

*Your*

An Argus Specialist Publication

OCTOBER 1984

80p

**NEW**

# COMMODORE

**YOUR BEST INDEPENDENT COMMODORE MAGAZINE**

**MACHINE CODE C8  
TOO MUCH?  
FIND RELIEF INSIDE...**

**LEARN ABOUT  
BASIC WITH OUR  
GREAT IN-DEPTH  
SERIES**

**GAMES AND  
UTILITIES  
TO TYPE  
IN**

**ADDICTED TO  
ADVENTURE?  
READ TALES FROM  
THE CRYPT**

**EA**

**9F**

**D2**

**16**

**DOWN TO  
BUSINESS:  
CBM8296  
HARDWARE  
REVIEW**



**PAGES PACKED WITH  
SOFTWARE REVIEWS**

# DON'T JUST SIT THERE - PLAY SOMETHING!

## FORBIDDEN FOREST

Forbidden Forest is more of a quest than just a game! The action takes place in a four dimensional scrolling forest landscape which many have entered, but none has returned. Yes, I told you FOUR dimensional! The quest is to seek out and slay Lady Ina right in the action website! The quest is to seek out and slay the Demogorgon, mystical ruler of the Forbidden Forest. Before you can even set eyes on her you will have to contend with his army of fearsome creatures, including mutant spiders, showers of giant frogs, snakes, dragons, skeleton warriors and more! You have only your trusty bow and arrows to slay them!



55018

## AZTEC CHALLENGE

A challenge on an epic scale! Aztec Challenge takes you on a journey to Mexico and the ancient pyramid of Teocapitlan. The ancient Aztec gods and their devotees have ensured that no ordinary human can learn the secrets of the temple and live to tell the tale. The pyramid is protected by all

manner of treacherous traps and hidden perils - an epic test of your courage and cunning. Aztec Challenge features no less than seven totally different screens - here are just three of them - each of which presents a brand new challenge. We hope your joystick can stand up to it!



55019

## SLINKY


Slinky, the spring, was having fun hopping about when suddenly he came upon a pile of all coloured blocks, so he thought he'd play croquet on them for a while. Much to his amusement he found that they changed colour when he landed on them. Wop! But unknown to him, the blocks belonged to the Wicked Wizard, who sent his friends along to teach our poor hero. Slinky is a real fun package with ninety-nine levels, amazing reward displays, and action replays. Where else could you meet such charming characters as Dinky the dust cloud, Merge the maggot, Ralph the random raindrop, and Lorenzo the chameleon hoppers!



55020

ON CASSETTE £8.95

ON DISK £12.95

FOR THE **commodore** 

**Audiogenic** LTD

P.O. BOX 88 READING, BERKS.

SEND FOR FREE COLOUR CATALOGUE!

# Our COMMENT

**Our Editor has taken time out from the daily drudgery of journalistic life (and generally enjoying himself) to introduce his new magazine.**



**CONGRATULATIONS!** You have had the good sense and judgment to pick up a copy of the first issue of a GREAT new magazine dedicated to the Commodore range of micro-computers. If you have actually purchased this copy and are now sitting in your armchair at home you can rest assured in the knowledge that this and future issues of Your Commodore will satisfy your thirst for information, games, serious software, education, news and all sorts of goodies that are part and parcel of the Commodore scene. On the other hand if you haven't already parted with money to buy this copy — Why not? We can assure you here and now that it will be money well spent — you'll gain an invaluable insight into your Commodore micro and what it can do for you!

## What can we offer?

Assuming that you have spent a couple of minutes flicking through this magazine and you are still not convinced that Your Commodore is the best thing since sliced silicon, spend a little more time in my company and let me try to change your mind. . .

Your Commodore will entertain, inform and educate you on all matters 'a la Commodore'. Each issue will have regular news pages to keep you informed of all the latest products and stories related to your computer; our intrepid software reviewers will be in loose each month on the latest packages around and let you in on their opinions before you actually hand over some cash; Runwater will be summoned from the Crypt to advocate the spirit of Adventure; and at least one major piece of hardware will be reviewed each issue.

Commodore are not exactly renowned for the quality and clarity of their manuals — we have series and articles on programming in both BASIC and machine code to help you write programs yourself and to help you understand the way other programs are written — whether you are a beginner or expert there's sure to be something here for you to learn!

For the average home computer user (although we of course accept that Your Commodore readers

will be well above 'average') games occupy a tremendous amount of the time and energy spent on the computer. Your Commodore will cater for the games player, as evidenced by our feature on games programming on the VIC and the fantastic games for the VIC and Commodore 64 in this issue. However we are trying not to go 'game-mad' and we appreciate that there are an awful lot (correct me if I'm wrong) of users out there in Commodore-land who have exhausted their trigger fingers and crashed all their spacehips!

## Seriously, though. . .

In Your Commodore we have put together a variety of articles for the more serious-minded amongst our readers: there are some really useful routines that are primarily intended for the Commodore 64, although they will run on other Commodore micros with a few alterations. The business users of Commodore micros have not been forgotten, either. Each issue of Your Commodore will have a number of pages devoted to the growing and generally under-served application area. — under the highly original title of "Down to Business". As you can see in this copy, we have a review of the CBM 8296-D and we take a look at how to set about writing your own business-type software.

## Not only. . .

It would be true to say that we could go on ad infinitum about the value, expertise and entertainment that you are going to get from Your Commodore, but. . .

All you really need to know is that if you have a Commodore micro (regardless of which model) and you want to keep informed on the latest happenings in the world of Commodore, then just buy, read and inwardly digest each fantastic issue of Your Commodore.

Now pay your money and enter the fascinating world of Your Commodore!

## Passing thought

Graham Davies, one of the merry contributors to this first action-packed issue thought that the following idea was worth passing on — why not link up your 55-64 to your normal video recorder? You can then see your TV screen instead of that tiny 5" screen. It works, you know!

## But wait. . .

Should you consider that there is something you would like to see in a future issue of Your Commodore, why not drop a line to the Editor? We obviously put a lot of blood, sweat and tears into putting together a balanced magazine, but why don't you let us have your opinions! Write to the Editor at the editorial address in London.





VOLUME 1  
NUMBER 4  
OCTOBER 1984  
Special Editor: Wendy J. Palmer  
Advertisement Manager: Mike  
Hager  
Advertisement Copy Control:  
Tom Crookston  
Managing Editor: Bob Harris  
Chairman: Jim Connell  
Organisation: Barry Topping  
Design: AM Design

Editorial & Advertisement Office  
No 7 Gifford Square,  
London W1R 3AQ  
Telephone: 01-437 0620  
Telex: 817186

Your Commodore is a monthly  
magazine appearing on the first  
Friday of each month.

Distribution by: Agon Press  
Sales & Distribution Ltd, 11-13  
Paul Street, London EC2A 4EJ.  
Printed by: Aldenat Kalamita  
& Sons Ltd, York, Yorkshire,  
York.

Subscription rates upon  
application to: Your  
Commodore Publications  
Department (Postal Ltd, Stone  
House, 179 The Malvern,  
Hemel Hempstead, Herts, HP1  
3SE.

The contents of this publication  
including all articles, designs,  
plans, drawings, and programs  
and all copyright and other  
intellectual property rights  
therein belong to Agon  
Specialist Publications Limited.  
All rights are reserved by the Law  
of Copyright and other  
intellectual property rights, and  
by virtue of international  
copyright conventions are  
specifically reserved to Agon  
Specialist Publications Limited  
and any reproduction, copying,  
the print or other content of the  
magazine in 1984 Agon Specialist  
Publications Limited.

## MASTERING MACHINE CODE 6

The mysteries of machine code are explained in this tremendous series.

## BATTLE ATAK 10

The drums are sounding and you are at war! Can you defeat the enemy before he gets you?

## MACHINE CODE TO BASIC 13

A great little utility that does exactly what you would think — converts machine code to BASIC!

## BASIC FACTS PART 1 14

BASIC is supposed to be such a simple language but few of us know and understand everything. We start at the beginning...

## FAST TAPE SEARCH 28

A great way of speeding up the Commodore tape system.

## SOFTWARE SPOTLIGHT 30

Our fearless reviewers have been let loose on some of the latest software around for the Commodore 64 and VIC 20. So read on before you spend out some money...

## YOUR OWN ARCADE GAME 40

So you think you can do better than the games you buy? We give you some general guidelines on writing your own!

## CONCORDE II 43

Fasten your seatbelts and prepare to land the fastest aircraft in the world — Concorde II.



## ARTISTIC STUFF 17

Fancy your chances as an artist? Our reviewer takes a look at a package that's almost as useful as a new paint brush.

## CODE BREAKER 18

Do you consider yourself endowed with a logical mind? See if you can beat the computer at a great game of deduction and guesswork!

## RENUMBERING 22

If your programs end up looking like a piece of numerical spaghetti, this useful routine could be just what you need.

## SPRITE DESIGNER 24

Be spritely and do fantastic things with your Commodore 64.

## TALES FROM THE CRYPT 46

Don't worry — it's not as frightening as it may sound! Bannister has been conjured up and has expounded on the spirit of Adventure.





**INTRODUCING MODEMS 48**

Modems have become rather fashionable in computing circles just lately...we let you into the secrets of their appeal.

**MAKING MUSIC 50**

You are chord-tally (!!) invited to make learning music that much easier with the use of your Commodore 64.

**INPUT/OUTPUT 52**

Some commonly asked technical enquiries answered in easy-to-understand language.

**TRACK KING 54**

Drive yourself crazy with this racy game for the VIC 20.

**LADDER MAZE 56**

Can you climb to great heights and get lost in the intricacies of this VIC 20 game?

**REFERENCE LIBRARY 58**

We've taken some of the Commodore books off our shelves and let you know whether they're worth reading.



**VIC GAMES PROGRAMMING 61**

We know there are thousands of VIC gamers out there and here we help you to write your own games.

**COMMODORE 64 EXPOSED 64**

We've taken more than a leaf out of the great book from Melbourne House and give you a taste of things to come from within the covers.

**RELATIVE FILES 70**

Relative files can be somewhat of a problem for some of you — but be confused no longer!

**DATA STATEMENTS 73**

All the news and information for Commodore owners.

**DOWN TO BUSINESS**

**DOING IT YOURSELF 80**

You shouldn't be put off writing your own business software — it's certainly not impossible and we try to show you how.

**IS BIGGER BETTER? 82**

The IBM 870-D is an attractive looking machine. Does its performance match its appearance?

**SUPERBASE OVERVIEW 86**

Freeman's Superbase has received a lot of exposure — what does our 'expert' think of it?

**CONTENTS**

Machine code is the lowest-level computer language but it is also the fastest. A P and D I Stephenson dispel some of its mysteries.

# MASTERING MACHINE CODE

NEWCOMERS TO THE ART must be forgiven for thinking that BASIC is somehow the natural language of the home or business micro-computer. The written information supplied with machines, such as the Commodore 64 User Manual, tend to give this impression. BASIC is certainly the most popular and the easiest to learn but it is a mistake to think of it as the natural language of the Commodore 64.

In fact, BASIC is not natural to any computer. This honesty is claimed by a language, if indeed it can be called a true language, known (loosely) as machine code. The microprocessor chip, which functions as the central control of a computer, can only respond to orders couched in machine code. Even when we write a program in BASIC, a complex program, called an interpreter, is working feverishly in the background, converting each line of the program into machine code ready for the attention of the microprocessor.

An obvious question now arises — if machine code is the natural language of a computer, why is BASIC so popular? The question is best answered by comparing their relative advantages and disadvantages.

- BASIC is easier to learn to use, than machine code.
- Programs written in BASIC are slow to execute. Machine code programs execute somewhere between a hundred and a thousand times faster than the same program written in BASIC.
- A machine code program occupies less memory space

100	CSC3		LNOPTR	=	#95
170	CSC3		PNLIN	=	#96
180	CSC3		PRSTG	=	#97
200	CSC3		!		
210	CSC3	EQ00	OLD	OPK	#00
220	CSC3	F003		BEG	\$000
230	CSC7	4000AF		JMP	#9F
240	CSCA	F001	DOOLD	LBR	#00
250	CSCC	F0		TRV	
260	CSCD	9128		STR	1020
350	CSE2	850F		STR	\$2F
370	CSE4	8501		STR	\$01
388	CSE8	F000		LDR	#00
398	CSE8	6569		WDC	\$69
400	CSE8	652E		STR	\$2E
410	CSEC	8539		STR	\$39
418	CSEC	F0		STR	\$32
428	CSEE	8532		RTS	
438	CSE0	68			
470	CSEF	2030D		JSR	LMB
480	CSE2	85FF		LDR	#FF
490	CSD4	8514		STR	\$14
500	CSD6	8515		STR	\$15
510	CSD0	2013AE		STR	\$1E
520	CSD8	F002		JSR	F002
530	CSD0	18		LDN	#00
540	CSD0	658F		CLC	
550	CSE8	8529		ADC	#2F
				STR	\$20

than the equivalent BASIC version.

- Learning machine code forces you to understand the inner workings of a computer. However clever you become with BASIC, the computer itself will still remain a mysterious grey box.
- Skill in machine code tends to inspire awe at the local computer club.

(Whether this is considered trivial or not depends on your temperament.)

It is clear that machine code scores on all counts except simplicity.

### How Hard Is Machine Code?

We have said that machine code is not simple, but this does not necessarily mean you will find it particularly

hard. It requires just that little more dedication and patience than BASIC requires, a different attitude of mind and, above all, a greater attention to detail. The language is more abstract and code-like in form. The example below illustrates the code-like form. It adds 3 and 5 together and stores the result in a machine address.

Assembly code	Pure machine code
10A #08	85 00
AD0 #3	68 03
ADC #5	68 05
STA \$4008	8D 08 40

The lines on the left are written in assembly code which is a superior type of machine code. (The difference is explained later.) The version on the right is the same program but written in pure machine code. In BASIC, the same effect can be achieved with, say,

2+5

You will have to understand binary arithmetic, hexadecimal code, contain logical operations and a few other bits of background knowledge. But none of these subjects need frighten you. They are not hard to learn and the rewards are well worth the effort. However, when you do become proficient, don't adopt a snobbish attitude towards BASIC. BASIC is a nice language and should not be thought of as a competitor of machine code. In fact, the aim of this series is directed towards the amalgamation of BASIC with machine code.



Although it was stated above that machine code is much faster than BASIC, it would be quite wrong to attach too much significance to it. If a particular program runs fast enough written in BASIC, there is little point in re-writing it in machine code. On the other hand, there may be certain parts of a BASIC program which run far too slow for comfort.

For example, sorting a long list of data into order can take several minutes or, in some cases, even hours in BASIC. A machine code *subroutine* could achieve the same result in a few seconds. Screen animation effects are far better in machine code than in BASIC. The movements are less jerky and the display can be given a more professional appearance in machine code. Robotics is becoming popular with computer hobbyists and machine code control of the robot's private parts is virtually a necessity in BASIC is seldom fast enough.

### Entering Machine Code

There are two methods — one is free but tiresome to use and terribly error-prone, the other is pleasant to use and far less error-prone but involves additional expenditure. The cheap method is by using the keyboard PCISII — we simply POKE every machine code instruction into memory, a byte at a time. Trying to follow how a "POKE" machine code program works, or to debug it if it doesn't work, is nothing short of sheer hell. To compare the difference, refer back to the short example which added two numbers together. The left hand version, although still code-like in form, will be shown later to be far more comprehensible than the numeric mess on the right.

By using a piece of software known as an *assembler*, the task is much easier, the machine code listing is easy to read or correct and you are relieved of much boring work. Although this series will cater for readers who are

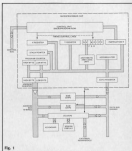


Fig. 1

content to POKE everything, all programming examples will be given in assembly language. It cannot be too strongly emphasized that serious machine code work demands the purchase of a good assembler. We have used the MICRO assembler, which is obtainable as a plug-in cartridge from: SUPERSOFT, Winchester House, Canning Road, Waddesdon, Harrow, Middlesex HA1 7SU.

### Hardware Knowledge Required

If 'handy' knowledge means familiarity with electronics, then you may rest assured you need hardly try. We shall, from time to time, be mentioning 'volts' but only in yes/no form. That is to say, either a certain voltage is present or it is not present. Computers happen to be electronic in nature, but, as far as programming in machine code is concerned, it wouldn't make all that much difference if they ran on North Sea gas. However, you will have to understand a little about the overall system of a computer, particularly the role of the microprocessor,

memory chips and a few other bits and pieces — not how they work but what they do. Figure 1 gives a rough idea of how some of the components are connected in most micro-computers. The microprocessor looks a bit lonesome at this stage but all should become clear as the series progresses.

Many readers will already be aware of the distinction between RAM chips and ROM chips. ROM stands for Read Only Memory and refers to memory chips which hold **permanent** information put there by the manufacturer. Information in ROM is known as **firmware**.

ROMs are said to be non-volatile, meaning that the stored information is still there after the machine is switched off. There are three ROMs in the Commodore 64:

- The 8K BASIC language interpreter occupying the hex addresses 4000 to 8FFF (4096 to 49151 decimal).
- The character generator ROM, responsible for arranging the correct pattern of screen dots for every character. There are two separate sets of 256 characters: one set covers the upper case and fixed keyboard graphics, the other set covers the normal upper and lower case typewriter-style characters.

RAM stands for Random Access Memory, an unfortunate title because ROM chips are also 'random access'. The essential feature of RAM chips is the ability to read existing information and write new information under computer control. You may notice from Fig. 1 that the control line, I/O, is connected to the RAM chips but not the ROMs. This is because ROMs are permanently in the 'read' state, whereas RAMs must be switched to a different state when information is to be read from them when they are to be written to. RAMs are more like the old school slate used in the Charles Dickens era — old material can be rubbed off and new material written.

