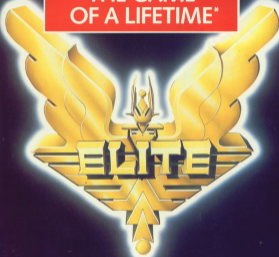
**DURELL** sales dept.,  
Castle Lodge, Castle Green, Taunton TA14AB

Write now for details of Durell's  
Great New Competition also  
T-shirts, Calendars  
and Posters.





THE GAME  
OF A LIFETIME\*



Available for Commodore 64, Spectrum-128K and Spectrum +. Coming soon: Amstrad and MSX.



**FIREBIRD**

4th Floor, 100, The Arcade, London EC2A 4DF, UK  
100, The Arcade, London EC2A 4DF, UK

**Gold  
Software**

# RUPERT

and the  
TOYMAKER'S PARTY



Jolly old Rupert can walk, jump, leap, climb stairs – and even fly – but can you get him to the party?

32 challenging screens of animation and fun.

**QUICKSTAVA**  
The most powerful games in the Universe

Quickstava Ltd, Liberty House,  
222 Regent Street, London W1R 7DB.  
Telephone 01-439 0666

Rupert Bear © Express Newspapers PLC

Spectrum 48K  
CBM64

£7.99

**Runecaster risks life and limb  
yet again in order to bring  
you his column.**

AS I AM SURE YOU ALREADY KNOW, adventure games can cost quite a lot of money! Generally speaking the more expensive games are also longer and provide you with many hours of pleasure for your money. The higher middle range of tapes (around £10 plus) seem to be the ones that sometimes make you think:

On the other hand there are a number of bargains around. My local Spectrum Chain shop had a couple of good buys... One would be loosely (very loosely) classed as an arcade adventure - *Mystic Mansion* - and the other, a well presented text adventure, was *Allard's Tomb*. Respectively, £2.99 and £3.99.

The first, *US Gold*, hardly comes within the range of this column but is worth looking at, especially if you have young children around. It is a 2D level, find the treasure, avoid the monsters, type of game with a good difficulty grading. - *Adult, Teenager or Child.*

The second, *Allard's Tomb*, is a real title adventure that does not appear to have any hidden inconsistencies and at the price is well worth buying. Fairly conventional scenario...dark dungeons, in which you have to find the shoreward Tomb, solving several ingenious puzzles in the process.

Keep your eyes skinned for these and other 'cheaper', they are often either newbies or are rock from defunct software houses. You will not find *Foodus: Move It*, *The Hobbit* or *Funke* but may well find something to keep you interested.



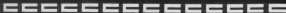
# SENSE OF ADVENTURE



## Kitchen spin-off...

Down in the kitchen something stirred...or rather squeaked! It was probably the lady of the house using one of those reusable shopping list boards - you know the type: plastic-based and supplied with a water soluble ink marker. Make notes throughout the week on what is required - buy them on Saturday and wipe the 'list' clean on Sunday.

What has this to do with playing adventures? Well, a firm in Wales, Mapit, has produced a plastic reusable Map Maker. It is A4 size (two pages of Your Commodore) and has a first goal of 14x27mm rectangles printed on it...just the thing for mapping all your trip adventures! There is space at the top of the board to make 'vital notes' and the pens supplied have a fine enough point to enable all the vital facts of a location to be recorded in the space provided.



The Map Maker costs £3.49 and therefore is a range of suitable prices in various colours at 79p each. A little more expensive than scrap paper but as an approach Christmas Magics can be found at: 356 Bolton Street, Finsbury (perhaps we should have a check-out as that's Mid Glamorgan CF10 3BA.

But... the great Crystal was stolen. The Red Moon is the story of how a Magician [you!] recovered this fabulous gem and restored hope to the World.

Not only are there hundreds of screen writing pictures but there are also the long and descriptive texts that have always been the hallmark of Level Nine games. Read, and wonder at it all... hope do they get it all into our computer's memory? The pictures are good, better than the rather unrealistic landscapes of Return to Eden. They are certainly not high, hi-res works of art but serve the very important purpose of visually triggering recognition of your present location.

As with all Level Nine games (and most other good adventures!) it is important to keep a careful map of your progress. There is a temptation when recognisable pictures are displayed, to skip the mapping procedures... might this temptation - show a map!

Unlike many games where an apparently invisible puzzle hints your progress within a few steps of your entry point, Red Moon has plenty of locations for even the absolute novice to explore right from the start. Do not think that this implies a beginner's game, even accomplished adventurers will be grasping their heads on occasions, for those that get totally lost and confused, Level Nine offers the best hint shown in the business. There are so laid out that it is not easy to read something you wish you had it.

Interact found soon after entry will include the inevitable lamp for subsequent exploration in dark and dangerous places. Dangerous? Yes! There are a number of decidedly aggressive creatures just waiting for you to stumble by. To be successful you will have to fight them. Bare hands are not to be recommended, find a spell of weapon and better still some armour to protect you. You start your trip into this other world with some 'hit points', it is surprising how quickly these can be used up. The use of Magic is permitted, even encouraged. To implement these spells you must enter a command of the type 'CAST ZAP', each spell may only be cast successfully if you have the object that is the 'focus' for that particular spell.

'CAST ZAP' is used to magically attack an enemy but you must have the provided digger for it to work. Casting spells also uses up some of your 'hit points', so try not to be too overzealous with them.

WALL and BUSTLE your current position are also counted as 'spells' in this case the 'focus' is required but it will save up your 'hit points'. An iron dagger this, as it stops us making dozens of 'safety saves'.

Whether can you cast a spell successfully if you have lost about your person (or at that location). You cannot fault the logic, even if it is a little frustrating at times!

The CD4 tape version utilizes a 'rapid loader' and takes just less than three minutes to load this monster program. However though the screens are usually available the game is fairly slow.

Level Nine is to be congratulated on keeping standards at such a high level for such an extended time... good luck and roll on the next one - but please give me a little time to solve this one first!

## Score fingers...

Although the range of good software grows as time goes by, there are always some people who like to 'do their own thing'. Obviously if you get that good at it we will start reading about your work in this column!

Many others spend enjoyable hours providing their keyboards typing in programs gleaned from magazines or books. This can often be very, very good programming practice. This is the way to learn how a program works (over the hand but permanent way - by having to debug it).

It is also often possible to see how to improve on the published listing. The original programmer was probably happily involved in his program, whereas you will be looking at each section in a fresh light.

There are already several books on the market that will help you write adventures games on the Commodore 64. On the other hand, new ones are always welcome, you never know what new routines you may find! Book is nowadays are rarely cheap but in terms of time spent at your computer are probably much cheaper than the average price of software. Two offerings have come out

## Red Moon



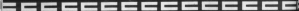
## Level 9 Computing

## Over the moon

Level Nine does it again. Red Moon is the latest from Britain's number one adventure software house. As usual the program are available for a number of computers but of interest to us here is the CD4 version.

Over 300 pictures can be found as you explore the 'magic' land of the Red Moon. Level Nine seems to improve with age, not only can you turn the pictures off, saving the 'drawing' time, but with this program you can even type in commands while the pictures are being created on the screen!

The Red Moon is about a land where Magic works, about nowhere nearly as strongly as in the past. In the past the moon was crimson red and the source of all the great Magic. As the moon faded to the grey we know today, so the Magic waned. A great concourse of Magicians created a substitute - the Red Moon Crystal. This was mounted in the Moon Tower of Baskalos and shone out over the kingdom, maintaining an island of Magic and religious Civilisation.



was recently, one American and one British.

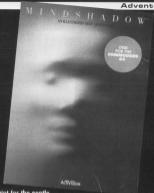
The first is Golden Flutes & Great Escapes by Delium 1 Home, published by Delium Press, ISBN 0-88098-010-7. There are over 200. Four complete listings are given together with hints, tips, flow charts and possible variations to the game play.

The book is written in clear English (American) and presents the writing of adventure games in a modular form that should be understood by the reader. The listings are in BASIC and are for text only displays. Although I have not tested any of them in, they look reasonably interesting if you are just starting out along this path. They should give you plenty to think about. Delium Press Publications are distributed in the United Kingdom by: Holt Saunders Ltd, 3, St Anne's Road, Eastbourne BN27 3JN. Holt Saunders has a number of titles dealing with computers and computing. Lists of these are obtainable from the above address.

The second book is Quicksave and Quickload by Bob Litch, published by Virgin Books, ISBN 0-86392-094-7 - 106 pages cost £5.99. This book is quite different to the one above as no explanation of each individual program is given.

Listings for 15 (1) different adventures are given together with a page UNDO/REDO or so to set the scene for each. Again the programs are in BASIC and are clearly set out for you to type in. The games are presented for you to play rather than learn from but the inevitable debugging is bound to teach you something.

Both books appear to offer good value for money - both out for them and let us know how you get on! Both publications offer copies of the games on cassette for the lay amongst us!



### Not for the gentle...

Do you remember Dallas by US Gold? It appeared in this country last year and was acclaimed by many reviewers at the time. If you have not played it, try to get your computer shop to demonstrate it for you. It is disc based only and has excellent graphics that are called from the disc as you proceed.

What made me think of Dallas is a recent issue from Activision - Mindshadow. It too is disc based and also has good graphics, again called from the disc. The plot is novel too, you are on a desert island... not knowing who you are!

The aim of the game is to discover your own identity and who left you in prison... As the story unfolds you find yourself travelling around the world in search of the answers. The program is well conceived and presented, the puzzles are good and the use of the function keys will thought out.

The only thing I have against Mindshadow is the underlying need to behave in a somewhat unorthodox manner to succeed. As a historian dabbling with a bit of sailing dilly through clangers - I have no qualms... but striking out with no provocation or sailing from a sleeping tramp... I have. That great quest that is a good adventure and the use of the function keys make it very easy to play. The most useful commands are immediately at your fingertips. SAVE, LOAD, REPEAT, HELP,

DROP, GET, QUICKSAVE and QUICKLOAD are all function key commands.

SAVE and LOAD allow you the option of 99 different game positions. These may be overwritten at any time you change your mind throughout the game. QUICKSAVE and QUICKLOAD are particularly nice features as they allow you to save your present position 'temporarily' at any time you think your next move may be your last!

HELP allows the assistance of a wise old bird - the Corvid - why a Corvid? I do not know but he can be damn useful! His help can only be given three times... so make use of his knowledge carefully. You can of course start from scratch and use the help gained in previous games to reach your last position, then get three new HELPS!

Although there are some obvious parts to Mindshadow, it is not a very difficult game to play. The number of locations in each section is not large and although mapping should always be an adventurer's first line of attack it is not always necessary here. You must solve each section before you can proceed, so at least you know how you are doing! If you are unable to go anywhere else then you know you have missed something!

For all that, I still do not know who I am, there are no prizes to anyone making any useful suggestions!



# SMALL PRINT



**Barry Miles has had a look at  
the C-128 Dot Matrix Printer  
— and gone overboard.**

THIS PRINTER IS BEING SOLD AS A companion to Commodore printers. It is fitted with 500 sockets and an internal interface making it directly plug-compatible with Commodore machines.

In addition, it responds to Commodore's control codes and Commodore's address. It is in the secondary range of printers has been extended. You can also switch the machine into an Epson 80-column mode. It is immediately apparent that serious thought must be given to this printer.

First impressions are very good. This is not a feeble-looking machine. It weighs in at a mere 8.275 pounds and comes in a box with a nylon carrying handle, which adds to its lightness. In fact it could easily be carried in a normal briefcase.

Additionally, first examination of the machine shows that very careful thought has been given to its design. It is both a tractor and friction-feed printer and has a carefully designed paper-support system, which keeps the paper flat. This is a great advantage, since it permits extraordinarily easy front-loading of

paper onto the tractor, and also enables you to insert cut sheets, of any thickness, (even glossed), without difficulty. Furthermore, the paper you are printing on does not get bent at all during the printing process.

Another great feature it appears in the way in which the controls have been placed fully within outside the case of the printer. You do not have to grope down into the bowels of the machine, looking for the lever to vary the force of impact, for the lever is fully accessible, fully visible, and sensibly labelled.

Similarly, the paper guides are marked making it easy to set up normal lining paper or A4 accurately.

Switching between tractor and friction is a similar trivial task making the printer a similar trial and pleasure to use.

The simplest and cleverest innovation is the sturdy foldaway wire stand which lifts the printer just a few inches off the table-top, thus making room for a convenient pile of paper.

You have only to trouble about the destination of the paper. If your table extends backwards further than the back of the printer, you can allow the paper to fall up there. Users who prefer to have built up their 1,000 or 2,000 sheets on the whole box of 1,000 or 2,000 sheets on the floor will find that when the stand is folded away, there is still clearance for the paper to pass through the printer. C-128 paper to pass through of anything.

The machine runs at 180 characters per second, and prints all 80 Commodore

graphic characters in addition to the 80 ASCII characters. The raise-wire head enables full descenders to be printed for good readability.

In use the printer was reasonably quiet, and extremely easy to operate. The adjustments to make more fundamental adjustments are easily accessible once the single self-tapping screw holding the flap in place is removed.

The features which are likely to impress most of all are the 80 length impress most of all are the 80 length graphics at 480 or 960 dots per line, and the complete EPSON compatibility. This means that all the fancy Control Codes can be invoked, giving double strike, emphasis, double emphasis, italic, compressed and expanded, subscript and superscript characters.

Near Letter Quality print, by means of multiple passes is not available, but as you will see from the sample printouts, all other likely requirements are catered for. Indicator LEDs show you when-globe is on, paper is out, and whether the printer is Online or not.

The Select Button toggles the printer on and off-line, and the Home Feed and Linefeed buttons perform their obvious functions.

Ribbon changing is a double because the ribbon is in a small rear cartridge, and the printhead is changed, after 50-million characters, by simply clipping it on and off.

I found the printer to be the simplest I have come across, and best from view. Highly recommended and very good value for money.



Set your  
Commodore 64  
FREE

# 1985 THE YEAR OF COMMUNICATIONS



A whole new world awaits you and your 64. The world of Protel, Microsoft, Viewfax, Homelink, Citelview.

This is the world of product reviews, mainframe games, time-saving, business information, commodity futures, animal shipping, real time conversation, electronic notice boards, helpful tips, KOPAC, news, Telex-talk, Mailbox, Christine Clatworthy. The entire world of software, with programmes to download into your 64 and save to disk or tape. Many specialist programmes are provided free of charge or at nominal cost.

Dialog with over 600 massive databases on call and almost unlimited information on just about any subject is accessible from your 64, together with BT Celer, Baseline, One-to-One etc, with electronic mail, business services and telex facilities.

You can link to Bulletinboards in your area for local news, club news, letters, jokes, poems, helpful tips, etc. It's a whole new experience.

Take the first step. Make a commitment for your 64. Communicate with Modem House and you'll soon be talking to other "64 users, linking to other networks and swapping files, tips, programmes or just chatting.

Modem House are the largest suppliers of Commodore communications equipment. We also supply complete systems for the Pro 2000, 3000, 4000 and 6000 series.

**MODEM HOUSE**  
Islands Drive, Exeter, Devon EX4 9BA, Tel: (0392) 69095

The most popular 64 micropack in use today, will link you to Protel, Microsoft, Viewfax 250, Homelink, Farmink, Citelview, etc. From Modem House, the complete system is on offer at the incredible price of £99.95 inc. VAT, p.4 p.

Optional terminal software to link you to BT Cost, Easylink, One to One, Dialog and Bulletin Boards is available at £29.95 inc. VAT, p.4 p.

Complete Commodore 64 micropack

special price **£99.95** inc. VAT p.4 p.  
for a limited period only.

Special, special offer, order your 64 micropack now and get a FREE quarter subscription to Microview 800 and Viewfax 250 (Protel Microcomputing).

Modem House, Islands Drive, Exeter, Devon EX4 9BA, Tel: (0392) 69095

Please send me \_\_\_\_\_ Commodore 64 Micropack £99.95 inc. VAT

\_\_\_\_\_ Optional Terminal Software £29.95

I enclose £ \_\_\_\_\_ Bill by Account No. \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Telephone No. \_\_\_\_\_

Please allow 28 day delivery

This offer applies only while stocks last

HC11.885



## 3-D GLOOPER

Superior 14.99



### CRS/Plus 4 joystick optional

This game, originally on the G4 has now been converted to the 16, in effect, the game is 3D Pac-man and you do actually stand in the maze. The scenario is as follows: You must collect

all the blue dots on the floor of the maze whilst avoiding the attention of the ghosts.

Three brown creatures patrol the maze and will chase you given half a chance. There are a number of red dots which empower you, for a limited time, to munch the glooper's. During this period, the glooper's are coloured yellow rather than their usual brown. Clear all the dots and you move onto another screen.

The screen shows a perspective view of the maze walls and side passages. A combination of block patterns and coloured shading heighten the effect. The glooper's are quite simple in design but great fun in use as you approach them, in all a surprisingly neat bit of graphics.

You can examine your immediate

surroundings on a small map which appears at the top of the screen. This shows the position of dots and glooper's. If a glooper starts to approach you, a warning sound is given. The keyboard or joystick is used to move about the maze. You can move at quite a speed but this isn't always advisable since you can easily turn a corner straight into the arms of a glooper.

In this world of more sophisticated software, it's nice to play a simple but addictive game. I found Pac-man rather tedious since the whole screen is visible. In this game, you have no idea where the glooper's are. This engenders a surprising amount of excitement. I found this game great fun to play and quite addictive. The graphics are simple but most effective. At the price, quite good value for money. **M.R.**

## SKYFOX

Advanced 19.95



### COM

As its title Sky Fox might suggest it is about a jet fighter but it's not one of the flight simulators that have become popular lately, more of an air combat simulator. Thus the pilot doesn't have to concern himself with all the intricacies of take off and landing but can concentrate on the important business of dipping bombs.

This brings us onto the game itself. The story goes that the facilities have invaded, either from another country or another planet, and so about the usual matter - the things such as killing and destroying. Your job is, of course, to - neutralise the invaders and save the world/country. In order to achieve this you are equipped with the Sky Fox fighter plane, a weapon capable of standing still in the air or staying along at speeds of 5000mph. The Sky Fox boasts a full complement of battle weapons ranging from general purpose laser guns to an anti-steering missile.

The screen displays a cockpit view that has three large windows and a radar display which shows wires as if round proximity loss or a sort of weaponry screen complete with gun sights. Also available is a direct link to the main battle computer in your base. When activated this brings out an expanding window to cover the whole display. This computer link supplies information about the progress of the war with the status of various installations being displayed so that you can judge where you are most urgently required.

Also in the data banks is a battle plan displaying the position and number of your base as well as the locations of your base and the other installations. This plan may be used to navigate to



using the auto pilot is much more fun as it will either take you to a specific location or it will find the nearest available base and zoom there at 5000mph.

Whilst being great fun and very convenient, this auto pilot knows nothing of air combat and thus usually engages you with the enemy from the worst possible position. Combat takes two forms. First is the ground force, these consist of groups of jet tanks. These appear on the display as tiny little spots on the horizon. At this stage the spots may be dispersed of with the laser very easily. This would lead to a status however detailed battle tanks that shoot at you as you approach and to miss this would spoil the fun. So my advice on the tanks would be to destroy all but one, by first that remaining the excellent graphics and then turn round and blow it to hell.

The other form of combat is in the air. As you climb from ground level you pass into the cloud layer and the screen goes black. At this point the sky is being accessed but I think the cloud idea is a nice cover for this. Once

through you are presented with a view of clouds and air is the domain of the enemy air fighters. These usually don't appear at all because the auto pilot tanks you in the fight with the missiles behind you. If you do manage to get one in front of you don't forget to be annoyed by the smooth animation of such a large and detailed shape before you target the fuel tank and dispatch it to the same fate that the tanks suffered. This aside, I should state that the air combat is not easy at all and took me many goes to get the hang of it.

The biggest and baddest of all are the nuclear ships. These are the large floating cities in the sky. The nuclear ship will do all the other duties as a good plan is to take on the nuclear ship first to prevent them reaching our repositories for the gas and jets that you destroy. As big as they are they can be taken out by laser work but a lot of careful flying is required to get past the internal defenses and they need to be hit a number of times due to their armour. Once you have exhausted your supplies of missiles and fuel you can land at your base to stock up. That is if it still exists!

The game has a large range of options such as difficulty levels from Novice to Ace of the base and practice levels involving only air or only ground combat. Also available is the suite of the tactics attack and the form of your task. These have titles such as 'Invaded', 'The Atomic and', 'My personal favourite', 'Massive Outright'.





## BEACH-HEAD II

US Gold £9.95

.....



### Q&A - 1 or 2 joystick?

The sequel to the successful Beach-Head is not just one game, but four games in one for C&A or two players.

The evil dictator known as the Dragon has captured some of your men and is holding them hostage in his heavily fortified island home. The task of the allied forces is to storm the island,

rescue the prisoners and escape with them by helicopter. You then do battle in single combat with the Dragon himself.

Three four screens may be played in any order, with three skill levels for each. You may choose to play the allied troops or the dictator's forces, so many permutations are available.

The attack phase involves landing your troops by parachute from a helicopter, then advancing them individually past two barriers under withering machine gun fire. You will lose many of your men, and will need to sacrifice some for the safety of others. Bonus points are scored for successful grenade attacks on the machine gun positions.

To rescue the hostages you use a captured machine gun against a bewildering array of forces, all intent on killing them. This is not easy, as the tanks and armoured trucks need to be

hit in exactly the right spot to knock them out.

In the escape stage you fly helicopters, loaded with escapees, past a series of obstacles while under constant fire. You can strike back, and you score points for targets destroyed as well as soldiers rescued. The Dragon sets the difficulty level for each of the three games, and the highest level is really hard!

Finally, having tracked the Dragon to his lair, you engage him in battle with wooden spears called Pterotas, from opposite banks of an underground river. You move, duck and jump, and can control the waves in flight, making them lethal. Four hits will be found, none of which must be played in all.

A tremendous game - challenging throughout, with excellent graphics and the best software speech I've heard. It deserves to be a great success!

P.B.E.

## PAINTBOX

Audiogenic

.....



### Q-16 - joystick optional

For those who like graphics without mathematics, PaintBox is a must for Commodore users; it says users as one side of the tape is for the C64 and the other for the C-16.

Loading is fast and efficient and the menu display scrolls smoothly along the bottom of the screen. At any point the menu is instantly accessible by pressing the spacebar. The menu controls a variety of options including saving to tape or disk and a fast erase.

Now what does the program offer? DRAWING is, of course, a standard feature using joystick or keyboard. To this is added:

Line - Allows a line to be drawn at any angle.

Lines - draws a series of lines starting from where the last one finished.

Flays - gives a series of lines starting from the same point.

Framing - provides a rectangle of any given size at any position on the screen, while BOX gives a solid rectangle of any chosen colour. CIRCLE at any place, any size and use DRAG to give a colour filled circle.

This sounds pretty much the same as usual but, believe me, the use of 'rubber-banding' technique gives you tremendous scope and versatility.

When we come to PAINTING the full colour range is available for filling enclosed areas while, in DRAWING mode eight options are available from very fine line to very thick pen which are really excellent. Drawing speed may also be controlled.

For the C64 alone there are extra

facilities allowing you to copy, SWAP and MOVE text pictures around building one upon the other.

Audiogenic has produced here a very good, fast acting, easily handled tool which can be used to create very intricate drawings, picture drawings, poster drawings etc.

PaintBox compares most favourably

with other graphics utilities and, indeed, would be the one of my choice by virtue of its speed, scope, ease of handling and well written manual; screen colour control is likewise excellent.

A joystick is preferable but keyboard control is fully functional and adequate to a beginner and advanced artist. Well recommended.

P.B.E.



## SPEED KING

Digital Integration (3.95)

3 1 7 7 1 5



### CM — joystick optional

If you have quick wits, cool hands and an iron nerve, then this is a game you must try! You can experience the thrill of high-performance motorcycle racing at any of the world's top 18 circuits.

from Brands Hatch to Daytona, at speeds of up to 250 miles per hour. And all this from the comfort of your own armchair!

Each of the circuits is faithfully reproduced, with good graphical representation of the scenery, the track and, of course, the bikes. You race against 19 other riders, starting at the back of the grid. If you collide with another bike you crash, but can get off again. Wandering off the roadway slows you down too, and eventually makes you lose control.

Your machine is equipped with six gears, which you can change up and down, and the resulting effects accompanying gear-changes are most realistic. Holding down the fire button for space bar causes constant acceleration. Releasing it slows you

down, but for heavy braking you will need to change down.

Various options are open to you. In addition to the choice of circuit, you may select novice, pro or champion skill levels, and you are given a preview of the chosen track. You may also set as many practice laps as you wish, before setting off under race conditions. The length of the race may be two, four or six laps. The computer records the total time for the race, and your fastest lap time.