RAMs are **volatile**. Any stored information is lost if the power is interrupted, even for a fraction of a second. RAM is for your use, although some of it is hogged by the operating system for screen display purposes. Input/output control and, for want of a better term, working space.

There is indeed 64K of RAM present in the Commodore 64 but, unless some fiddling around is done (explained later in the series) and some ROM facilities sacrificed, only 32K is normally available to the user. It is difficult to proceed any further with computer components at machine code until the binary counting system is explained together with the **hexadecimal** notation. Unfortunately, the mere mention of binary is enough to invoke a pawn from those

who already know it and a gasp of fear from those who don't.

## The Binary Number System

There are two kinds of computing machines, the analogue kind (which is of no interest whatsoever to us) and the digital kind. So, when we speak of 'computers', it is taken for granted we are referring to the digital kind. A computer is essentially a system of switches — silent electronic versions of the ordinary household on/off switch. There are hardly any smoothly varying voltages. Voltages are either in the HIGH state (about 5V) or the LOW state (nearly zero volts). Due to this essential two-state nature of the computer, it is natural to base all arithmetic and other forms of processing on a counting system which uses only two characters. Binary is such a system because it only uses the characters 1 and 0, allowing us to represent a HIGH voltage by 1 and a low voltage by 0.

Any number, however large, can be formed by a string of 1's and 0's called bits. Binary, like the familiar decimal system, uses the normal place-weighting system but, instead of each place being worth 10 times the value of the number on its immediate right, it is only worth twice as much. For example 111 in binary is read as one 1, one 2 and one 4 which, in decimal, is 7. To help you get the feel of binary, study the following binary numbers with their decimal equivalents:

1111 = 15  
1010 = 10  
1110 = 14  
0011 = 3  
1111 1111 = 255  
1000 0000 = 128  
1000 0011 = 131

Note that when there are more than four bits, it is conventional to separate them into groups of four because it is easier for humans to read them that way.



Here are a few commonly used terms:

- A **bit** is either a 0 or 1.
- A **nibble** is a group of four bits.
- A **byte** is a group of eight bits.
- The **LSB** (least significant bit) is the one on the extreme right.
- The **MSB** (most significant bit) is the one on the extreme left.
- The **2<sup>n</sup> = 1024 (2<sup>10</sup>, two to the tenth power) nearest power of 2 to 1000.**

The following is of great importance:

The number of ways of arranging N bits is 2<sup>n</sup> = N

For example, three bits can only be arranged in 2<sup>3</sup> = 8 ways as can be seen from the following table:

binary	decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Since a byte is the most commonly encountered number of bits, it follows that there are 2<sup>8</sup> = 256 ways of arranging the bits — there are 256 combinations. Referring back to Fig. 1, you will notice 16 address wires coming out of the microprocessor. We can deduce from this that only 2<sup>16</sup> = 65,536 different address combinations are possible. Now 1K = 1024, so there are only 64K valid addresses. This should explain why we cannot have 64K of addressable RAM as well as ROM.

## Hexadecimal Notation

The sorts of numbers we can be dealing with in computing get very unwieldy when expressed in binary notation; for example, a large number on the address bus could be:

```
1001 1111 0001 1010
```

We poor humans need a more compact way of presenting large binary numbers. Well, the four-bit groupings mentioned earlier correspond to a base-16 system of counting (since 2<sup>4</sup> = 16). This counting system is called hexadecimal; if you gain experience with the hexadecimal equivalent, you would be able to see almost immediately that the above binary number is DF1A. Just as base two required only two types of digit, 1 and 0, so hexadecimal uses the characters 0 to 9 and the six letters, A to F, to represent a group of four bits. The following table shows the value of each character:

hex	decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Note that the bit patterns for decimal 0 to 9 are the normal binary equivalents. The last six patterns, which have a decimal value of 10 to 15, are represented by the letters A to F respectively. Once these six extra patterns are mastered (or memorised), it is easy to express a bit pattern, however long and cumbersome-looking, in a few hex characters. Remember that each hex character represents one nibble and the following examples should be easy to understand:





```

1111 0101 1100 0000 1010
 F  S  E  O  A
 0011  0101
   1     1
  
```

## Changing Hex To Decimal

Each hex digit is worth sixteen times as much as the digit on its right. In other words, the system is based on powers of 16 instead of the normal 10 as in decimal. The place weightings are shown below, in exponential as well as decimal form:

```

160 161 162 163
(0006) (0064) (1024) (16384)
  
```

Examples based on above weighting:

Hex 111 = 0006+0064+1024 = 1094 decimal.

Hex 3012 = 0006+(3x16)+1 = 4186 decimal.

Hex 21AF = (2x0006)+0064+(18x16)+15=0623 decimal.

Hex FF7F = (15x0006)+(15x16)+15=1035 decimal.

After glancing at the above "horrible-looking jumble," it may be a relief to learn that only occasionally will it be necessary for you to convert hex to decimal. In fact, you will gradually realize that decimal is alien to machine code. It is actually easier and much quicker, to think in hex, particularly if you do most in an assembler.

For example, in machine code work, machine addresses are frequently involved. In hex, these addresses within the range 0000 to FFFF hex, in decimal, the same range is 0 to 65,535. Decimal is clumsy and quite unsuited to machine code; hex is concise and tailor-made for it. Memory locations are each one byte "wide" so the binary contents can be expressed very neatly by two hex characters. The last address in memory is FFFF. There is something final about FFFF but there is nothing final about 65,535!

It does take a little time to get hex into your blood, in the initial stages it comes as a shock to discover that, say 10 in hex is not ten — it is sixteen. Similarly, 13 in hex is not thirty-two — it is fifty. In speech, don't speak of 10

hex as "ten". Call it "one zero" to avoid confusion.

## Adding Hex Numbers

Some examples are given below without explanation as an exercise:

```

  13    2F
  15    8D
  2E    31
  ---    ---
  FF    CD
  91    1A
  183    17
  
```

## Binary Arithmetic

The pencil and paper procedure for adding binary is the same as for decimal, providing you realize that a carry to the next digit is worth 2 (instead of 10, for example):

```

  3 0011
add 3 0101
----
total 1000
  
```

Users of BASIC are usually unaware of the computer's internal arithmetic. Numbers of enormous size are casually entered and the correct answers are taken for granted. When we first take a look into machine code arithmetic, the situation looks decidedly bleak. The 6510 microprocessor is only capable of handling eight bits — in fact, it is called an "eight-bit" chip if all eight bits are binary 1's, the largest absolute number it can handle is FF hex or 255 decimal. This may seem a depressing start.

If both positive and negative numbers are to be handled, the situation is even worse because one of the bits is used to indicate the sign of the number. This reduces the maximum positive number to only 127 decimal and the maximum negative number to -128. However, things are not so bad as they appear because, as we shall see later in the series, it is possible to increase the range of numbers by employing some crafty programming tricks.

## Two's Complement Notation

Two's complement notation is employed in nearly all computers for the following reasons:

- It enables positive and negative numbers to be handled in an efficient manner.
- It simplifies the hardware concerned with addition and subtraction. Only a circuit capable of adding binary numbers is required; subtraction is achieved by adding the two's complement of the number.

The most significant bit in the byte, called the **sign bit**, has the following significance:

The sign bit=0, if the number is positive, and 1 if it is negative.

Examples of positive numbers are given under the following table.

Some examples follow of Two's complement.

```

Original          0111 1111
Two's complement 1000 0000
  
```

```

Original          1100 0000
Two's complement 0100 0000
  
```

```

Original          0000 0000
Two's complement 1111 1111
  
```

```

0000 1001 = 99 hex = 48 decimal
0111 1111 = 7F hex = 127 decimal
  
```

Negative numbers are not so straightforward because it is not just a question of changing the sign bit. Before we give the rule, try and work it out by studying the following examples:

```

0000 0001 = +1
0000 1010 = +10
0100 0000 = +64
1111 1111 = -1
1111 0100 = -10
1100 0000 = -64
  
```

The term 'flip' in the following rule means change '1's' and '0's' and vice versa. To find the equivalent negative, flip all the bits and

then add 1 more, ignoring any final carry. For example:

```

0000 0011 = +3
Flip all the bits .... 1111 1100
Add 1 .....          1
                          1111 1101 =-3
  
```

When we flip all the bits, the result is called the **one's complement** but, adding the final 1, converts it to the **two's complement**. The process works both ways. The rule still applies for changing a negative number back into positive. Try using the rule to change -3 back to +3.

To avoid adding the 1, which may often involve the propagation of a carry, there is an easier way. Assume we start with the positive number:

To obtain the two's complement, start from the right and copy down up to, and including, the first 1. Thereafter, flip the remaining bits.

The first rule we gave is the academic version, the second is the useful one — it is easy to use. Sometimes, we need to find the two's complement of a hex number. The easiest way is to write out the binary bits, use the second rule and put the result back into hex. For example:

```

If hex = 0811 1111,
Two's complement = 1000 0000=C1 hex.
  
```

This concludes Part 1 of the series and most of the boring background knowledge. We could have stopped a lot of it but this would only have caused trouble later on in the series. A working knowledge of binary and hex is essential for machine code programming.

Get to be a real

Commodore and keep  
the high seas safe with  
this game from Jamie

Clyde.

# BATTLE ATAK

BATTLE ATAK IS A computerized version of the popular game 'Battleships'. It runs on the standard Commodore 64 in no more than 10K of memory and uses user-defined graphics.

The scenario is as follows: you are in command of a submarine equipped with torpedoes and a battle computer. Your task is to defend your fleet of surface ships with your torpedoes and destroy the enemy fleet before their torpedoes wipe you out. You are shown your ships on one grid and where your shots hit on the other. Unfortunately you are unable to determine the locations of the enemy vessels, so the second grid starts off blank at the beginning of the game. Both fleets are identical and consist of the following:

- |                    |             |
|--------------------|-------------|
| 1 Aircraft Carrier | (4 squares) |
| 2 Battleships      | (2 squares) |
| 3 Submarines       | (3 squares) |
| 4 Cruisers         | (2 squares) |

The figure stated after the ship is the number of squares which it occupies on the grid.

The computer marks up on the grid any developments as the game progresses and the game will finish when either side's fleet has been completely destroyed.

## When you run the program

You will first be shown a title page while the computer loads in the user-defined graphics into its memory. To proceed, press the Spacebar when told and you will be asked for a skill level and then speed level. Level 1 is the easiest game, and beyond level 4 the computer



will automatically have another shot if it hits a ship. As far as speed levels are concerned, '1' is the fastest and '5' the slowest — this governs the length of time between each shot.

Once you have entered the desired levels, the grids will be displayed. But be-

fore the game can commence, you have to place the ships on your grid which is left of the coordinates in the computer, three characters must be entered. The first is the letter for the distance across, and the others for the figure for the distance

down — for example, the coordinate for the top left-hand corner would be 'A11'.

So, to enter a ship on the grid you type the coordinate where you want the base of the ship to appear, and you then must say if you want the ship to be facing



### Organization of the Lines in the program, from 0 to 30000

Lines	Function
0 - 300	Title page & Set up graphics
370 - 450	Set up screen & variables
460 - 630	Place Player's ships on board
640 - 730	Set up computer's ships
740 - 1000	Finalize screen
1000 - 1050	Main Play routine
1100 - 1800	Player's shot
2000 - 3000	Computer's shot
5000 - 6000	End of game routine
7000 - 8000	Set Levels
10000 - 11000	Sound routines
12000 - 20000	Extra Subroutines
20000 - 30000	Data for User-Defined Characters & Graphics



up or along. If it cannot fit into your position, an error will appear and you will have to try another point. When all the ships are entered, the game can commence. To play, you enter the coordinates of the point where you want the shot when the top left-hand

box says "Your Turn". If it is a hit, the part of the ship will be displayed on the right-hand grid. If a computer's shot hits one of your ships, it will turn red.

If a whole fleet is destroyed, the screen clears, and you can have another game if wanted.

### Program Listing

```

100 REM ***** TITLE PAGE *****
110 PRINT "***** TITLE PAGE *****"
120 PRINT "***** TITLE PAGE *****"
130 PRINT "***** TITLE PAGE *****"
140 PRINT "***** TITLE PAGE *****"
150 PRINT "***** TITLE PAGE *****"
160 PRINT "***** TITLE PAGE *****"
170 PRINT "***** TITLE PAGE *****"
180 PRINT "***** TITLE PAGE *****"
190 PRINT "***** TITLE PAGE *****"
200 PRINT "***** TITLE PAGE *****"
210 PRINT "***** TITLE PAGE *****"
220 PRINT "***** TITLE PAGE *****"
230 PRINT "***** TITLE PAGE *****"
240 PRINT "***** TITLE PAGE *****"
250 PRINT "***** TITLE PAGE *****"
260 PRINT "***** TITLE PAGE *****"
270 PRINT "***** TITLE PAGE *****"
280 PRINT "***** TITLE PAGE *****"
290 PRINT "***** TITLE PAGE *****"
300 PRINT "***** TITLE PAGE *****"
310 PRINT "***** TITLE PAGE *****"
320 PRINT "***** TITLE PAGE *****"
330 PRINT "***** TITLE PAGE *****"
340 PRINT "***** TITLE PAGE *****"
350 PRINT "***** TITLE PAGE *****"
360 PRINT "***** TITLE PAGE *****"
370 PRINT "***** TITLE PAGE *****"
380 PRINT "***** TITLE PAGE *****"
390 PRINT "***** TITLE PAGE *****"
400 PRINT "***** TITLE PAGE *****"
410 PRINT "***** TITLE PAGE *****"
420 PRINT "***** TITLE PAGE *****"
430 PRINT "***** TITLE PAGE *****"
440 PRINT "***** TITLE PAGE *****"
450 PRINT "***** TITLE PAGE *****"
460 PRINT "***** TITLE PAGE *****"
470 PRINT "***** TITLE PAGE *****"
480 PRINT "***** TITLE PAGE *****"
490 PRINT "***** TITLE PAGE *****"
500 PRINT "***** TITLE PAGE *****"
510 PRINT "***** TITLE PAGE *****"
520 PRINT "***** TITLE PAGE *****"
530 PRINT "***** TITLE PAGE *****"
540 PRINT "***** TITLE PAGE *****"
550 PRINT "***** TITLE PAGE *****"
560 PRINT "***** TITLE PAGE *****"
570 PRINT "***** TITLE PAGE *****"
580 PRINT "***** TITLE PAGE *****"
590 PRINT "***** TITLE PAGE *****"
600 PRINT "***** TITLE PAGE *****"
610 PRINT "***** TITLE PAGE *****"
620 PRINT "***** TITLE PAGE *****"
630 PRINT "***** TITLE PAGE *****"
640 PRINT "***** TITLE PAGE *****"
650 PRINT "***** TITLE PAGE *****"
660 PRINT "***** TITLE PAGE *****"
670 PRINT "***** TITLE PAGE *****"
680 PRINT "***** TITLE PAGE *****"
690 PRINT "***** TITLE PAGE *****"
700 PRINT "***** TITLE PAGE *****"
710 PRINT "***** TITLE PAGE *****"
720 PRINT "***** TITLE PAGE *****"
730 PRINT "***** TITLE PAGE *****"
740 PRINT "***** TITLE PAGE *****"
750 PRINT "***** TITLE PAGE *****"
760 PRINT "***** TITLE PAGE *****"
770 PRINT "***** TITLE PAGE *****"
780 PRINT "***** TITLE PAGE *****"
790 PRINT "***** TITLE PAGE *****"
800 PRINT "***** TITLE PAGE *****"
810 PRINT "***** TITLE PAGE *****"
820 PRINT "***** TITLE PAGE *****"
830 PRINT "***** TITLE PAGE *****"
840 PRINT "***** TITLE PAGE *****"
850 PRINT "***** TITLE PAGE *****"
860 PRINT "***** TITLE PAGE *****"
870 PRINT "***** TITLE PAGE *****"
880 PRINT "***** TITLE PAGE *****"
890 PRINT "***** TITLE PAGE *****"
900 PRINT "***** TITLE PAGE *****"
910 PRINT "***** TITLE PAGE *****"
920 PRINT "***** TITLE PAGE *****"
930 PRINT "***** TITLE PAGE *****"
940 PRINT "***** TITLE PAGE *****"
950 PRINT "***** TITLE PAGE *****"
960 PRINT "***** TITLE PAGE *****"
970 PRINT "***** TITLE PAGE *****"
980 PRINT "***** TITLE PAGE *****"
990 PRINT "***** TITLE PAGE *****"

```



60400 0000  
 60401 0000  
 60402 0000  
 60403 0000  
 60404 0000  
 60405 0000  
 60406 0000  
 60407 0000  
 60408 0000  
 60409 0000  
 60410 0000  
 60411 0000  
 60412 0000  
 60413 0000  
 60414 0000  
 60415 0000  
 60416 0000  
 60417 0000  
 60418 0000  
 60419 0000  
 60420 0000  
 60421 0000  
 60422 0000  
 60423 0000  
 60424 0000  
 60425 0000  
 60426 0000  
 60427 0000  
 60428 0000  
 60429 0000  
 60430 0000  
 60431 0000  
 60432 0000  
 60433 0000  
 60434 0000  
 60435 0000  
 60436 0000  
 60437 0000  
 60438 0000  
 60439 0000  
 60440 0000  
 60441 0000  
 60442 0000  
 60443 0000  
 60444 0000  
 60445 0000  
 60446 0000  
 60447 0000  
 60448 0000  
 60449 0000  
 60450 0000  
 60451 0000  
 60452 0000  
 60453 0000  
 60454 0000  
 60455 0000  
 60456 0000  
 60457 0000  
 60458 0000  
 60459 0000  
 60460 0000  
 60461 0000  
 60462 0000  
 60463 0000  
 60464 0000  
 60465 0000  
 60466 0000  
 60467 0000  
 60468 0000  
 60469 0000  
 60470 0000  
 60471 0000  
 60472 0000  
 60473 0000  
 60474 0000  
 60475 0000  
 60476 0000  
 60477 0000  
 60478 0000  
 60479 0000  
 60480 0000  
 60481 0000  
 60482 0000  
 60483 0000  
 60484 0000  
 60485 0000  
 60486 0000  
 60487 0000  
 60488 0000  
 60489 0000  
 60490 0000  
 60491 0000  
 60492 0000  
 60493 0000  
 60494 0000  
 60495 0000  
 60496 0000  
 60497 0000  
 60498 0000  
 60499 0000  
 60500 0000

60501 0000  
 60502 0000  
 60503 0000  
 60504 0000  
 60505 0000  
 60506 0000  
 60507 0000  
 60508 0000  
 60509 0000  
 60510 0000  
 60511 0000  
 60512 0000  
 60513 0000  
 60514 0000  
 60515 0000  
 60516 0000  
 60517 0000  
 60518 0000  
 60519 0000  
 60520 0000  
 60521 0000  
 60522 0000  
 60523 0000  
 60524 0000  
 60525 0000  
 60526 0000  
 60527 0000  
 60528 0000  
 60529 0000  
 60530 0000  
 60531 0000  
 60532 0000  
 60533 0000  
 60534 0000  
 60535 0000  
 60536 0000  
 60537 0000  
 60538 0000  
 60539 0000  
 60540 0000  
 60541 0000  
 60542 0000  
 60543 0000  
 60544 0000  
 60545 0000  
 60546 0000  
 60547 0000  
 60548 0000  
 60549 0000  
 60550 0000  
 60551 0000  
 60552 0000  
 60553 0000  
 60554 0000  
 60555 0000  
 60556 0000  
 60557 0000  
 60558 0000  
 60559 0000  
 60560 0000  
 60561 0000  
 60562 0000  
 60563 0000  
 60564 0000  
 60565 0000  
 60566 0000  
 60567 0000  
 60568 0000  
 60569 0000  
 60570 0000  
 60571 0000  
 60572 0000  
 60573 0000  
 60574 0000  
 60575 0000  
 60576 0000  
 60577 0000  
 60578 0000  
 60579 0000  
 60580 0000  
 60581 0000  
 60582 0000  
 60583 0000  
 60584 0000  
 60585 0000  
 60586 0000  
 60587 0000  
 60588 0000  
 60589 0000  
 60590 0000  
 60591 0000  
 60592 0000  
 60593 0000  
 60594 0000  
 60595 0000  
 60596 0000  
 60597 0000  
 60598 0000  
 60599 0000  
 60600 0000

A wondrous utility  
from Mike Roberts  
converts machine  
code into BASIC  
for those of you with  
problems.

# MACHINE CODE TO BASIC

WHILE I WAS WRITING the renumber program elsewhere in this issue it became necessary to create a BASIC listing from an area of memory as a list of DATA statements. Also I was doing some work with character generators at the time so I also needed a program that could read the character generator ROM as well (very difficult).

The result is a program that asks you for the start line number of the resulting program and the increment between the lines. Program lines will then begin to flash on the screen. After a while the program will end and the original program will be a lot longer. If you specify a start line number that the program already uses it will overwrite the program and will irreversibly crash.

The start location for the code to be converted is stored in line 20 (currently 40153) the end location is

stored in line 55 (currently 49340-1).

The end result is the original program, which can

now be deleted, and a long list of DATA statements containing the code. A POKER can be put as the first

line to POKER in the DATA (like line 10 of renumber) and hey presto there you have it.



## The conversion program

```

1 DEF FNH(X)=INT(X/256)
2 DEF FNL(X)=INT(X-(INT(X/256)*256))
3 DEF FND(X)=PEEK(X)+(PEEK(X)*256)
10 INPUT "*****START NUMBER, INCREMENT";Z,X
20 POKES6304,PEEK(56304)+FNH(Z)*X+PEEK(1)+FNH(Z)*X+49152:REM START LOCATION
30 ODSUR100:PRINT"Z,X,Z," DATA "+##:PRINT"00709"
32 POKES632,FNL(Z)+POKE633,FNH(Z)+POKE634,FNL(X)+POKE635,FNH(X)
33 POKES36,FNL(C)+POKE637,FNL(C)
35 POKE199,3+POKE631,19+POKE632,19+POKE633,19
40 POKEL,PEEK(1)+OR4+POKE56334,PEEK(56334)+OR1:STOP
50 POKES6334,PEEK(56334)+FNH(Z)*X+PEEK(1)+FNH(Z)*X
51 DEF FNH(X)=INT(X/256)
52 DEF FNL(X)=INT(X-(INT(X/256)*256))
53 DEF FND(X)=PEEK(X)+(PEEK(X)*256)
55 Z=FNH(6323)+FNH(634)+FNH(635)+Z+X:IFC493240THEOR36:REM END LOCATION
60 POKEL,PEEK(1)+OR4+POKE56334,PEEK(56334)+OR1:END
100 RM=""
110 FORI=40109:DOPEEK(C+I)+RM=RM+STR$(C)+", " :NEXT I:RM=LEFT$(RM,LEN(RM)-1)
120 C=C+19:RETURN
  
```



Like any language,  
BASIC can be used  
well or badly. A P  
and D | Stephenson  
brush up your  
grammar.

# THE BASIC FACTS PT. 1

BEGINNING A NEW SERIES on BASIC for the 64, or indeed any machine, is a problem. If the standard is pitched too high, new converts are disadvantaged; if too low, the growing mass of knowledgeable readers are inclined to get a little touchy and start muttering things about their intelligence being insulted. Another problem is the Commodore User Manual supplied with the CBM 64. It is badly written, it is improved version. But it isn't. However, it is rather thin volume and some users may feel that some of the subjects are given too sparse a treatment for comfort.

It is hoped that this series will help to fill some of the gaps. A certain amount of repetition is inevitable — that is to say, certain descriptions of the BASIC keywords will be little more than a restatement of the Commodore User Manual. However, repetition is not always useless providing it is not literal repetition. Sometimes, the same facts expressed differently can change darkness into light — or at least twilight. However, it would be pointless to begin this series by defining or describing the purpose of every keyword in the BASIC vocabulary. It is expected that the reader has at least glanced through the Commodore User Manual and that it is kept available as a reference text when following this series.

## 14 What is BASIC?

If we define 'language' as a means of communicating, then BASIC is a primitive language. It is a set of rigidly defined keywords which, by



virtue of a special translation program, can communicate our orders to the computer. The list of keywords is the vocabulary of the language and the computer will recognise no others. It is impossible to overstate the fact that computers have no initiative whatsoever. If you use a word that is nearly right, the computer's morose intellect is immediately over-stretched so it gives up. The same thing happens if a colon is used when the thing expects a semicolon.

Everything you write must be precise with exact spelling and exact punctuation. The computer doesn't really understand anything — at least not in the normal meaning of the word. It is you, the writer of the program which injects intelligence. Computers themselves may start exhibiting primitive intelligence when the new Fifth Generation machines are launched but, for the moment, we must resign ourselves to the unpalatable fact that all computers, including the CBM 64, have zero intelligence. In spite of this, computers do seem to obey Benjamin Sed's fifth law which reads, "if a computer has a flaw, a program will delight in finding it".

The translation program is essential because the

computer, or rather the microprocessor chip within the computer, can only respond to a lower level language called 'machine code' (which is covered elsewhere in this book). As far as BASIC is concerned, it hasn't got a clue. The translation program in the CBM 64 is buried within a ROM (Read Only Memory) chip. Each line in your

BASIC program is translated (the technical term is **interpreted**) by the ROM into machine code before it is executed.

This takes place at electronic speed which, in most cases, is so fast that you may be completely unaware of the lengthy translation going on behind the scenes. The translation is said to be 'transparent' to the user. And yet it does take time — particularly when your program contains mathematical angles within loops (loops are bits of program which are executed over and over again until some exit condition is satisfied). Small times (milliseconds) add up to large times (minutes or even hours) if there are enough of them so the translation time can, in some programs, reach an unacceptable limit.

BASIC is not without its critics. In fact it has become fashionable in learned quarters to decry the language. Some of the criticism is justified: BASIC can be slow at times and it is not easy to write programs which obey the rules of good structure (structure is a set of programming guidelines designed to make programs easier to read and debug). There is also an element of snobishness in some of the criticism. BASIC was designed for ease of learning



and there is little doubt that the original aim has been met. It is the easiest of all computer languages to convert microcomputers over a debt to the originators, John Kemeny and Thomas Kurtz of Dartmouth College, USA.

It has been said that the microprocessor brought computers to the high street. This may be true but without BASIC, it is doubtful if the man or woman in the high street would have bought many of them. The static languages COBOL, FORTRAN, ALGOL and Pascal have many virtues but simplicity isn't one of 'em. Of course, if you are not interested in programming and treat the computer as a tool to run professional software or to play games, the computer language is of no consequence — but then this series will not interest you anyway.

## BASIC Dialects

Natives of London and Glasgow sometimes have communication problems. They both speak the same language but use different dialects. And so it is with BASIC. There are many variations on the original Dartmouth version. The essential features of the original language are preserved but there are some differences. Some dialects



may have additional keywords in the vocabulary or may employ different shades of syntax (grammar). This is understandable. BASIC was first launched in 1964 and the computing scene has changed almost beyond recognition since then. Hardware changes, such as high resolution graphics, joystick controls, graphics 'printers', sound synthesizers and speech recognition chips, all need appropriate activating keywords.

The more sophisticated the translation, the bigger the ROM. The software in a big ROM costs more and, what is more important, uses up more of the **addressable space** of the computer. For example, the CBM 64 uses a 65700A microprocessor. In common with most other eight-bit microprocessors, only 18 address wires are available, limiting the number of directly addressable memory locations, including both RAM and ROM, to  $2^{18} = 64K$ . Apparently, CBM considered RAM space more important than ROM sophistication so they decided to cut down the BASIC interpreter ROM to 8K. Many would say this is a pity. It is inevitable that penny pinching in this area has led to a rather stony dialect of BASIC, the best that can be said for it is that it is just adequate.

These gentle (?) criticisms should not be taken to heart. Complex programming projects can be undertaken whatever BASIC dialect is resident in ROM providing we work just a bit harder and be a little more methodical than we write programs.

### Programming Guide Lines

If the CBM 64 is your first machine, and you have



ironed out the last remaining bug in your first program then, by all means, feel elated. But don't let the elation last too long. The fact that a program works should be considered the first, rather than the last, step. In all probability, the listing will be fouled by scar tissue resulting from multiple alterations and insertions. In all probability, there will be redundant lines and evidence of excessive zeal in the use of GOTOs.

It may be argued that, providing a program works, why bother with niceties? Your first program was probably short and niceties won't have mattered very much. The trouble starts when your confidence increases and you have an urge to tackle more ambitious projects. If you carry on in the original undisciplined manner you will soon regret it. The bug-hunt stage becomes more and more tedious. The process of curing one bug often leads to the introduction of two more. The outcome, after hours or even weeks of work, can end in hours of verbal obscenities or, even worse, lead to permanent change in the personality.

The answer to all this is to start out on the right lines. Try and make the listing beautiful as well as functional. It is too early in the series to start laying down the precise rules of programming structure but, in the meantime, you would do well to take notice of the guide lines which follow.

### Use Of Remarks

Use REMs liberally. It is often thought that remarks are for the benefit of other people trying to decipher how your program works. This is only partially true. They are more of an aid to the programmer. Never rely on memory for understanding the purpose of a routine. It may be perfectly clear at the time of writing, but even a few hours afterwards, you may not be quite so certain how it worked or, in some cases, even what it was

supposed to do! REMs use up memory but this doesn't matter in the slightest when you are developing a program. When it is finally finished, the original copy can be kept in its REM'd state and a copy, naked of REMs, kept as a worksheet.

### Preliminary diagrams

Scrbble out a rough diagram, showing the order in which the various parts of the program are to be carried out. The type of diagram is unimportant to start with. There are things called 'flow charts' and 'top down structure charts' which use established symbolism but there is no urgent need to learn, or even stick to them. Providing you understand your own diagram, that is all that matters. Too much establishment guidance applied too early can crucify enthusiasm and destroy initiative.

### Variable names

Decide carefully on **variable names** and write them down as a list. Unfortunately, CBM BASIC only allows two characters for a variable name, the first of which must be an upper case letter and the second a letter or one of the numbers 0 to 9. Some BASICs allow an unlimited number of characters in a variable name so it is easy to choose them meaningful. For example, INPUT-VOLTAGE is meaningful but on the 64 we would have to be content with, say, IV instead.

Because of this restriction, it pays to spend some time on the choice of variable names. Although only two characters are allowed, the law of combinations allows you choose any one of 26 combinations. This can easily be demonstrated: the first character must be a letter, so there are 26 ways of choosing one. The second character can be a letter or a numeric figure so there are 36 ways of choosing it. The number of ways in which

two characters can be chosen is therefore  $26 \times 36 = 936$  combinations.

Sometimes, it will be necessary to use a name without any meaning (no mnemonic value) in order to avoid clashing with another name. However, whatever names you give them, the most important thing is to write them down first and stick to them like glue. If you neglect this discipline, programming time will squandered by having to continually scroll



back the listing to see if you have used the same letters. Giving the same name to what should be two different variables can cause distress to the programmer and pleasure to malicious computers.

### Modular construction

As far as possible, try and arrange your program as a set of self-contained **subroutines**, each capable of separate testing outside the program. This is sometimes difficult but will pay dividends in the long term. If the subroutine requires some values normally obtained from another part of the program (as it probably will) then write a short trigger program which hands over test numbers, entered from the keyboard. When eventually, you persuade it to pass this test, it can safely be inserted in the program with the knowledge that, whatever else is wrong in the final version that particular subroutine is beyond suspicion. In this way, your program grows as a coherent structure of tested building bricks. (The technical term is **modules**.)

## The hated GOTO

Go easy on the GOTOs. The most violent critics of BASIC has always been directed against the overuse of the GOTO keyword. The GOTO is a jump to a line number; for example, GOTO 560 then, a few lines later, GOTO 120. Too much of this is called "spagetti programming". It can be absolute hell to follow if carried to excess. GOTOs can never be avoided in old BASIC but, with a little care, their population can be decreased.

In any case, don't even think of jumping out of a loop with a GOTO before the normal exit. This is the worst crime in the book, even if the program does work. (There are one or two exceptions to this but this is not the proper time to discuss them.)

## Screen messages

However brilliant the program its effect on the user will depend ultimately on the quality of the screen display. There is only one thing worse than a scruffy display and that is one that is over-embellished. Messages to the operator should be as short as possible but not at the expense of clarity. Excesses should be avoided. For example, a message such as "AGE?" is certainly short but it may not be immediately obvious that it is an invitation to input your age at the key board. Noisy messages of this kind often lead to a situation in which screen and user stare at each other blankly, each waiting for the other to do something.

On the other hand, "Would you be kind enough to supply the computer with information as to your age and then press RETURN" is an example of the other extreme and liable to induce projectile vomiting all over the poor keyboard.

The CBM has a liberal allotment of graphic keys and there is a natural temptation in the early stages to start up every message with a coloured border of circles, squares or asterisks. Over enthusiasm

in this respect can reduce the impact of the message by swamping it with the kaleidoscopic beauty (?) of the border.

## Programming philosophy

Always program with the idea that the finished product will, in the light of experience, require modifications or additions. This attitude will drive you to plan the program in module form. It will also provide the incentive to include plenty of REMs and, most important of all, it will teach you to cut down on the GOTOs and to keep a register of the variables used. In short, it will help you to write good programs — not just ones that work.

## Finding And Curing Bugs

To start with bug finding before we have treated programming in detail may seem like putting the cart before the horse but it is based on sound psychological reasoning. Programming is unlike any other discipline because it demands a strange mixture of skills. Attempts to quantify the mixture will inevitably invite argument, but for what it's worth, programming is 70% science, 30% art. Imagination is also an important ingredient pro-



viding it is supported by a strong sense of method. But, above all, a programmer must always be on guard against the evils of frustration. It is ridiculously easy to be frustrated, particularly in the early stage of learning. The following advice (which should not be taken too literally) may prove comforting!



1. No program works first time.
2. If it does, it was too simple anyway.

There are two classes of bug:

- Programming errors which are unacceptable to the computer. These are relatively easy to cure because the computer outputs a screen message indicating the nature of the crime. A full list of the error messages are given on pages 151-152 of the Commodore User Manual.

- Programming errors which the computer accepts but cause the program to behave other than the intended behaviour. These bugs are more serious and could take a long time to find.

The most common bug in the first group is due to the use of incorrect grammar. The computer obligingly informs the user that a "SYNTAX ERROR" has been discovered. The offence could be caused by an incorrectly spelt keyword, wrong punctuation, unrecognisable keyword, bracket (parenthesis) ending or one more than there should be, or the incorrect use of arithmetic operators. With regard to brackets, remember there must always be an even number of them. An incorrect arithmetic operator can be more difficult to locate, particularly if the question is at all complex.

The most common error is likely to arise over multiplication. When we write  $Z = XY$  in ordinary mathematical notation, it is implied that X and Y are to be multiplied together. The computer does not respond to implications. It wants to know exactly, by using "\*" as the multiplication operator, so we must write  $Z = X*Y$ .