I was impressed with everything about this game. It is very well programmed, fast and exciting. There is plenty of scope for improving your skill by practicing, and plenty of variety. It sets a standard for motorcycle games which will be difficult to beat. So get on your crash-helmet and get started! **P.A.B.**

## RACING DESTRUCTION SET

Amsoft (14.95)

9 1 8 7 1 9

### CM

Have you ever had the urge to get in a powerful car (and race round a track as quickly as possible)? Destruction Set will allow you to indulge all your fantasies and more!

The game consists of two race cars on the same track. Each car's progress is displayed on a separate window on the screen so if one driver gets ahead of his opponent he doesn't disappear off the screen. The cars are faster like slot cars as they will burn corners on their own and the only control required is to change speed and accelerate.

So what is the Construction in the title for? Well this is because RDS allows the user to either choose one of about 20 pre-formed race tracks or even

to make one up himself. This allows all sorts of differing terrains and thus the Game can change from a Formula One at Silverstone to a dirt track rally on the moon!

To enable the cars to race on the different surfaces the user can also control the design and specification of his car. This allows such things as a Concorde Stingray with an 8.2 litre engine right down to a 2000cc Bike.

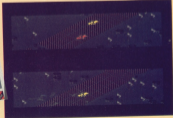
Cars has to be taken when choosing a vehicle as it is very easy to construct a car that is too heavy to get up the steep hills or not powerful enough to get off the line. It is great fun however to pit two entirely different cars against each other on a variable track and see one catch up on the straights due to its power but

then fall behind on the hills due to its weight!

What about the destruction bit. I hear you ask? Well this is great fun as well because the game offers options to carry oil for slowing in the path of your opponent and land mines to blow him up! The problem with both the oil and the mines is that they stay where they are and stay ready as well so if you drop a land mine in a place where your opponent can't miss it he will blow up, but on the next lap if you can't get past it, so will you!

Although the game does a lot of very slow disk accessing, I think that it's excellent and I recommend it.

J.G.B.



## KARTUS

Supersoft £4.95

• • • •



### CM Plus 4 — joystick optional

This is another CM conversion from Supersoft. From memory, I don't think there have been many alterations from the original.

The plot is somewhat unusual in that you must protect a cactus from the unfriendly attentions of a swarm of

wasps and hares. Your efforts are hindered by interfering moss and vultures. In effect this is a shoot-em-up game similar in flavour to Centipede but with its own attraction.

On screen you get a side view of the cactus and the ground level. You may move above or below the ground via holes in the ground. The worms of eggs and hares appear from above and fly from side to side slowly moving downwards. You must ground them from reaching the base of the cactus where they can do the most damage.

Unfortunately, it isn't a simple matter of shooting the wasps. Moss moves from side to side trying to fill in the holes in the ground and cut off your means of access. You, of course, can shoot them. After seven one, vultures fly across dropping eggs. These bounce about and will destroy you on contact.

Graphically, the game is simple with limited use of the CM's colour abilities. The animation of the wasps is quite good but overall the graphics don't really look like the creatures they are supposed to be. Sound is limited to sipping noises although a nice version of *Invincible* Peacher starts the game.

While this game doesn't break new ground in sophistication, it presents a real challenge. The first screen is deceptively easy, but our friends the vultures make life pretty busy later. They seem to aim their eggs accurately and the bouncing bomb behaviour of the eggs is tricky to judge.

Overall, not a bad game especially bearing in mind the lack of material for the CM. The rather primitive graphics do the game no favours but it does offer a good old fashioned rapping session.

M.R.C.

## WORD PERFECT

Supersoft £16.95 disk

• • • •



### CM

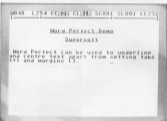
I dare say that a good many owners of the 64 have considered the benefits to be derived from using a word processor but have been deterred by the high price of most of the programs currently available.

Even if you don't own a printer, a word processor can still be extremely useful. How many of you, for instance, can sit down to write a letter and get it right first time without having to retype paragraphs, correct spelling or syntax errors, or even re-write the entire thing, perhaps several times? With the aid of a word processor and a little typing practice the whole task becomes very much easier. Using the screen as a sort of electronic note pad, you can organise your thoughts and correct any errors, all without committing anything to paper.

However, while appreciating the above, many people will be unable to justify the expense of a program which although very useful is somewhat limited in its application in the home.

Word Perfect from Supersoft has recently been released and seems to be aimed at just such a market. The program is available in both disk and tape format, the disk version being the one reviewed here.

Word Perfect is a scaled down word processor and as such it is suited to any of the uses to which you would put a typewriter, but with most of the advantages of word processing. The program loaded quickly and reliably and resides in the protected RAM above Basic starting at 49132. Most of the important features of a word processor are there, although some of them such



as block move and copy are implemented in a somewhat crude fashion. You may search for a specific word or phrase, centre, underline, set tabs and merge previously saved text into the current work. You have the option of saving your finished document to tape or disk.

Although some of the more usual commands are missing, right justify and search with replace for instance, I don't see that their absence will cause too much inconvenience.

The program allows the use of a wide range of printers although, as with all software that uses a printer, it would be

prudent to check that your particular set up works before buying.

Word Perfect comes complete with a well written 16 page booklet that should enable even the novice to quickly get to grips with the program. While perhaps not suited to serious office work, I feel it has a lot to offer the Home user. This review was written with the aid of Word Perfect and I shall continue to use it in the future.

My copy of the manual has an error on page eight. To reformat a paragraph you should press DELETE in CONTROL Mode, not F as stated.

B.J.T.

## MAIL ORDER MONSTERS

AmigaSoft 014.05

10 19 6 8



### 034 *Urban Defenders*

Although the title is different, this is one of the excellent Electronic Arts "Construction" titles.

The game starts by telling you to select a Monster, i.e. a really tough type. This varies from the Human (which is weak and strange) to the like creatures.

Next, weapons must be selected to arm your particular beast. These range from laser rifles and grenades for the

humans to rocket splat from the worms and of course deadly claws from the lizards.

Then the creature's level to the teleport chamber where they are left until the controllers decide what type of combat is to be engaged. One player will decide the terrain, and this should be chosen to give your creature the maximum advantage and your opponents the least. So if an owner has an amphibious creature he would choose a terrain with lots of water thus confining his opponent to the land.

The other owner must try to nullify the disadvantages of the terrain by selecting the type of combat. The first type is the simple demolition combat where each player will fight to the death with the creatures found in the arena and the opponent.

The second requires flags to be selected in order. Each flag is defended by an arena creature.

The third type is called the horde.

This is what both creatures must work together to defeat an invading horde that comes from the top of the screen and is attempting to reach the bottom. The one who kills most horlings wins.

As I mentioned, the arena contains Urban Defenders. In a one player game all these creatures are played by the computer but in a two player game the creature is played by your opponent. So if a creature runs into an urban defender, the screen clears to a control screen and the two players battle to the death. This is done by selecting weapons and timing of disability. This requires a great deal of dexterity and practice is recommended. If the player is victorious in his attack on the urban defender then the screen changes back to the large map and the game goes on.

So praise on this game isn't stunning but some of the construction areas are. The game is a different matter as it is excellent and I found it extremely addictive. **D.S.D.**

## THE GREAT AMERICAN CROSS COUNTRY ROAD RACE

AmigaSoft 03.05

4 17 7 7

### 044

At first glance the Great American Cross Country Road Race - T.G.A.C.C.R. is a fun one on, but it takes too long to write - is like Subnautica's Tallgrass, but after closer scrutiny I found some pleasing additions.

After you've chosen your route, which is at first a touch confusing in regards to not trying to follow the intricacies of a British Rail line later, you are shown the map which roughly shows your route between cities and the weather conditions you are likely to encounter. You can stretch across the great American countryside (there's snow, rain, etc., from 500 sprawling desolate desertification like San Francisco to another like Washington, experiencing sun and snow).

To proceed I chose the US Tour route which, as the name suggests, shows you to be right around the continent.

After the prologue, you're faced with the old hat task, don't which you must guide you low speed speed machine.

You have a certain amount of time to get from one check point to the next and you're expected to change gear using the joystick, while keeping an eye on your fuel. If you allow your RPM to stay in the "red-line" low long your engine overheats but it doesn't take too much skill to avoid this catastrophe.

Fuel consumption is, I found, another story. This particular feature really wound me up. When you run out of gas you have to "pull" your car to the next station. You do this by continuously pressing the fire button - I actually

boke my joystick doing this. I believe that this would not have been necessary if the gas stations were more clearly defined, because if you only just miss one you've got to go to the next one which is 100 miles down the road - infuriating!

The clever touches in this arcade game are what make it different. As you travel the road, conditions change along

with the scenery and it gets dark - I liked that. If you get too fast you attract the attention of the fuzz and they pull you over.

Altogether this is just another road-race game and, although devoid of places with good scrolling graphics, I feel that once you've seen one there isn't a lot of difference in the rest - but this is certainly one of the best. **D.P.**





## PINBALL CONSTRUCTION SET

Activision £14.95



### CS4 • Joyстик

The idea behind the construction series of games is that people could adapt a game which they find to take on many different forms and levels of difficulty to suit to involve their enjoyment. This isn't the case with Pinball C.S.

The game is a representation of a pinball board which can be set up to the user's requirements. All manner of bumpers, flippers and fancy bits may be put together on the board in an attempt to find an interesting game. This is where the construction bit falls down as most fun is to be had in the construction of weird and

wonderful boards. Playing these boards soon becomes very boring indeed! As all the fun is real pinball is in the flashing lights and ping, pong, ping noise.

However, back to the game. As I mentioned, the game allows care of bumpers and flippers. It also has many advanced features such as a ball flipper which collects balls, and it's full at which point it releases them all putting four balls in play at once. Another interesting feature is the AND gate that allows special scores to be awarded for good play.

All in all I was disappointed by this. Nice idea but a boring game. J.G.B.

## THE RATS

Hodder and Stoughton £7.95



### CS4

As reported in the October issue of Your Commodore, Hodder and Stoughton Software has released the computer version of that now age old novel story *The Rats* by James Herbert.

When I got my grubby little paws on, the game I could hardly wait to get it in the CS4 and have a look - you see,

dear reader, I'd read the book and seen the film already and thought they were both terrific - so maybe I expected too much.

In the game you take the roles of various characters in the original story - except the rats, of course. You can, for instance, be anyone from the commander of defense to a civilian who, altogether, adds to the general confusion, and this, in my opinion, is the only atmosphere generated by this attempt to imitate all the aspects of micro-gaming in one package. I think it's a shame that the game doesn't live up to my expectations especially after the stunning graphics that are displayed after the animation sequence.

The game is staged in central London where the rats are swiftly taking over and it's your task, with the help of the defense forces and various elements from the research and development people, to contain the hideous monster

within London, whilst keeping the three main characters alive and not at the expense of every last man jack in your defense forces. To fail in any part results in defeat and some supposedly horrid graphics.

To avert the destruction of the rats you are given various tools and professional forces to help you and you must put into action (perhaps) some of your main character's strategic battle to do this. This is done by showing a cursor again to pick the words (your choice of which is madamingly limited) to build the command. Needless to say, I was shocked very quickly and often.

If it is a good idea, but with the lack of atmosphere and the limitations imposed to the program I found it hard work and I lost my interest very quickly. Like I said, I'd read the book, seen the film, and now I've played the game. I think I would have preferred the part—

D.P.

## KAYAK

Creative Sparks — Sparklers £2.95



### CS4 — Joyстик

If you find it exciting to have a cold bath on a foggy Sunday afternoon in November, while Chopin's funeral march plays in the background, then this is the game for you! According to the rather ungrammatical cassette insert, your

mouth will be dry and your arms shaking with excitement — see, all of it is free!

The idea of the game is that you paddle a canoe in the World White Water Slalom Championship. This involves two forces along a stretch of winding river, with 25 gates to negotiate, endeavouring to keep the white pole to your left at each. The control you have over your craft is minimal. Attempting to turn quickly ends with you ramming the bank, at which point the program is as likely as not to take up altogether.

The "superb scoring 3D" graphics, mentioned on the packaging, are in fact crude and jerky, with the price of your boat bobbing up and down in the foreground and slalom gates appearing as if by magic, just in front of you. Making headless against the current is

practically impossible, and response to the joystick is very slow indeed. The game proceeds at the pace of a snail with coral.

The only sound-effect, apart from a discordant fanfare at the start, is like nothing so much as brushing a pair of shoes, or perhaps a veteran washing-machine. There is a high-score table, of a sort, but even that is multi-coloured. The only text "Will you have the stamina and strength to last the whole course?" - I very much doubt it!

If I seem rather unkind, it is well deserved. This is the poorest game I have seen for a long time, and the best thing to do with it is cover the tabs on the cassette and record something else over the tape. Unless, perhaps, you suffer from insomnia?

P.R.B.

This month Margaret Webb explains how adventure games can be used as a valuable educational aid.

## Pet

Involving the manipulation of spatial concepts.

The mapping operations and the actual solution of the game need a rather special form of skill—the ability to think logically and apply lateral reasoning. A simple example is how to get past a fire-breathing serpent which blocks your way when you're carrying a sword, a bucket of water and wearing running shoes. You could kill the dragon, put his fire out with the water, cut his hair to see your way and find another route.

The problem is to find the correct solution and the means to achieve it using thought, trial and error and patience. All useful skills for real life.

Most of the best adventures use text to convey information and accept instructions. These programs will help improve reading, increase vocabulary, spelling, comprehension and creative writing, but not to an extensive degree.

That's the boring bit out of the way, let us look at what's available. It might be easier to deal with them in rough age groups: this list is not exhaustive, simply a guide.

### Under-tens

**The Magic Sword** (Database Publications) is a clay table book plus a simple adventure. The adventure uses simple text with graphics and the instructions are mostly key commands. You must find the prince (who has been turned into a frog) so that he can save Princess Poppy from the wicked witch.

**Grumpy Mouse in the Black Forest** (Creative Sparks) is a graphics text adventure with many clever comments on culture. There are lots of wily puns and great fun, pretty too. Help Grumpy Mouse save the world from the Pheasants.

### Early Secondary (and Smart Little-Ums)

This age group seems to prefer arcade type action and there are a

range of tough graphics-only games to choose from. All of these require problem solving and lateral thought. Some worthy titles include: **Imperable Adolescence** (Ipp), an arcade style game requiring both physical and mental agility; **Ball of Fire** (Ultimate), a magic type game involving exploration of a tomb and the lighting of fireworks magic; **Blades** (Daphne) in which you guide the fairy on her search for magic potions and plants; **Dragon's Den** of **Be Quickies**, search the dungeons, find the treasure kill the monster; **Shadowline** (Beyond) uses advanced graphics techniques. You control an intergalactic 'A' team trying to rescue an ambassador; **Dr Na Nog** (Cargyle), a Cobi legend in which you help the hero find the trail of Calver's family; **Ultima IV** (US Gold), **Dragons and Demons** style, you lead your band of heroes on a quest.

### Tens to Old-Ums

The most challenging games tend to be text or text/graphics adventures. While graphics help brighten up a game, you should be aware that the pictures only rarely give clues to the solution of the game. The list of such games is endless. There are some excellent software houses who are almost guaranteed to produce excellent games.

Those of note are: **Infocom**—disk only, heavy use of detailed and absorbing text—a sense of human wit. Those worth looking at are **The Dark Elogy** (wizard and dragons), **Sherlock** and **Suspended** (crime fiction) and **Wish Upon a Guide to the Cullin** (horror). **Level Nine**—early games text only, later games use graphics. Complex games with many locations—text compression measures detailed descriptions.

**Adventure International**—not the most complex games but have a very high content of puzzles. Great variety covering many types of scenarios. Later games include **Greenies** and **Incredible Path**.

**Individual games of note are:** **The Hobbit** (Middle-earth House)—on the book, quite tough.

**Seven Kingdoms Valley** (Bug Bytes)—sports graphics, complex and quite tough.

**Fourth Protocol** (Thunderbox)—amazing graphics, month deep thought and a devious mind.

### THIS MONTH I INTEND TO DISCUSS

The value of adventures as educational aids. In contrast to the monthly adventure columnist, with whom I don't seem to compete, I will be advising the uninitiated, not preaching to the converted.

Many parents feel that programs other than educational software cannot be of any benefit to their children. This is simply untrue since we learn something no matter what task we're performing. The nobles about the alien game may seem pointless but it does teach eye-hand co-ordination and helps improve reaction times. Adventures are a more subtle means of combining enjoyment with learning.

Before diving into the material, it would be best to try to describe what is meant by an adventure. Initially, adventures were simply word-games programmed on main frame computers. With the development of home micros, the games have changed form. First the text appeared graphics, then animation and interaction was developed and finally a rich/real time features appeared. All after similar benefits about in differing proportions.

Most adventures involve placing the player, or his alter-ego, in an artificial scenario. By use of various forms of instruction, the player embarks on to solve certain objectives or achieve certain goals. The scenario can basically be anywhere and can be mapped on a piece of paper. A building, for example, can be drawn in terms of rooms and passages.

The first task usually attempted in an adventure is to map, by exploration, the area where you find yourself—just as you would in real life. This means that you may need to know such concepts as the points of the compass and left, right etc. The scenario is normally too complex to memorise, forcing you to draw some form of map. Again this is a tricky task

# RITEMAN:C+

THE ULTIMATE ENHANCEMENT  
TO YOUR COMMODORE®

- 105 cps and bidirectional logic seeking
- Supports Commodore 82-character graphics set, plus 95-character ASCII with true descenders
- Ten print modes, including standard, bold, expanded, condensed, emphasis, italics and super/subscript
- Commodore 64 and VC serial interfaces
- Four international character sets - USA, UK, France and Germany plus Spanish and Scandinavian options
- 4" to 18" wide paper, traction or friction drive
- Super quiet
- One million character ribbon life
- Plus Epson® code compatibility



Advanced design features unique to the Riteman C+ give you big, fast, printer performance with Commodore graphics, in a super-compact, ready to use, portable desktop unit - making it first choice for your Commodore system.

A revolutionary front-loading design eliminates paper load problems, and any weight of paper - even card - can be used.

The Riteman C+ also gives you a super-compact printer footprint. A built-in collapsible stand enables all paper to be stored under the printer, giving you 50% more desk space.

C. Hart Electronics Co. Ltd,  
Riverton House, 26/28 Weybridge Road,  
London TW20 2EL.  
Tel: 01-840 4960, Telex: 0815476L



I want to know more about the Riteman C+, please rush me a datasheet.

(Please indicate: End User  Dealer  OEM

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

# Build a better BASIC

**Nick Hampshire adds more commands to your ever-increasing Basic.**

IN LAST MONTH'S ISSUE I gave all the initialisation and wedge routines needed to add extra commands to the Basic of a C64 computer. Also included in that article was a single command CTL. In this second article I am giving four new commands, APPEND, CHANGE, DUMP and FIND. These are very useful "toolkit" type commands for editing a program and are consequently all used in direct mode.

These four new commands all require the wedge and initialisation code - given last month - to be present in memory at the correct locations and that their command names and entry points are stored in the correct tables. It should also be noted that all these commands use common routines within each other and should therefore always be used in a set. To ensure that you have these correctly positioned the Basic loader at the end of this article is a repeat of last month's with the three new commands added.

## CHANGE

**Abbreviated entry:** CHANG  
**Affected Basic abbreviations:** CHRG - CH (edit)  
**Notes:** Hex 00, 00, 00, Decimal 0000  
**Mode:** Direct and program  
**Recommended modes:** Direct only

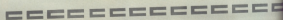
**Purpose:** To change all occurrences of a string or command to something else.

## CHANGE

```

0000 CHANGE C00 000000 ;NEXT LINE LINE
0010 J00 F10014 ;SET CURRENT CHAR
0020 ST0 A01 ;STORE IN FLAG
0030 L00 0000
0040 J00 F10017 ;SET SEARCH STRING
0050 L01 0001
0060 J00 00007 ;SET STRING TO CHANGE
0070 ST0 W01 ;STORE LENGTH OF CHANGE STRING
0080 J00 F10010 ;SETUP POINTERS
0090 SET
0100 L00 0000
0110 ST0 F10000
0120 L00 0000
0130 ST0 F10001
0140 L00 0000
0150 ST0 F10001
0160 L00 0000
0170 ST0 A01
0180 SET
0190 J00 F10006 ;FIND STRING
0200 CHANGE J00 CHANGE ;CHANGE
0210 CHANGE J00 F10013 ;LIST LINE
0220 J00 F10017 ;FIND STRING
0230 J00 CHANGE ;NO REPEAT
0240 ?
0250 CHANGE L00 W01 ;LENGTH OF CHANGE STRING
0260 SET
0270 S00 A01 0 - LENGTH OF FIND
0280 S00 CHANGE ;THEY ARE EQUAL
0290 J00 CHANGE ;ELSE CHANGE W10
0300 CHANGE J00 A01 ;ORDER TO LINE
0310 L00 0000 ;ORDER TO CHANGE STRING
0320 L00 00
0330 W00 0000 ;OUT BASIC ROM
0340 ST0 001
0350 CHANGE L00 0000,0 ;SET CHANGE CHAR
0360 S00 CHANGE ;END OF STRING
0370 ST0 A01 ;REPLACE CHAR
0380 S01 ;NEXT CHAR
0390 S01 ;NEXT BYTE
0400 J00 CHANGE ;NO MATCH
0410 CHANGE L00 001
0420 S00 0000 ;ON BASIC ROM
0430 ST0 001
0440 SET
0450 ST0 A01 ;STORE LINE INDEX
0460 J00 CHANGE ;NO NEXT FIND
0470 ?
0480 CHANGE J00 F10015 ;SET NEXT CHAR
0490 S00 W01 ;END OF FLAG
0500 S00 CHANGE ;YES, GET STRING
0510 J00 0000
0520 CHANGE J00 F10013 ;SET NEXT CHAR
0530 S00 CHANGE ;END OF LINE
0540 S00 W01 ;END OF STRING
0550 S00 CHANGE ;YES
0560 ST0 A01 ;STORE CHAR
0570 SET
0580 S01 0000 ;STRING TOO LONG
0590 S00 CHANGE ;NO
0600 L00 0001 ;STRING TOO LONG
0610 J00 00007 ;OUTPUT CHAR
0620 CHANGE L00 0000 ;STRING TERMINATED
0630 ST0 A01 ;STORE IT
0640 SET
0650 ?
0660 CHANGE L01 0001
0670 L00 0001,0 ;SET LOWER LB
0680 ST0 A01 ;STORE IT
0690 SET
0700 L00 0001,0 ;SET LOWER HC
0710 ST0 A01 ;STORE IT
0720 L00 0000
0730 CHANGE J00
0740 S01 001 ;REACHED STACK
0750 S00 CHANGE ;YES, INSERT IT
0760 L00 0001,0 ;SET PROGRAM BYTE
0770 ST0 0000,0 ;STORE IN BUFFER
0780 SET
0790 S01 0001 ;BUFFER TOO LARGE?
0800 S00 CHANGE ;NOT SET
0810 CHANGE L00 001
0820 W00 0000 ;OUT BASIC ROM
0830 ST0 001
0840 L01 0000
0850 CHANGE L00 0000,0 ;SET CHANGE STRING BYTE
0860 S00 CHANGE ;END OF STRING
0870 ST0 0000,0 ;STORE IN BUFFER
0880 S01 ;NEXT CHAR
0890 S01 ;END OF PROGRAM BYTE
0900 S01 0001 ;END OF BUFFER
0910 S00 CHANGE ;NO
0920 CHANGE L00 001
0930 S00 0001 ;ON BASIC ROM

```





```

1760 STA #01
1765 LDA #02 ;CALCULATE START
1770 CLC ; SET REST OF PROGRAM LINE
1775 ADD #02 ;AFTER INVERTING PH
1780 TRY ;CHANGE STRING
1785 LDA #03
2000 CLC
2010 NEG #02
2020 STA #03
2030 SEC #02
2040 CHRSZ LDA #02;L ;SET PROGRAM BYTE
2050 STA #02;L ;STORE IN BUFFER
2060 JMP #02;L ;JMP
2070 STA #02;L ;STORE IN BUFFER
2080 CMP #00 ;END OF LINE?
2090 BEQ CHRSZ;L ;NO
2100 CPI #02 ;END OF BUFFER
2110 BNE CHRSZ;L ;NOT YET
2120 LDA #00 ;END OF END OF BUFFER
2130 STA #02;L ;STORE IT
2140 JMP #02
2150 CHRSZ STA CHRSZ ;STORE LENGTH OF
2160 TBA ;LINE
2170 CLC
2180 ADD #04
2190 STA #02
2200 LDA #02;L
2210 STA CHRSZ
2220 LDA #02;L
2230 STA CHRSZ+1
2240 LDA #CHRSZ;L ;ASCII ADDR START
2250 STA ADDR ;ASCII ADDR POINT
2260 LDA #CHRSZ;L
2270 STA ADDR
2280 STA ADDR
2290 JMP ADDR ;GO TO NEXT PROGRAM LINE
2300 CHRSZ LDA CHRSZ ;GET ASCII VALUE
2310 STA ADDR ;ASCII ADDR
2320 BEQ CHRSZ;L ;YES
2330 CHRSZ LDA CHRSZ ;GET ASCII VALUE
2340 STA ADDR ;ASCII ADDR
2350 CHRSZ LDA CHRSZ ;GET ASCII VALUE
2360 STA ADDR ;ASCII ADDR
2370 STA ADDR
2380 JMP CHRSZ;L ;GO NEXT WITHOUT LIST
2390 CHRSZ JMP #02;L ;EXIT CHANGE
2400 CHRSZ ;JIT 0
2410 CHRSZ ;JIT 0
2420 ;JIT 0
2430 ;JIT 0

```

Each line that is changed is listed if there is anything left to do.