It should be mentioned here that failure to include the asterisk in this example would not necessarily result

in a syntax error because  $Z = XY$  is recognised as valid syntax. It will be interpreted as a simple assignment, in which the variable named XY is to be assigned to the variable named Z. The example provides a good illustration of the second type of bug. That is to say, the computer accepts it as a valid order but, if the programmer intended multiplication, the result would be quite different to that intended. This kind of error can take some spotting until you get used to arithmetic operators.

While on this subject, it is worth discussing the concept of operator precedence. Page 134 of the Commodore User Manual gives the order of precedence in which arithmetical operations are carried out. For example:

$$V1 = 3+8*2/2 \text{ will make } V1 = 20$$

$$V1 = 3+8/2/2 \text{ will make } V1 = 3$$

When you are learning, it is safer to forget precedence and rely on parentheses to indicate the order in which the arithmetic is to be applied. The two previous examples would be safer written:

$$V1 = 3+(8*(2/2))$$

$$V2 = 3+(8/2)/2$$

Although parentheses are safer for the inexperienced, they do increase translation time so, in programs where execution time is critical, it is better to revert to precedence.

Syntax errors are responsible for much of the early frustration, mainly because humans are not used to dealing with such fussy animals. It takes some time to come to terms with such unaccustomed intolerance, but RETURN and RETURN the program eventually reaches the last line number and then stops. The frustration should then be replaced by a warm glow of pride.



Our intrepid reviewer, Sue Duffield, was very impressed with this artistic program.

# ARTISTIC STUFF

WELL, COULD A COMPUTER teach me what nearly ten years of schooling couldn't, namely to paint? Could it and how? Move over Bambi — here comes Paintpic.

Paintpic is a complete painting and drawing system developed by the New Zealand company Kuma Computers in conjunction with Kuma Computers.

Claimed by the makers to be capable of producing outstanding results, I dutifully worked through the 30-page manual provided to see if I could. Thankfully the manual is very definitely written in layman's terms — it even tells you how to load the cassette which is a boon for the simple-minded. It gently takes the user step by step through the basics of the program, always working by example and you quickly reach the point where you actually start to paint.

The quick load (seven minutes) and the question and answer type entry bring you straight into the drawing mode.

And there you are — Turner's seascapes, Constable's landscapes or even Picasso's madness — all at your fingertips.

## The details

So how does it work?

When the program is loaded, a crosshair appears on the TV screen. You begin to paint by moving this joystick or a series of designated keys. A trail of minute dots follow whichever direction you choose to go in: left, right, up, down, and even diagonal. These painting methods are available, pen, brush or text mode.

Using the pen mode there are around 32,000 dots in the painting area



from which you make up your pictures. This means your painting can be both intricate and delicate but at the same time it is both painstaking and slow — not recommended for the splashy among you.

The brush mode means you can colour large areas of the screen quickly and there are individually definable brushes and brushes which you can store and retrieve at will.

The text mode is, as the name suggests, simply the ability to reproduce on the screen any of the keys, letters, figures and symbols, accessed through the keyboard.

You paint in four colours at any given time, chosen from the full colour range of the Commodore 64.

Once you start to paint, if you make a mistake or change your mind, you don't have to wipe the

screen clean and start again. Erasing is simply a matter of retracing your steps using a different colour. Who knows how Van Gogh's paintings would have turned out if he'd had this facility?

You can also nuke all the hard work out of moving the crosshair round and round the screen as you paint because there are nine home positions, accessible by a single key stroke, which take away the necessity of going over and over the same area to get to where you want to be.

A little patience and in no time at all you'll be producing results to stun your friends. And you don't need to lose the masterpiece when you switch the TV off and go to bed. You can save all or part of your painting on tape and retrieve it when you want. The manual also gives

detailed instructions on how to photograph your efforts direct from the TV so you can keep them forever.

Very young children can play happily at this game, using the joystick, with no mess or washing up and low tears.

Adults can get equally hooked creating and recreating at little cost to either pocket or temper. Even professional artists, claim the manual, can use the system to produce accurate publishable visual work.

Truly versatile, totally addictive and just fascinating, this is one of the best 'games' I've played for a long time.

Paintpic is for the Commodore 64, is priced at £19.95 and comes from Kuma Computers.



Can you outguess the machine? Derek Waldron's programmed a CBM 64 version of this popular game.

# CODE BREAKER

THE PROGRAM IS A version of the popular type of code-breaking game. It is a game for two players who take the roles of 'codebreaker' and 'codebreaker'. In any version for the Commodore, the computer is given the role of codebreaker, leaving you the task of cracking the code.

The code consists of four colours from a choice of eight. Not only do you have to guess which colours have been used, but you also have to determine where each colour has been placed within the code. To help you reach this goal, the computer will mark each guess that you make, according to the following rules: for each colour and position guessed correctly, a black 'peg' will be awarded. Where a colour is guessed correctly, but the position does not match that in the code, a white 'peg' will be awarded. A mark of 4 black 'pegs' therefore means you have cracked the code!

In view of this assistance, you are only allowed 10 attempts in which to break the code. If, after this time, you have been unsuccessful in your attempts, the computer will reveal to you what the code was.

To enter your guess, keys 1-8 should be used, the colour chosen is passed, a square on the grid will be filled with the appropriate colour, working left to right as each colour is entered. It should be noted that once a colour has been entered it cannot be deleted, so careful thought should be given before entering any colour.

Once the fourth colour has been entered, the computer will automatically



reveal your guess and print the appropriate mark in the answer box alongside the line you have entered. The answer will be in the form of black and/or white 'pegs' in accordance with the rules already specified. Please note that the position of any black or white 'peg' in the answer bears no relation to the position of the colour in your guess. This means that a black 'peg' in the left-most position of the answer does not necessarily mean that the left-most colour of your guess is the one that is right.

## Watch Out

The relevant points to watch for in each answer given by the computer are the number of 'pegs' and their colours.

If any of your guesses are totally unsuccessful, obviously no pegs will be awarded in the marking, which will be backed up by an unpleasant noise. Do not be disheartened; you will come to learn that an unsuccessful guess will often give you more information than one rewarded with, say, one or two 'pegs'.

The scope for using sound within a program of this nature is limited, and has been restricted to the already mentioned 'noise' for an unsuccessful guess, together with different pitched notes for black and white 'pegs', when marked. When you are able to crack the code within the 10 guesses allowed, you will also hear a short, familiar tune.

A summary of the instructions to play the game are included in the program.



## The Program

Line 18	Cleans the screen and changes the background to grey, with a different shade grey border	Line 803	Checks if the answer is all black 'pegs'
Lines 28-30	Prints the first page defining the object of the game	Line 810	Checks if the answer is entirely wrong
Lines 100-110	Requests a key to be pressed to continue	Line 820	Sets up a loop to be repeated for as many times as there are 'pegs' to be printed
Lines 113-140	Prints the second page giving instructions to play	Line 830	POKEs a circle [arg] directly to the screen
Lines 158-160	Requests a key to be pressed to continue	Line 840	Checks if any black 'pegs' are to be awarded
Line 178	Sets up the initial values for the variable array GP(n)	Line 850	Decrements the black 'peg' counter by one, POKEs the 'peg' position with the colour black, goes to the 'black peg note' subroutine
Line 179	Defines A4	Line 860	GOTO 880
Line 179	Defines the variables used for producing sound	Line 870	POKEs the position of the peg with the colour white and then goes to the 'white peg note' subroutine
Line 180	Sets up the initial values for the variable array AP(n)	Line 880	Repeats the loop
Line 190	Sets CT equal to 4	Line 890	Prints 4 black pegs on the answer pad and goes to the 'play a tune' subroutine
Lines 308-410	Cleans the screen and prints the grids on which the guesses and answers are placed	Line 900	Adds if you want another go at the bottom of the screen
Line 508	Generates a random code of four colours stored in the array C, 00	Line 910	Asks for a key to be pressed
Line 548	Increases CT by one and checks if 10 guesses have been made	Line 940	Checks if the key was a 'Y' and replays the game if it was
Line 558	Cleans GB and sets up a loop which will be repeated 4 times	Line 950	If the key pressed was not an 'N' the program goes back to request a key
Line 568	Requests a key to be pressed	Line 960	ends the program
Line 582	Converts the key pressed on an ASCII code, which is assigned to the variable CC	Line 978	Cleans the 'another go' question from the bottom of the screen and resets the program to line 178
Line 578	Checks that the key pressed is allowable (keys 1-8)	Line 1000	Sets volume to maximum, sets waveform to pulse, sets high and low pulse rates
Line 588	Converts the key pressed to a number representing the colour chosen, which is then assigned to the variable C(n)	Line 1010	Sets high and low frequency values, sets up sustain value
Line 598	Repeats the above loop until C(n) contains 4 values representing the colours of your guess	Line 1020	Play note
Line 608	Cleans the flags used in checking your guess	Line 1030	Resets high and low frequencies and sustain value
Line 618	Sets up a loop to be repeated four times	Line 1040	Play note and then turns voice off
Line 628	If guess does not equal code, repeats loop	Line 1100	Sets volume to maximum selects sawtooth waveform, sets attack wave
Line 638	Sets the relevant guess and code flags and increments the black 'peg' counter by one	Line 1120	Sets high and low frequency, and sustain level
Line 648	Repeats the loop	Line 1210	Play note and turns voice off
Line 658	Sets up two loops used in checking out of position colours	Line 1220	Sets volume to mid level, selects sawtooth waveform, sets attack value
Line 668	Checks if the flags have been set	Line 1280	Sets high and low frequency, and sustain level
Line 678	If guess does not equal code, repeats the loop	Line 1295	Play note and turns voice off
Line 688	Sets the flags and increments the white peg counter	Line 1310	Delay loop
Line 698	Repeats the loop	Line 1320	Sets volume to maximum, selects sawtooth waveform, sets attack level
Line 708	Goes to the 'print answer' subroutine and then goes back to the request input for next guess	Line 1330	Resets note values and duration from data statements
Line 718	Answers variable GP(n) which is used as a POKE address for printing joint guess POKEs directly to the screen a 'reversed space'	Line 1340	Checks, if all notes have been read
Line 728	POKEs the relevant colour to the position POKEd with a reverse space in the above line	Line 1350	Sets high and low frequency of note
Line 738	Returns from subroutine	Line 1360	Sets sustain value
Line 748	Answers the address to which the answer are to be POKEd	Line 1370	Play note, turns voice off, goes back to play next note
Line 758		Line 1380	Data containing values of notes to be played in turn
Line 768		Line 1390	Updates the position in the array AP(1)-AP(4)
Line 778		Line 1400	Prints the code if you were unsuccessful in cracking it
Line 788		Line 1420	Gets the colours of the code
Line 800		Line 1430	Delay loop before returning to ask if you would like another go.





Mike Roberts has  
come to the rescue of  
jangled line numbers  
with this great  
renumbering utility  
for the CBM 64.

# RENUMBERING RENUMBERING RENUMBERING RENUMBERING RENUMBERING

1234567890E  
1234567890E

COMMODORE BASIC IS lacking in a great deal of functions. Graphics commands have been well documented and most magazines have published programs that do hi-res, sprites, sound etc.

Utilities tend to be a bit thin on the ground as everybody's efforts are directed against the graphics. While using SuperSoft's 'Winks' assembler I suddenly needed a renumber program. The following machine code program was the result. As I only use it for renumbering assembler programs I'm afraid that there is no facility for renumbering GOTO, GOSUB etc. as this would make the program quite long and complex.

As it stands it is just as good as the renumber program in Commodore's line's BASIC cartridge.

## Setting Great Store

To explain how the program works it is first necessary to go into detail about how BASIC stores a program line. The first byte in a program line is zero (not the last byte as most people seem to think), the next two bytes contain the address of the start of the next program line. This is called the link address and is used by the interpreter only while running a program and inserting a new line. The next pair of bytes is the pair that will be inserted in — the line number. After the line number is the rest of the BASIC line.

The line number is stored as a 16 bit binary unsigned integer, so it can be from 0 to 65535, so for a renumber program all we have to do is scan through the program picking out the

zero bytes, indicating the start of a line, skip a few and modify the two bytes containing the line number — easy!

Of course there are some options. You can dictate the line number that the program starts with and the difference between them. This is done by POKEing four bytes into the 64's memory:

```
POKE 348,INT(X/256):POKE
349,INT(X/256)*256
```

This will set the start number for the program where 'X' should be replaced by the number that you want the program to start with.

```
POKE350,INT(Y/256):POKE
349,INT(Y/256)*256
```

This will set up the difference between the lines, where 'Y' should be replaced by the increment.

The machine code should then be called with 595 49132, after a short pause the program will be renumbered. The POKE'd values will be preserved, so you only have to POKE them the first time, the next time they will still be there and you just need to do the 595 call.

## Machine Code Program

```
10 B=0000          250 RDC INCR+1
20 START=00F7     260 STR COUNT+1
30 INCR=00F9      270 LOOP JSR INPT
40 PTR=00F8       280 LDR (PTR),Y
50 COUNT=00F8     290 CFF #000
60 LNK START      300 BNE LOOP
70 STR COUNT      310 INH
80 LNK START+1    320 LDR (PTR),Y
90 STR COUNT+1    330 CFF #000
100 LDR #000      340 BNE HKT
110 STR PTR+1     350 INH
120 LNK #000      360 LDR (PTR),Y
130 STR PTR       370 CFF #000
140 LDR #000      380 BEO ENDT
150 NEXTLN LDR COUNT 390 HKT LDR #000
160 STR (PTR),Y   400 JSR INPT
170 JSR INPT      410 JSR INPT
180 LDR COUNT+1   420 JSR INPT
190 STR (PTR),Y   430 JMP NEXTLN
200 CLC           440 ENDT RTS
210 LDR COUNT     450 INPT INC PTR
220 RDC INCR      460 BNE NEXTP
230 STR COUNT     470 INC PTR+1
240 LDR COUNT+1   480 NEXTP RTS
```

## BASIC loader

```
1 REM RENUMBER BY MIKE ROBERTS JULY 1984
2 FORI=49132TO49248:REDOJ:POKEI,J:NDOT
3 DATA 166, 247, 134, 253, 166, 248, 134, 254, 163, 0
4 DATA 134, 253, 162, 0, 134, 251, 168, 0, 165, 253
5 DATA 148, 251, 32, 0, 192, 165, 254, 145, 251, 24
6 DATA 165, 253, 181, 249, 133, 253, 165, 254, 181, 250
7 DATA 139, 254, 32, 0, 192, 177, 251, 281, 0, 280
8 DATA 247, 208, 177, 251, 281, 0, 280, 7, 280, 177
9 DATA 251, 281, 0, 248, 14, 168, 0, 32, 0, 192
0 DATA 32, 0, 192, 32, 0, 192, 76, 18, 192, 96
100 DATA 238, 251, 288, 2, 238, 252, 96, 255, 255, 255
```



# Lamasoft

ORIGINAL SOFTWARE DESIGN

48 MOUNT PLEASANT, TADLET, HAMTS. RG25 6DP.



SEND S.A.E. FOR FREE NEWS LETTER "THE NATURE OF THE BEAST".



DEALERS may order direct from CBS Telebooks 01-940 2195 quoting account number, UK code number and quantities. Orders delivered within 48 hours.

Painting in the pixels

was never easier!

Design your own

sprites with this

package from

J McHale

and A Carton.

# SPRITE DESIGNER '64

TYPE IN THE PROGRAM provided and save it on a blank cassette. Now RUN it and if all has gone as planned, a message will appear on the screen, telling you to place a blank cassette in your C2N unit so that 'Sprite Designer '64' may be SAVED to tape as a machine code file.

After it has been SAVED, verify it to ensure that there are no errors present.

To load 'Sprite Designer '64', type 'LOAD', '1' — then type 'SPS (64/38)' to reset the CBM 64's pointers before using the program.

As 'Sprite Designer' is located at \$4080 (4092 decimal), you may use it in conjunction with a BASIC program of up to 128.

Sprites designed by this program are located between pages 230 and 254 inclusive in (230x64) — (254x64) x 63.

You may design a total of 25 sprites which should be sufficient.

SPS (5000) initialises the sprite editor.

Movement is by means of the cursor keys, which I found to be the most

practical, with the Shift key playing its usual role in CURSOR = Right; CESC + SHFT = Left.

When initialised, 'Sprite Designer '64' displays a grid and four sprites on the right hand side of the screen, with the markings 'X', 'Y', 'Y' & 'X' displayed alongside.

These letters stand for: standard sprite, X — expanded, Y — expanded and X&Y expanded respectively.

Current sprite status & Page No. is displayed at the top of the screen and you should use a cursor flashing in the top left hand corner of the 24 x 21 grid.

Every bit set in the sprite block will be displayed on the grid as a 'x' and every blank bit, a 'y'.

Remember to be careful when using the 'clear grid' function as it also wipes the sprite definition.

It is a good idea to familiarise yourself with the keys before starting to define sprites.

RUN/STOP — Restore has been disabled for programming purposes but is re-enabled on exiting to BASIC.

## Useful Locations

(\$CBAF) — 31967 is the only location to use to the user. It is used in the timing of keyboard response etc.

The value of this location is set at 40.

Changing this value will make key response faster or slower, 1 = fastest; 255 = slowest.

## Explanation

Function	Description	Key Used:
Clear Grid.	Clears sprite grid and current sprite block.	CLR/ HOME
Page Plus.	Advances to next sprite page.	'>
Page Minus.	Lowers sprite page.	'<
Exit.	Exit back to BASIC.	'X'
Save.	Save sprite definitions to tape.	'S'
Enable Multicolour.	Enable sprite Multicolour mode.	'M'
Disable Multicolour.	Disable sprite Multicolour mode.	'M'
Change Sprite colour.		'C'
Change Miscel Reg 1.		'1'
Change Miscel Reg 2.		'2'
Rotate Sprite.	Rotates sprite through 180° (Horizontally)	'R'
Invert Sprite.	Rotates sprite through 180° (Vertically)	'I'
Fill.	Fills in one bit of the sprite definition.	'SPC'
Delete.	Runs out one bit of the sprite definition.	'SPC'





## Program Listing

```

10 REM *****
15 REM #
20 REM # SPRITE DESIGNER '64 #
25 REM #
30 REM #          BY          #
35 REM #
40 REM # J. MC HALE & R. CARTON #
45 REM #
50 REM # SL100-EIRE ..... #
55 REM #
60 REM *****
100 FOR%3281,3:FOR%3280,3
110 PRINT"SPRITE DESIGNER '64 WRITTEN BY J.MC HALE"
120 PRINT"(<25>: "CHR%3:FOR%31>:"R,CARTON"
130 GOSUB%60:GOSUB%60
140 PRINT"*****PLEASE WAIT - LOADING CODE INTO MEMORY!"
150 SYS%3000
160 OR%40152:TL%0
170 FOR%0:DFH%7:WT%THE%000
180 FOR%40107
190 H%FC%01:SR%00:CR%+1,17>+40:SF%9TH%0000+7
200 L%FC%01:SR%00:CR%+2,17>+40:DF%0TH%0000+7
210 FC%00+L:FOR%0+C,RC:TL%TL+RC:NR%TC:SR%00+G0T%01/01
220 FOR%0000,40:FOR%0000,224:SYS%0000
230 PRINT"0":
240 DF%0+016000*TL%000013TH%0000+REM # TRAP ERRORS #
250 REM # EVERYTHING BLURRY #
260 FOR%3280,14:FOR%3281,6
280 PRINT"THE CODE HAS BEEN LOADED INTO MEMORY. IT
290 PRINT"PLEASE BLANK CASSETTE INTO YOUR COM UNIT."
300 PRINT"WHEN READY, PRESS '003' & 'SPRITE DESIGNER'
310 PRINT"WILL BE SAVED TO TAPE AS A MACHINE CODE."
320 PRINT"FILE #....."
330 FOR%0001373
340 DF%0+40TH%0000
350 SYS%2016:FOR%0+0T%0000+NEXT
360 PRINT"SPRITE DESIGNER HAS BEEN SAVED ON TAPE":
370 PRINT"IN MACHINE CODE FORMAT, TO RE-LOAD, TYPE IT
380 PRINT"LOAD"CHR%34"CHR%34"01".5.1"
390 PRINT"*****THE(13):"CHR%34"URGENT"
400 PRINT"VERIFY SPRITE DESIGNER TO ENSURE THAT"
410 PRINT"IT HAS BEEN SAVED WITHOUT ERRORS ....."
420 FOR%0000,21:FOR%0000,287:NR%0000
430 END
440 REM # ERROR TRAPPING #
510 PRINT"0":
520 IF%01600TH%PRINT"INSUFFICIENT DATA ERROR ?":G0T%500
530 IF%01600TH%PRINT"EXCESSIVE DATA ERROR ?"
540 PRINT"CHECK DATA LIST CAREFULLY"
550 IF TL%00013TH%0000
560 PRINT"CHECKSUM ERROR ? - CHECK DATA CAREFULLY"
570 PRINT"FOR INCORRECT ENTRIES....."
580 END
590 REM # SAVE TO TAPE ROUTINE #
610 FOR%0T%04:RE%0:FOR%0000+T,R:NEXT-RETURN
620 DTR%0,00,02,73,04,09,02,08,09,03,73,71,78,02,04,02
630 DTR%03,0,130,240,160,132,133,240,160,1,160,170
640 DTR%02,100,250,160,16,162,02,160,200,02,100,250
650 DTR%03,240,162,224,160,201,02,210,250,96
660 REM # INTERRUPT ROUTINE #
910 FOR%0T%000:RE%0:FOR%0000+R,R:NEXT-RETURN
920 DTR%120,160,21,141,20,0,160,207,141,21,0,00,96,130,2
930 DTR%000,0,76,49,204,160,0,133,2,162,40
940 DTR%107,223,0,73,128,107,223,0
950 DTR%002,200,240,76,49,204

```





This 64 utility from  
Mike Roberts will  
really speed up your  
tape system.

# FAST TAPE SEARCH



## First on

My program is designed to be recorded as the first program on a tape. The procedure when wanting to load a program is to rewind the tape to the beginning, load and run the directory program **bat**, and this is important. **DO NOT HOLD THE STOP/EJECT BUTTON.** When the program runs it will ask you what file you want, and whether you are going to save to it or load from it. Next comes the important bit — you must hold down the **PLAY** key and press down the **F/PAID** key at the same time so that they are both down at once, now you must hold down the **F/PAID** key and press the **STOP/EJECT** key. The end result of all this maneuvering is a silent tape recorder and a depressed **F/PAID** key.

Now if you merely press the Return button, the tape recorder will magically start to make a whizzing sound. After a number of seconds, depending on which file you want, the tape recorder will stop and you can proceed as normal.

This system will find programs on tape a lot faster than any normal method.

## But wait...

There are one or two restrictions though:

The tape must be blank when you start to save programs on it, you can't just tack the catalogue on the beginning of a tape of programs and hope it will find them. The program

expects the programs at evenly spaced intervals, that's how it works. The program is currently set up for 10 files of about 8K each on a C80 cassette. These parameters can be easily changed to suit your own tastes.

The number of programs on the tape is stored in a

**DATA** statement in line 238. This must be equalled by the following number of **DATA** statements. The **DATA** statements are padded out with spaces so that if you add a new program name to the list and re-record it at the beginning of the tape it will be the same length as it originally was and not

overwrite the first file.

The length of the sectors (currently 8K, or 6 minutes) is dictated by the "16" in line 160. Double it for 16K sectors, halve it for 4K etc.