**Syntax:** CHANGE char delin2d - where d is a delimiter character that does not appear in either of the strings del1 or del2.

**Errors:** Syntax error - if the format is not as above, string too long - if either del1 or del2 are longer than 40 characters. Line CHANGE has a number of uses. An example would be:

CHANGE PRINT PRINTIN.

To change all occurrences of PRINT to PRINTIN, or

CHANGE "PRINT" "PRINTIN."

which will change all occurrences of the text PRINT to the text PRINTIN.

**Note:** Not all delimiter characters will work (all cases, for example).

CHANGE /BINARY/

As the character '/' has two values the first is the token for divide and the second is just the ASCII slash character.

The same is true of DATA. Other characters that will have the same effect are: '%', '=', '!',

## DUMP

```

0000 DUMP LDA #02 ;GET START OF VARIABLES
0010 STA #02 ; AND STORE IN REGISTER
0020 LDA #02 ; LOOKING
0030 STA #02
0040 ;
0050 DUMPI BEQ ;START OF MAIN LOOP
0060 SEC #02 ;END OF VARIABLES?
0070 LDA #02
0080 SEC #02
0090 BEQ DUMPI ;NO
0100 JMP DUMPI ;YES, DISPLAY ADDR +1
0110 ;
0120 DUMPI LDA DUMPI ;GET ADDR NAME
0130 LDA #02 ;REAL?
0140 BEQ DUMPI ;YES
0150 JMP #02 ;FUNCTION?
0160 BEQ DUMPI ;YES
0170 JMP #02 ;FUNCTION?
0180 BEQ DUMPI ;YES
0190 LDA #02 ;MUST BE DEFINED
0200 JMP #02 ;PRINT '0'
0210 JMP DUMPI ;ANY EXTRA SPACES
0220 LDA #02
0230 JMP #02 ;PRINT ' '
0240 LDA #CHRSZ;L ;GET ASCII VALUE
0250 STA ADDR ;ASCII ADDR POINT
0260 LDA #CHRSZ;L
0270 STA ADDR
0280 STA ADDR
0290 JMP ADDR ;GO TO NEXT PROGRAM LINE
0300 CHRSZ LDA CHRSZ ;GET ASCII VALUE
0310 STA ADDR ;ASCII ADDR
0320 BEQ CHRSZ;L ;YES
0330 CHRSZ LDA CHRSZ ;GET ASCII VALUE
0340 STA ADDR ;ASCII ADDR
0350 CHRSZ LDA CHRSZ ;GET ASCII VALUE
0360 STA ADDR ;ASCII ADDR
0370 STA ADDR
0380 JMP CHRSZ;L ;GO NEXT WITHOUT LIST
0390 CHRSZ JMP #02;L ;EXIT CHANGE
0400 CHRSZ ;JIT 0
0410 CHRSZ ;JIT 0
0420 ;JIT 0
0430 ;JIT 0

```

## Baseline entry points 0000

**Baseline operations:** CHANGE uses most of the HAND routines to find del1 and list the line.

CHANGE reads in the delimiter byte and stores it away. The string to be changed is then read in until the second delimiter character is reached and stored. The next character is checked to see that it equals the delimiter character and if so the string to change to is read in until the delimiter character is found again or the end of command.

The rest of the routine is just a loop finding all occurrences, changing them and listing until the end of the program.

The actual routine that changes the string uses the Basic input buffer and the Basic routines to change a line. The routine copies the line up to del1 into the buffer, the change string (del2) is then copied to the buffer and the remainder of the line is copied over, the pointers are then set so that the next byte to check is the one following del2.

## DUMP

**Abbreviated entry:** D[un]D[ump]  
**Affected Basic abbreviations:**  
None  
**Options:** Hex 00, 00C Decimal  
100,10

**Modes:** Direct and program.  
**Recommended mode:** Direct  
**Purpose:** To display the values of all simple variables, name-functions, and display the dimensions of arrays.

**Syntax:** DUMP

**Errors:** None

**Uses:** For debugging basic programs, the DUMP command may be used after the program has run to get a list of all variables and their values. As an added bonus, not found in any other DUMP command for the Commodore 64, all array dimensions are also given. The DUMP command will also display function names.

**Realtime entry point:** \$B0E1

**Realtime operation:** The DUMP routine sets a pointer to the start of variables and checks for the end of variables. If it does not find any, the variable name is read in and displayed, the variable type is determined, and the display is produced according to which type is required. When all simple variables have been processed, arrays are handled. The array names are read and displayed in the same way as the simple variables and the number of dimensions read off. The pointer is then set to the end of the dimension string and, reading backwards, the dimensions are read and displayed.

## FIND

**Abbreviated entry:** F[ind]  
**Affected Basic abbreviations:**  
None  
**Options:** Hex 01,00E Decimal  
100,14

**Modes:** Direct and program.  
**Recommended mode:** Direct only

**Purpose:** To find all occurrences of a string or command inside a basic program.

**Syntax:** FIND string -where str is the delimiter character as in CHANGE

**Errors:** Syntax error - if the syntax is not as above, string too long - if the string is longer than 40 characters.

**Uses:** FIND is another useful routine for debugging and

## DUMP

```
0400 DDP #0004 :PRINT NUM00
0410 DDP #0007 :DD NEXT VAR
0420 :
0430 :FUNCTION
0435 :
0440 DDP%26 DDP DDP%15 :PAD NAME
0450 :DD #FUNCTION :PRINT TO
0460 :DD #FUNCTION :FUNCTION'
0470 DDP NAME :PRINT STRING
0480 DDP DDP%07 :DD NEXT VAR
0490 FUNCTION :BYT ' + FUNCTION',ADD
0500 :
0510 :
0520 :FUNCTION VARIABLE
0530 :
0540 DDP%08 LDD #000 :DDDP TO PRINT 'n '
0550 DDP%02 LDD DDP%16
0560 :DD #DD%02
0570 DDP #000
0580 :DD #DD%02
0590 :DD DDP%16 :PAD FOR NAME
0600 DDP%04 DD
0610 #L DDP%02 :COMPLETE LOOP
0620 :DD #DD%04 :SET ADDRESS OF #DD%04
0630 :DD #DD%17
0640 :DD #DD%02
0650 DD
0660 LDD #DD%17 :LENGTH
0670 DDP #0004 :PRINT #DD%04 FROM #DD%04
0680 :DD #DD%02 :AND LENGTH IN #L
0690 DDP #DD%02 :PRINT ' '
0700 :
0710 :PRINT CARRIAGE RETURN AND DD NEXT
0720 :
0730 DDP%02 LDD #000
0740 :DD #DD%02 :PRINT RETURN
0750 DDP%02 DDP #DD%15 :STOP NEXT
0760 #DD DDP%16 :DD
0770 DDP%02 #DD :DD%17 TO '0000'
0780 DDP%02 LDD #DD%02 TO NEXT VAR
0790 :DD #DD #DD
0800 :DD #DD #DD
0810 :DD #DD #DD
0820 :DD #DD #DD
0830 :DD #DD #DD
0840 :DD #DD #DD
0850 :DD #DD #DD
0860 :DD #DD #DD
0870 :DD #DD #DD
0880 :DD #DD #DD
0890 :DD #DD #DD
0900 :DD #DD #DD
0910 :DD #DD #DD
0920 :DD #DD #DD
0930 :DD #DD #DD
0940 :DD #DD #DD
0950 :DD #DD #DD
0960 :DD #DD #DD
0970 :DD #DD #DD
0980 :DD #DD #DD
0990 :DD #DD #DD
1000 :DD #DD #DD
1010 :DD #DD #DD
1020 :DD #DD #DD
1030 :DD #DD #DD
1040 :DD #DD #DD
1050 :DD #DD #DD
1060 :DD #DD #DD
1070 :DD #DD #DD
1080 :DD #DD #DD
1090 :DD #DD #DD
1100 :DD #DD #DD
1110 :DD #DD #DD
1120 :DD #DD #DD
1130 :DD #DD #DD
1140 :DD #DD #DD
1150 :DD #DD #DD
1160 :DD #DD #DD
1170 :DD #DD #DD
1180 :DD #DD #DD
1190 :DD #DD #DD
1200 :DD #DD #DD
1210 :DD #DD #DD
1220 :DD #DD #DD
1230 :DD #DD #DD
1240 :DD #DD #DD
1250 :DD #DD #DD
1260 :DD #DD #DD
1270 :DD #DD #DD
1280 :DD #DD #DD
1290 :DD #DD #DD
1300 :DD #DD #DD
1310 :DD #DD #DD
1320 :DD #DD #DD
1330 :DD #DD #DD
1340 :DD #DD #DD
1350 :DD #DD #DD
1360 :DD #DD #DD
1370 :DD #DD #DD
1380 :DD #DD #DD
1390 :DD #DD #DD
1400 :DD #DD #DD
1410 :DD #DD #DD
1420 :DD #DD #DD
1430 :DD #DD #DD
1440 :DD #DD #DD
1450 :DD #DD #DD
1460 :DD #DD #DD
1470 :DD #DD #DD
1480 :DD #DD #DD
1490 :DD #DD #DD
1500 :DD #DD #DD
1510 :DD #DD #DD
1520 :DD #DD #DD
1530 :DD #DD #DD
1540 :DD #DD #DD
1550 :DD #DD #DD
1560 :DD #DD #DD
1570 :DD #DD #DD
1580 :DD #DD #DD
1590 :DD #DD #DD
1600 :DD #DD #DD
1610 :DD #DD #DD
1620 :DD #DD #DD
1630 :DD #DD #DD
1640 :DD #DD #DD
1650 :DD #DD #DD
1660 :DD #DD #DD
1670 :DD #DD #DD
1680 :DD #DD #DD
1690 :DD #DD #DD
1700 :DD #DD #DD
1710 :DD #DD #DD
1720 :DD #DD #DD
1730 :DD #DD #DD
1740 :DD #DD #DD
1750 :DD #DD #DD
1760 :DD #DD #DD
1770 :DD #DD #DD
1780 :DD #DD #DD
1790 :DD #DD #DD
1800 :DD #DD #DD
1810 :DD #DD #DD
1820 :DD #DD #DD
1830 :DD #DD #DD
1840 :DD #DD #DD
1850 :DD #DD #DD
1860 :DD #DD #DD
1870 :DD #DD #DD
1880 :DD #DD #DD
1890 :DD #DD #DD
1900 :DD #DD #DD
1910 :DD #DD #DD
1920 :DD #DD #DD
1930 :DD #DD #DD
1940 :DD #DD #DD
1950 :DD #DD #DD
1960 :DD #DD #DD
1970 :DD #DD #DD
1980 :DD #DD #DD
1990 :DD #DD #DD
```

## DUMP

```

0100 CLR
0110 ADD #N001,007 POINTER TO END
0120 STB #01,01 OF ARRAY ENTRY FOR
0130 LDB #01,01 DISPLAY OF SIZE
0140 ADD #N001
0150 STB #01
0160 LDB #N01
0170 LDB #N01," OF DIMENSIONS
0180 STB #01
0190 LDB #N001
0200 STB #01
0210 LDB #N001
0220 STB #01
0230 LDB #01 #TIMES 2
0240 STB #01
0250 LDB #01 #PLUS END VALUE
0260 CLR
0270 ADD #01
0280 STB #01
0290 LDB #01
0300 ADD #01
0310 STB #01
0320 STB #01
0330 LDB #N01
0340 STB #N01
0350 LDB #N01
0360 STB #N01
0370 STB #N01
0380 LDB #N01
0390 STB #N01
0400 LDB #N01
0410 STB #N01
0420 STB #N01
0430 LDB #N01
0440 STB #N01
0450 STB #N01
0460 STB #N01
0470 STB #N01
0480 STB #N01
0490 STB #N01
0500 STB #N01
0510 STB #N01
0520 STB #N01
0530 STB #N01
0540 STB #N01
0550 STB #N01
0560 STB #N01
0570 STB #N01
0580 STB #N01
0590 STB #N01
0600 STB #N01
0610 STB #N01
0620 STB #N01
0630 STB #N01
0640 STB #N01
0650 STB #N01
0660 STB #N01
0670 STB #N01
0680 STB #N01
0690 STB #N01
0700 STB #N01
0710 STB #N01
0720 STB #N01
0730 STB #N01
0740 STB #N01
0750 STB #N01
0760 STB #N01
0770 STB #N01
0780 STB #N01
0790 STB #N01
0800 STB #N01
0810 STB #N01
0820 STB #N01
0830 STB #N01
0840 STB #N01
0850 STB #N01
0860 STB #N01
0870 STB #N01
0880 STB #N01
0890 STB #N01
0900 STB #N01
0910 STB #N01
0920 STB #N01
0930 STB #N01
0940 STB #N01
0950 STB #N01
0960 STB #N01
0970 STB #N01
0980 STB #N01
0990 STB #N01
1000 STB #N01

```

## FIND

```

1000 FIND 200 FIND1,007 CHARACTER
1010 STB #01,0100 IN FLAG
1020 LDB #N001
1030 JMP FIND001,007 SEARCH STRING
1040 JMP FIND002,007 OF POINTERS
1050 STB
1060 LDB #N001
1070 STB #N001
1080 LDB #N001
1090 STB #N001
1100 LDB #N001
1110 STB #N001
1120 LDB #N001
1130 STB #N001
1140 STB #N001
1150 STB #N001
1160 STB #N001
1170 STB #N001
1180 STB #N001
1190 STB #N001
1200 STB #N001
1210 STB #N001
1220 STB #N001
1230 STB #N001
1240 STB #N001
1250 STB #N001
1260 STB #N001
1270 STB #N001
1280 STB #N001
1290 STB #N001
1300 STB #N001
1310 STB #N001
1320 STB #N001
1330 STB #N001
1340 STB #N001
1350 STB #N001
1360 STB #N001
1370 STB #N001
1380 STB #N001
1390 STB #N001
1400 STB #N001
1410 STB #N001
1420 STB #N001
1430 STB #N001
1440 STB #N001
1450 STB #N001
1460 STB #N001
1470 STB #N001
1480 STB #N001
1490 STB #N001
1500 STB #N001
1510 STB #N001
1520 STB #N001
1530 STB #N001
1540 STB #N001
1550 STB #N001
1560 STB #N001
1570 STB #N001
1580 STB #N001
1590 STB #N001
1600 STB #N001
1610 STB #N001
1620 STB #N001
1630 STB #N001
1640 STB #N001
1650 STB #N001
1660 STB #N001
1670 STB #N001
1680 STB #N001
1690 STB #N001
1700 STB #N001
1710 STB #N001
1720 STB #N001
1730 STB #N001
1740 STB #N001
1750 STB #N001
1760 STB #N001
1770 STB #N001
1780 STB #N001
1790 STB #N001
1800 STB #N001
1810 STB #N001
1820 STB #N001
1830 STB #N001
1840 STB #N001
1850 STB #N001
1860 STB #N001
1870 STB #N001
1880 STB #N001
1890 STB #N001
1900 STB #N001
1910 STB #N001
1920 STB #N001
1930 STB #N001
1940 STB #N001
1950 STB #N001
1960 STB #N001
1970 STB #N001
1980 STB #N001
1990 STB #N001
2000 STB #N001

```

Checking Basic programs, for example:

## FIND PRINT

which will find and list all lines containing the command PRINT. If PRINT occurs more than once on a line, the line will be listed each time it is found with the exception of the last line where the line will be listed only once.

**Routine entry point:** #0001  
**Routine operations:** The string to be found is read in within quotes, including spaces and colons and stored away. The rest of the program is a loop that searches the program until the string has been found, lists the line, and starts searching from the next character.

The error message vector is stored away and replaced with a jump to an RTS so that LIST will return to the routine.

## APPEND

**Abbreviated entry:** #4341P  
**Altered Basic abbreviations:**  
 #N01A

**Tokens:** Hex \$11,801 Decimal 2561

**Mode:** Direct and program

**Recommended mode:** Direct

**Purpose:** To load a program into memory so that it appears "on top" of the current program. This routine will work with both disk and cassette and the variable pointers when loaded are set to the end of the combined program. When this routine is used, you should check that the line numbers of the APPENDED program are larger than the line numbers of the program in memory.

**Syntax:** APPEND [filename],  
 [d],  
 [s] - where d is the device number and s is the secondary address.

**Errors:** The same errors will be encountered as in the Basic command LOAD.

**Use:** This routine would be used mostly to add Basic library routines onto the end of your programs. It would be used rather than MERGE because APPEND is much faster.

**Routine entry point:** #0002

**Routine operations:** The APPEND routine uses LOAD's parameter parsing routine to get the filename etc. then set the secondary address so that it loads at the end of the Basic program in memory. The load

routine is then called and the program re-compiled and variable pointers set.

These extended Basic routines are all taken from the book *Advanced Commodore 64 Basic Revealed* by Nick Humphreys, published by Collins.



## FIND

```

1200 STR WSTR0,0 :STORE IN SEARCH STRING
1210 END
1220 CPU WSTR0 :STRING TOO LONG?
1230 BND FIND0 :NO
1240 LDA WSTR :STRING TOO LONG
1250 JMP WSTRT :INPUT ERROR
1260 FIND0 LDA WSTR :PERMUTATE TO STRING
1270 STR WSTR0,0 :STORE IT
1280 STR WSTR :STORE STRING LENGTH
1290 RND WSTR
1300 :
1310 FIND00 LDA WSTR :GET START OF PROGRAM
1320 CLC
1330 ADC WSTR :PLUS 2
1340 STR WSTR
1350 LDA WSTR :GET START OF PROG END
1360 WDC WSTR
1370 STR WSTR :STORE IT
1380 WDC
1390 FIND00 LDA WSTR :INDEX TO STRING
1400 LDA WSTR :INDEX TO LINE
1410 STR WSTR
1420 FIND0T LDA WDC
1430 WDC WSTR :GET BASIC ROM
1440 STR WDC
1450 LDA WSTRT,F :GET BYTE
1460 BND FIND00 :END PROGRAM LONG OF LINE
1470 CPU WSTR0,0 :SAME AS STRING?
1480 JMP
1490 LDA WDC
1500 ORA WSTR :IN BASIC ROM
1510 STR WDC
1520 JMP
1530 BND FIND00 :NOT MATCHED
1540 END :GET BYTE
1550 ORA WSTR :SAME
1560 CPU WDC :STRING MATCHES?
1570 BND FIND00T :NO
1580 STR WDC
1590 FIND00T LDA WDC :START AT NEXT BYTE
1600 LDA WDC
1610 LDA WSTR0 :AND START OF STRING
1620 LDA WSTRT,F :GET BYTE
1630 WDC FIND00T :END OF LINE
1640 JMP FIND0T :TRY AGAIN
1650 FIND0T LDA WDC
1660 ORA WSTR :IN BASIC ROM
1670 STR WDC
1680 ORA WDC
1690 WDC WSTR
1700 STR WDC :GET NEXT LINE
1710 FIND0T CLC
1720 LDA FIND00 :CHECK BRANCH LINE
1730 STR WDC
1740 LDA FIND0T+1
1750 STR WDC
1760 CLC
1770 JMP WSTRT,F :GET NEXT LINE
1780 FIND0T CLC
1790 LDA FIND00 :CHECK BRANCH LINE
1800 STR WDC
1810 LDA FIND0T+1
1820 STR WDC
1830 CLC
1840 JMP WSTRT,F :GET NEXT LINE
1850 :
1860 FIND0T LDA WSTR
1870 ORA WSTR :SAME POINTERS
1880 LDA WSTR :COUNTER UP
1890 STR WSTR :POINT IT
1900 LDA WSTRT,F :GET LINE0 LD
1910 STR WSTR :STORE IT
1920 JMP
1930 LDA WSTRT,F :GET LINE0 LD

```

## APPEND

```

1000 APPEND LDA #000          1180 SEC ; POINTERS TO END OF
1001 STA #004                1181 SEC #002 ; NEW PROGRAM
1002 JBR #C04 ; SET FILE PARAMETERS 1200 STA #07
1003 LDA #000                1201 TFR
1004 STA #001 ; SET SA FOR ALT LOAD 1220 SEC #000
1005 LDA #02                 1230 STA #07*0
1006 SEC                    1240 RESUME LDA #000 ; FIND END OF PROGRAM
1007 SEC #001 ; SET LOAD ADDRESS 1250 LDA #C07.7 ; AND SET VARIABLE
1008 TAC ; DIRECTLY AFTER RESUME 1260 END RESUME ; POINTERS
1009 LDA #00+1 ; PROGRAM.      1270 JRF
1010 SEC #000                1280 LDA #C07.7
1011 TAF                    1290 END RESUME
1012 LDA #00                 1300 LDA #07
1013 JBR #T00 ; LOAD        1310 CLC
1014 ;                      1320 ADD #001
1015 RESUME JBR #C07 ; 80-CARD LINES 1330 STA #02
1016 LDA #00                 1340 STA #07
1017 LDA #00+1 ; RESET VARIABLE 1350 STA #02

```

## BASIC LOADER

```

1000 REM *****
1010 REM #BASIC LOADER FOR THE BASIC #
1020 REM #EXTENSION PACKAGE #
1030 REM #INCLUDES MESSAGES AND THE #
1040 REM #COMMANDS #
1050 REM #APPEND, CHANGE, CTL, DUMP AND#
1060 REM #FIND #
1070 REM #COPYRIGHT 20,8,85 #
1080 REM #NECK HAMPSHIRE #
1090 REM *****
200 I=1;X=0;L=32768
210 READA I#A-999THENX300
220 POKEI,R
230 L=L+1;I=I+1;X=X+A
240 GOTO210
260 IF(X>32767)THENPRINT"NUMBER OF
DATA ENTRIES ERROR "I" SHOULD
BE 2907" END
310 IF(X<341614) THENPRINT"CHECKSUM
ERROR, VALUE "X" SHOULD BE
341614" END
320 REM TO RUN ROUTINES SYS(64720)
330 REM
340 END
1000 DATA122,128,57,128,195,194,205
1010 DATA56,49,139,227,131,164,201
1020 DATA129,158,138,247,139,59,131
1030 DATA76,72,178,8,49,234,60
1040 DATA128,71,254,74,243,145,242
1050 DATA14,242,88,242,51,243,241
1060 DATA131,280,241,237,246,62,241
1070 DATA47,243,68,129,165,244,237
1080 DATA245,32,188,246,32,225,255

```



```

1090 DATA240,3,76,114,254,32,169
1100 DATA253,32,24,229,32,93,128
1110 DATA32,284,255,169,8,133,19
1120 DATA32,122,160,88,162,128,76
1130 DATA136,227,162,21,168,129,134
1140 DATA135,132,196,169,25,177,156
1150 DATA153,16,3,136,16,249,169
1160 DATA118,160,13,141,143,2,140
1170 DATA144,2,96,142,22,208,32
1180 DATA163,253,32,88,253,32,91
1190 DATA255,32,93,128,88,32,229,7
1200 DATA128,32,191,227,169,128,133
1210 DATA32,133,54,133,56,169,0
1220 DATA133,51,133,53,133,53,169
1230 DATA172,168,128,32,45,228,162
1240 DATA251,154,209,173,147,13,32
1250 DATA32,32,32,42,42,42,42
1260 DATA32,69,88,84,69,78,68
1270 DATA69,68,32,84,92,32,68
1280 DATA65,83,73,67,32,82,48
1290 DATA49,32,42,42,42,42,13
1300 DATA13,32,54,52,73,32,62
1310 DATA65,77,32,83,69,83,64