This is an invaluable utility and I have been using it on the PEI, VIC, and now the '64 for about the past five years.

```

10 PRINT "F"
20 GOTO 25
30 PRINT "C"
40 PRINT "T"
50 PRINT "L"
60 PRINT "M"
70 PRINT "N"
80 PRINT "O"
90 PRINT "P"
100 PRINT "Q"
110 PRINT "R"
120 PRINT "S"
130 PRINT "T"
140 PRINT "U"
150 PRINT "V"
160 PRINT "W"
170 PRINT "X"
180 PRINT "Y"
190 PRINT "Z"
200 PRINT "0"
210 PRINT "1"
220 PRINT "2"
230 PRINT "3"
240 PRINT "4"
250 PRINT "5"
260 PRINT "6"
270 PRINT "7"
280 PRINT "8"
290 PRINT "9"
300 PRINT "A"
310 PRINT "B"
320 PRINT "C"
330 PRINT "D"
340 PRINT "E"
350 PRINT "F"
360 PRINT "G"
370 PRINT "H"
380 PRINT "I"
390 PRINT "J"
400 PRINT "K"
410 PRINT "L"
420 PRINT "M"
430 PRINT "N"
440 PRINT "O"
450 PRINT "P"
460 PRINT "Q"
470 PRINT "R"
480 PRINT "S"
490 PRINT "T"
500 PRINT "U"
510 PRINT "V"
520 PRINT "W"
530 PRINT "X"
540 PRINT "Y"
550 PRINT "Z"
560 PRINT "0"
570 PRINT "1"
580 PRINT "2"
590 PRINT "3"
600 PRINT "4"
610 PRINT "5"
620 PRINT "6"
630 PRINT "7"
640 PRINT "8"
650 PRINT "9"
660 PRINT "A"
670 PRINT "B"
680 PRINT "C"
690 PRINT "D"
700 PRINT "E"
710 PRINT "F"
720 PRINT "G"
730 PRINT "H"
740 PRINT "I"
750 PRINT "J"
760 PRINT "K"
770 PRINT "L"
780 PRINT "M"
790 PRINT "N"
800 PRINT "O"
810 PRINT "P"
820 PRINT "Q"
830 PRINT "R"
840 PRINT "S"
850 PRINT "T"
860 PRINT "U"
870 PRINT "V"
880 PRINT "W"
890 PRINT "X"
900 PRINT "Y"
910 PRINT "Z"
920 PRINT "0"
930 PRINT "1"
940 PRINT "2"
950 PRINT "3"
960 PRINT "4"
970 PRINT "5"
980 PRINT "6"
990 PRINT "7"
1000 PRINT "8"
1010 PRINT "9"
1020 PRINT "A"
1030 PRINT "B"
1040 PRINT "C"
1050 PRINT "D"
1060 PRINT "E"
1070 PRINT "F"
1080 PRINT "G"
1090 PRINT "H"
1100 PRINT "I"
1110 PRINT "J"
1120 PRINT "K"
1130 PRINT "L"
1140 PRINT "M"
1150 PRINT "N"
1160 PRINT "O"
1170 PRINT "P"
1180 PRINT "Q"
1190 PRINT "R"
1200 PRINT "S"
1210 PRINT "T"
1220 PRINT "U"
1230 PRINT "V"
1240 PRINT "W"
1250 PRINT "X"
1260 PRINT "Y"
1270 PRINT "Z"
1280 PRINT "0"
1290 PRINT "1"
1300 PRINT "2"
1310 PRINT "3"
1320 PRINT "4"
1330 PRINT "5"
1340 PRINT "6"
1350 PRINT "7"
1360 PRINT "8"
1370 PRINT "9"
1380 PRINT "A"
1390 PRINT "B"
1400 PRINT "C"
1410 PRINT "D"
1420 PRINT "E"
1430 PRINT "F"
1440 PRINT "G"
1450 PRINT "H"
1460 PRINT "I"
1470 PRINT "J"
1480 PRINT "K"
1490 PRINT "L"
1500 PRINT "M"
1510 PRINT "N"
1520 PRINT "O"
1530 PRINT "P"
1540 PRINT "Q"
1550 PRINT "R"
1560 PRINT "S"
1570 PRINT "T"
1580 PRINT "U"
1590 PRINT "V"
1600 PRINT "W"
1610 PRINT "X"
1620 PRINT "Y"
1630 PRINT "Z"
1640 PRINT "0"
1650 PRINT "1"
1660 PRINT "2"
1670 PRINT "3"
1680 PRINT "4"
1690 PRINT "5"
1700 PRINT "6"
1710 PRINT "7"
1720 PRINT "8"
1730 PRINT "9"
1740 PRINT "A"
1750 PRINT "B"
1760 PRINT "C"
1770 PRINT "D"
1780 PRINT "E"
1790 PRINT "F"
1800 PRINT "G"
1810 PRINT "H"
1820 PRINT "I"
1830 PRINT "J"
1840 PRINT "K"
1850 PRINT "L"
1860 PRINT "M"
1870 PRINT "N"
1880 PRINT "O"
1890 PRINT "P"
1900 PRINT "Q"
1910 PRINT "R"
1920 PRINT "S"
1930 PRINT "T"
1940 PRINT "U"
1950 PRINT "V"
1960 PRINT "W"
1970 PRINT "X"
1980 PRINT "Y"
1990 PRINT "Z"
2000 PRINT "0"
2010 PRINT "1"
2020 PRINT "2"
2030 PRINT "3"
2040 PRINT "4"
2050 PRINT "5"
2060 PRINT "6"
2070 PRINT "7"
2080 PRINT "8"
2090 PRINT "9"
2100 PRINT "A"
2110 PRINT "B"
2120 PRINT "C"
2130 PRINT "D"
2140 PRINT "E"
2150 PRINT "F"
2160 PRINT "G"
2170 PRINT "H"
2180 PRINT "I"
2190 PRINT "J"
2200 PRINT "K"
2210 PRINT "L"
2220 PRINT "M"
2230 PRINT "N"
2240 PRINT "O"
2250 PRINT "P"
2260 PRINT "Q"
2270 PRINT "R"
2280 PRINT "S"
2290 PRINT "T"
2300 PRINT "U"
2310 PRINT "V"
2320 PRINT "W"
2330 PRINT "X"
2340 PRINT "Y"
2350 PRINT "Z"
2360 PRINT "0"
2370 PRINT "1"
2380 PRINT "2"
2390 PRINT "3"
2400 PRINT "4"
2410 PRINT "5"
2420 PRINT "6"
2430 PRINT "7"
2440 PRINT "8"
2450 PRINT "9"
2460 PRINT "A"
2470 PRINT "B"
2480 PRINT "C"
2490 PRINT "D"
2500 PRINT "E"
2510 PRINT "F"
2520 PRINT "G"
2530 PRINT "H"
2540 PRINT "I"
2550 PRINT "J"
2560 PRINT "K"
2570 PRINT "L"
2580 PRINT "M"
2590 PRINT "N"
2600 PRINT "O"
2610 PRINT "P"
2620 PRINT "Q"
2630 PRINT "R"
2640 PRINT "S"
2650 PRINT "T"
2660 PRINT "U"
2670 PRINT "V"
2680 PRINT "W"
2690 PRINT "X"
2700 PRINT "Y"
2710 PRINT "Z"
2720 PRINT "0"
2730 PRINT "1"
2740 PRINT "2"
2750 PRINT "3"
2760 PRINT "4"
2770 PRINT "5"
2780 PRINT "6"
2790 PRINT "7"
2800 PRINT "8"
2810 PRINT "9"
2820 PRINT "A"
2830 PRINT "B"
2840 PRINT "C"
2850 PRINT "D"
2860 PRINT "E"
2870 PRINT "F"
2880 PRINT "G"
2890 PRINT "H"
2900 PRINT "I"
2910 PRINT "J"
2920 PRINT "K"
2930 PRINT "L"
2940 PRINT "M"
2950 PRINT "N"
2960 PRINT "O"
2970 PRINT "P"
2980 PRINT "Q"
2990 PRINT "R"
3000 PRINT "S"
3010 PRINT "T"
3020 PRINT "U"
3030 PRINT "V"
3040 PRINT "W"
3050 PRINT "X"
3060 PRINT "Y"
3070 PRINT "Z"
3080 PRINT "0"
3090 PRINT "1"
3100 PRINT "2"
3110 PRINT "3"
3120 PRINT "4"
3130 PRINT "5"
3140 PRINT "6"
3150 PRINT "7"
3160 PRINT "8"
3170 PRINT "9"
3180 PRINT "A"
3190 PRINT "B"
3200 PRINT "C"
3210 PRINT "D"
3220 PRINT "E"
3230 PRINT "F"
3240 PRINT "G"
3250 PRINT "H"
3260 PRINT "I"
3270 PRINT "J"
3280 PRINT "K"
3290 PRINT "L"
3300 PRINT "M"
3310 PRINT "N"
3320 PRINT "O"
3330 PRINT "P"
3340 PRINT "Q"
3350 PRINT "R"
3360 PRINT "S"
3370 PRINT "T"
3380 PRINT "U"
3390 PRINT "V"
3400 PRINT "W"
3410 PRINT "X"
3420 PRINT "Y"
3430 PRINT "Z"
3440 PRINT "0"
3450 PRINT "1"
3460 PRINT "2"
3470 PRINT "3"
3480 PRINT "4"
3490 PRINT "5"
3500 PRINT "6"
3510 PRINT "7"
3520 PRINT "8"
3530 PRINT "9"
3540 PRINT "A"
3550 PRINT "B"
3560 PRINT "C"
3570 PRINT "D"
3580 PRINT "E"
3590 PRINT "F"
3600 PRINT "G"
3610 PRINT "H"
3620 PRINT "I"
3630 PRINT "J"
3640 PRINT "K"
3650 PRINT "L"
3660 PRINT "M"
3670 PRINT "N"
3680 PRINT "O"
3690 PRINT "P"
3700 PRINT "Q"
3710 PRINT "R"
3720 PRINT "S"
3730 PRINT "T"
3740 PRINT "U"
3750 PRINT "V"
3760 PRINT "W"
3770 PRINT "X"
3780 PRINT "Y"
3790 PRINT "Z"
3800 PRINT "0"
3810 PRINT "1"
3820 PRINT "2"
3830 PRINT "3"
3840 PRINT "4"
3850 PRINT "5"
3860 PRINT "6"
3870 PRINT "7"
3880 PRINT "8"
3890 PRINT "9"
3900 PRINT "A"
3910 PRINT "B"
3920 PRINT "C"
3930 PRINT "D"
3940 PRINT "E"
3950 PRINT "F"
3960 PRINT "G"
3970 PRINT "H"
3980 PRINT "I"
3990 PRINT "J"
4000 PRINT "K"
4010 PRINT "L"
4020 PRINT "M"
4030 PRINT "N"
4040 PRINT "O"
4050 PRINT "P"
4060 PRINT "Q"
4070 PRINT "R"
4080 PRINT "S"
4090 PRINT "T"
4100 PRINT "U"
4110 PRINT "V"
4120 PRINT "W"
4130 PRINT "X"
4140 PRINT "Y"
4150 PRINT "Z"
4160 PRINT "0"
4170 PRINT "1"
4180 PRINT "2"
4190 PRINT "3"
4200 PRINT "4"
4210 PRINT "5"
4220 PRINT "6"
4230 PRINT "7"
4240 PRINT "8"
4250 PRINT "9"
4260 PRINT "A"
4270 PRINT "B"
4280 PRINT "C"
4290 PRINT "D"
4300 PRINT "E"
4310 PRINT "F"
4320 PRINT "G"
4330 PRINT "H"
4340 PRINT "I"
4350 PRINT "J"
4360 PRINT "K"
4370 PRINT "L"
4380 PRINT "M"
4390 PRINT "N"
4400 PRINT "O"
4410 PRINT "P"
4420 PRINT "Q"
4430 PRINT "R"
4440 PRINT "S"
4450 PRINT "T"
4460 PRINT "U"
4470 PRINT "V"
4480 PRINT "W"
4490 PRINT "X"
4500 PRINT "Y"
4510 PRINT "Z"
4520 PRINT "0"
4530 PRINT "1"
4540 PRINT "2"
4550 PRINT "3"
4560 PRINT "4"
4570 PRINT "5"
4580 PRINT "6"
4590 PRINT "7"
4600 PRINT "8"
4610 PRINT "9"
4620 PRINT "A"
4630 PRINT "B"
4640 PRINT "C"
4650 PRINT "D"
4660 PRINT "E"
4670 PRINT "F"
4680 PRINT "G"
4690 PRINT "H"
4700 PRINT "I"
4710 PRINT "J"
4720 PRINT "K"
4730 PRINT "L"
4740 PRINT "M"
4750 PRINT "N"
4760 PRINT "O"
4770 PRINT "P"
4780 PRINT "Q"
4790 PRINT "R"
4800 PRINT "S"
4810 PRINT "T"
4820 PRINT "U"
4830 PRINT "V"
4840 PRINT "W"
4850 PRINT "X"
4860 PRINT "Y"
4870 PRINT "Z"
4880 PRINT "0"
4890 PRINT "1"
4900 PRINT "2"
4910 PRINT "3"
4920 PRINT "4"
4930 PRINT "5"
4940 PRINT "6"
4950 PRINT "7"
4960 PRINT "8"
4970 PRINT "9"
4980 PRINT "A"
4990 PRINT "B"
5000 PRINT "C"
5010 PRINT "D"
5020 PRINT "E"
5030 PRINT "F"
5040 PRINT "G"
5050 PRINT "H"
5060 PRINT "I"
5070 PRINT "J"
5080 PRINT "K"
5090 PRINT "L"
5100 PRINT "M"
5110 PRINT "N"
5120 PRINT "O"
5130 PRINT "P"
5140 PRINT "Q"
5150 PRINT "R"
5160 PRINT "S"
5170 PRINT "T"
5180 PRINT "U"
5190 PRINT "V"
5200 PRINT "W"
5210 PRINT "X"
5220 PRINT "Y"
5230 PRINT "Z"
5240 PRINT "0"
5250 PRINT "1"
5260 PRINT "2"
5270 PRINT "3"
5280 PRINT "4"
5290 PRINT "5"
5300 PRINT "6"
5310 PRINT "7"
5320 PRINT "8"
5330 PRINT "9"
5340 PRINT "A"
5350 PRINT "B"
5360 PRINT "C"
5370 PRINT "D"
5380 PRINT "E"
5390 PRINT "F"
5400 PRINT "G"
5410 PRINT "H"
5420 PRINT "I"
5430 PRINT "J"
5440 PRINT "K"
5450 PRINT "L"
5460 PRINT "M"
5470 PRINT "N"
5480 PRINT "O"
5490 PRINT "P"
5500 PRINT "Q"
5510 PRINT "R"
5520 PRINT "S"
5530 PRINT "T"
5540 PRINT "U"
5550 PRINT "V"
5560 PRINT "W"
5570 PRINT "X"
5580 PRINT "Y"
5590 PRINT "Z"
5600 PRINT "0"
5610 PRINT "1"
5620 PRINT "2"
5630 PRINT "3"
5640 PRINT "4"
5650 PRINT "5"
5660 PRINT "6"
5670 PRINT "7"
5680 PRINT "8"
5690 PRINT "9"
5700 PRINT "A"
5710 PRINT "B"
5720 PRINT "C"
5730 PRINT "D"
5740 PRINT "E"
5750 PRINT "F"
5760 PRINT "G"
5770 PRINT "H"
5780 PRINT "I"
5790 PRINT "J"
5800 PRINT "K"
5810 PRINT "L"
5820 PRINT "M"
5830 PRINT "N"
5840 PRINT "O"
5850 PRINT "P"
5860 PRINT "Q"
5870 PRINT "R"
5880 PRINT "S"
5890 PRINT "T"
5900 PRINT "U"
5910 PRINT "V"
5920 PRINT "W"
5930 PRINT "X"
5940 PRINT "Y"
5950 PRINT "Z"
5960 PRINT "0"
5970 PRINT "1"
5980 PRINT "2"
5990 PRINT "3"
6000 PRINT "4"
6010 PRINT "5"
6020 PRINT "6"
6030 PRINT "7"
6040 PRINT "8"
6050 PRINT "9"
6060 PRINT "A"
6070 PRINT "B"
6080 PRINT "C"
6090 PRINT "D"
6100 PRINT "E"
6110 PRINT "F"
6120 PRINT "G"
6130 PRINT "H"
6140 PRINT "I"
6150 PRINT "J"
6160 PRINT "K"
6170 PRINT "L"
6180 PRINT "M"
6190 PRINT "N"
6200 PRINT "O"
6210 PRINT "P"
6220 PRINT "Q"
6230 PRINT "R"
6240 PRINT "S"
6250 PRINT "T"
6260 PRINT "U"
6270 PRINT "V"
6280 PRINT "W"
6290 PRINT "X"
6300 PRINT "Y"
6310 PRINT "Z"
6320 PRINT "0"
6330 PRINT "1"
6340 PRINT "2"
6350 PRINT "3"
6360 PRINT "4"
6370 PRINT "5"
6380 PRINT "6"
6390 PRINT "7"
6400 PRINT "8"
6410 PRINT "9"
6420 PRINT "A"
6430 PRINT "B"
6440 PRINT "C"
6450 PRINT "D"
6460 PRINT "E"
6470 PRINT "F"
6480 PRINT "G"
6490 PRINT "H"
6500 PRINT "I"
6510 PRINT "J"
6520 PRINT "K"
6530 PRINT "L"
6540 PRINT "M"
6550 PRINT "N"
6560 PRINT "O"
6570 PRINT "P"
6580 PRINT "Q"
6590 PRINT "R"
6600 PRINT "S"
6610 PRINT "T"
6620 PRINT "U"
6630 PRINT "V"
6640 PRINT "W"
6650 PRINT "X"
6660 PRINT "Y"
6670 PRINT "Z"
6680 PRINT "0"
6690 PRINT "1"
6700 PRINT "2"
6710 PRINT "3"
6720 PRINT "4"
6730 PRINT "5"
6740 PRINT "6"
6750 PRINT "7"
6760 PRINT "8"
6770 PRINT "9"
6780 PRINT "A"
6790 PRINT "B"
6800 PRINT "C"
6810 PRINT "D"
6820 PRINT "E"
6830 PRINT "F"
6840 PRINT "G"
6850 PRINT "H"
6860 PRINT "I"
6870 PRINT "J"
6880 PRINT "K"
6890 PRINT "L"
6900 PRINT "M"
6910 PRINT "N"
6920 PRINT "O"
6930 PRINT "P"
6940 PRINT "Q"
6950 PRINT "R"
6960 PRINT "S"
6970 PRINT "T"
6980 PRINT "U"
6990 PRINT "V"
7000 PRINT "W"
7010 PRINT "X"
7020 PRINT "Y"
7030 PRINT "Z"
7040 PRINT "0"
7050 PRINT "1"
7060 PRINT "2"
7070 PRINT "3"
7080 PRINT "4"
7090 PRINT "5"
7100 PRINT "6"
7110 PRINT "7"
7120 PRINT "8"
7130 PRINT "9"
7140 PRINT "A"
7150 PRINT "B"
7160 PRINT "C"
7170 PRINT "D"
7180 PRINT "E"
7190 PRINT "F"
7200 PRINT "G"
7210 PRINT "H"
7220 PRINT "I"
7230 PRINT "J"
7240 PRINT "K"
7250 PRINT "L"
7260 PRINT "M"
7270 PRINT "N"
7280 PRINT "O"
7290 PRINT "P"
7300 PRINT "Q"
7310 PRINT "R"
7320 PRINT "S"
7330 PRINT "T"
7340 PRINT "U"
7350 PRINT "V"
7360 PRINT "W"
7370 PRINT "X"
7380 PRINT "Y"
7390 PRINT "Z"
7400 PRINT "0"
7410 PRINT "1"
7420 PRINT "2"
7430 PRINT "3"
7440 PRINT "4"
7450 PRINT "5"
7460 PRINT "6"
7470 PRINT "7"
7480 PRINT "8"
7490 PRINT "9"
7500 PRINT "A"
7510 PRINT "B"
7520 PRINT "C"
7530 PRINT "D"
7540 PRINT "E"
7550 PRINT "F"
7560 PRINT "G"
7570 PRINT "H"
7580 PRINT "I"
7590 PRINT "J"
7600 PRINT "K"
7610 PRINT "L"
7620 PRINT "M"
7630 PRINT "N"
7640 PRINT "O"
7650 PRINT "P"
7660 PRINT "Q"
7670 PRINT "R"
7680 PRINT "S"
7690 PRINT "T"
7700 PRINT "U"
7710 PRINT "V"
7720 PRINT "W"
7730 PRINT "X"
7740 PRINT "Y"
7750 PRINT "Z"
7760 PRINT "0"
7770 PRINT "1"
7780 PRINT "2"
7790 PRINT "3"
7800 PRINT "4"
7810 PRINT "5"
7820 PRINT "6"
7830 PRINT "7"
7840 PRINT "8"
7850 PRINT "9"
7860 PRINT "A"
7870 PRINT "B"
7880 PRINT "C"
7890 PRINT "D"
7900 PRINT "E"
7910 PRINT "F"
7920 PRINT "G"
7930 PRINT "H"
7940 PRINT "I"
7950 PRINT "J"
7960 PRINT "K"
7970 PRINT "L"
7980 PRINT "M"
7990 PRINT "N"
8000 PRINT "O"
8010 PRINT "P"
8020 PRINT "Q"
8030 PRINT "R"
8040 PRINT "S"
8050 PRINT "T"
8060 PRINT "U"
8070 PRINT "V"
8080 PRINT "W"
8090 PRINT "X"
8100 PRINT "Y"
8110 PRINT "Z"
8120 PRINT "0"
8130 PRINT "1"
8140 PRINT "2"
8150 PRINT "3"
8160 PRINT "4"
8170 PRINT "5"
8180 PRINT "6"
8190 PRINT "7"
8200 PRINT "8"
8210 PRINT "9"
8220 PRINT "A"
8230 PRINT "B"
8240 PRINT "C"
8250 PRINT "D"
8260 PRINT "E"
8270 PRINT "F"
8280 PRINT "G"
8290 PRINT "H"
8300 PRINT "I"
8310 PRINT "J"
8320 PRINT "K"
8330 PRINT "L"
8340 PRINT "M"
8350 PRINT "N"
8360 PRINT "O"
8370 PRINT "P"
8380 PRINT "Q"
8390 PRINT "R"
8400 PRINT "S"
8410 PRINT "T"
8420 PRINT "U"
8430 PRINT "V"
8440 PRINT "W"
8450 PRINT "X"
8460 PRINT "Y"
8470 PRINT "Z"
8480 PRINT "0"
8490 PRINT "1"
8500 PRINT "2"
8510 PRINT "3"
8520 PRINT "4"
8530 PRINT "5"
8540 PRINT "6"
8550 PRINT "7"
8560 PRINT "8"
8570 PRINT "9"
8580 PRINT "A"
8590 PRINT "B"
8600 PRINT "C"
8610 PRINT "D"
8620 PRINT "E"
8630 PRINT "F"
8640 PRINT "G"
8650 PRINT "H"
8660 PRINT "I"
8670 PRINT "J"
8680 PRINT "K"
8690 PRINT "L"
8700 PRINT "M"
8710 PRINT "N"
8720 PRINT "O"
8730 PRINT "P"
8740 PRINT "Q"
8750 PRINT "R"
8760 PRINT "S"
8770 PRINT "T"
8780 PRINT "U"
8790 PRINT "V"
8800 PRINT "W"
8810 PRINT "X"
8820 PRINT "Y"
8830 PRINT "Z"
8840 PRINT "0"
8850 PRINT "1"
8860 PRINT "2"
8870 PRINT "3"
8880 PRINT "4"
8890 PRINT "5"
8900 PRINT "6"
8910 PRINT "7"
8920 PRINT "8"
8930 PRINT "9"
8940 PRINT "A"
8950 PRINT "B"
8960 PRINT "C"
8970 PRINT "D"
8980 PRINT "E"
8990 PRINT "F"
9000 PRINT "G"
9010 PRINT "H"
9020 PRINT "I"
9030 PRINT "J"
9040 PRINT "K"
9050 PRINT "L"
9060 PRINT "M"
9070 PRINT "N"
9080 PRINT "O"
9090 PRINT "P"
9100 PRINT "Q"
9110 PRINT "R"
9120 PRINT "S"
9130 PRINT "T"
9140 PRINT "U"
9150 PRINT "V"
9160 PRINT "W"
9170 PRINT "X"
9180 PRINT "Y"
9190 PRINT "Z"
9200 PRINT "0"
9210 PRINT "1"
9220 PRINT "2"
9230 PRINT "3"
9240 PRINT "4"
9250 PRINT "5"
9260 PRINT "6"
9270 PRINT "7"
9280 PRINT "8"
9290 PRINT "9"
9300 PRINT "A"
9310 PRINT "B"
9320 PRINT "C"
9330 PRINT "D"
9340 PRINT "E"
9350 PRINT "F"
9360 PRINT "G"
9370 PRINT "H"
9380 PRINT "I"
9390 PRINT "J"
9400 PRINT "K"
9410 PRINT "L"
9420 PRINT "M"
9430 PRINT "N"
9440 PRINT "O"
9450 PRINT "P"
9460 PRINT "Q"
9470 PRINT "R"
9480 PRINT "S"
9490 PRINT "T"
9500 PRINT "U"
9510 PRINT "V"
9520 PRINT "W"
9530 PRINT "X"
9540 PRINT "Y"
9550 PRINT "Z"
9560 PRINT "0"
9570 PRINT "1"
9580 PRINT "2"
9590 PRINT "3"
9600 PRINT "4"
9610 PRINT "5"
9620 PRINT "6"
9630 PRINT "7"
9640 PRINT "8"
9650 PRINT "9"
9660 PRINT "A"
9670 PRINT "B"
9680 PRINT "C"
9690 PRINT "D"
9700 PRINT "E"
9710 PRINT "F"
9720 PRINT "G"
9730 PRINT "H"
9740 PRINT "I"
9750 PRINT "J"
9760 PRINT "K"
9770 PRINT "L"
9780 PRINT "M"
9790 PRINT "N"
9800 PRINT "O"
9810 PRINT "P"
9820 PRINT "Q"
9830 PRINT "R"
9840 PRINT "S"
9850 PRINT "T"
9860 PRINT "U"
9870 PRINT "V"
9880 PRINT "W"
9890 PRINT "X"
9900 PRINT "Y"
9910 PRINT "Z"
9920 PRINT "0"
9930 PRINT "1"
9940 PRINT "2"
9950 PRINT "3"
9960 PRINT "4"
9970 PRINT "5"
9980 PRINT "6"
9990 PRINT "7"

```

ORDER TODAY  
PRINT TOMORROW  
at special discounts

# SUMMER MADNESS SALE FROM SCI(UK)

OPEN  
7 DAYS  
A WEEK

## EPSON PRICE SPECIALS



EPSON RX80 (DOT MATRIX) . . .	£149.00	£199 + VAT = £220.50
EPSON RX80FT (DOT MATRIX) . . .	£185.00	£239 + VAT = £260.50
EPSON FX80 (DOT MATRIX) . . .	£438.00	£534 + VAT = £572.50
EPSON MX100 (DOT MATRIX) . . .	£475.00	£585 + VAT = £627.50
EPSON RX100 (DOT MATRIX) . . .	£459.00	£585 + VAT = £642.50
EPSON FX100(DOT MATRIX) . . .	£569.00	£699 + VAT = £731.50

## DAISYWHEELS...at an incredible new LOW PRICE!



### JUKI 6100.....just £329 + VAT = £378.25

20CPI: Bi-Directional & Logic Printing  
 10, 12, 15 & Proportional Spacing  
 Keyboard Compatible  
 2K Buffer: 15 Inch Platen  
 Underline: Backspace + Ltr more  
 Centronics Interface Standard

THE DAISYWHEEL THAT  
HAS NO COMPETITION

OPTIONAL RS 232 TRACTOR AND SHEET FEEDER

*We will match any Genuine Price Advertised—  
SCI(UK) IS NEVER BEATEN ON PRICE*



MANY MORE PRINTERS AVAILABLE - 1000s of SCI(UK) BARGAINS  
send now for the FAMOUS SCI(UK) Catalogue

for the cheapest prices telephone 0730 68521 or 0730 68522



## MORE SCI(UK) BARGAINS

SHINWA CP80 . . . £179.00 + VAT = £200.50



Printer and tractor feed as standard  
 80cps  
 Bi-Directional logic coding  
 21 x 9 dot matrix giving clear descenders  
 Bold and superscripts  
 Studio printing and auto underlining  
 Condensed, emphasized, expanded and  
 double width printing (can be mixed in a line)  
 Parallel interface fitted as standard

FIDELITY 14"  
 COLOUR  
 MONITOR  
 & COMPOSITE  
 VIDEO



£189.00 + VAT = £210.50

New from the world famous CANON Company

CANON 1080a NLQ DM best value ever at.....£289.00 + VAT = £322.50

We have interfaces for all types of computers,  
including CBM 64, VIC 20, APPLE, TRS 80,  
IBM, BBC, SPECTRUM, QL, etc.



INCLUDE DELIVERY DELIVERY 29.50 plus VAT \* BANKERS ORDERS, BUILDING SOCIETY CHEQUES, POSTAL  
ORDERS — SAME DAY DISPATCH \* ALL ORDERS COVERED BY THE MAIL ORDER PROTECTION SCHEME  
NATIONWIDE MAINTENANCE CONTRACTS ARRANGED \* EDUCATIONAL DISCOUNTS VERY WELCOME  
It's SUNDAY - Do you realise you can order NOW - We are open 7 Days a Week.

BANKERS LTD. Trading as SCI(UK)

# SCI(UK)

101-102, HENRIETTA ST, THE SHOP MARKET,  
PETERSFIELD, HANTS. GU32 3JG

0730 68521

0730 68522

EXPRESS DELIVERY  
FOR THE  
QUALITY OF COST

WELLS BRANCH  
WELCOMES  
NEW FOR 1982

Large showrooms now open at 12 High Street, PETERSFIELD, Hants. GU32 3JG Telex 88628 MYNEWS G

We've searched the software shelves and thrown the spotlight on some available packages. See what our intrepid reviewers think of them before you part with your money.



**SHADES OF BATTLE ZONE** this one! Confronted with a 3-D plain (yes, it is 3-D), you hear the aliens approaching. Diamond shaped, fast moving aliens which don't fire at first, appear and are easy prey. Then, if you have the sound up on your TV, you will hear at times a low humming noise. This is when your shooting accuracy counts. The noise signifies that there is a suicidal alien missile coming towards you. One shot is enough but it's getting that one shot that is the problem.

Once you dispatch the alien, a door or vortex opens on the screen. When you travel through it you will have to steer round a star field making sure you don't hit anything. When you accomplish this minor (I) task another door opens onto the second level. This level is a little harder than the first with the alien shooting back and aliens which explode on their own.

Definitely a good game, having eight landscapes and three skill levels, it is very challenging. Hitting the sound effects is quite helpful as they tell you what to expect from simple aliens to a blipping exploding alien on level four.

S.L.P.P.

# SOFTWARE



# SPOTLIGHT

**OUR REVIEWERS HAVE** spent long hours poring over the software to be assessed. They have tried to hear in mind a number of factors that will be of interest to you, the reader, such as the use of sound and graphics, how quickly the game became boring or addictive and to give you an overall impression of what

to expect should you decide to spend some of your hard-earned cash.

Each review has been given a star rating out of a maximum of five stars (meaning brilliant, why haven't you rushed to your local store already?) down to one star (which means that the chances are you won't be ecstatic with your

new purchase!). Always read the review and do not judge the package solely on the star rating — you may not be too bothered about the poor graphics and sound that have earned the package this opinion. And always remember that the review is only one man's opinion — one man's meat, etc. . . .



**YOU HAVE ENTERED THE** mountain ridge in your helicopter determined to stop the trucks carrying the arms from getting through to the enemies' base. They have heat seeking missiles which they fire at you. This is the setting from Sundlock's game, Chopper. This is done by moving your chopper left, right, up and down, dropping bombs on the trucks.

The graphics are jerky, the sound is not very good

and the animation is fair. Loading is quite simple. It is an addictive game and very quick. The action is fast and it gets faster as the game progresses. It's a pity there are no skill levels. There are other good points especially the loading you have when you have beaten the guided missiles. The scoring is very simple, 50 for a large and small truck, 100 for a tank and 250 for a missile launcher. The key choice is good and joystick response is also good. This is a brilliant idea for a game, however it's a great pity that it was not thought out more constructively. Overall a fast and reasonably enjoyable game from Sundlock but for £5.95 it doesn't appeal to me as value for money. P.B.W.



## Maziacs

\*\*\*  
**DIETRONICS**  
 \$5.95  
 CBM 64 • Joystick (Optional)

THE IDEA OF THE GAME IS to go into a maze and collect the treasure in the smallest possible amount of moves. Preventing you from doing this are perhaps the meanest bunch of snakes a video gamer could hope to run up against, the Maziacs.

The game starts with your character clutching a sword ready to do battle with the Maziacs for the treasure, which is encarcerated somewhere in the maze. To enable you to fight, you have to be strong and to keep your strength up you will find there is food around the maze. You will also find prisoners in the maze who are willing to divulge the whereabouts of the gold. The other way to

find the treasure is by pressing "V" which will display a portion of the total maze.

When you fight with the Maziacs you will find that you lose your sword if you kill it. This isn't a problem as long as there is a spare sword lying about, which there usually is, if there isn't then you may have to fight with your bare hands and then you stand a chance of losing your life.

It is worth paying attention to the graphics in the fight because of the intricate detail that has been put in. It is a good game, but no two mazes the same.

S.L.P.

Commodore 64

## Maziacs



## Scramble



## Scramble

\*\*\*  
**RABBIT SOFTWARE**  
 \$5.95  
 CBM 64 • Joystick (Optional)

HAVING SEEN VARIOUS software companies make attempts at the game "Scramble" and being honest, not succeeding, I was quite surprised and excited at seeing this version.

Simply, it is even better than the original arcade machine game (yes, it's possible). To start with, the quality of graphics used in this game surpasses all other iterations. The game sequences are kept as close to the original as possible with few exceptions.

The first is that when playing you will notice some

musical notes scattered on the floor of the landscape. Shoot these and the music is turned on or off. This is a good idea, since the only drawback is having "star lines" playing in the background. This spoils the sound effects.

The second deviation from the original is screen 3. This screen contains mobile tanks that shoot at an angle, and it's quite difficult to fire at them and maintain a good supply of fuel.

It is a hard, if not harder, than the original but the only high score feature is when you're playing. It's definitely well worth the money as it is an example of state of the art games. I look forward to more projects from Nigel Rowland of Rabbit Software.

S.L.P.



## Triad

\*\*\*  
**LIVENTRE**  
 \$5.95  
 CBM 64 • Joystick (Optional)

TRIAD COMES IN A BOX containing no end of instructions, feature listings and pictures of the game. I couldn't wait to grab my joystick and start playing. On the box it said there was "3D total perspective graphics". "Smooth multicolour graphics, music and sound effects" etc, etc.

The game itself took a long time to load but I could wait! The description of the game was that the galactic

merchants had constructed nine routes in which they could travel. Unfortunately, the Triads kept on attacking without warning.

This is where your fleet of pilots come in. It is your job, firstly, to clear a patrol of Triads who advance on you at an alarming rate (on the easiest level) and, secondly, to clear a path through the nebulae. Once both screens are accomplished, you see a merchant vessel disappear behind a planet and land safely, then you return back to the first screen again!

The graphics on this game are not exactly the best, although the laser sight is quite good as it is shown in perspective as you move.

S.L.P.

## Alpha Blaster

\*\*\*  
**Sumtek**  
 \$5.95  
 VIC 20 • Optional joystick

THIS NEW GAME FROM Sumtek puts you in control of the latest space fighter; it

is a version of the arcade game Astroblaster. This version has no speech or warp factor. The basic idea is to shoot down everything in sight. This includes spaceships of different kinds and sizes. They do come down in a uniform area of movement and you have to dodge the

unshootable asteroids. After going through the asteroids you dock into the mothership; this is not a hard task because the computer automatically docks you. This is the basic idea of the game.

The sound is very good and the spaceships are in different colours. It is a fast

demanding game. A novice at shoot 'em up games is in for a hard time and even an expert won't find it easy.

The key choice, 2 right C left and Shift to fire, is good and makes it easy to play.

My only criticism is one bug. Overall this is a very good, fast moving game.

P.W.W.

# SOFTWARE SPOTLIGHT



## Skramble

★★★  
Bandlock  
\$7.95  
VIC 20 + joystick (Optional)

THIS IS A GAME IN THEIR "LiveWire" Series. Since loaded, which it did first time, the screen displayed the keys F1, and . For movement up, left, right and down shift to fire and CTRL to bombs. Key C starts the game. This is a version of the arcade game of the same name, graphics are fair and colour is reasonable. The game is written in machine code but is fairly slow, movements tend to be jerky, sound is used repetitively and continuously and becomes

irritating. The idea of the game is fully to use spaceship horizontally through a series of Gaves, avoiding mountains, rockets, meteors and other hazards, using your lasers and bombs to destroy the enemy's missiles and fuel dumps. This is explained in detail on the colourful insert.

Considering this is for the unexpanded VIC 20 they have made good use of the memory. However, the end result is a game which is disappointing and soon becomes tedious.

It is quite a difficult game to play and the lack of a high score or ball of fame doesn't encourage you to keep trying to improve your skills.

P.W.M.

## Metrolibit

★★★  
PWA  
C64/64  
£7.95

Two basic rapping games seem to have the lasting appeal of good old fashioned Space Invaders. However, as a variation on the theme, Metrolibit will have you instantly hooked and as a piece of arcade action should have you practising in your sleep.

You are the sole defender of the metropolis which is under attack from waves after of kamikaze aliens intent on destroying the city below you. Your space ship is in perpetual motion and you have to prove your prowess by dodging be-

twens all the alien craft and collaborating them with your superior fire power before they destroy all the buildings.

Apparently there are twenty four waves of aliens to defeat and naturally points for every alien craft you zap plus an additional variable bonus at the end of each wave depending on how many you hit. At wave 23 you enter the advanced stage and that's when they home in on you as well as the city, pass onto the championship challenge at wave 19 and the termination attack at wave 22.

In short, it's fast, it's vigorous and it's energetic and I lost count of the variety of alien craft and the number of times it's broken my joystick. Whimper! K.M.

## Horace Goes Skiing

★★★  
Melbourne House  
C64/64  
£5.95

HORACE IS NO STRANGER when it comes to computer games. Hungry Horace was his first starring role. This time he's determined to display his prowess on the piste. Regardless of the quality of the game, one of the prime advantages is the use of a fast loading system which loads in less than two minutes. There's nothing worse than hanging around

for ages waiting for a game to load! But it digress. Despite a partial similarity in the initial stages to those frustrating 'frogger' games, Horace Goes Skiing is an enjoyable and skilful game. The objective of the game is to get that cheeky little character across an extremely busy main road avoiding the juggernauts, cars and motorbikes and into the shed on the other side to pick up his skis. To reach the ski slopes of the Harewood run you have to get Horace back across the road which gets busier as time passes. What happens

if Horace gets splattered? Well it's not the end of the game. At the start of the game Horace has \$40 in his pocket and every time he gets run over it costs him \$10 for the ambulance ride. It also costs \$10 for the skis so Horace has a few chances. Extra cash can be accumulated by continually crossing the road for which points are gained and a \$10 bonus is given for every 1000 points. Once on the ski slopes the fun begins as Horace has to ski between the red and blue flags paralleling left and right and losing or gaining points

according to his success. But there are obstacles. Strategically placed conifers get in the way and if Horace hits one, there is a chance he will have to start all over again if his skis break. There are also cleverly placed rebounds to throw Horace off balance.

At the end of the run Horace simply has to cross the road again to reach the next slope. Of course there is more to this game than immediately meets the eye and despite its apparent simplicity has a high level of lasting appeal.

K.M.



## Flight Path 737

★ ★ ★ ★  
 Amigos  
 £7.95  
 VIC 20 + 16K RAM + joystick

THIS IS ONE OF THE ONLY flight simulations I have seen for the VIC and proved to be excellent. You have a choice of 6 levels, including a first solo flight (easier) to test pilot (hardest). The

game has an inflight clock, fuel indicator, altitude meter and many other dials found on the flight deck of an aircraft. On taking off you must have the flaps down and be at 180mph. The screen shows the runway which scrolls towards you in 3D perspective. The visual effects are good and are quite realistic. When airborne you see a

backdrop of mountains. The graphics tend not to be as good as the runway but are effective. Whilst cruising you may have to extinguish a fire in the aircraft. This is done by pressing key E. You now come to the difficult bit, landing. It starts with ease but the element of difficulty soon appears when an adjustment of your instruments is necessary.

Once landed, you are told how well you have, or have not, done and you are given a score accordingly.

The sound is very realistic and adds flavour to the simulation. This game loads very quickly and is one in a series of flight simulations to come for the VIC. I look forward to purchasing the next one out.

P.W.W

## Star Commando

★ ★  
 Terminal Software  
 £7.95  
 £8M 64

STAR COMMANDO IS one of those games that looks like it will be an instant winner. But when I actually started playing there was something about it that was not quite right. I couldn't put my finger on it at first. Then it hit me slap bang between the eyes. The thing you'd expect to be able to control, the gunight, was rigid and unmovable and, instead, the joystick controlled the background. That was it. The game was ruined.

Nevertheless, onward in the face of adversity. As the noble in the star fleet your mission is to repel the cosmic invasion force consisting of the Galactic Pirates and Sestonic Raiders

and clear the space lanes of mines. Every sector of the galaxy has to be searched and cleared. The only problem is that your energy level is decreasing all the time and you need to find your mother ship to re-energise.

You do have an advantage, however, a long range scanner which can be consulted in any clear sector. The only problem is that the information is never complete.

Points are scored for each sector cleared and each invader blasted. The more points you score the higher your ranking at the end of the game — Captain, Commodore or Star Commando only skill dictates but the last does give you access to the secret code enabling you to claim your wings from Terminal.

Overall impressions: well nice concept, shame about the gunight. K.M.

## Cavelon

★ ★ ★  
 Ocean Software  
 £5.95  
 £8M 64

WHO SAID THE AGE OF chivalry is dead! Who wouldn't come to the rescue of a damsel in distress, answering her piteous cries for help? So you lady your chariot do you! Then enter the castle of Cavelon. The only problem is that it's not the easiest of

tasks climbing your way to the top of the turret.

Castle Cavelon is a stronghold guarded by an army of knights and archers who wander the stone-walled corridors eager to deter any intruders. To reach the top of the castle turret you have to ascend six levels of stairs each of which is hidden behind a doorway at the end of a maze of corridors. To open the door you have to have all the pieces of the door in your possession. Search the

corridors well because that is where they are hidden. Also floating around the corridors are a number of swords. Collect them all if you can. Their magic will render you immune for a while from the knights and archers. There's no doubt you'll need them all if you are to claim your prize.

On screen data indicates your points total and the comparative achievements of your rival if you chose the two player option, as well as the pieces of the door you

have collected, the number of lives you have left and how many magical Excalibur swords you have managed to find.

You'll not be surprised to discover that this is yet another variation on the illustrious Pacman theme. However that is not to denigrate what it is. In fact, it holds its own well and is quite an enjoyable game. And anyway, who is their right mind would leave Cavelon crying in vain for assistance. K.M.



**Flight Path 737**  
 ANIKOG  
 £7.95  
 CBM 64 + joystick

THANK GOD THIS ISN'T real. If it was, either the airline would have run out of undamaged passengers by now or my exploits would have been turned into one of those sky high Hollywood disaster movies. Needless to say this is not

one of the easiest games around but it has to be one of the more addictive. There's no giving up on it until what at first seems like the impossible has been achieved, even if it is only getting the jet safely off the runway and into the air without smashing the flags to smithereens or breaking the undercarriage off.

The trouble is, that is only the beginning. Even in the easiest level there's the mountain to get over and then there's the problem of getting the plane down

again safely on the runway without trying to tunnel beneath it. Like any flight simulator, Flight Path 737 requires a high degree of eye, hand, memory and general brain co-ordination.

As a representation of an aircraft flight deck, the screen is crammed full of flight information, heading, speed, altitude, fuel, flaps and under-carriage status and distance from home — including a view from the cockpit window.

Flying the 737 requires a combination of joystick and keyboard control plus a propensity not to panic, but in case, by some fluke chance, you manage to software spotlight 8 cope with your first solo flight, there are five other skill levels to tackle, all of which progressively shortens the runway and increase the height of the mountains, whilst throwing in the odd on-board fire and last minute crosswinds on

landing. Getting your wings won't be easy, so whatever you do, don't panic.

If I had to criticise the game at all, and I feel really reluctant to do so, it would be on the basis that there is a tendency to master each level a stage at a time. If it's the landing you're always having trouble with, getting to that stage can be a bit of an unnecessary obstacle. The addition of a practice level would be a welcome solution.

R.M.

# SOFTWARE SPOTLIGHT



**Mr. Mephisto**  
 ★★★★★  
 EURO-BYTE  
 CBM 64 + joystick

TO BE HONEST, I DIDN'T know what to expect. Once loaded, using Euro-Byte's fast loader system called 'Overdrive', I was confronted by a cinema front with a 'Now Showing' sign including a piece of music the same of which is on the tip of my tongue! Very nice.

Now the game... Dave Lucas who is the Author of the game, even went to the extent of composing a poem in which there are clues to guide Hugo (that's you) through a tortuous, frustrating graphic arcade type adventure. I spent the first half hour, with beads of sweat on my forehead,

trying to get past the first level!

The object of Level 1 is to get up the down escalator whilst trying to avoid the demons coming down from the top of the screen. You have to get through the door and that is not easy. I had to read the second verse for this screen (no more clues!)

Second level is a little more difficult, with yet more down escalators to go up! This time you have got three exits to go through. I went through the wrong one and had to start again... on Level 2!

It's a must for any computer game enthusiast who thinks he or she can play games well. It is well worth the money and you will be writing to Euro-Byte for more clues when you can't figure it out. S.L.P.P.

**Marooned**  
 ★★★★★  
 Bantam  
 VIC 20 + 128 RAM

THIS IS A 128 TEXT adventure. You have crash landed on the planet Mars and the only way to get home is to find a missing part of your space ship.