```

**BASIC LOADER**

```

1300 DATA69.77.32.32.0.162.11
1300 DATA169.9.128.157.0.3.202
1340 DATA16.247.96.82.85.206.67
1350 DATA94.204.65.80.80.69.70
1360 DATA196.63.85.84.207.67.63
1370 DATA94.65.76.79.199.67.72
1380 DATA65.78.71.197.67.72.65
1390 DATA73.206.67.82.85.78.67
1400 DATA200.69.69.76.69.84.197
1410 DATA68.73.83.203.68.79.75
1420 DATA197.69.65.77.208.69.80
1430 DATA69.195.79.73.78.196.71
1440 DATA69.212.73.69.217.77.63
1450 DATA212.77.69.82.71.197.79
1460 DATA76.196.80.79.200.80.85
1470 DATA212.82.69.79.85.77.66
1480 DATA69.218.82.69.80.69.65
1490 DATA212.83.79.82.212.84.82
1500 DATA65.67.69.73.206.84.82
1510 DATA69.67.69.79.70.190.84
1520 DATA69.68.197.85.78.84.73
1530 DATA204.68.69.69.203.72.73
1540 DATA77.69.205.76.79.77.69
1550 DATA205.86.65.82.88.84.210
1560 DATA0.36.139.138.138.197.130
1570 DATA39.139.42.139.116.134.45
1580 DATA39.48.139.51.139.54.139
1590 DATA57.139.187.139.68.139.139
1600 DATA137.63.139.66.139.69.139
1610 DATA72.139.75.139.78.139.61
1620 DATA39.84.139.87.139.90.139
1630 DATA93.139.96.139.99.139.102
1640 DATA39.105.139.108.139.111.139
1650 DATA114.139.106.132.160.4.132
1660 DATA15.169.0.2.16.7.201
1670 DATA255.240.43.232.208.244.201
1680 DATA32.240.36.139.0.201.34
1690 DATA248.71.36.15.112.26.201
1700 DATA23.208.4.169.153.208.18
1710 DATA201.48.144.4.201.68.144
1720 DATA10.76.70.130.169.238.44
1730 DATA5.11.164.113.232.208.153
1740 DATA251.1.201.238.248.49.185
1750 DATA251.1.248.34.56.233.98
1760 DATA248.4.201.73.208.2.133
1770 DATA15.56.233.85.208.174.133
1780 DATA9.189.0.2.248.219.197
1790 DATA6.248.215.208.153.251.1
1800 DATA232.208.248.153.251.1.196
1810 DATA133.169.255.133.122.36.165
1820 DATA11.208.193.251.1.76.207
1830 DATA129.132.113.168.255.134.122
1840 DATA202.169.1.133.11.200.232
1850 DATA189.0.2.56.249.241.128
1860 DATA248.245.201.128.248.156.169
1870 DATA122.250.11.208.165.248.128
1880 DATA16.250.185.241.128.208.228
1890 DATA160.0.132.11.136.166.122
1900 DATA282.200.232.189.0.3.96
1910 DATA249.158.168.240.245.201.128
1920 DATA288.3.76.255.129.166.122
1930 DATA288.11.208.185.157.168.16
1940 DATA258.185.158.168.288.225.189
1950 DATA0.2.76.1.130.48.3
1960 DATA76.243.168.201.255.240.249
1970 DATA36.15.48.245.201.238.240
1980 DATA3.32.217.130.48.3.32
1990 DATA186.130.76.239.166.200.177
2000 DATA95.170.132.73.168.255.202
2001 =>N=0
2010 DATA240.8.200.185.241.128.16
2020 DATA258.48.245.208.185.241.128
2030 DATA69.5.32.218.255.208.245
2040 DATA96.56.233.127.178.132.73
2050 DATA160.255.262.248.8.200.185
2060 DATA158.160.16.250.48.245.200
2070 DATA185.158.168.48.238.32.218
2080 DATA255.208.245.32.115.0.201
2090 DATA238.240.18.201.153.240.38
2100 DATA32.121.0.76.231.167.32
2110 DATA94.131.76.174.167.208.122
2120 DATA208.2.230.123.168.0.177
2130 DATA128.56.233.1.18.168.185
2140 DATA98.129.72.185.137.129.72
2150 DATA76.115.0.32.46.131.76
2160 DATA174.167.173.58.131.76.173
2170 DATA57.131.72.76.115.0.235
2180 DATA32.169.0.133.13.32.115
2190 DATA201.238.240.6.32.121
2200 DATA0.76.141.174.238.122.209
2210 DATA2.230.123.168.0.177.122
2220 DATA201.29.176.9.76.0.175
2230 DATA133.36.169.173.72.169.140
2240 DATA72.198.36.169.36.18.178
2250 DATA189.198.129.72.189.137.149
2260 DATA72.76.185.0.165.157.240
2270 DATA16.163.1.36.212.208.10
2280 DATA165.208.201.3.144.4.201
2290 DATA7.144.3.76.72.235.197
2300 DATA197.240.249.169.0.133.232
2310 DATA133.251.169.1.44.141.2
2320 DATA240.4.169.32.133.251.169
2330 DATA151.133.252.169.192.24.181
2340 DATA251.133.251.165.209.201.3
2350 DATA208.4.169.24.208.18.201
2360 DATA6.208.4.169.16.208.18
2370 DATA281.5.208.4.169.8.208
2380 DATA2.169.0.4.201.251.133
2390 DATA1.160.0.169.94.133.1
2400 DATA177.251.240.8.153.119.2
2410 DATA200.192.8.208.244.132.198

```



## BASIC LOADERS

```

2420 DATA63.95,133.1,165.203,133
2430 DATA197,173,141.2,141,142.2
2440 DATA96,165,159,209,4,149,157
2450 DATA289,3,76,87,244,165,211
2460 DATA133,202,165,214,133,201,152
2470 DATA72,138,72,165,209,240,6
2480 DATA76,98,230,32,22,231,163
2490 DATA198,133,204,141,146,2,240
2500 DATA247,120,165,207,240,12,163
2510 DATA206,174,135,2,160,0,132
2520 DATA207,32,19,234,32,180,229
2530 DATA201,131,200,16,162,9,120
2540 DATA134,198,163,230,236,157,110
2550 DATA2,202,208,247,240,207,201
2560 DATA13,200,3,76,2,230,201
2570 DATA17,200,133,166,214,224,24
2580 DATA349,3,76,15,132,162,24
2590 DATA160,0,34,32,240,255,230
2600 DATA20,200,2,230,21,32,19
2610 DATA166,160,1,177,95,200,16
2620 DATA163,205,133,20,133,21,169
2630 DATA185,160,132,32,30,171,76
2640 DATA18,132,160,2,177,95,133
2650 DATA20,200,177,95,133,21,160
2660 DATA162,141,0,3,169,132,141
2670 DATA1,3,104,141,163,132,104
2680 DATA141,104,132,160,1,132,15
2690 DATA76,215,166,169,139,141,0
2700 DATA3,169,227,141,1,3,173
2710 DATA104,132,7,173,163,132,72
2720 DATA76,10,132,0,1,13,13
2730 DATA10,42,42,42,42,42,42
2740 DATA42,42,42,42,42,42,32
2750 DATA69,78,68,32,79,70,32
2760 DATA90,82,79,71,92,65,77
2770 DATA32,42,42,42,42,42,42
2780 DATA42,42,42,42,42,42,13
2790 DATA0,32,33,171,32,121,0
2800 DATA240,98,240,94,201,163,240
2810 DATA107,201,166,24,240,162,201
2820 DATA230,200,20,160,1,177,122
2830 DATA201,2,208,12,32,115,0
2840 DATA32,115,0,32,139,133,76
2850 DATA233,132,32,121,0,201,44
2860 DATA240,55,201,59,240,97,32
2870 DATA150,173,36,13,49,193,32
2880 DATA221,169,32,135,100,32,33
2890 DATA171,32,59,171,200,104,169
2900 DATA0,157,0,2,162,255,160
2910 DATA1,163,19,200,16,169,13
2920 DATA32,71,171,36,15,16,5
2930 DATA165,10,32,71,171,73,255
2940 DATA96,56,32,240,255,152,56
2950 DATA233,10,176,252,73,255,165
2960 DATA1,200,25,0,56,32,240
2970 DATA255,132,9,32,153,169,201
2980 DATA41,240,3,76,0,175,40
2990 DATA144,6,138,229,9,144,5
3000 DATA170,232,202,200,6,32,113
3010 DATA0,76,230,132,32,59,171
3020 DATA208,242,76,30,171,32,12
3030 DATA134,32,250,174,32,121,0
3040 DATA32,49,134,176,0,32,69
3050 DATA134,142,111,134,176,66,32
3060 DATA46,134,176,0,32,70,134
3070 DATA142,112,134,176,53,32,46
3080 DATA134,176,0,32,72,134,142
3090 DATA113,134,176,40,32,46,134
3100 DATA176,0,32,72,134,142,114
3110 DATA134,176,27,32,46,134,176
3120 DATA0,32,72,134,142,115,134
3130 DATA176,14,32,46,134,144,3
3140 DATA76,8,175,32,75,134,142
3150 DATA116,134,32,247,174,173,116
3160 DATA134,240,5,169,147,32,32
3170 DATA231,173,113,134,141,134,2
3180 DATA173,114,134,141,33,200,173
3190 DATA115,134,141,32,200,172,111
3200 DATA134,174,112,134,24,76,240
3210 DATA355,56,32,240,255,140,111
3220 DATA134,142,112,134,173,33,200
3230 DATA141,114,134,173,32,200,114
3240 DATA115,134,173,134,2,141,113
3250 DATA134,169,0,141,116,134,96
3260 DATA32,115,0,201,44,200,2
3270 DATA56,96,201,41,240,2,24
3280 DATA96,104,104,32,115,0,76
3290 DATA230,133,169,40,44,169,16
3300 DATA44,169,2,44,169,25,141
3310 DATA110,134,32,150,163,236,110
3320 DATA134,176,14,32,121,0,201
3330 DATA41,240,211,201,44,240,213
3340 DATA76,0,175,162,14,76,55
3350 DATA64,0,0,0,0,0,0
3360 DATA0,32,223,130,32,134,130
3370 DATA133,89,162,0,32,100,137
3380 DATA162,0,32,223,134,134,252
3390 DATA32,218,137,120,173,0,3
3400 DATA141,196,138,173,1,3,141,3
3410 DATA197,130,169,92,141,0,3
3420 DATA169,130,141,1,3,89,32
3430 DATA232,137,76,179,134,32,93
3440 DATA130,32,230,137,76,167,134
3450 DATA165,252,56,225,34,240,3
3460 DATA76,5,129,164,35,162,64
3470 DATA165,1,41,254,133,1,169
3480 DATA64,191,240,7,145,67,232
3490 DATA200,76,199,134,163,1,9
3500 DATA1,133,1,136,132,25,76
3510 DATA170,134,32,120,130,197,89
3520 DATA240,3,76,0,175,32,120
3530 DATA138,240,17,197,89,240,13

```

## BASIC LOADER

```

3540 DATA157,128,131,232,224,64,288
3550 DATA235,162,23,76,55,164,169
3560 DATA98,197,128,191,96,168,8
3570 DATA177,87,133,20,208,177,87
3580 DATA133,21,162,8,288,196,35
3590 DATA249,18,177,87,157,8,2
3600 DATA232,224,88,208,241,165,1
3610 DATA1,254,133,1,168,8,185
3620 DATA128,191,248,9,157,8,2
3630 DATA232,208,224,87,288,242,165
3640 DATA1,9,1,133,1,165,35
3650 DATA24,181,34,168,165,35,24
3660 DATA181,252,133,35,198,35,177
3670 DATA87,157,8,2,288,232,281
3680 DATA8,248,18,224,88,288,241
3690 DATA169,8,157,8,2,232,142
3700 DATA185,135,138,24,185,4,133
3710 DATA11,173,2,3,141,186,135
3720 DATA173,3,3,141,187,135,169
3730 DATA136,141,2,3,169,135,141
3740 DATA3,3,32,139,138,164,11
3750 DATA76,164,164,173,186,135,141
3760 DATA2,3,173,187,135,141,3
3770 DATA3,32,165,138,165,87,197
3780 DATA45,288,8,165,88,197,46
3790 DATA248,19,173,185,135,281,1
3800 DATA248,3,76,178,134,168,2
3810 DATA132,35,162,8,76,173,134
3820 DATA76,75,138,8,8,8,165
3830 DATA46,133,96,165,45,133,95
3840 DATA56,229,47,165,96,229,48
3850 DATA144,3,76,167,136,32,127
3860 DATA136,165,37,248,43,281,1
3870 DATA248,71,281,2,248,92,169
3880 DATA37,32,218,255,32,157,136
3890 DATA169,61,32,218,255,168,2
3900 DATA177,95,78,288,177,95,168
3910 DATA184,32,149,179,32,221,189
3920 DATA32,38,171,76,89,136,169
3930 DATA32,32,218,255,32,157,136
3940 DATA169,61,32,218,255,32,133
3950 DATA177,165,78,164,72,32,162
3960 DATA187,32,221,189,32,218,189
3970 DATA78,99,136,32,157,136,169
3980 DATA47,168,136,32,38,171,76
3990 DATA99,136,32,61,32,78,85
4000 DATA78,67,84,73,79,78,8
4010 DATA162,3,189,132,137,32,218
4020 DATA255,224,3,288,3,32,157
4030 DATA136,282,16,248,168,4,177
4040 DATA95,133,35,136,177,95,133
4050 DATA34,136,177,95,32,36,171
4060 DATA169,34,32,218,255,169,13
4070 DATA32,218,255,32,225,255,288
4080 DATA1,96,24,165,95,165,7
4090 DATA133,95,166,96,144,1,232
4100 DATA134,96,76,196,135,168,8
4110 DATA132,37,288,177,95,18,38
4120 DATA37,74,153,69,8,136,16
4130 DATA244,165,69,32,218,255,165
4140 DATA78,248,3,32,218,255,96
4150 DATA165,78,288,5,169,32,32
4160 DATA218,255,96,169,13,32,218
4170 DATA225,165,47,133,95,165,48
4180 DATA133,96,165,96,197,98,288
4190 DATA6,165,95,197,48,248,173
4200 DATA32,225,255,248,168,32,127
4210 DATA136,165,37,248,18,281,2
4220 DATA288,3,169,36,44,169,37
4230 DATA44,163,32,32,218,255,32
4240 DATA197,136,165,32,32,218,255
4250 DATA169,48,32,218,255,165,95
4260 DATA24,185,3,133,291,165,96
4270 DATA185,8,133,252,188,1,177
4280 DATA251,133,252,169,8,133,254
4290 DATA6,253,88,254,165,253,24
4300 DATA181,251,133,253,165,254,181
4310 DATA252,133,254,168,8,177,253
4320 DATA141,131,137,288,177,253,141
4330 DATA138,137,288,3,288,131,137
4340 DATA288,138,137,173,131,137,174
4350 DATA138,137,164,95,148,138,137
4360 DATA164,96,148,131,137,32,285
4370 DATA189,172,138,137,132,95,172
4380 DATA131,137,132,96,56,165,253
4390 DATA232,2,133,253,165,254,233
4400 DATA8,133,254,197,252,288,6
4410 DATA165,253,197,251,248,8,169
4420 DATA44,32,218,255,76,16,137
4430 DATA168,3,177,95,133,251,136
4440 DATA177,95,24,181,95,133,95
4450 DATA165,96,181,251,133,96,169
4460 DATA41,32,218,255,169,17,32
4470 DATA218,255,76,188,136,8,8
4480 DATA34,32,61,36,32,134,138
4490 DATA133,69,162,8,32,188,137
4500 DATA32,218,137,128,173,8,3
4510 DATA141,196,138,173,1,3,141
4520 DATA197,138,169,92,141,8,3
4530 DATA169,138,141,1,3,88,32
4540 DATA232,137,32,93,136,32,238
4550 DATA137,76,176,137,76,8,175
4560 DATA32,128,138,248,248,197,83
4570 DATA248,13,157,64,191,252,224
4580 DATA64,288,239,162,23,78,55
4590 DATA164,169,8,157,64,191,134
4600 DATA34,96,165,43,24,185,2
4610 DATA133,87,165,44,185,8,133
4620 DATA88,96,162,8,168,2,132
4630 DATA88,165,1,41,254,133,1
4640 DATA177,87,248,33,221,64,191
4650 DATA8,165,1,9,1,133,1

```



## BASIC LOADS

4660	DAT148.208.7.208.232.228.34	4920	DAT889.173.195.138.133.252.96
4670	DAT208.227.96.230.5.164.25	4930	DAT8.0.0.0.0.0.0
4680	DAT162.0.177.87.240.3.76	4940	DAT169.0.133.10.32.212.225
4690	DAT239.137.165.1.9.1.133	4950	DAT169.0.133.165.165.45.56
4700	DAT1.165.87.56.233.2.133	4960	DAT239.2.179.165.46.233.9
4710	DAT97.165.88.233.0.133.88	4970	DAT169.165.10.32.213.255.32
4720	DAT168.0.177.87.133.89.288	4980	DAT51.165.165.45.164.45.56
4730	DAT177.87.133.88.5.89.248	4990	DAT233.2.133.87.152.233.9
4740	DAT16.165.83.24.165.2.133	5000	DAT133.88.168.0.177.87.288
4750	DAT97.165.88.165.0.133.88	5010	DAT27.288.177.87.289.22.165
4760	DAT76.232.137.128.173.196.138	5020	DAT87.24.185.2.133.45.133
4770	DAT141.0.3.173.197.138.141	5030	DAT47.133.49.165.88.165.0
4780	DAT1.3.88.76.116.164.96	5040	DAT133.46.133.48.133.50.96
4790	DAT168.0.32.139.138.169.145	5050	DAT168.0.177.87.133.89.288
4800	DAT32.218.255.177.87.133.28	5060	DAT177.87.133.88.165.89.133
4810	DAT288.177.87.133.21.32.19	5070	DAT87.76.242.138.76.113.168
4820	DAT166.32.281.166.32.165.138	5080	DAT76.8.175.76.8.175.76
4830	DAT288.33.164.35.162.0.96	5090	DAT8.175.76.8.175.76.0
4840	DAT248.122.288.2.238.123.168	5100	DAT175.76.8.175.76.8.175
4850	DAT8.177.122.96.165.34.141	5110	DAT76.8.175.76.8.175.76
4860	DAT191.138.165.35.141.192.138	5120	DAT8.175.76.8.175.76.0
4870	DAT163.87.141.193.138.165.88	5130	DAT175.76.8.175.76.8.175
4880	DAT141.194.138.165.252.141.135	5140	DAT76.8.175.76.8.175.76
4890	DAT138.36.173.191.138.133.34	5150	DAT8.175.76.8.175.76.8
4900	DAT173.192.138.133.35.173.193	5160	DAT175.76.8.175.76.8.175
4910	DAT138.133.87.173.194.138.133	5170	DAT76.8.175.76.8.175.76.999

## COLLINS - THE BEST IN COMPUTING

Advanced Commodore 64 BASIC Revealed  
Nick Hampshire

Although it is relatively easy to learn to program the Commodore 64 in BASIC, advanced programmers need to know much more than how to use the fundamental commands. This book explains all these details and sets out a unique library of routines to add extra commands to BASIC. \$10.95 (HARD) \$7.95 (SOFT)

Advanced Commodore 64  
Graphics and Sound  
Nick Hampshire

Any programmer wishing to make full use of the graphics and sound capability of the Commodore 64 will find this book an invaluable store of essential information - much of it previously unpublished. \$10.95 (HARD) \$7.95 (SOFT)



## ORDERING SYSTEM

Please order from your local bookseller or write to our head office at the address on Page 46. Cash, Postorder and Advance Bank Orders Welcome. Contact 011 541.

## Price only

ADVANCED COMMODORE 64  
BASIC REVEALED (HARD) \$10.95

ADVANCED COMMODORE 64  
GRAPHICS AND SOUND (HARD) \$10.95

## Minimum requirements for

these titles: Commodore 64 or compatible computer with 64K RAM. MS-DOS 2.0 or later software. Major text additional cost.

## Name

## Address

## City

## State

## Postcode

## Country

## \*\* TAPES BACKUP DEVICES \*\*

DCL -  
Interface

The DCL-1 uses two magnetic tape drives to backup any DOS-based computer. DCL-1 (HARD) \$75.15 (SOFT) \$45.00  
DCL-1 (HARD) \$75.15 (SOFT) \$45.00

DCL4  
Interface

The DCL-4 uses two magnetic tape drives and one hard disk to backup any DOS-based computer. DCL-4 (HARD) \$125.15 (SOFT) \$75.00  
DCL-4 (HARD) \$125.15 (SOFT) \$75.00

DCL4A  
Interface

Similar to the DCL-4 but uses one hard disk only.  
DCL-4A (HARD) \$125.15 (SOFT) \$75.00

## \*\* BACKUP PROBLEMS SOLVED \*\*

Alignment  
Tape

Aligns your floppy disks to the correct magnetic field. Includes a test tape to check disk alignment. \$15.00 (HARD) \$10.00 (SOFT) \$10.00

Tape Head  
Designation

The most accurate designation of tape head models. Includes a test tape to check head designation. \$15.00 (HARD) \$10.00 (SOFT) \$10.00

## \*\* 64 LOADER CARTRIDGE \*\* for 1641 80000 \*\*

64T Loader  
Cartridge

The 64T Loader Cartridge is a 1641 cartridge that loads any 64K program onto a 1641 cartridge. Includes a test tape to check cartridge operation. \$15.00 (HARD) \$10.00 (SOFT) \$10.00

## \*\* FREE CATALOGUE \*\* Please send 13p stamp

Free  
Catalogue

Our latest catalogue is FREE! Simply request a copy from: COLLEGE COMPUTERS, 25-26 THE SQUARE, BATH, AVON BA1 1UN, ENGLAND. Tel: 01225 310111. Fax: 01225 310112. We also have a FREE 64T Loader Cartridge with your copy of the catalogue. Please send 13p stamp.

## \*\* 30 DAY MONEY BACK GUARANTEE ON ALL HARDWARE \*\*

## Terms:

30 DAY MONEY BACK GUARANTEE ON ALL HARDWARE. All other goods sold as shown. Delivery and VAT extra. Cash on delivery.

## TRILEGIC

## Dept 94

## Bath, Avon

25-26 THE SQUARE, BATH, AVON

BA1 1UN

TEL: 01225 310111

Your Commodore's man in  
the States, Burton Rubin,  
attended the launch of the  
Amiga. Here's his report plus  
Brendin Lewis's impressions  
of Metacomco — makers of  
Amigados.

### US View

FOR THE PAST TWO YEARS, WHISP-OF rumours, as insubstantial as smoke, have appeared in various magazines concerning the talked, long-awaited, Amiga computer. Originally called the Lorraine, the product of the Amiga Corporation, the machine, a 32-bit computer with fabulous graphic and sound capability, was to be the ultimate home computer.

Silicon Valley, as we all know, is littered with the bones of "ultimate home

computers". Time-sharing, Texas Instruments, Coleco, Mattel, and even the mighty IBM have all seen their offerings for the home market wither away to a dusty death. When Jack Trammell left Commodore, there was speculation that his plan might be to purchase Amiga, (and with it, the rights to the wonderful Lorraine) and use the new machine as the spigot of the almost modified Atari. By purchasing Amiga, Commodore beat Atari to the punch.

Commodore first exhibited the machine at the June, 1984 Consumer

# THE AMIGA



### The Amiga Specifications

<b>Machine:</b>	Commodore Amiga
<b>Processor:</b>	68008 at 7MHz
<b>RAM:</b>	256K standard; 256K expansion slot; Up to 512K 1536 operating system.
<b>ROM:</b>	320 x 200 12-colours; 320 x 400 12-colours.
<b>Graphics:</b>	640 x 200 16-colours; 640 x 400 16-colours. Colours selected from a palette of 4096.
<b>Sound:</b>	Four audio channels including voice and speech.
<b>Disk:</b>	3 1/2 inch floppy 800kb.
<b>Software:</b>	Amigados, speech, Basic, Graphics.
<b>Miscellaneous:</b>	Peripherals: Video/tek unit, hard disk plus 1M byte RAM, colour monitor, mouse. Three custom chips inside the Amiga take much of the processing burden away from the 68000. The very high speed graphics are handled by one of these chips.

Electronics show and its specifications were impressive. The Motorola 68000 was chosen for the central processing unit. This is a 32 bit chip with a 16 bit bus. Capable of addressing up to 16 megabytes of memory, it is the same chip that powers the Apple Macintosh. Tandy's response was the 10051 and 52051, 32 bit machines running CPM. The Commodore camp responded with... silence. Sure, there was plenty on the sparkling, national new 128, and even pictures and publicity on the new notebook computer. But Commodore breathed easy a word about

Logic and Logi will be available at the time of introduction, as well as a very powerful version of Microsoft Basic. Third party publishers will have more than 20 games available when the machine hits the shelves. Adronics has already completed work on a videotext interface.

The Amiga works through the new familiar system of menus, windows, icons, multitasking, and a mouse - first pioneered by Apple with the Lisa. Up to 16 windows can be open and running on the Amiga, though this is obviously more than anyone can manipulate without going

to Oliver QED.

Irving Gould, Chairman of the company, sees the future of Commodore in "sophisticated, high end systems", with "excellent price/performance ratios" and "a full, rich, product line". That doesn't sound like the marketing philosophy that we've all grown to know and love.

The Amiga, though, is a machine capable of changing the philosophy of a company. Where the trusty old II was a Ford, and the Plus Four an Opel, the Amiga is a Ferrari. The ultimate decision will be made in the marketplace. Don't bet against the Amiga.

## The UK Connection

Tucked away in the corner of a small square in Bristol is the software house Metacom. What is Metacom? It is the company which has written AmigaDOS, the operating system for the new Commodore Amiga. Metacom is not a large company - with a staff of 25 - but it does have a good track record, working on software for both the Sinclair QL and the Atari 1000.

Upon meeting a few of the staff it's quite easy to see that the firm's success is based on three main factors - sound management, expertise and, most notably, enthusiasm. Even some members of the senior management seem like small children playing with a new toy whenever the Amiga is mentioned. Even this cynical reporter was surprised at its performance. Though, as ever I'm still not totally convinced, I'll reserve final judgement for the full production model. Metacom's staff, on the other hand, have had the word cynical totally erased from their memories. It was difficult for them to see a market into which the Amiga could not fit.

The whole story really started about three years ago when Amiga Inc. started work on a new machine. In November 1984, Commodore took over Amiga and then the machine. Previous to this though, Commodore had already approached Metacom concerning Topos (which is the framework around which AmigaDOS is built). From here, Metacom has never looked back and has written various bits of software for the Amiga, including a version of Basic.

Although, for most of the day it was difficult to stop our host talking, I did finally get an amiable silence when I broached the subject of Amiga II. The only reply I did get was that, due to the open technology of the Amiga, the Japanese would be the first to produce an Amiga look-a-like and that ideas were already in the pipeline for something within the next 12 months.

Finally, my thanks to all at Metacom for a very enjoyable day, and for providing what Commodore UK could not (or would not), a look at the Amiga.

# UNVEILED

the Lorraine, and the rumours continued to fly.

At the official premiere of the Amiga computer, hosted quite grandly by Commodore at the Vivian Beaumont theatre at Lincoln Centre, the rumours all came true.

The specifications do not convey the power of the machine. The standard configuration of the Amiga includes a Motorola 68000 microprocessor, running at almost eight MHz, with 256K of RAM, internally expandable to 512K. The speed of this microprocessor is further abetted by a proprietary three chip set which links the 68000 from routine graphic and I/O tasks.

The Amiga is controlled by an 89 key keyboard with numeric keypad, cursor and special function keys, or a two-button mouse. It features a built in three and a half inch disk drive (800K formatted), 80" x 25 line text display, 640 x 400 resolution and a palette of 4,096 colours (of which, any 16 can be on the screen at one time in high-resolution modes). There are parallel, serial, and second drive ports, two reconfigurable joystick ports, as well as text to voice and professional quality four channel multi-voice music synthesis capability. AtariC, Amiga CDS, and Amiga Tutor will be bundled with the machine.

Before availability, the bare of all new computer introductions, would seem not to be a problem here. Thanks to the Sinclair - a software option - the Amiga can run IBM PC compatible software packages like Lotus 1-2-3, Wordstar, and D Base III, in either 1.5 or 5.25 inch disk format.

In addition, when introduced in September, the Amiga will have available more than 20 programs including word processing, accounting, productivity, education, speech synthesis, teleconferencing, paint, animation, and graphic programs. Assembler, C, Pascal,

Basic. The Intuition operating system, working through a system of "gadgets" makes moving an easier and quicker task than it is on the Macintosh, or with the CPM operating system of the Atari.

The animation power of the machine is almost beyond description. It should open new vistas for anyone who needs professional visual aids, its value to the small advertising agency, or any small business should be incalculable. Combined with the phenomenal multitasking capability, Amiga should be a formidable weapon in the business wars.

With business applications firmly in mind, Commodore has chosen to merchandise the machine through specialist dealers. Long ago, Commodore pulled the plug on computer stores, in favour of mass market merchants.

The parents at Commodore are reversing the decision that made them such a commercial success and critical failure. It remains to be seen just how easily Commodore will be accepted by the same computer stores which were abandoned a few years ago.

At a list price of \$1299, the Amiga represents excellent value. However, it's unlikely to be found at the local K Mart.

My feeling is that acceptance in the computer stores may come gradually, but it will definitely come. The machine is simply too good to be ignored. The \$1299 price tag includes a healthy mark-up for the dealers, and Commodore has signed up 588 RCA service locations to provide service support. With the advent of the Amiga, Commodore has both Apple and IBM lined up in its sights.

Thomas Hartigan, president of Commodore North America, is talking tough. "Commodore" he says, "is a strong, lean, aggressive organisation", and he intends to have the Amiga showcased in 12,000 outlets within a year of its introduction. He feels that the Amiga is a great leap forward (plaudits of



## BASIC Loader (continued)

```

110 DATA162,16,173,174,2,72,183,83,141,1
111
115 DATA174,2,104,157,63,141,202,173,33,
208
120 DATA72,183,83,141,141,33,208,104,157
,83
125 DATA141,202,173,32,208,72,183,63,141
,141
130 DATA32,208,104,157,63,141,202,181,43
,72
135 DATA83,63,141,149,43,104,157,63,141
,202
140 DATA16,241,162,0,183,0,4,72,183,82
145 DATA141,157,0,4,104,157,82,141,183,0
150 DATA216,72,183,83,146,157,0,218,104,
157
155 DATA33,146,183,0,6,72,183,82,142,157
160 DATA0,6,104,157,82,142,183,0,217,72
165 DATA83,83,146,157,0,217,104,157,33,
146
170 DATA183,0,6,72,183,82,143,157,0,6
175 DATA104,157,82,143,183,0,218,72,183,
33
180 DATA147,157,0,218,104,157,33,147,183
,0
185 DATA7,72,183,82,144,157,0,7,104,157
190 DATA82,144,183,0,219,72,183,83,146,1
57
195 DATA0,219,104,157,83,146,232,224,0,2
08
200 DATA33,96,1,6,3,8,3,8,3,8
205 DATA0,74,0,74,0,74,224,246,14,1
210 DATA74,3,74,3,74,3,74,0,140,0
215 DATA140,0,140,246,240,13,0,0,-1

```

## Entering the Program

To enter the program type in the Basic loader program, then SAVE it. Type RUN and the screen should flash, then clear to the normal Basic start up message. Note the much reduced free memory. Type in a short Basic program and RUN it. Stop the program, hold down the shift key and press control. The screen should now change colour and have the Basic start up message on it.

You are now in the second area LIST and there should be no trace of the program you typed in. Main control should take you back to the first screen and LIST will show your

program is still there. If you wish to make a machine code copy of split 64 then first make sure you are in program area (a) before entering your machine code monitor. The start address is \$0C00 and the end address \$0D01.

## How it Works — General

split 64 takes advantage of the fact that the CG4 operating system allows Basic to work within any free area of memory. To write your programs in another area of memory it is only necessary to change the pointers used by Basic in zero page. Change the

pointers back and any program that was there before will still be there. It is also possible to change the area of memory used by the screen but when writing this program it was decided to store the second colour map along with the second colour map above Basic memory. The shift control switch is operated by a simple interrupt wedge.

## How it Works — Machine Code

## Initialise Routine

This routine sets up area (a) colours, memory start, memory

end, and variable pointers from values fixed in Table 1. It then puts zero in the first three locations of Basic memory for areas (a) and (b). It then calls the swap routine, which stores the current screen and loads the contents of the second screen (probably at this time). Having swapped screens the Basic start up program is called which clears the screen and prints the start up message. This procedure is then repeated for area (a). Finally the wedge routine address is inserted into the IRQ vector, the normal vector address having been saved, then back to basic.

## Wedge Routine

The wedge routine is entered on every IRQ. It first checks location \$0800 to see if the shift control keys are pressed. If not it jumps to the normal IRQ routine. If the keys are pressed, it checks that the cursor is off and that the computer is not in RUN mode. When all three of these conditions are satisfied it calls the swap routine. After swapping areas a large delay loop is executed to avoid multiple swaps. Control is then passed to the normal IRQ routine.

## Swap Basic

This is the first part of the swap sub-routine. A loop is used to exchange the current screen, loader and character colours, with the contents of Table 1. The same loop is used to exchange the Basic pointers (\$20-\$35) again with the contents of Table 1. On start up Table 2 contains the filled values for area (b).

## Swap Screen

Once more a loop is employed to exchange the current screen and colour map stored above Basic memory. The exchange is done right bytes at a time. Not the most elegant way to swap four areas of memory but it was chosen to cause minimum disruption to the screen during the swap.

Table 1

## INITIALISE ROUTINE

8c00	a2 10	ldr #410	:index to table1
8c02	bd 2e 8d	lda #8d2e.x	:table1
8c05	8d 86 02	sta #0286	:char. colour area a
8c08	ca	dex	:next
8c09	bd 2e 8d	lda #8d2e.x	:table1
8c0c	8d 21 d0	sta #d021	:increase colour area a
8c0f	ca	dex	:next
8c10	bd 2e 8d	lda #8d2e.x	:table1
8c13	8d 20 d0	sta #d020	:border colour area a
8c16	ca	dex	:next
8c17	bd 2e 8d	lda #8d2e.x	:table1
8c1a	95 2b	sta #2b.x	:basic memory size area a
8c1c	ca	dex	:
8c1d	10 f8	bpl #8c17	:next
8c1f	a2 02	ldr #402	:
8c21	a9 00	lda #400	:zero first 3 bytes in
8c23	9d 00 08	sta #0800.x	:both area a&b
8c26	9d 00 4a	sta #4a00.x	:
8c29	ca	dex	:
8c2a	10 f7	bpl #8c23	:
8c2c	20 22 e4	jer #e422	:basic start up a
8c2f	20 78 8c	jer #8c78	:swap area
8c32	20 22 e4	jer #e422	:basic start up b
8c35	20 78 8c	jer #8c78	:swap back
8c38	78	sei	:
8c39	ad 14 03	lda #0314	:irq low
8c3c	48	pha	:save it
8c3d	ad 15 03	lda #0315	:irq high
8c40	48	pha	:save it
8c41	a9 57	lda #457	:low address wedge
8c43	8d 14 03	sta #0314	:irq vector
8c46	a9 8c	lda #88c	:high address wedge
8c48	8d 15 03	sta #0315	:irq vector
8c4b	68	pla	:recover address irq high
8c4c	8d 51 8d	sta #8d51	:store
8c4f	68	pla	:recover address irq low
8c50	8d 50 8d	sta #8d50	:store
8c53	58	cli	:
8c54	4c 74 a4	jmp #a474	:ready for basic

Table 2

## WEDGE ROUTINE

```

8c57 ad 8d 02    lda #028d      :control shift pressed
8c5a c9 05      cmp ##05      :
8c5c d0 17      bne #8c75     :if not goto normal irq
8c5e a5 cf      lda #cf       :cursor off
8c60 d0 13      bne #8c75     :if not goto normal irq
8c62 a5 9d      lda #9d       :direct mode
8c64 c9 80      cmp ##80      :
8c66 d0 0d      bne #8c75     :if not goto normal irq
8c68 20 78 8c   jsr #8c78     :swap area
8c6b a2 ff      lda ##ff      :
8c6d a0 ff      ldy ##ff      :
8c6f 00         dey          :delay to avoid key bounce
8c70 d0 fd      bne #8c6f     :
8c72 ca        dex          :
8c73 d0 f8      bne #8c6d     :
8c75 6c 50 8d   jmp (#8d50)   :jump to normal irq routine

```

Table 3

## SWAP BASIC 2.PAGE LOCATIONS

```

8c78 a2 10      ldx ##10      :index to table2
8c7a ad 86 02   lda #0286     :char colour current
8c7d 48        pha          :save it
8c7e b4 3f 8d   lda #8d3f.: :table2
8c81 8d 86 02   stx #0286     :change char colour
8c84 68        pla          :recover char colour
8c85 9d 3f 8d   stx #8d3f.: :store in table2
8c88 ca        dex          :next
8c89 ad 21 d0   lda #d021     :current screen colour
8c8c 48        pha          :save it
8c8d b4 3f 8d   lda #8d3f.: :table2
8c90 8d 21 d0   stx #d021     :change screen colour
8c93 68        pla          :recover screen colour
8c94 9d 3f 8d   stx #8d3f.: :store in table2
8c97 ca        dex          :next
8c98 ad 20 d0   lda #d020     :current border colour
8c9b 48        pha          :save it
8c9c b4 3f 8d   lda #8d3f.: :table2
8c9f 8d 20 d0   stx #d020     :change border colour

```

Table 3 continued

8ca2 48		pla	:recover border colour
8ca3 9d 3f 8d		sta #0d3f,x	:store in table2
8ca6 ca		dex	:next
8ca7 b5 2b		lda #2b,x	:basic memory size
8ca9 48		pha	:save
8caa bd 3f 8d		lda #0d3f,x	:table2
8cad 95 2b		sta #2b,x	:basic memory size...
8caf 68		pla	:recover
8cb0 9d 3f 8d		sta #0d3f,x	:store in table2
8cb3 ca		dex	:next
8cb4 10 f1		bpl #0ca7	:repeat for other 2 page locations

Table 4

## SCREEN SWAP ROUTINE

8cb6 a2 00		ldx #800	:index for screen+colour mem
8cb8 bd 00 04		lda #0400,x	:screen mem
8cbb 48		pha	:save it
8cbc bd 52 8d		lda #0d52,x	:second screen
8cbf 9d 00 04		sta #0400,x	:store in screen
8cc2 68		pla	:recover
8cc3 9d 52 8d		sta #0d52,x	:store in second screen
8cc6 bd 00 48		lda #4800,x	:colour mem
8cc9 48		pha	:save it
8cca bd 5d 91		lda #915d,x	:second colour mem
8ccd 9d 00 48		sta #4800,x	:store in colour mem
8cd0 68		pla	:recover
8cd1 9d 5d 91		sta #915d,x	:store in second colour mem
8cd4 bd 00 05		lda #0500,x	: -----
8cd7 48		pha	:
8cd8 bd 52 8e		lda #8e52,x	:
8cdb 9d 00 05		sta #0500,x	:
8cde 68		pla	:
8cdf 9d 52 8e		sta #8e52,x	: same but plus 256
8ce2 bd 00 49		lda #d900,x	:
8ce5 48		pha	:
8ce6 bd 5d 92		lda #925d,x	:
8ce9 9d 00 49		sta #d900,x	:
8cec 68		pla	:
8ced 9d 5d 92		sta #925d,x	: -----
8cf0 bd 00 06		lda #0600,x	:
8cf3 48		pha	:



Table 4 (continued)

8c14	bd	52	8f	lda	#0152.x	:	
8c17	9d	00	06	sta	#0600.x	:	
8c1a	68			pla		:	same but plus 512
8c1b	9d	52	8f	sta	#0152.x	:	
8c1e	bd	00	da	lda	#da00.x	:	
8d01	48			pha		:	
8d02	bd	5d	93	lda	#935d.x	:	
8d05	9d	00	da	sta	#da00.x	:	
8d08	68			pla		:	
8d09	9d	5d	93	sta	#935d.x	:	
8d0c	bd	00	07	lda	#0700.x	:	
8d0f	48			pha		:	
8d10	bd	52	90	lda	#9052.x	:	
8d13	9d	00	07	sta	#0700.x	:	
8d16	68			pla		:	same but plus 768
8d17	9d	52	90	sta	#9052.x	:	
8d1a	bd	00	db	lda	#db00.x	:	
8d1d	48			pha		:	
8d1e	bd	5d	94	lda	#945d.x	:	
8d21	9d	00	db	sta	#db00.x	:	
8d24	68			pla		:	
8d25	9d	5d	94	sta	#945d.x	:	
8d28	e8			inx		:	!next
8d29	e0	00		cpx	#000	:	!255 done in each block
8d2b	d0	8b		bne	#8cb8	:	!if not then go back
8d2d	60			rts		:	!return

Table 5

TABLES

Contents shown as at start.

Table1	Table2		
8d2e 01	flow s.a	8d36 00	flow b/inactive string
8d2f 08	thigh "	8d37 4a	thigh " "
8d30 03	flow s.a	8d38 00	flow tr " "
8d31 08	thigh "	8d39 4a	thigh " "
8d32 03	flow s.arrays	8d3a 00	flow see top
8d33 08	thigh "	8d3b 4a	thigh " "
8d34 03	flow s.arrays	8d3c fe	tscreen colour
8d35 08	thigh "	8d3d fe	tborder colour
		8d3e 0c	tchar colour
			Address for IR0.
		8d50 31	
		8d51 ea	



This month Garry

Marshall shows you

how to create artificial  
landscapes on your

64.

# PROGRAMMING PROJECTS

## Graphic landscapes

### The project

COMPUTER-GENERATED GRAPHICS are increasingly finding their way into films. Among the most impressive and realistic effects that have been seen are the entirely artificial landscapes created for some of the Star Line films. This month's project involves the creation of an artificial landscape of this kind.

Of course, far more sophisticated (and expensive!) equipment than a C64 is needed to display graphics of a quality that is suitable for use in films. But, as we shall see, it is not too difficult to write programs that produce the basic effects, and the quality of the results is surprisingly good.

The theory that provides the basis for the creation of these landscapes is known as fractal geometry. Without going too deeply into the theory, the idea behind a fractal curve is that it is not one or two-dimensional, but has a dimension which is a fraction. Curves of this kind describe, for example, rough surfaces, with the degree of roughness determining the fractal dimension giving the dimension of the fractal curve. They also describe many other naturally occurring curves, such as coast lines and hillside, and this is what makes them eminently suitable for simulating landscapes.

The project involves starting with a triangle and, by means of a simple procedure deriving from fractal geometry, sub-dividing it into a graphic display that resembles a mountain. The further the sub-division process is carried out, the more realistic the result will appear.

### The solution

The basic idea of the procedure for creating the mountain landscape is illustrated in figures 1 and 2. The first figure shows that taking a triangle, finding the mid-points of its sides and joining them gives four smaller triangles. Repeating the process on each of the smaller triangles gives a four triangular mesh, and the more it is repeated, the

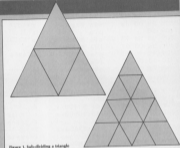


Figure 1. Sub-division of a triangle

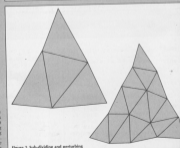
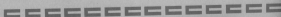


Figure 2. Sub-division and perturbing



line the triangular mesh becomes. But the results of this procedure do not resemble any natural phenomenon, simply because of their regularity. This is not a property to be found in nature, which is characterized by randomness.

We can produce an element of randomness by perturbing the mid-point of each side by a random amount the variation of which is proportional to the length of its side. The left-hand illustration in Figure 2 shows the result of doing this and drawing the four smaller triangles produced by one sub-division. The right-hand illustration shows what happens after two sub-divisions, demonstrating that none of the triangles can be drawn until all the mid-points have been found and then perturbed. Continuing this process will give an increasingly mountain-like display. The range of the perturbation for the mid-point of each side or, if you like, the degree of proportionality between the range and the length of the side, effectively determines the fractional dimension of the final result. Different values for the range give quite different appearances to the mountain landscapes.

This gives us the following form for a graphics program to draw an artificial mountain landscape. It should declare the arrays to be used, and there will be several, because we must store the points for all the triangles in the final display as we cannot draw any of them until all their positions have been computed. Then it must read the co-ordinates of the corners of the basic triangle, prepare the high-resolution graphics screen, and carry out the sub-division of each side to give the points at the corners of the resulting triangles. To begin with, we will ignore the perturbations of the mid-points that introduce the randomness. This simplifies the program a little, gives results such as those in Figure 3, but provides a program which we can easily generalize to produce results like those in Figure 2. Finally, when the program has found the points for all the triangles, it only remains to plot them.

The main program based on this scheme is:

```

10 DIM(X(48), Y(48), S(3), T(3), U(12), W(12),
  X1(48), Y1(48))
20 FOR K=1 TO 3
30 READ X(K), Y(K)
40 NEXT K
50 DATA 150, 50, 250, 150, 50, 150
60 GOSUB 600: REM HI-RES SCREEN
70 FOR N=0 TO 1
80 FOR K=1 TO 4*N
90 GOSUB 4000: REM SUB-DIVIDE THE
  TRIANGLES
100 NEXT K
110 FOR K=1 TO 4*(N+1)*3
120 X(K)=X(T(K)): Y(K)=Y(T(K))
130 NEXT K
140 NEXT N

```

## Program Listing

```

15 DIM(X(48), Y(48), S(3), T(3), U(12), W(12),
  X1(48), Y1(48))
20 FOR K=1 TO 3
30 READ X(K), Y(K)
40 NEXT K
50 DATA 150, 50, 250, 150, 50, 150
60 GOSUB 600: REM HI-RES SCREEN
70 FOR N=0 TO 1
80 FOR K=1 TO 4*N
90 GOSUB 4000: REM SUB-DIVIDE THE TRIANGLES
100 NEXT K
110 FOR K=1 TO 4*(N+1)*3
120 X(K)=X(T(K)): Y(K)=Y(T(K))
130 NEXT K
140 NEXT N
150 FOR K=1 TO 18
160 FOR J=1 TO 3
170 S(J)=S(J)+3*(K-1): T(J)=Y(J)+3*(K-1)
180 NEXT J
190 GOSUB 3000: REM PLOT EACH TRIANGLE
200 NEXT K
210 END
500 POKE 53272, PEEK(53272) OR 8
510 POKE 53265, PEEK(53265) OR 32
520 FOR I=8192 TO 16192: POKE I, 0: NEXT I
530 FOR I=16192 TO 8092: POKE I, 255: NEXT I
540 RETURN
1000 RD=INT(R/8): CD=INT(C/8)
1010 L=R AND 7
1020 BIT=7 - (C AND 7)
1030 BYTE=BIT2 + RD*320 + CD*8 + L
1040 POKE BYTE, PEEK(BYTE) OR 2*BIT
1050 RETURN
2000 EX=X2-X1: DY=Y2-Y1
2010 IF EX=0 THEN 2070
2020 FOR C=X1 TO X2 STEP 50*(EX)
2030 R=INT(Y1+(C-X1)*DY/EX)
2040 GOSUB 1000: REM PLOT POINT
2050 NEXT C
2060 RETURN
2070 C=X1
2080 FOR R=Y1 TO Y2 STEP 50*(DY)
2090 GOSUB 1000: REM PLOT POINT
2100 NEXT R
2110 RETURN
3000 X1=S(3): Y1=T(3)
3010 FOR P=1 TO 3
3020 X2=S(P): Y2=T(P)
3030 GOSUB 2000: REM DRAW LINE
3040 X1=X2: Y1=Y2
3050 NEXT P
3060 RETURN
4000 FOR N=1 TO 3
4010 S(N)=X(N)+3*(K-1): T(N)=Y(N)+3*(K-1)
4020 NEXT N

```

```

100 FOR M=1 TO 15
110 FOR N=1 TO 3
120 GOTO(275,315,355) T0=N+P-4*(N-1)

```

```

130 NEXT I
140 GOSUB 300: REM: PLOT EACH TRIANGLE
150 NEXT K
170 END

```

Here, the arrays are declared in line 10. The data, which is at line 30 and gives the positions of the corners of the initial triangle, is read by lines 20 to 40. Line 40 calls the now-familiar sub routine, starting at line 300, for preparing the high-resolution graphics screen.

Lines 70 to 140 calculate and store the positions of the corners of all the triangles that result from the sub-dividing process. The outer loop variable, M, determines how many stages of sub-division occur. In the program as presented, there are two stages, which are necessary to bring us to the position shown in the right-hand illustration of Figure 1. The first sub-division is done with N=0 and the second with N=1.

The inner loop variable, K, counts the number of triangles to be sub-divided. Initially, there will be one, and the value of K when N=0 is 1. After the first sub-division, there will be four, the value

#### Program Listing (cont.)

```

4000 GOSUB 6000: REM SUB-DIVIDE THIS TRIANGLE
4010 FOR M=1 TO 12
4020 XT(M)=12*(E-1)+UM(M): YT(M)=12*(K-1)+VM(M)
4030 NEXT M
4070 RETURN
5000 A1=(S(1)-S(2))*D.2*(RND(0))-D.5)
5010 B1=(S(1)-S(2))*D.2*(RND(0))-D.5)
5020 C1=(S(2)-S(3))*D.2*(RND(0))-D.5)
5030 A=D.5*(S(1)+S(2))+A1
5040 B=D.5*(T(1)+T(2))+B1
5050 D=D.5*(S(1)+S(3))+D1
5060 E=D.5*(T(1)+T(3))+E1
5070 C=D.5*(S(2)+S(3))+C1
5080 F=D.5*(T(2)+T(3))+C1
5090 UC1)=S(1): UC4)=S(2): UC7)=S(3)
5100 UC1)=T(1): UC4)=T(2): UC7)=T(3)
5110 UC2)=A: UC5)=B: UC10)=A
5120 UC2)=D: UC6)=D: UC12)=D
5130 UC3)=B: UC8)=B: UC11)=B
5140 UC3)=E: UC9)=E: UC11)=E
5150 UC5)=C: UC8)=C: UC10)=C
5160 UC5)=F: UC9)=F: UC10)=F
5170 RETURN

```

READY.

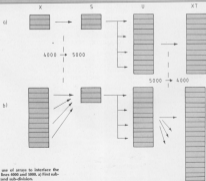
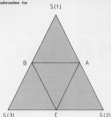


Figure 1. The use of arrays to interface the sub-routines at lines 5000 and 5050. a) first sub-division b) second sub-division.

Figure 4. Names used by subroutines for subdividing a triangle.



of 4° when  $N=1$ . (Although the program, as listed, goes no farther than this, after the second sub-division each of four triangles will have been divided into four to give 4°4 or 4° triangles, and so on. In this way, the program is ready to be extended to create more detailed graphics.)

Line 80 calls a subroutine starting at line 4000 to sub-divide the triangles that are repeatedly passed to it. This subroutine places the *x* and *y* co-ordinates at the corners of the triangle that is being dealt with in the arrays *I* and *J*. Then it calls the subroutine starting at line 6000 to do the actual sub-division. After it has done the dividing up, this subroutine leaves the *x* and *y* co-ordinates of the corners of the four triangles that result from the sub-division in the arrays *I* and *J*, respectively. The contents of these arrays are then transferred to the arrays *X* and *Y*. This gives us the subroutine as

```
4000 FOR N=1 TO 3
4010  S(4)=S(3)+I*(J-1): T(4)=Y(4)+I*(J-1)
4020 NEXT M
4030 GOTO 3000: REM SUB-DIVIDE TRIANGLE
4040 FOR A=1 TO 12
4050  S(1)=I(1)-I*(J+1)/3: S(2)=I(2)-I*(J+1)/3
4060 NEXT A
4070 RETURN
```

The interface between the subroutines starting at 4000 and 6000 is illustrated in Figure 5, showing how they communicate by means of the arrays. The top illustration shows what happens with the initial sub-division. This does not exactly make it clear that all the arrays are needed, but the bottom illustration does so by showing how the arrays are used for

the second sub-division.

Figure 4 shows how the *x* co-ordinates of the corners of a triangle are named during a sub-division as performed by the subroutine starting at line 6000. The subroutine itself is:

```
6000 A=0.5*(I(1)+I(2))
6010 D=0.5*(I(1)+I(2))
6020 E=0.5*(I(1)+I(2))
6030 I(4)=0.5*(I(1)+I(2))
6040 C=0.5*(I(2)+I(3))
6050 I(5)=0.5*(I(2)+I(3))
6060 I(3)=I(3): I(4)=I(4): I(5)=I(5)
6070 Y(1)=Y(1): Y(4)=Y(4): Y(5)=Y(4)
6080 I(2)=A: I(4)=A: I(5)=A
6090 Y(1)=C: Y(4)=C: Y(5)=C
6100 I(3)=B: I(4)=B: I(5)=B
6110 I(3)=C: I(4)=C: I(5)=C
6120 Y(1)=E: Y(4)=E: Y(5)=E
6130 RETURN
```

After this, lines 770 to 130 copy copy the positions of the triangles from *X* and *Y* to *X* and *Y* so that they will be available for the next round of sub-division.

When the positions of all the triangles have been found they are plotted by lines 150 to 200. Here the inner loop from line 160 to 180, controlled by *J*, passes the co-ordinates of each triangle in turn to the arrays *X* and *Y* so that the subroutine starting at line 8000 can be called to plot the triangle. The outer loop, controlled by *I*, ensures that all the triangles are dealt with.

The triangle-plotting routine simply calls our much used line-drawing subroutine (which begins at line 2000) three times. The line drawing routine draws a line from (*X*<sub>1</sub>, *Y*<sub>1</sub>) to (*X*<sub>2</sub>, *Y*<sub>2</sub>) and this gives the triangle-plotting routine as

```
3000 S(1)=I(1): Y(1)=I(2)
3010 FOR P=1 TO 3
```

```
3020  S(2)=P(1): Y(2)=P(2)
3030  C=200:R=200: REM DRAW LINE
3040  S(1)=S(2): Y(1)=Y(2)
3050  NEXT P
3060 RETURN
```

Apart from the previously used routines, this gives the complete program for creating the regular triangular mesh. It is now a simple matter to perturb the points at the corners of the triangles randomly before the triangles are plotted, so that we get our 'rough'. The perturbing can be done in the innermost subroutine, the one starting at line 6000, by adding to the co-ordinates a random amount with a variation that is proportional to the length of the side being bisected. For simplicity, the perturbing has been taken as a random amount between  $\pm 0.1\%$  length of the line in the *x*-direction. This gives the amended subroutine as:

```
5000 A(1)=I(1)-S(2)*(0.2*(RND(0))-0.5)
5010 B(1)=I(1)-S(2)*(0.2*(RND(0))-0.5)
5020 C(1)=I(2)-S(2)*(0.2*(RND(0))-0.5)
5030 A(2)=I(1)+S(2): A(1)
5040 B(2)=I(1)+S(2): B(1)
5050 C(2)=I(2)+S(2): C(1)
5060 P=0.5*(I(2)+I(3))+C(1)
5070 I(3)=I(3): I(4)=S(2): I(5)=I(3)
5100 Y(1)=I(1): Y(4)=I(2): Y(5)=I(1)
5110 I(2)=A: I(4)=A: I(5)=A
5120 Y(2)=I(2): Y(4)=I(2): Y(5)=I(2)
5130 I(3)=B: I(4)=B: I(5)=B
5140 Y(3)=I(3): Y(4)=I(3): Y(5)=I(3)
5150 I(3)=C: I(4)=C: I(5)=C
5160 Y(4)=I(4): Y(5)=I(4)
5170 RETURN
```

The complete listing of the program to create the artificial landscape is given as Figure 5.

## Further developments

The program can be extended to take the sub-division process further, so producing a more detailed landscape. The program only needs a few numerical changes for this which follow at once from the number of triangles there will be in the new arrangement. Don't forget to change the dimensions of the arrays! The way that the randomness is introduced can be made much more general, and the constant of proportionality for the range of the perturbations can be changed. You may like to experiment with this to see if you can characterize the differences between the types of landscapes that are produced by significantly different values for the constant of proportionality. Remember that because the perturbations are random, each run of a program gives a different picture, although one that belongs to the same family.

If you want to design  
your own games or  
use high-resolution  
and multi-colour  
characters, then this  
program by John  
McHale could be just  
what you're looking  
for.

### Setting up

TYPE IN THE LISTING PROVIDED and save from a disk or tape before attempting to run it.

Have RUN it and if all has gone smoothly, you will be given the option of saving the .M. Machine Code file to the director of your preference.

You should make it a habit, to 'verify' all programs that you save so as not to end up being disappointed, if you are not able to reload the program. It would also be a good idea to make a second copy, just in case you 'scrash' the first by accident.

For those of you who are not fortunate enough to enter the data correctly first time, you will be given one of two possible error reports (as maybe both).

These two error reports are as follows:

1. Invalident/Too Many Data Items Error.
2. Checksum Error.

If you get error 1, then you will almost certainly get error 2 as well.

If you get error 2 on its own, then you have entered some of the Data Items incorrectly e.g. you may have entered 255 instead of 240. Error 1 is an explanation.

It would be useful to have a friend close by, to read out the data statements to you, in order to minimise the risk of error.

### Using the Program

Load the code by typing -

Load " " 1,1 or Lad "CERN".,1 for the tape version  
Load "CERN".,1 for the disk version

# IN CHARACTER

Now type 'SIS 6478' to cold start the machine before using the program. Type 'SIS 6288' to enter the Character Generator.

The Program resides at \$C000 (print) decimal which is well out of the reach of Basic.

The user has the ability, using the versatile utility, to write a Basic program of up to 100 and at the same time, being able to enter and exit the Character Generator at will.

It is important to note that the data for your user defined graphics, resides in a 2K block - (\$3000-\$37FF) or 12888 - 14335 decimal and the complete set may be accessed by typing 'P0K3 53272,8'.

Users who delight in designing games, using high-resolution and multi-colour characters, will find this program an invaluable aid for their graphics. A maximum of 256 Characters (which is more than enough) may be redefined. I have allocated these 256 Characters into four subsets of 64 Characters each.

Alert users will notice, that when you enter the Character Generator for the first time, the standard PET character set is loaded into the user definable area, but this only happens once.

To do this exercise, switch off the Machine, Load in the Character Generator. Type 'SIS 6478' and 'P0K3 53272,8'. You will not note that all the characters have turned to garbage. Type 'SIS 6288' and press 'Q' for quit. You should be back in Basic at this stage.

Now type 'P0K3 53272,8'. Notice that the PET character set has now been loaded down. Type 'P0K3 53288,255' and press '-' and you will notice a straight line going across the top of the

### Character Generator — User Function List

Function	Name	Keypress(es)
01.	cursor home	'C'/'Home'
02.	clear grid	'Shift + C'/'Home'
03.	cursor right	'Tab L/'
04.	cursor left	'Shift+Cap L/'
05.	cursor down	'Crs w/d'
06.	cursor up	'Shift+Crs w/d'
07.	fill	','
08.	clear	'Del'
09.	space	'Spa'
10.	new line	'Ret'
11.	quick fill	'F'
12.	quick rub	'D'
13.	enable wrap mode	'W'
14.	disable wrap mode	'W'
15.	select character	'/'
16.	on+	'/'
17.	on-	'/'
18.	next char	'/'
19.	last char	'/'
20.	enable multicolour	'M'
21.	disable multicolour	'M'
22.	speed cursor	','
23.	slow cursor	','
24.	update cursor colour	'/'
25.	update multicolour 1	'1'
26.	update multicolour 2	'2'
27.	update screen Colour	','
28.	shift right	'R'
29.	shift left	'L'
30.	shift down	'D'
31.	shift up	'U'
32.	60° Rotate	'R' or 'C'
33.	invert character	'I'
34.	180° rotate	'R'
35.	transfer character	'T'
36.	transfer set	'CSE'/'T'
37.	reverse video	'V'
38.	recall	'C'
39.	load characters	'L'
40.	save characters	'S'
41.	quit	'Q'



' signs. Now type SYS 12882 and you will notice that this character has not been changed.

The Program uses a very powerful raster interrupt, to enable the CBM set and the user defined graphics to be displayed at the same time. This routine also handles such effects as split screen colour etc.

Reading the next section should convince you of the value of this extremely powerful Character Generator.

## Select

When you first use the Character Generator, you will notice four lines to the right of the grid, which generate status reports. The most significant of these is the line, i.e. Character Mode. On running the program, this is set to 'edit', which is the default mode. In this mode, you are able to access all of the listed functions. However, if you press ' ', then the character mode will change to select.

Immediately, you will see that the cursor is no longer on the grid, but is now flashing on a line of characters which are directly below the grid. The only keys valid in this mode are 'left', 'right', 'up', 'down', ' ', ' ', ' ', ' '.

Use the 'right' key in the normal way (in conjunction with the 'left' key) to locate the character you wish to edit. Now press ' ' and the character mode will return to 'edit' and you will be able to experiment with the character you have selected.

Alternatively, as mentioned in the Function List, you may use functions 'W' and 'V' (Next Char & Last Char) to achieve the same result without ever having to enter the Select Mode.

Using the functions 'Set #' and 'Set -', choose the set that you want the original to be transferred to.

Now press 'Return', the transfer will be completed and normal operation resumed.

## Recall

Have you ever made a complete mess of a sprite or LUDG that you were designing and wished you could restore it to its original form. This is what 'Recall' is for.

Every character you use is automatically 'buffered' so that in the event of you using the functions excessively, pressing 'Recall' will restore the character to its original form.

## A Finishing Note

Do not try to define your first block/typing character set immediately. Mess around with the various functions, until you become familiar with them. I hope you enjoy using it.

## Description

Positions cursor in top left of grid.  
As above and clears current character.  
Moves cursor right one space.  
Moves cursor left one space.  
Moves cursor down one space.  
Moves cursor up one space.  
Fills space at current cursor location.  
Deletes space to left of cursor.  
Rubs space to right of cursor.  
Places cursor on left of next line down.  
Fills left-right on current line.  
Deletes left-right on current line.  
Allows mode to move off the edge of the grid and appear on the opposite side.  
Keeps cursor within grid boundaries.  
Enables user to select the next character to be edited.  
Advances to next subset of 64 chars.  
Returns to previous subset.  
Allows the user to move to the next character without having to enter 'Select Mode' - see notes on 'V'.  
As above but moves backwards rather than forwards.  
Enables character multicolour mode.  
Disables character multicolour mode.  
Speeds cursor's response.  
Slows cursor response.  
Self explanatory.  
Self explanatory.  
Self explanatory.  
Self explanatory (split screen colour).  
Moves complete character right (1 bit).  
Moves complete character left (1 bit).  
Moves complete character down (1 bit).  
Moves complete character up (1 bit).  
Rotate character through 90 degrees 'A' - anticlockwise 'C' clockwise.  
Flips character upside down.  
Creates Mirror image of character through the vertical axis. (Works in hi-res & multicolour modes).  
Copies one character in the current set into another character in the same set.  
Transfers one complete set to another.  
'W' = Next Char - 'V' = Last Char - 'B' = Set.  
Recalls 'buffered' character (See special notes on this function).  
Load character sets from tape or disk.  
Saves character sets to tape or disk. (note: pressing 'V' will save the current subset (64 chars.); pressing 'Ctrl+V' will save the entire four sets).  
Exit to Basic.

## Transfer Set

To transfer a set of 64 characters to another location, first press 'Ctrl+T'. You should now see the Character Set no. (fourth status line) flashing.

## Program Listing

```

0 REM .....
1 REM .....
2 REM : CHARACTER GENERATOR :
3 REM :
4 REM : 4K PURE M/CODE :
5 REM :
6 REM : OVER 40 FUNCTIONS. :
7 REM :
8 REM : WRITTEN BY J.P.C HALE :
9 REM :
10 REM : STRANDHILL RD., SLIGO. :
11 REM :
12 REM : REP. IRELAND. :
13 REM :
14 REM : DEDICATED TO. :
15 REM :
16 REM : JENNIFER ..... :
17 REM :
18 REM .....

```

Program Listing (cont.)

```

80 PRINT"CLEAR":REM CLR/ROME
90 POKES3991,0:POKES3000,0
40 PRINT"YELLOW:PLEASE WAIT, WRIT
ING CODE TO MEMORY ."
50 SA=19150:BC=0:TL=0
60 READ:IFA=-1THEN690
70 POKESA+BC,A:BC=BC+1:TL=TL+A:GOT
O60
80 REM * ERROR TRAPPING *
90 IFBC<>3002THEN6400
100 IFTL<>490302THEN6400
110 POKES3200,14:POKES3201,6:PRINT
"CLEAR:G 71,REM CLR/ROME & LIGH
T BLUE ( CLR
E+7")
120 PRINT"DOWN:OKAY = CODE ENTERE
D CORRECTLY ."
130 PRINT"DOWN:YOU MAY NOW SAVE T
HE PROGRAM CODE TO "
140 PRINT"EITHER TAPE OR DISK."
150 DA=1
160 INPUT"DOWN:WHICH DEVICE ( T O
R D ) : "IDB
170 IFDB<>"D"ANDDB<>"T"THEN160
180 IFDB="D"THENDU=6
190 IFDU=6THEN220
200 PRINT"DOWN:PLACE IN/SONOBLANK
IN/SONOFF) CASSETTE IN CON UNIT, REM
IND"
210 PRINT"AND PRESS STOP/EJECT.">D
OTD230
220 PRINT"PLACE (R/SON)FORMATTEDIE
USOFF) DISK IN DRIVE#0 (DEV #). "
230 PRINT"DOWN:DOWN:PRESS '*' WE
EN READY .(DOWN:DOWN)"
240 POKE190,0
250 GETA:IFAS<>"*"THEN250
260 POKE190,0:POKE1,DA:SYSS2999
300 PRINT"DOWN:PLEASE VERIFY CODE
BY TYPING ."
310 PRINT"DOWN:TAPE VERSION = "VE
RIFY"CHR$(24)CHR$(24)",1,1"
320 PRINT"DOWN:DISK VERSION = "VE
RIFY"CHR$(24)"SEN"EN."CHR$(24)",B,
1"
330 PRINT"DOWN:THIS IS A PREDEFIN
EDRY MEASURE TO MAKE "
340 PRINT"SURE THAT THE CODE HAS B
EEN SAVED "
350 PRINT" CORRECTLY. "
360 END
400 REM * ANALYSE ERRORS AND REPOR
T *
410 IFTL<>490302THENPRINT"DOWN:ICH
ECKSUM ERROR."
420 IFBC=3002THEN460
430 IFBC<3002THENPRINT"DOWN:INSUF
FICIENT "":GOTO450
440 PRINT"DOWN:ITD MAYV ";
450 PRINT"DATA ITEMS."
460 PRINT"DOWN:CHECK DATA S
TATEMENTS CAREFULLY.":STOP
1000 DATA 100, 151, 146, 137, 148
, 148, 133, 142, 160, 130
1010 DATA 163, 160, 160, 106, 160
, 160, 138, 143, 136, 142
1020 DATA 180, 141, 131, 160, 136
, 129, 140, 133, 174, 160
1030 DATA 188, 131, 168, 160, 160
, 147, 133, 144, 148, 133
1040 DATA 141, 130, 133, 148, 160
, 180, 177, 185, 164, 180
1050 DATA 180, 175, 160, 160, 160
, 160, 160, 160, 131, 136
1060 DATA 129, 146, 129, 131, 148
, 133, 146, 160, 141, 143
1070 DATA 132, 130, 160, 106, 141
, 173, 133, 143, 140, 143
1080 DATA 148, 146, 160, 160, 141
, 143, 132, 133, 160, 168
1090 DATA 131, 136, 129, 148, 129
, 131, 146, 133, 146, 160
1100 DATA 160, 147, 133, 148, 160
, 166, 133, 132, 137, 148
1110 DATA 160, 160, 147, 133, 140
, 133, 131, 148, 143, 134
1120 DATA 134, 143, 142, 160, 168
, 147, 32, 210, 225, 163
1130 DATA 6, 141, 33, 208, 73, 8,
141, 32, 209, 162
1140 DATA 40, 163, 160, 157, 67,
6, 169, 7, 157, 67
1150 DATA 218, 208, 208, 243, 103
, 0, 162, 157, 11, 7
1160 DATA 169, 29, 152, 157, 91,
9, 169, 1, 157, 11
1170 DATA 216, 163, 3, 157, 91, 2

```



## Program Listing (cont.)

```

16. 169, 169, 167, 171
1180 DATA 4, 157, 211, 4, 157, 25
1, 4, 157, 35, 5
1190 DATA 157, 75, 5, 157, 171, 2
16. 73, 1, 157, 75
1200 DATA 217, 157, 211, 216, 157
, 251, 216, 157, 35, 217
1210 DATA 232, 224, 24, 208, 195,
169, 0, 133, 251, 133
1220 DATA 253, 170, 189, 4, 133,
252, 169, 216, 133, 254
1230 DATA 160, 0, 169, 160, 145,
251, 169, 4, 145, 253
1240 DATA 200, 192, 11, 208, 243,
169, 251, 24, 105, 40
1250 DATA 133, 251, 133, 253, 169
, 252, 105, 0, 133, 252
1260 DATA 105, 212, 133, 254, 252
, 224, 11, 208, 217, 169
1270 DATA 72, 133, 251, 169, 4, 1
33, 252, 162, 176, 160
1280 DATA 9, 138, 145, 251, 163,
40, 4, 165, 251, 24
1290 DATA 105, 41, 133, 251, 165,
252, 105, 0, 133, 252
1300 DATA 136, 232, 224, 184, 208
, 231, 162, 0, 189, 58
1310 DATA 182, 167, 211, 4, 189,
74, 182, 157, 251, 4
1320 DATA 189, 188, 193, 157, 35,
5, 232, 224, 16, 208
1330 DATA 233, 36, 120, 173, 14,
220, 41, 254, 141, 14
1340 DATA 220, 173, 17, 208, 41,
127, 141, 17, 208, 163
1350 DATA 132, 141, 20, 3, 169, 1
53, 141, 21, 3, 169
1360 DATA 177, 141, 18, 208, 173,
25, 208, 3, 1, 141
1370 DATA 25, 208, 88, 86, 173, 2
5, 208, 3, 1, 141
1380 DATA 25, 208, 104, 168, 104,
170, 104, 84, 168, 1
1390 DATA 44, 25, 208, 240, 243,
173, 24, 208, 41, 8
1400 DATA 208, 40, 180, 28, 174,
18, 207, 169, 48, 141
1410 DATA 18, 208, 173, 14, 207,
208, 8, 173, 22, 208
1420 DATA 41, 239, 76, 174, 193,
173, 22, 208, 3, 16
1430 DATA 141, 22, 208, 142, 33,
208, 140, 24, 208, 76
1440 DATA 118, 133, 162, 8, 180,
20, 169, 177, 141, 18
1450 DATA 208, 76, 161, 193, 151,
148, 129, 144, 129, 146
1460 DATA 143, 149, 142, 132, 160
, 160, 180, 150, 180, 188
1470 DATA 133, 142, 129, 130, 140
, 133, 132, 180, 132, 137
1480 DATA 147, 123, 130, 140, 139
, 132, 120, 163, 42, 141
1490 DATA 20, 3, 169, 234, 141, 2
1, 3, 173, 14, 220
1500 DATA 8, 1, 141, 14, 220, 88,
58, 0, 0, 0
1510 DATA 173, 14, 220, 41, 254,
141, 14, 220, 185, 1
1520 DATA 41, 251, 133, 1, 163, 0
, 139, 251, 133, 253
1530 DATA 169, 208, 133, 252, 169
, 48, 133, 254, 162, 0
1540 DATA 160, 0, 177, 251, 145,
253, 200, 208, 249, 239
1550 DATA 252, 230, 254, 232, 224
, 8, 208, 238, 185, 1
1560 DATA 9, 4, 133, 1, 173, 14,
220, 5, 1, 141
1570 DATA 14, 220, 234, 86, 168,
0, 133, 254, 182, 8
1580 DATA 10, 38, 254, 8, 252, 14
4, 7, 24, 101, 251
1590 DATA 144, 2, 230, 254, 202,
208, 239, 139, 253, 96
1600 DATA 173, 12, 207, 130, 251,
163, 8, 133, 252, 32
1610 DATA 82, 194, 165, 254, 24,
303, 48, 133, 254, 169
1620 DATA 82, 133, 248, 133, 251,
163, 4, 133, 250, 169
1630 DATA 216, 133, 252, 160, 0,
177, 253, 170, 152, 72
1640 DATA 136, 162, 8, 160, 0, 10
, 72, 144, 8, 168
1650 DATA 81, 145, 248, 145, 251,
208, 8, 168, 43, 145
1660 DATA 248, 168, 0, 145, 251,
104, 200, 208, 208, 231
1670 DATA 165, 248, 24, 105, 40,
133, 249, 133, 251, 166
1680 DATA 250, 165, 0, 133, 250,
303, 212, 133, 252, 194
1690 DATA 168, 200, 182, 8, 208,
185, 234, 96, 163, 64
1700 DATA 133, 251, 173, 13, 207,

```

Program Listing (cont.)

```

133, 252, 32, 62, 124
1710 DATA 162, 0, 160, 4, 163, 1,
    163, 204, 217, 153
1720 DATA 0, 218, 138, 24, 101, 2
53, 163, 224, 5, 24
1730 DATA 105, 32, 153, 0, 6, 232
    200, 182, 36, 208
1740 DATA 228, 189, 189, 133, 248
    133, 250, 188, 6, 133
1750 DATA 248, 188, 218, 133, 251
    163, 0, 133, 2, 162
1760 DATA 0, 160, 0, 138, 24, 101
    253, 145, 248, 173
1770 DATA 18, 207, 72, 173, 14, 2
    07, 240, 5, 104, 9
1780 DATA 0, 208, 1, 104, 145, 25
    0, 232, 200, 182, 18
1790 DATA 208, 227, 185, 248, 24,
    108, 80, 133, 248, 133
1800 DATA 250, 185, 245, 105, 0,
    133, 248, 105, 212, 133
1810 DATA 251, 230, 2, 162, 2, 20
    1, 4, 208, 198, 96
1820 DATA 163, 163, 133, 250, 133
    252, 163, 6, 133, 251
1830 DATA 163, 218, 133, 253, 163
    0, 133, 2, 170, 160
1840 DATA 0, 173, 12, 207, 145, 2
    50, 138, 145, 252, 232
1850 DATA 200, 200, 182, 0, 208,
    241, 185, 250, 24, 105
1860 DATA 60, 133, 250, 133, 252,
    165, 251, 108, 0, 133
1870 DATA 251, 105, 212, 133, 253
    230, 0, 185, 2, 201
1880 DATA 4, 208, 212, 96, 173, 4
    207, 133, 250, 173
1890 DATA 5, 207, 133, 251, 172,
    7, 207, 177, 250, 41
1900 DATA 63, 141, 12, 207, 169,
    64, 133, 251, 173, 13
1910 DATA 207, 133, 252, 32, 62,
    194, 173, 12, 207, 24
1920 DATA 101, 253, 141, 12, 207,
    98, 173, 0, 207, 133
1930 DATA 250, 173, 5, 207, 133,
    251, 172, 10, 207, 96
1940 DATA 32, 158, 125, 160, 0, 1
    63, 0, 133, 2, 177
1950 DATA 250, 72, 74, 74, 74, 74
    74, 74, 24, 101
1960 DATA 2, 133, 2, 104, 72, 10,
    10, 10, 10, 10
1970 DATA 10, 24, 101, 2, 133, 2,
    104, 72, 41, 48
1980 DATA 74, 74, 24, 101, 2, 133
    2, 104, 41, 12
1990 DATA 10, 10, 24, 101, 2, 145
    250, 200, 182, 0
2000 DATA 208, 189, 96, 32, 158,
    185, 180, 0, 188, 0
2010 DATA 133, 2, 162, 0, 177, 25
    0, 10, 72, 144, 9
2020 DATA 185, 2, 74, 5, 128, 133
    2, 208, 2, 70
2030 DATA 2, 104, 202, 208, 237,
    165, 2, 145, 250, 200
2040 DATA 182, 0, 208, 220, 96, 3
    2, 158, 185, 185, 7
2050 DATA 133, 2, 160, 0, 152, 72
    177, 250, 170, 184
2060 DATA 2, 177, 250, 72, 138, 1
    45, 250, 104, 170, 104
2070 DATA 168, 138, 145, 250, 138
    2, 200, 152, 4, 208
2080 DATA 228, 96, 152, 72, 177,
    250, 164, 2, 145, 250
2090 DATA 104, 168, 96, 32, 158,
    185, 188, 7, 183, 2
2100 DATA 160, 6, 32, 58, 186, 16
    9, 0, 145, 250, 188
2110 DATA 2, 136, 182, 256, 208,
    242, 96, 32, 158, 185
2120 DATA 162, 0, 134, 2, 160, 1,
    32, 58, 186, 230
2130 DATA 2, 200, 182, 0, 208, 24
    6, 136, 138, 145, 250
2140 DATA 86, 32, 158, 185, 180,
    0, 177, 250, 10, 145
2150 DATA 250, 200, 182, 0, 208,
    248, 96, 32, 158, 185
2160 DATA 163, 128, 133, 2, 160,
    0, 163, 0, 153, 247
2170 DATA 207, 135, 208, 250, 152
    72, 177, 250, 160, 0
2180 DATA 74, 144, 10, 72, 186, 2
    48, 207, 5, 2, 153
2190 DATA 248, 207, 104, 200, 182
    0, 208, 238, 70, 2
2200 DATA 104, 168, 200, 182, 0,
    208, 203, 160, 0, 182
2210 DATA 248, 207, 145, 250, 200
    198, 0, 208, 246, 96
2220 DATA 238, 18, 207, 96, 238,
    34, 208, 96, 238, 96
2230 DATA 208, 96, 238, 38, 208,

```

## Program Listing (cont.)

```

2200 DATA 19, 207, 95
2210 DATA 173, 17, 207, 95, 233,
1, 208, 1, 95, 141
2220 DATA 17, 207, 95, 173, 17, 2
27, 24, 105, 1, 144
2230 DATA 244, 95, 173, 14, 207,
73, 1, 141, 14, 207
2270 DATA 95, 101, 153, 95, 172,
3, 207, 135, 208, 21
2280 DATA 173, 23, 207, 240, 15,
159, 1, 141, 11, 207
2290 DATA 153, 95, 141, 0, 208, 1
88, 8, 141, 3, 207
2300 DATA 95, 140, 3, 207, 14, 11
, 207, 173, 0, 208
2310 DATA 95, 233, 8, 141, 0, 208
, 95, 172, 3, 207
2320 DATA 208, 152, 9, 208, 21, 1
73, 23, 207, 240, 15
2330 DATA 159, 128, 141, 11, 207,
153, 40, 141, 0, 208
2340 DATA 153, 1, 141, 3, 207, 95
, 140, 3, 207, 78
2350 DATA 11, 207, 173, 0, 208, 2
4, 105, 8, 141, 0
2360 DATA 208, 95, 174, 2, 207, 2
62, 208, 20, 173, 23
2370 DATA 207, 240, 24, 159, 105,
141, 0, 207, 159, 5
2380 DATA 141, 1, 207, 152, 8, 14
2, 2, 207, 202, 142
2390 DATA 10, 207, 159, 122, 141,
1, 208, 95, 142, 2
2400 DATA 207, 208, 10, 207, 173,
0, 207, 95, 233, 40
2410 DATA 141, 0, 207, 173, 1, 20
7, 233, 0, 141, 1
2420 DATA 207, 173, 1, 208, 95, 2
33, 8, 141, 1, 208
2430 DATA 95, 174, 2, 207, 232, 2
24, 8, 208, 30, 173
2440 DATA 23, 207, 240, 24, 159,
81, 141, 0, 207, 159
2450 DATA 8, 141, 1, 207, 159, 1,
142, 2, 207, 202
2460 DATA 142, 10, 207, 159, 95,
141, 1, 208, 95, 142
2470 DATA 2, 207, 233, 10, 207, 1
73, 0, 207, 24, 105
2480 DATA 40, 141, 0, 207, 173, 1
, 207, 105, 0, 141
2490 DATA 1, 207, 173, 1, 208, 24
, 105, 8, 141, 1
2500 DATA 208, 95, 173, 141, 2, 4
1, 1, 208, 4, 39
2510 DATA 37, 187, 95, 32, 250, 1
95, 95, 173, 141, 2
2520 DATA 41, 1, 208, 4, 32, 151,
157, 95, 32, 82
2530 DATA 157, 95, 152, 63, 159,
0, 157, 64, 3, 202
2540 DATA 208, 250, 73, 255, 157,
64, 3, 232, 232, 232
2550 DATA 224, 24, 208, 245, 95,
159, 128, 152, 74, 159
2560 DATA 70, 141, 137, 195, 142,
155, 135, 140, 174, 195
2570 DATA 32, 133, 195, 95, 159,
1, 152, 10, 159, 5
2580 DATA 75, 25, 195, 173, 0, 20
7, 133, 250, 173, 1
2590 DATA 207, 133, 251, 172, 3,
207, 95, 32, 159, 152
2600 DATA 159, 0, 177, 250, 73, 2
55, 145, 250, 200, 199
2610 DATA 8, 208, 245, 95, 159, 0
, 133, 250, 173, 13
2620 DATA 207, 10, 24, 105, 48, 1
33, 251, 152, 0, 159
2630 DATA 0, 177, 250, 73, 255, 1
45, 250, 200, 208, 247
2640 DATA 230, 251, 232, 224, 2,
208, 238, 95, 173, 141
2650 DATA 2, 41, 4, 208, 4, 32, 8
1, 158, 95, 32
2660 DATA 78, 195, 95, 173, 21, 2
68, 73, 1, 141, 21
2670 DATA 208, 95, 173, 81, 208,
73, 2, 141, 21, 208
2680 DATA 95, 152, 10, 141, 124,
195, 32, 117, 195, 95
2690 DATA 159, 74, 208, 245, 32,
212, 135, 75, 195, 194
2700 DATA 173, 4, 207, 130, 250,
173, 5, 207, 133, 251
2710 DATA 172, 7, 207, 95, 32, 15
8, 195, 153, 0, 145
2720 DATA 250, 95, 32, 159, 155,
153, 255, 145, 250, 95
2730 DATA 173, 14, 207, 208, 4, 3
2, 235, 155, 95, 32
2740 DATA 172, 195, 95, 159, 62,
141, 1, 208, 153, 40
2750 DATA 141, 0, 208, 159, 81, 1
41, 0, 207, 159, 4
2760 DATA 141, 1, 207, 152, 1, 14

```

Program Listing (cont.)

```

2, 2, 207, 142, 3
2770 DATA 207, 202, 142, 10, 207,
189, 189, 141, 11, 207
2780 DATA 173, 141, 2, 41, 1, 208
, 1, 35, 32, 158
2790 DATA 155, 189, 0, 145, 200,
200, 152, 5, 208, 249
2800 DATA 95, 182, 0, 189, 90, 19
2, 157, 75, 5, 232
2810 DATA 224, 18, 208, 245, 95,
173, 12, 207, 24, 105
2820 DATA 175, 141, 32, 5, 182, 0
, 173, 15, 207, 208
2830 DATA 5, 189, 105, 132, 208,
3, 182, 112, 132, 157
2840 DATA 225, 4, 232, 224, 6, 20
8, 235, 173, 23, 207
2850 DATA 208, 5, 189, 215, 133,
208, 3, 182, 208, 193
2860 DATA 157, 45, 5, 232, 224, 1
4, 208, 235, 173, 14
2870 DATA 207, 208, 5, 182, 104,
182, 208, 3, 189, 107
2880 DATA 192, 157, 204, 4, 232,
224, 17, 208, 235, 32
2890 DATA 9, 189, 35, 32, 23, 158
, 208, 19, 207, 32
2900 DATA 199, 199, 95, 173, 23,
207, 73, 1, 141, 23
2910 DATA 207, 32, 23, 199, 35, 1
73, 12, 207, 133, 251
2920 DATA 189, 5, 133, 252, 32, 6
2, 194, 155, 253, 141
2930 DATA 8, 207, 183, 254, 24, 1
05, 48, 141, 3, 207
2940 DATA 25, 32, 185, 154, 32, 2
3, 189, 32, 115, 135
2950 DATA 32, 123, 199, 32, 88, 1
24, 32, 52, 195, 95
2960 DATA 173, 13, 207, 24, 105,
1, 201, 7, 208, 1
2970 DATA 95, 141, 13, 207, 76, 5
3, 202, 173, 13, 207
2980 DATA 55, 233, 1, 201, 259, 2
08, 240, 95, 173, 21
2990 DATA 208, 41, 1, 141, 21, 20
8, 95, 173, 21, 208
3000 DATA 3, 2, 141, 21, 208, 95,
172, 7, 207, 200
3010 DATA 183, 23, 208, 90, 174,
5, 207, 232, 224, 3
3020 DATA 208, 48, 173, 23, 207,
208, 1, 95, 32, 185
3030 DATA 199, 183, 227, 141, 4,
207, 189, 5, 141, 5
3040 DATA 207, 189, 145, 141, 3,
208, 182, 1, 141, 8
3050 DATA 207, 141, 7, 207, 189,
55, 141, 2, 208, 189
3060 DATA 0, 141, 18, 208, 32, 20
5, 195, 95, 142, 8
3070 DATA 207, 32, 195, 199, 173,
4, 207, 24, 105, 40
3080 DATA 141, 4, 207, 173, 5, 20
7, 105, 0, 141, 5
3090 DATA 207, 173, 2, 208
, 24, 1
3100 DATA 24, 105, 5, 72, 144, 8,
173, 15, 208, 3
3110 DATA 2, 141, 18, 208, 104, 1
41, 2, 208, 95, 172
3120 DATA 7, 207, 135, 208, 88, 1
74, 5, 207, 202, 208
3130 DATA 95, 173, 23, 207, 208,
1, 35, 32, 155, 159
3140 DATA 183, 11, 141, 4, 207, 1
69, 6, 141, 5, 207
3150 DATA 183, 124, 141, 3, 208,
163, 2, 141, 8, 207
3160 DATA 183, 32, 141, 7, 207, 1
69, 48, 141, 2, 208
3170 DATA 183, 2, 141, 18, 208, 3
2, 205, 129, 25, 142
3180 DATA 5, 207, 32, 195, 199, 1
73, 4, 207, 55, 233
3190 DATA 40, 141, 4, 207, 173, 5
, 207, 233, 0, 141
3200 DATA 5, 207, 173, 3, 208, 95
, 233, 8, 141, 3
3210 DATA 208, 208, 203, 140, 7,
207, 173, 2, 208, 55
3220 DATA 233, 8, 72, 175, 5, 163
, 0, 141, 18, 208
3230 DATA 104, 141, 2, 208, 95, 1
69, 1, 141, 15, 207
3240 DATA 32, 23, 195, 32, 159, 2
55, 185, 157, 201, 2
3250 DATA 208, 28, 173, 141, 2, 4
1, 1, 208, 5, 32
3260 DATA 214, 199, 75, 230, 200,
32, 81, 200, 32, 2
3270 DATA 201, 32, 135, 199, 32,
155, 199, 75, 205, 200
3280 DATA 201, 14, 208, 240, 169,

```

## Program Listing (cont.)

```

0, 141, 15, 207, 32
3300 DATA 83, 189, 32, 14, 201, 9
6, 172, 17, 207, 162
3310 DATA 0, 202, 208, 253, 136,
209, 248, 95, 32, 158
3320 DATA 195, 160, 0, 177, 250,
153, 240, 207, 200, 188
3330 DATA 8, 208, 246, 96, 32, 15
8, 185, 160, 0, 185
3340 DATA 240, 207, 145, 250, 200
, 182, 8, 208, 246, 96
3350 DATA 32, 214, 133, 78, 47, 2
02, 32, 61, 200, 76
3360 DATA 47, 202, 32, 158, 135,
173, 11, 207, 73, 255
3370 DATA 49, 250, 145, 250, 96,
32, 158, 195, 177, 250
3380 DATA 13, 15, 207, 145, 250,
32, 37, 197, 76, 99
3390 DATA 194, 32, 250, 195, 32,
58, 201, 78, 88, 184
3400 DATA 32, 58, 201, 32, 37, 12
7, 78, 88, 134, 32
3410 DATA 50, 137, 76, 151, 137,
0, 51, 2, 7, 50
3420 DATA 94, 0, 1, 21, 18, 20, 4
5, 53, 48, 46
3430 DATA 48, 54, 40, 43, 17, 33,
10, 20, 6, 3
3440 DATA 4, 5, 32, 57, 36, 58, 5
9, 8, 9, 62
3450 DATA 42, 13, 22, 0, 0, 0, 20
7, 222, 237, 56
3460 DATA 71, 87, 105, 188, 178,
218, 229, 208, 30, 104
3470 DATA 52, 48, 188, 183, 134,
21, 18, 38, 145, 154
3480 DATA 83, 68, 112, 188, 238,
200, 204, 138, 111, 170
3490 DATA 58, 23, 31, 0, 0, 0, 13
8, 137, 137, 201
3500 DATA 201, 201, 201, 199, 199
, 195, 195, 196, 201, 205
3510 DATA 201, 201, 199, 199, 199
, 196, 199, 199, 199, 199
3520 DATA 186, 186, 199, 199, 199
, 195, 195, 196, 195, 202
3530 DATA 205, 205, 205, 0, 0, 0,
32, 153, 255, 162
3540 DATA 0, 155, 137, 221, 112,
201, 208, 15, 189, 152
3550 DATA 201, 141, 48, 3, 189, 1
99, 201, 141, 47, 3
3560 DATA 108, 48, 3, 232, 224, 3
7, 208, 231, 35, 81
3570 DATA 4, 1, 3, 227, 5, 1, 1,
0, 48, 0
3580 DATA 188, 0, 0, 0, 0, 1, 40,
10, 0, 0
3590 DATA 1, 0, 1, 182, 0, 183, 2
, 202, 157, 0
3600 DATA 207, 232, 224, 24, 208,
245, 56, 32, 155, 133
3610 DATA 78, 14, 201, 32, 143, 1
99, 78, 14, 201, 163
3620 DATA 0, 32, 210, 255, 162, 0
, 32, 207, 255, 201
3630 DATA 13, 210, 10, 157, 32, 2
07, 232, 224, 15, 208
3640 DATA 241, 163, 13, 142, 22,
207, 32, 210, 255, 96
3650 DATA 163, 0, 141, 25, 208, 1
41, 21, 208, 163, 6
3660 DATA 141, 32, 208, 73, 8, 14
1, 32, 208, 169, 147
3670 DATA 32, 210, 255, 56, 163,
28, 141, 24, 208, 162
3680 DATA 0, 138, 157, 0, 4, 163,
1, 157, 0, 218
3690 DATA 232, 208, 244, 162, 0,
188, 145, 202, 32, 210
3700 DATA 255, 232, 224, 21, 208,
245, 108, 2, 3, 17
3710 DATA 17, 17, 17, 17, 17, 17,
13, 81, 85, 73
3720 DATA 84, 48, 13, 62, 63, 65,
68, 69, 48, 13
3730 DATA 32, 30, 202, 76, 125, 2
05, 12, 15, 1, 4
3740 DATA 76, 13, 1, 22, 5, 46, 1
73, 14, 220, 41
3750 DATA 224, 141, 14, 220, 165,
1, 41, 251, 133, 1
3760 DATA 168, 48, 133, 247, 183,
207, 133, 248, 162, 0
3770 DATA 138, 78, 188, 178, 202,
133, 251, 169, 8, 133
3780 DATA 252, 224, 32, 62, 184,
155, 224, 24, 105, 208
3790 DATA 133, 224, 160, 0, 177,
253, 145, 247, 200, 182
3800 DATA 8, 208, 247, 188, 247,
24, 105, 8, 133, 247
3810 DATA 165, 218, 105, 0, 133,
248, 104, 170, 232, 224
3820 DATA 10, 208, 203, 165, 1, 9

```

Program Listing (cont.)

```

. 7, 133, 1, 173
3830 DATA 14, 220, 5, 1, 141, 14,
220, 55, 173, 13
3840 DATA 207, 133, 255, 32, 159,
255, 155, 197, 201, 1
3850 DATA 240, 31, 201, 40, 208,
5, 32, 168, 199, 78
3860 DATA 53, 209, 201, 43, 208,
3, 37, 185, 189, 173
3870 DATA 92, 5, 73, 128, 141, 52
. 5, 32, 2, 201
3880 DATA 78, 27, 203, 173, 13, 2
07, 10, 24, 105, 48
3890 DATA 133, 253, 169, 0, 133,
250, 133, 252, 155, 255
3900 DATA 10, 24, 105, 48, 133, 2
51, 122, 0, 169, 0
3910 DATA 177, 240, 145, 252, 200
, 208, 249, 230, 251, 230
3920 DATA 253, 232, 224, 2, 208,
238, 155, 255, 141, 13
3930 DATA 207, 173, 92, 5, 9, 128
, 141, 92, 5, 78
3940 DATA 53, 202, 188, 80, 133,
247, 133, 248, 133, 251
3950 DATA 169, 4, 133, 248, 133,
250, 162, 0, 189, 72
3960 DATA 160, 0, 162, 72, 177, 2
53, 160, 0, 10, 144
3970 DATA 17, 72, 162, 180, 145,
245, 185, 250, 24, 105
3980 DATA 212, 133, 252, 168, 1,
145, 251, 104, 200, 182
3990 DATA 8, 208, 231, 185, 245,
24, 105, 40, 133, 248
4000 DATA 133, 251, 165, 250, 105
, 0, 133, 250, 104, 188
4010 DATA 200, 192, 0, 208, 203,
165, 247, 24, 105, 0
4020 DATA 133, 247, 133, 248, 133
, 251, 165, 248, 105, 0
4030 DATA 133, 248, 133, 250, 165
, 253, 24, 105, 0, 133
4040 DATA 253, 165, 254, 105, 0,
133, 254, 104, 170, 232
4050 DATA 224, 5, 208, 180, 58, 1
53, 207, 133, 254, 173
4060 DATA 20, 207, 208, 4, 162, 4
8, 208, 2, 169, 68
4070 DATA 133, 253, 78, 125, 203,
169, 0, 133, 250, 133
4080 DATA 252, 185, 170, 240, 5,
162, 48, 188, 55, 208
4090 DATA 8, 173, 13, 207, 32, 37
, 205, 224, 224, 133
4100 DATA 253, 134, 251, 55, 32,
32, 39, 17, 35, 32
4110 DATA 61, 32, 17, 21, 9, 20,
32, 59, 32, 39
4120 DATA 43, 39, 32, 61, 32, 20,
1, 18, 5, 32
4130 DATA 53, 32, 39, 55, 39, 32,
61, 32, 4, 9
4140 DATA 19, 11, 32, 32, 162, 40
, 189, 31, 204, 157
4150 DATA 183, 3, 183, 7, 167, 18
3, 217, 202, 208, 242
4160 DATA 55, 32, 2, 201, 162, 40
, 189, 183, 3, 73
4170 DATA 128, 157, 183, 5, 202,
208, 245, 55, 0, 28
4180 DATA 29, 29, 68, 78, 84, 65,
62, 32, 70, 73
4190 DATA 78, 69, 78, 65, 77, 68,
32, 65, 78, 68
4200 DATA 32, 60, 62, 68, 83, 83,
32, 39, 62, 69
4210 DATA 84, 65, 62, 78, 38, 13,
13, 25, 25, 23
4220 DATA 0, 70, 73, 76, 68, 78,
65, 77, 65, 32
4230 DATA 58, 32, 162, 13, 128, 3
2, 218, 255, 202, 208
4240 DATA 250, 188, 108, 204, 32,
210, 255, 232, 224, 54
4250 DATA 208, 245, 78, 59, 202,
32, 115, 205, 183, 0
4260 DATA 141, 32, 208, 141, 33,
208, 32, 239, 203, 32
4270 DATA 180, 204, 32, 72, 204,
231, 234, 231, 231, 231
4280 DATA 234, 234, 32, 159, 255,
185, 187, 201, 62, 208
4290 DATA 3, 76, 252, 206, 201, 5
5, 208, 5, 169, 1
4300 DATA 78, 243, 204, 201, 27,
240, 5, 32, 89, 204
4310 DATA 78, 210, 204, 188, 8, 1
41, 21, 207, 189, 1
4320 DATA 188, 174, 21, 207, 32,
188, 255, 173, 22, 207
4330 DATA 152, 32, 180, 207, 32,
185, 255, 173, 20, 207
4340 DATA 208, 6, 32, 219, 258, 7
5, 32, 206, 32, 3
4350 DATA 204, 188, 250, 188, 232

```



## Program Listing (cont.)

```

, 184, 253, 32, 215, 255
4360 DATA 104, 104, 76, 252, 206,
10, 24, 105, 48, 170
4370 DATA 106, 2, 95, 169, 0, 141
, 20, 207, 76, 183
4380 DATA 204, 169, 1, 208, 246,
173, 8, 207, 72, 173
4390 DATA 9, 207, 72, 32, 197, 20
0, 32, 168, 195, 104
4400 DATA 133, 253, 104, 133, 252
, 180, 0, 177, 252, 145
4410 DATA 250, 200, 152, 8, 208,
247, 76, 14, 201, 173
4420 DATA 141, 2, 41, 4, 208, 3,
76, 39, 206, 76
4430 DATA 22, 203, 173, 21, 208,
9, 2, 76, 17, 206
4440 DATA 76, 197, 200, 32, 90, 2
02, 189, 0, 139, 199
4450 DATA 78, 230, 183, 32, 230,
183, 183, 0, 133, 158
4460 DATA 104, 104, 76, 114, 202,
169, 0, 141, 16, 208
4470 DATA 141, 21, 208, 141, 23,
209, 141, 29, 209, 32
4480 DATA 252, 197, 169, 13, 141,
249, 7, 141, 249, 7
4490 DATA 169, 1, 141, 39, 208, 1
41, 40, 209, 169, 40
4500 DATA 141, 0, 208, 169, 66, 1
41, 1, 209, 169, 58
4510 DATA 141, 2, 208, 169, 146,
141, 3, 209, 234, 66
4520 DATA 234, 234, 234, 234, 234,
, 234, 234, 234, 32, 188
4530 DATA 202, 32, 137, 206, 92,
33, 202, 162, 8, 189
4540 DATA 255, 47, 157, 239, 207,
202, 209, 247, 32, 124
4550 DATA 192, 32, 76, 193, 32, 1
78, 189, 173, 21, 208
4560 DATA 41, 1, 141, 21, 208, 32
, 232, 201, 32, 88
4570 DATA 194, 32, 2, 201, 32, 12
7, 188, 76, 229, 205
4580 DATA 169, 2, 139, 2, 160, 0,
162, 0, 202, 209
4590 DATA 253, 136, 208, 248, 199
, 2, 208, 242, 78, 192
4600 DATA 205, 41, 2, 141, 21, 20
8, 76, 197, 200, 163
4610 DATA 0, 133, 170, 173, 141,
2, 41, 4, 133, 170
4620 DATA 78, 50, 208, 173, 21, 2
08, 41, 2, 141, 21
4630 DATA 208, 78, 27, 205, 169,
20, 141, 21, 208, 76
4640 DATA 53, 205, 169, 20, 141,
24, 208, 76, 76, 206
4650 DATA 3, 32, 8, 32, 1, 32, 18
, 32, 1, 32
4660 DATA 3, 32, 20, 32, 5, 32, 1
8, 32, 32, 7
4670 DATA 32, 5, 32, 14, 32, 5, 3
2, 18, 32, 1
4680 DATA 32, 20, 32, 15, 32, 18,
32, 38, 54, 52
4690 DATA 16, 32, 18, 32, 5, 32,
18, 32, 18, 32
4700 DATA 32, 1, 32, 14, 32, 25,
32, 32, 111, 32
4710 DATA 5, 32, 25, 32, 32, 32,
20, 32, 15, 32
4720 DATA 32, 19, 32, 20, 32, 1,
32, 18, 32, 20
4730 DATA 169, 48, 141, 17, 207,
234, 234, 234, 169, 147
4740 DATA 32, 210, 255, 169, 142,
32, 210, 255, 169, 8
4750 DATA 32, 210, 255, 169, 0, 1
41, 32, 208, 141, 33
4760 DATA 208, 189, 64, 133, 197,
182, 40, 189, 105, 208
4770 DATA 157, 183, 5, 169, 1, 15
7, 183, 217, 189, 65
4780 DATA 206, 157, 205, 3, 169,
7, 157, 205, 215, 202
4790 DATA 209, 231, 165, 187, 204
, 64, 240, 3, 76, 236
4800 DATA 206, 32, 89, 204, 162,
40, 189, 255, 3, 73
4810 DATA 129, 157, 255, 3, 202,
209, 245, 76, 208, 208
4820 DATA 173, 111, 201, 72, 208,
3, 32, 254, 193, 104
4830 DATA 8, 1, 141, 111, 201, 23
4, 234, 76, 192, 205
4840 DATA 169, 1, 188, 2, 180, 1,
32, 188, 255, 169
4850 DATA 7, 182, 35, 180, 207, 3
2, 189, 255, 169, 0
4860 DATA 133, 250, 189, 132, 133
, 251, 169, 250, 162, 0
4870 DATA 160, 207, 76, 216, 255,
71, 69, 76, 39, 54
4880 DATA 62, 46, -1

```

# PROGRAMMED TO WIN!



## Win a unique Austin Metro worth £5,500 with the October Issue of

### 64 TAPF COMPUTING

Get motoring this Winter with a special edition Austin Metro. You choose the tires and fittings for a real eye-catcher. All you have to do is become our Tape Magazine Computer Champion.

We are having a two-part competition in our October and November issues.

It's easy to enter and free.

Just answer the set of computer-related questions in each competition issue and enter your Tie Breaker.

Can you make the grade and become our Computer Champion?

The competition closes on 30 November 1985.

Get motoring and order your issues now... before they sell out.



Available every month from all branches of W H Smiths, John Menzies and other leading newsagents.

Argus Tape Magazines produced by 64 Tape Computing Infonet Limited Times House 179 The Marlowes Hemel Hempstead Herts HP1 1BB

GREAT GAMES - GREAT UTILITIES - GREAT VALUE



# COMMODORE PLUS 4 AND C16

CAN YOU MEET THE CHALLENGE OF:-

# HIT & MISS

HIT & MISS IS THE CHALLENGE  
BUT - THE QUESTION IS -  
CAN YOU MEET IT?

FREE A3 POSTER  
FREE PACKAGE &  
POSTAGE  
NEW - IDEAS IN ACTION  
ONLY - FOR PLUS 4/C16

TAPE  
£7.99



WARNING - PLAYING

## HIT & MISS

COULD UNRESTURB DAMAGE YOUR  
HEALTH, BUT IT CERTAINLY WILL NOT  
DAMAGE YOUR WEALTH

"MORE ADDICTIVE THAN  
ANY ARCADE GAME, THE  
ONLY C16/PLUS 4 SOFTWARE  
OFFERING A REAL  
CHALLENGE"

Design by Nigel Whitbourn Graphic Design, Leeds

HIT & MISS DEFINITELY IS A MIND CHALLENGE OF MEGA  
PROPORTION THAT ALL YOUR FAMILY AND FRIENDS CAN TAKE  
PART IN. FROM SINGLE PLAY TO DOUBLES IT WILL STRETCH YOUR  
MENTAL DEXTERITY TO THE FULL.....

BUT REMEMBER -  
IT TAKES MORE THAN JUST GUESSEWORK TO  
BECOME A GENIUS WITH HIT & MISS

**VENTUREGATE LTD**

AVAILABLE AT MOST GOOD SOFTWARE OUTLETS

CHECK YOUR COPY  
ROAD INFORMATION, 17 MARSH ROAD, WYKESHIRE, LEIC. LE14 1LW

NAME (BLOCK LETTERS) \_\_\_\_\_

ADDRESS \_\_\_\_\_

COUNTRY \_\_\_\_\_

POSTCODE \_\_\_\_\_

PLEASE SEND ME \_\_\_\_\_ COPIES OF "HIT & MISS" AT  
£7.99 COMPLETE WITH A FREE POSTER WITH EVERY COPY.

PRICE A3 POSTER AND TAPES £14.95



**Mike Hart provides a couple of handy plotting routines for the Vic 20 and C64.**

# RELIABLE ROUTINES

AT ONE TIME OR ANOTHER, MANY people must have experimented with their machines to see if there is an easy way to draw graphs or plots on the screen. There are three ways in which this can be done and I will call them low resolution, medium resolution and high resolution respectively.

In low-resolution plotting, one merely takes the screen as a grid and plot points using asterisks or a similar graphic characters. In the case of a C64 this would obviously be a grid of  $40 \times 25$  giving 1000 potential plot points.

If one wished to use a high resolution screen then on the C64 one would use bit-mapped graphics which allows each individual 'dot' on the screen to be controlled. Using this mode will increase the resolution quite dramatically to  $320 \times 200$  which is 64000 addressable dots. However using the C64 in this mode is quite complicated and needs to be approached with a degree of caution. The approach which I am going to adopt here is one that gives pleasant 'chunky' graphics points and is half-way between these two extremes and which, therefore, I shall call medium-resolution graphics.

If you were to PRINT CHR\$(64), CHR\$(68), CHR\$(72) and CHR\$(76) as a brief experiment you would see that a graphic character is generated which consists of a quarter square either by itself or in combination with another 'quarter-square' to make up a full square. There are 16 of these combinations altogether and together with space and square space we have a total of sixteen permutations of 'quarter-square' graphics. This enables us to make a resolution which is twice that of the normal screen and therefore instead of having  $40 \times 25$  we can increase this to  $80 \times 50$  giving us a resolution of 4000 plot points.

Let us suppose that we wish to plot a point at the bottom left hand corner of the screen. The complicating factor that we have to take into account is that there might already be a graphics character already occupying that position and we would wish to preserve the point that it represents. The trick here is to PEEK the screen at that particular point, read the value of the character at that point, look up its value in a table of potential values and then work out from the same table the value of the new character to be plotted back into the screen which took place the 'new' point and the value of the existing point. It is a little hard to visualise these any of Ramo-Went's books have an excellent explanation under the

## Program Listing

### MEDIUM-RES PLOT

```

1 REM *** MEDIUM-RES PLOT (BASIC) ***
2
3 REM **      N. C. HART      **
4
5 DIM C(125),R(1,1)
6 FOR J=0 TO 124:READ C(J):NEXT J
7 DATA 32,123,188,228,188,87,127,222
8 DATA 124,222,222,224,222,222,221,122
9
10 R(0,0)=1:R(0,1)=2:R(1,0)=3:R(1,1)=4
11
12 FOR J=0 TO 99:READ H:POKE J,H:NEXT J
13 DATA 32,241,123,123,122,122,122,122,122,122
14 DATA 218,127,2,217,127,2,218,127
15 DATA 2,218,222,222,241,222
16 REM DATA 2,121,222,222,241,222
17
18
19 PRINT CHR$(147):CHR$(144):REM BLACK
20 COL=0:REM BLACK
21 POKE32226,12:POKE32261,12:REM GREY
22 REM VIC POKE 32672,22
23
24 SYS 940,COL:REM SET COLOUR MEMORY
25 FOR J=1 TO 2:FOR H=0 TO 75
26 REM VIC FOR J=1 TO 2:FOR H=0 TO 29
27 T=22+21+2:IN(R(0,0))+225:22 1222
28 NEXT X:T=NOT(T):NEXT J
29 GOTO 2222
30 END REM VIC ONLY
31
32 REM PLOT SUBROUTINE
33
34 NL=INT(X/2)+1:NR=N-NL
35 YL=INT(Y/2)+1:YR=Y-INT(Y/2)+1
36 N=(YR,NS)
37 P=124-40+YL*NL
38 REM VIC BK+ P=4222-22*YL*NL
39 REM VIC CR P=2124-22*YL*NL
40 FOR I=0 TO 15:IF PEEK(I) < 120:GOTO 12:THEN NEXT I
41 IF C OR NR OR T=-1 THEN I=1 AND NOT(I)
42 POKE P,C(I):RETURN
43
44
45

```





Nick Hampshire reveals the mysteries of the TED chip in the C-16 and Plus/4.

# TED CHIP

THE GRAPHICS DISPLAY, SOUND GENERATION and internal clock/timers of the C-16 and Plus/4 computers are controlled by a single integrated circuit, the so-called TED chip. This is a complex device, and, unfortunately, rather difficult to use.

An equally unfortunate circumstance is that no information on the use of this chip is provided in Commodore's manual. This is presumably in the belief that the graphics and sound commands supplied in the extended Basic are adequate. However, most advanced programmers,

especially those writing machine code programs, will want direct access to the registers of this device.

The TED chips a rather strange device. It is located in the middle of the internal ROM area and overlays this ROM so that the ROM area covered by TED is inaccessible. In addition, the TED registers are not grouped in one continuous area of memory. We located TED registers in the area \$1000 to \$103F. The reason for this is obscure and probably related to a quirk in the chip's design.

In operation the TED chip is not unlike the VIC and SID chips in the C64 and it is worth studying one of the advanced books on the 64 (for instance, *Advanced Commodore 64: Graphics and Sound* and *The Commodore 64: Circuit and Hardware Revealed* - both by Nick Hampshire).

The following table shows the locations in TED which we have uncovered together with the function of each register and the bits within each location.

## TED Graphics/sound/keyboard control.

FF00	— Timer 1 low	5	1 = Voice 2 tone enable
FF01	— Timer 1 high	4	1 = Voice 1 enable
FF02	— Timer 2 low	3-0	Volume (0-4 only)
FF03	— Timer 2 high	FF10	Bit 3
FF04	— Timer 3 low	2	Bit map base
FF05	— Timer 3 high	1-0	1 = chars from ROM, 0 = chars from RAM
FF06	— Video control 1	FF11	Bit 3
Bit 7	Not used	2	1 = Voice 1 high
6	1 = extended background	FF12	Bit 7-3
5	1 = Bit map	2	— Character base address
4	0 = blank screen	1	Address of UD5c (28 steps)
3	1 = 25 lines, 0 = 24 lines	FF14	Bit 7-3
2-0	Vertical smooth scroll pos	2	— Screen base address
FF07	— Video control 2	FF15	Bit 7
Not used	Not used	6-4	— Address of colour memory
6	1 = 48 columns, 0 = 38 columns	3-0	(28 steps, screen 15 above colour)
5	1 = 48 columns, 0 = 38 columns	FF16	Bit 7
4-3	Horizontal smooth scroll pos	Not used	— Background colour
FF08	— Out 1 keyboard column or joystick (PD or PA) in keyboard row or joystick switches	6-4	Not used
FF09	— Interrupt control	3-0	Luminance (0-7)
Bit 5	T1 has run out	FF17	Bit 7
4	T2 has run out	6-4	— Extended back 1/4M col 1
3	T3 has run out	3-0	Not used
1	Raster compare occurred	FF18	Bit 7
FF0A	— Interrupt enable	6-4	Luminance (0-7)
Bit 5	1 = T1 enable	3-0	Colour (0-15)
4	1 = T2 enable	FF19	Bit 7
3	1 = T3 enable	6-4	— Extended back 1
1	1 = Raster enable	3-0	Not used
FF0B	— Raster compare low byte	FF1A	Bit 7
FF0C	— Screen offset from base for cursor (high byte)	6-4	Luminance (0-7)
FF0D	— Screen offset from base for cursor (low byte)	3-0	Colour (0-15)
FF0E	— Voice 1 low byte	FF1B	— Border colour
FF0F	— Voice 2 low byte	Not used	Not used
FF10	— Bits 1-0 voice 2 high	6-4	Luminance (0-7)
FF11	— Sound control	3-0	Colour (0-15)
Bit 7	Disable sound	FF1C	— Bit 0: Raster position high bit
6	1 = Pause on voice 2	FF1D	— Raster position low byte
		FF1E	— ROM in when written to
		FF1F	— ROM out when written to
		FF20	— Bit 2: cassette switch sense (0=down)
		FF21-FF2F	— res. low, FF40=high

# Scratchpad

**A fistful of DATAs — more of your useful programming goodies presented by Max Phillips.**

WELCOME TO THE PAGE FOR YOUR programming bits and pieces. We're interested in anything, useful or amusing, from a few FORs to a short utility and we'll pay for anything we use, just send us your program (either as a listing or on cassette or disk) and some notes as to what it does and how it does it. Past year contributions to Scratchpad: Your Commodore, No 1 Golden Square, London W1R 3AD.

## That 'T'! Sleepin Keyboard

Andreas Weinand ticks off this month with a machine code utility that produces a beep every time a key is pressed. It might be a help for anyone who has a clicks keyboard or can't manage two finger typing but watch out — it does mean everyone else can hear what your typing speed is like!

Andreas' program takes over the C64's regular interrupt and uses voice 1 on the

NO chip to provide an expensive sounding beeper.

## Clean Living

Wash is a handy disk utility supplied by Clifford Hanger of Manningtree in Essex. It lets you step through all the files on a disk one by one using the space bar. You can delete the current file by pressing X or rename it with R. Even better, pressing Y lets you take a peek at what's in the file so you can figure out what it is and then scratch it if you don't need it.

The VIEW command works with all files though it cuts out any control characters and replaces them with full stops. Even so, you should be able to get a good idea of what the file is. One problem — after viewing a file, Wash sometimes goes back to the title of the disk — just hit space a few times to go on again. It just goes to show — you can write a useful utility in Basic!

## Oh No, Not Again...

Printing at a particular position on the screen is still the most talked about subject in C64 programming. Graham Bligh of Earleigh in Hampshire has pointed out that David Reed's routine in Your 64 issue 13 won't let you print on the top line of the screen because you've got

to POKE 214,0-1 and then PRINT. Oh yes — I wondered when someone would spot that!

Graham has a neat solution — call the routine in the kernel ROM which recalculates the cursor position. The PRINT-AT routine then becomes:

```
POKE 214,POKE 214/255:MD10
```

Meanwhile, Asmat Ullah from Glasgow has a remarkably clever machine code solution. It's only 12 bytes of code so it should be no trouble to include it in your installation section. Once loaded, you can point anywhere on the screen with:

```
SYSTEM%Y,"MESSAGE"
```

Now huh? Asmat does point out that there's no error checking in the routine in order to keep it short so you should make sure that X is less than or equal to 40 and Y is less than or equal to 25. Even so, it has to be the best yet. Unless, of course, you know different...

```
1 REM PRINT-AT ASMAT ULLAH
2 FOR %X=0 TO 63:READ Y:POKE
  X,Y:NEXT
3 DATA 32,124,163,194,26,24,32,340,35,7
```

## Clean Cut Characters

Paul Barnham of Darwen in Lancashire has sent in a delightful utility that replaces all those hard to read control codes in listings with clear mnemonics such as [CLR], [END], [UP] and so on. Just enter POKE 56,15:CLR and run the listing below. You'll find LIST a changed command...

### ANDREAS WEINAND LISTING

```
100 REM KEY BEEP BY ANDREAS WEINAND
110 REM START WITH SYS 836
120 FOR I=800 TO 990
130 READ X:POKE I,X:SYS+X:NEXT
140 DATA 88, 18,141, 24,212,165, 64,141,
  9,212,188, 88
150 DATA 94, 8,218,188, 80,162,157,141,
  1,212,170, 0
160 DATA 22,188,105,188, 3,141, 80, 3,
  142, 21, 3, 80
170 DATA 22,203,201, 84,208, 3, 78, 48,
  239,162, 17,162
180 DATA 9,212,141, 87, 27,188,139,188,
  3,141, 20, 3
190 DATA 12, 21, 3,188, 12,141, 88, 87,
  78, 10,239,185
200 DATA 203,205, 87, 27,208, 12,174, 88,
  27,202,210, 8
210 DATA 162, 80, 27, 78, 48,239,188, 0,
  141, 9,212,168
220 DATA 100,162, 3,141, 80, 3,142, 21,
  3, 78, 48,239
230 DATA 0
240 IF %K=1:GOSUB 250:PRINT "ERROR IN DAT
  A":GOTO
250 PRINT "OK"
```

### CLIFFORD HANGER LISTING

```
10 REM DIRECTORY WASH: C HANGER 1985
20 PRINT "C",TAB(18)," WASH",PRINT
30 PRINT "... FOR CLEANER DIRECTIONS"
40 REM READ DIR
50 OPEN "B,15
60 OPEN L,"B,0","NO"
70 GET#1,AS,PS
80 GET#1,AS,PS,AS,PS
90 C=2:IF AS="*" THEN C=ASC(AS)
100 IF AS="*" THEN C=ASC(PS)+255
110 CB=STR$(C):CB=LEFT$( " ",1-LEN(CB)
  )+CB
120 GET#0,SS:IF SS="0 THEN PRINT:PRINT C
  " * BLANKS FREE":CLOSE 1:GOTO 60
130 IF SS="CHR$(39)" THEN 150
140 PS
150 GET#0,AS:IF AS="CHR$(39)" THEN PS=PS
  +SS:GOTO 150
160 SET#0,AS:IF AS=" " THEN 180
170 TS=PS
180 SET#0,AS:IF AS="*" THEN TS=TS+AS:GO
  TOS0
190 IF C=0 THEN PRINT:PRINT "TITLE" ,"F
  %":PRINT:GOTO 60
200 PRINT PS,TAB(18)," WASH",CB,END(C?)
  ,"STR$( " ",LEFT$(TS,3)
210 PRINT:PRINT "RECURSION DEPTH:";DUM
```





# YOUR COMMODORE

Lineage: 60p per word.

Serial display: £9.00 per single column centimetre

Ring for information on series bookings/discounts



## 01-437 0699

EXT. 312

Send your requirements to:  
**DAVID HORNE**  
ASP LTD, 1 GOLDEN SQUARE,  
LONDON W1

All advertisements in this section must be prepaid.  
Advertisements are accepted subject to the terms and conditions  
printed on the advertisement rate card (available on request)

### HARDWARE

## NEW STAR SG-10C PRINTER

ONLY £248 TO INCLUDE VAT & NEXT DAY  
DELIVERY BY SECURON

- 100% compatible with your Commodore computer.
- 12lines draft mode.
- 6pin and 8pin foot (not standard).
- Paper release (Stacy's mode included).
- 150 graphics mode characters.
- 150 business mode characters.
- 85 data characters.
- 80 NLD characters.
- Manual 80 pin Commodore 1301 not.
- Changed 25, 35 & 45.
- Supplied with automatic cable and main plug READY TO GO!

### STRONG COMPUTER SYSTEMS

8000A CONTACT: PUNEL, CANNATHAN, 0710 3402 701  
TEL: 0203 21246 FOR ASSAULTS!!

014

### • BACK-UP SYSTEMS FOR CBM 64 & VIC-20 •

The Back-Up Station 21 is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20.

The Back-Up Station 21 is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20.

The Back-Up Station 21 is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20.

The Back-Up Station 21 is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20.

The Back-Up Station 21 is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20.

The Back-Up Station 21 is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20. It is a complete self-contained system for the Commodore 64 and VIC-20.

### FOR SALE

#### ELECTRICAL CONTRACTORS

Company with 20 years experience in the electrical industry. We are now looking for a new area to expand our business. We are looking for a new area to expand our business. We are looking for a new area to expand our business.

**TONY SPRY,**  
13 Tador Drive,  
Barnstable, Devon.

**CBM 64** quality colour monitor plus TV complete system\* including good software library for sale. £195.00. CBM 6400 Ring Road, Chesham, Bucks. HP83 8JN. Tel: 0494 211111.

**1000 PRINTER** immaculate condition. Handy. Used. Power supply at £200.00. Telephone calls 01273 856 700.

**SOCCER MANAGER** (Commodore 64) Manager your own team! Four divisions. F.A. Cup. Transfers etc. Send £3.50. Palmer, 26 Park Road, Shefford S6 7JG.

#### WOODRAKER 05.95

1000 PRINTER. Immaculate condition. Handy. Used. Power supply at £200.00. Telephone calls 01273 856 700.

## Super Breaker Back-Up Board

For the Commodore 64, C64C and 6400, and offers a complete back-up system for your Commodore 64. It is a complete self-contained system for the Commodore 64. It is a complete self-contained system for the Commodore 64.

The Super Breaker Back-Up Board is a complete self-contained system for the Commodore 64. It is a complete self-contained system for the Commodore 64. It is a complete self-contained system for the Commodore 64.

For enquiries telephone 01437 0699

### REPAIRS

#### COMMODORE REPAIRS

Commodore repairs to Commodore computers, monitors, modems, printers, etc. We are now looking for a new area to expand our business. We are looking for a new area to expand our business.

For fast reliable repair service send you R60-00 modulator (7.50), and R60-00 from £13.00 and Commodore 64 from £16.00 plus postage and packing, to:

**HEMEL COMPUTER CENTRE**  
52 High Street  
Hemel Hempstead  
Herts HP1 1AP  
Tel: (0462) 212426

#### COMMODORE C11

DUAL CASSETTE INTERFACE. Transfer any file cassette program buffer. Works with all Commodore 64. It is a complete self-contained system for the Commodore 64. It is a complete self-contained system for the Commodore 64.

**J. FITZPATRICK (YOL)**  
1 Ashley Ave, Widdington,  
Manchester M20 9UG.

## TO FILL THIS SPACE RING

01-437 0699

### ACCESSORIES

#### MAPMAKER

Simply the best way of making addresses, maps, etc. on your Commodore 64. It is a complete self-contained system for the Commodore 64. It is a complete self-contained system for the Commodore 64.

### FOR SALE

#### GADGET H 05.95

1000 PRINTER. Immaculate condition. Handy. Used. Power supply at £200.00. Telephone calls 01273 856 700.





# COMMODORE 64

# SPEED KING™

BY MERVYN ESTCOURT

**SPEEDKING** - the motorcycle racing game with thrilling high speed wheel-to-wheel action against 19 other determined riders. Work your way up to world class with the 3 skill levels on 10 world famous circuits. Put on your helmet and enjoy the exhilarating challenge of high performance bike racing at 250 mph!

- 10 world famous circuits
- 3 skill levels
- High performance bike with 6 gears
- 19 other riders
- Display on best position, fastest time, lap record
- Joystick compatible



Available by mail purchase of Home Super Store

Order system for Digital Integration Ltd. Form

Order Reference ID: \_\_\_\_\_

Name: \_\_\_\_\_

Address: \_\_\_\_\_

Or add my Account Amount to: \_\_\_\_\_

Please specify Digital Integration, Washburn Trade Centre, Washburn Road, Cambridge, Surrey GU1 1AA. VAT and 6% Insurance on (where applicable) on request.

Trade and Retail supplies welcome. Lifetime Guarantee.

**DIGITAL INTEGRATION**

Washburn Trade Centre,  
Washburn Road,  
Cambridge, Surrey  
GU1 1AA

<b>COMMODORE 64</b>	Speed King 100	£19.95
	Speed King Complete	£24.95
	Figure Rider 100	£19.95
	Figure Rider Complete	£24.95
<b>SPECTRUM</b>	Speed King 100	£19.95
	Speed King Complete	£24.95
	Figure Rider 100	£19.95
	Figure Rider Complete	£24.95

# Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play Space Invaders.

Financial planning is a rather grand name for something you've been doing all your life — making ends meet! Perhaps if Mr Moneybags had used **BASICALC 3**, he would have been able to balance the books a little better.

For home, club or small business use **BASICALC 3** should pay for itself in no time at all; for larger companies we recommend **BASICALC 3**, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a powerful musical instrument you need to be a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER**.

To use **MUSIC MASTER** requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming either! **MUSIC MASTER** is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

**INTERSECTOR PILOT** is a space fight simulator. Nowadays simulators are widely used to train pilots and astronauts because — to be frank — it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 22nd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilisations is being fought. A shortage of trained pilots has prompted the Federation to develop a computer simulation that allows new recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 your son can learn to pilot the Intersector MK 3 craft. But be warned — this is no game!

Other SUPERSOFT products include the **MIKRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional needs of our competition (see IFI). The **VECTOR** cartridge adds dozens of commands to Basic including toolbar aids and disk commands, or on disk there's **MASTER 64**, a really comprehensive package for the home prep teacher.

Of course, we do also publish games programs, and with titles like **STAR, GUNN** and **KAMIKAZE** in our range we are one of the market leaders. But we most enjoy coming up with the sort of programs that are going to be in use for months and years, not hours and days — the sort of programs that make you glad that you bought a computer — and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket! But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque (pre-paid orders are post free), by calling at our offices, or over the telephone using your ACCESS card.

## SUPERSOFT

SUPERSOFT, Winchester House, Cannon Road,  
Woolston, Harrow, Middlesex HA1 1TA  
Telephone: 01-891 1166