The game is written in BASIC and with ingenuity you can break in and cheat. Even though written in BASIC, responses are fast and this is not a problem. The instructions are brief and typed into the insert. There is no sound except for the introductory text. The text is black on white and in the cases, white on black. The program gives a description of the location including obvious exits and

objects near by. Commands are given by usual verb noun combination although you can use one letter entry for N.S.I.W.L. and D. Pressing R gives a description of the situation that you are in. Pressing I gives you an inventory of the things you have, the game loads easily with a very snappy time on the title page. The game itself is difficult but that is partly due to limited vocabulary and BASIC language analysis. The problems themselves range from easy to difficult. There is no save to tape option like on many VIC ventures.

There is a sense of humour. I found this adventure addictive. It caused me sleepless nights and many cups of black coffee.

P.W.W.



## Bongo

★ ★ ★ ★  
 Aiming  
 £7.95  
 VIC 20 + ME RAM + joystick

A NEW GAME FROM Aiming, and one of the best I have seen. You are Bongo the super mouse, always wanted to be a mouse. You have to rescue the princess and win her heart. It is definitely changing into a mouse. The only way to win the princess is to find the king's lost diamonds. Bongo sets out and eventually arrives at the thieves' hideout on the riverbank. This is where you take over. You control Bongo with the joystick which is very sensitive to the slightest touch. The game uses the whole of the screen; the graphics are very large and superbly designed. The

game itself is in six parts. The layout is Kong/Mario/Miller type; in other words, walkways, ladders and trampolines and an added bonus of a slide feature which only you can go down jobiously the robbers suffer from vertigo). There is a two-player option and three levels. There are six screens, which makes it addictive trying to get to the next screen. The sound is very good with a catch tune at the beginning.

It does take a while to load the game but it is well worth waiting for.

All in all an excellent game and should remain a firm family favourite.

P.W.W.



them on the battlefields which are guarding the next chamber. Diggy can only pick up the crystals when they are in the harmless ones. If Diggy picks up a crystal before it is harmless then some of the shield power is used up. The shield power is always decreasing because of the aliens bumping into you.

If you reach zero energy before leaving the chamber, the game ends with you exploding and, assuming you score more than 30,000 points, your score is given a code which can then be sent into the software company. If your score is high enough, they will put you on their 1,000 top scores list published every 6 months.

Generally, it is a very good game with reasonable sound effects.

S.L.F.P.

## Multi Tron

★ ★ ★ ★  
 Simlock  
 £2.95  
 VIC 20 + Optional joystick

ANOTHER SHOOT'EM UP from Simlock. A superb game which has several screens and is very fast. Your task is to penetrate the enemy defences and destroy all aliens at each stage. The first screen takes you onto the main area; these you can shoot. The next screen is the warp stinger; these are very clever as they wait for your bullets to go past until they move. After this are the star-hoppers; these move diagonally down the screen and bounce off any obstruction except your bullets and you. The fourth screen are the missiles which come darting down until you hit them. This is fairly easy as you can send up showers of bullets.

The penultimate screen is by far the most difficult. Space turtles hide in their shells until, for a brief moment, they come out and you get the opportunity to shoot at them. At last you reach the last screen. The Cosmic phoenix's stand still and shower bullets down. The graphics on this screen are superb as the birds flap their wings. You are left with one final task which is to travel through the space corridor. This is easy and if you have managed to get this far you are granted a bonus ship. Back to screen one you are taken with an army of more furious aliens.

This really is a superb game and credit must be given to the writer's ability to fit it all into the unexpanded VIC.

P.W.W.



## Plumb Crazy

★ ★ ★ ★  
 Terminal Software  
 £8.94  
 £7.95

GOT A BIT OF A DOGGY boiler have you? Fancy doing a bit of plumbing? Not really? Then why not get old George to do it for you. He's willing, one of the best drip fixes in the business and, what's more, runs around at your beck and call.

With the water temperature in the boiler rising rapidly and threatening to blow a stop cock at any minute, you have to get George to plumb a relief pipe from the valve at the bottom left hand corner of the screen to the boiler in the top right.

Easy? Well not quite. With time gently ticking away, you have to identify all the right bits of pipe at

the bottom of the screen to weave the relief pipe around the boulders indiscriminately littered in your path. Mind you, if you do get stuck there's always that handy piece of dynamite to blast your way through.

But be on your guard. The ghost of George's former employer is on the rampage and dying to get him a bad name in the business by preventing him from completing the relief pipe in time.

Naturally the game gets progressively more difficult to the point where it is practically impossible to see the boiler for the boulders. To pick up your blow torch and get plumbing. It is,

## The Pyramid

★ ★ ★ ★  
 FANTASY SOFTWARE  
 £8.50  
 CBN 64 + joystick (Optional)

THE PYRAMID, LIKE MOST new pieces of software, comes complete with a high-speed loader to enable short waiting times for the player to play the game. Another nice aspect of this game is the abundance of instructions and information included. A high score form can also be found with the literature.

The program itself is about 'Diggy' (that's you), making his way through 120 chambers of weird and wonderful creatures, all of which are dangerous to touch. The way to get through the Chambers is by collecting crystals when they appear, and dropping

# SOFTWARE

# SPOTLIGHT



## Jumpin' Jack

Standard  
C1-35  
VIC 20 + joystick Optional

THIS IS A VERSION OF THE arcade favourite Frogger. If you have not heard of it, the basic aim is to get a frog across a road, watching out for traffic. Once over the road you have to find your way to the lily pond, to get there you must hop onto turtles and logs making sure that you don't jump into the water. The obstacles gradually move faster making it quite fun to play. Alas in this version the graphics are very jerky and the animation leaves a lot to be desired. The choice of keys are good: Forward/Back, Z Left and C Right; this makes it very easy to play.

The joystick movements are also very sensitive.

The game is written in full 6502 machine code. While the game is being played there is a 'tone' which soon becomes irritating. Fortunately you can always turn it down.

Unfortunately there is a 'bug'. It is possible to go off the bottom of the screen and reappear at the top without having to cross over the road or jump onto the logs or turtles — therefore the game is much easier to complete and high levels can be achieved with great ease.

This is a disappointing version of the arcade game but is one of the few versions of Frogger around for the VIC 20. It certainly won't wear out your trigger finger, but it does provide good family entertainment. P.W.W.

## Zodiac A & P SOFTWARE C7-95 CBM 64 - joystick

ANYONE WHO HAS SEEN the game "Shamus" on the Atari will notice more than a passing resemblance with this offering from Amig. As with most other games, there is a disappoive face-on-the-knee. This time they come in the shape of Masters of Black Magic.

The Masters have stolen the twelve signs of the Zodiac and thrown them down in a 400 corridor maze, and it is your job to retrieve them and place the signs in the central room. But it is not as straight forward as that, because you have to fight the demon slaves which inhabit all the corridors. Be careful if you

re-enter a corridor because there will be another set of demons to cope with.

If the demon catches you or you run into a wall, your character is immediately vaporised and all that is left is your hat!

While playing this game I did come across some interesting features. The first was that after I had watched a sign and had run into the next room, I found exactly the same sign again with more monsters! The second was a little bit more disconcerting. As my character ran from one corridor to another through a door, I materialised in a wall and got disintegrated yet again!

It is a good game with very good sprite graphics but it doesn't have that certain little something that makes it truly addictive.

S.L.P.R.

## Chuckie Egg

A & P Software  
C7-96  
CBM 64

WHAT A PLEASANT surprise for Chuckie Egg read Donkey Kong through said. I suppose, like I need to go further! Perhaps I'd better.

You are Hen House Harry and the object of the game is to guide him around the hen house

collecting the dozen eggs that have been laid. Yes, it's always a dozen and some are messily suspended in mid air. There is also a lot of corn to collect and all the while you have to keep old Harry out of the way of the ducklings which are busy gobbling up the corn and chasing Harry intent on pecking him to death. Why ducklings and not chickens I don't know!

Anyway, just like Donkey Kong there are platforms for Harry to walk

along, ladders to climb up and down, and lifts to hitch a ride on providing you can get him to jump accurately.

There is a choice of six game speeds ranging from 'for those who prefer adventures' and 'for the faint hearted' through to 'for the general folk' and 'for the suicidal maniac'. In addition, there are a variety of skill levels to progress through on each speed level but beware, on skill level nine the mother duck is let loose to run riot.

There's nothing at all pretentious about this game. It's a good solid arcade action which follows a tried and tested formula which has proved very successful over the past year or so. But you need to know it for what it is otherwise, with the absence of any real description on the packaging, I suspect many a games addict would be disappointed in having bought something they already own but is branded with a different title. K.A.T.





### Gridtrap

★★  
Live Wire  
\$8.95  
CBM 64

FIRST IMPRESSIONS ARE rarely wrong and in the case of Gridtrap it was pretty good. However, before I get too negative I have to admit that the graphic representations are extremely good indeed. But, having said that, it is the execution of

the game that really counts and that's where Gridtrap falls down miserably.

Despite some lengthy instructions, the game at first seems like a hopelessly overcrowded mess with a hideous musical accompaniment. It takes a while to fully realize what the hell is going on.

The object of the game is to manoeuvre a character called Mr Live Wire from box to box in the grid to reach and diffuse indi-

criminated placed bombs. Points are scored depending on how quickly you reach the bombs and how many bonus scoring flagged squares you take in the way. There are dangers of course—the skull and crossbones are out of bounds and several big bomb skulls around waiting to kick you into touch. Once a square has been used it disappears from the screen although there is a facility to scroll the

remaining squares in a line to help movement around the grid.

After every series of five bombs is defused, a new screen is presented and the game continues at a much greater difficulty level. Obviously the idea is to amass the highest score. At least the two player option brings an element of human challenge into the game which suffers dreadfully from trying to be too clever by half. **K.M.**

### Bathtime

★★  
PSS  
£7.95  
CBM 64 - joystick (1 or 2)

THERE ARE GOOD SILLY games and bad silly games. The trouble is that I can't decide which this one is. It is very subjective as to whether you like violent or non-violent games. Bath-time is a non-violent game, which has reasonably good graphics and sound.

The main feature is its large use of sprite graphics including an elephant, a boy in swimming trunks, two heavenly cherubs, a goldfish and a swan. The game itself has an interesting concept, as follows: The fish and swan are having a bath and controlling the flow of water are the two cherubs; they, being you and another player, or the computer. It appears that

the two cherubs have had a little argument and are dead set on making life difficult for each other.

Player one in the two player game runs the water into the bath while player two runs the water out. Player one has to either let the bath flood so that the swan sets away or let the bath run dry and kill the fish. Player two has to prevent this by monitoring the water flow. There is a time limit on each game played so as to let player two win if both swan and fish are still present.

To make things a little more difficult there is an elephant which keeps on appearing wanting a drink, and a boy replenishing from a bucket of water. The towers and raises the water level respectively.

A good game for the very young but not for the space insider enthusiast.

S.L.P.

### Cybotron

★★★  
Ardisk Software  
£7.95  
CBM 64 (1 or 2 joystick)

SOME TIME IN THE FUTURE robots will become endowed with greater self-determination and will decide that humans are an unnecessary evil — exterminate the warm blooded ones... Fortunately Cybotron — a somewhat up-rated human — can save the human race... so can he [you]?

There are several games with a similar theme on the market and although I can only get to level 7 with difficulty (there are 1001 there is the feeling of a reasonably balanced game — that with a little more luck [practice], I could get further. Only joysticks may be used — no keyboard —

but there is the option to use two... one for movement and one for firing at the 'baddies'. Who says computing is a solitary pastime! If you have time to admire the graphics you will find there are 4 different robots, 5 types of electrode (stationary objects that you can destroy but must not touch), the odd human to rescue and of course your 'alter-ego'. Cybotron, 64 scores are recorded and you have 3 lives to gamble with. Bonus waves, where there are a higher ratio of humans (to rescue) to robots will help boost your score — and to rest your shattered nerves, there is a pause option. Cybotron has 'Turbo Load' to cut down on loading time... and this worked perfectly every time. A game for the 'hard and shoot' brigade with quick reflexes — well balanced and worth a try... **P.F.**

# SOFTWARE SPOTLIGHT

Dr. Watson, computer learning series — basic adventure part 1  
 CJanop Publishers Ltd  
 ISBN 90782 76 2  
 £3.95

THE "SLEEVE" ON THIS product has several — let us be generous — ambiguous statements. Let us look at those and try and decide what we have in our hands.

First: "The revolutionary new concept in computer learning provides a FUN way for children to step into the 31st century world of BASIC programming..." So far, so good — the book does present a novel teaching concept. Programs and exercises are introduced in a storybook manner that might well catch a child's imagination.

Second: "Part One of the book is an exciting adventure... Your heroes must learn to operate the ship's computer if they are

to escape!" We now come to an ambiguous bit! The word adventure to most computer owners implies that the user has to solve an adventure — not so, the book works its way carefully through examples of PROMPT, INPUT and String handling and in the last two pages of the story our intrepid Dr Watson jury rigs a spaceship's controls — with no reference to our previous examples of computer programming — and sets it in motion. Not quite what we expected!

Third: "This unique adventure is specially designed to teach the fundamentals of 64 BASIC by way of example — the way children learn best." Well...perhaps — but for the use of that word "adventure" again.

Fourth: "Accompanying this is a tape containing the same programs as are on the spaceship's computer, and some 80WILL's teaching

programs so your children can learn as they play."

CLICHE! Yes there are three simple programs that are referred to in the book — very short but yes in the book! We then have two copies of a "Hangman" program — a little odd but it maintains that one can have a lot of fun with these sort of guessing games — BUT with a total vocabulary of four (4) words! AND one of those is GAMB — not exactly inspiring, it also makes one wonder about who is kidding where at the price of £3.95.

Having turned the tape over and LOADED the first program, we are presented with a menu. This lists the programs on this side of the tape and on selecting option 'x' you are given a very brief resume of what you will learn... then there is a "Message from Dr Watson" — this turns out to be pages of errors from the book! With no package there was

no indication that this was there — not even a cheap bit of paper...

The teaching programs request the user to type in the correct format for PRINTING various examples — followed by similar exercises on String handling and use of Line Numbers — quite a nice idea and well thought out...except...with a teaching program such as this one does not expect to be able to "crash" the program if given a wrong answer — OR, I tried to do it but....

Fifth: "Every BASIC command covered is also given a separate careful explanation in the second part of the book..." Yes it is...except for the errors mentioned above, it's possible.

In conclusion, all I can say is... wait for Mr Holmes's book — he always knew better than Dr Watson!

P.J.

Booga-Boo  
 Quackolite  
 £1.95  
 ISBN 84 (Joystick only)

THIS MUST BE ONE OF THE most frustrating games I have ever played for a long time! You are a flea (!) trying to get out of a cavern by jumping from ledge to ledge, to reach the exit above you. The "power" of your jump is controlled by how long you hold the

joystick left (or right) before releasing the stick — then you jump. It takes a little while to get the hang of judging how much power to put into a jump, in order not to overshoot or undershoot your target, just to add to the excitement — there is a large dragon roving around the cavern, ready and willing to gobble you up! You have only one life and having lost that I did, with monotonous regularity, you have to wait through the 78 seconds

"intro" as Booga-boo falls into the cavern once again.

Time elapsed, Level and Hi Score are indicated at the bottom of the screen. Level is the height you are at, above the floor of the cavern. Hi Score does not indicate until you have escaped the cavern, so no matter how long you manage to evade the dragon, or how high you climb, you have no personal best score to strive for in the early stages... it could have done with a little something

to spur me on after being gobbled up for the fifth time! Graphics are superb, I just wish it was a bit easier at the start, to help inspire me to greater heights! There is no pause option — but then if you don't last long enough to achieve a Hi Score, you can always drink your coffee in the 78 second wait following death!

P.J.



### Typing Wizard

\*\*\*  
Severe Software  
CBM 64

WHAT A SMASHING TITLE page — I always did like watching wizards throwing lightning bolts...out of the field of fire, naturally! What a pity,

one of the menu options is not 'wizard' itself. This program sets out to teach you touch-typing — a very commendable project, especially as I don't! I would guess it was written using a monitor, as the choice of colours for some instructions was orange letters on a medium-grey background — not the easiest to read, even if you turn the colour off at

the TV. Instructions shown like this are few and anyway you want to get on with your typing so perhaps lessons may be forgiven — a thought for the future could well be user definable colours!

Menu driven, you are offered nine options — 1 is the Introduction, from that on you are told how you should place your fingers

and thumb for different rows on the keyboard — starting with the 'home' row ASDF.

This sort of program is really only as good as your will to learn! — if you have that will, Typing Wizard can only help.

S.L.P.

### Space Pilot

\*\*\*  
ANIBOG SOFTWARE  
127/95  
CBM 64 - joystick (optional)

PACKAGED IN THE NOW standard library case, Anibog's "Space Pilot" comes with a picture showing aircraft of different eras fighting each other.

In this game you are an ordinary 1960 Jet Pilot. The very fact that you are from that particular time in aviation development is a definite help on the first 3 screens. In the first screen

you are faced with a 58 strong squadron of bi-plane fighters (the year being 1918). Easy you think, and easy it is, for these pilots do not stand a chance against your gleaming 1960 fighter. All that is left to do on that screen is to tackle a single Zeppelin which comes floating across your screen, but be careful because it can release a very little salvo of bullets when you are not expecting it. Once you have cleared the screen, you are transported to another time — 1940. Again, you must dispatch a large number of aircraft, but they are getting a bit more nasty.

Clear that screen and

you are launched into 1970 where you meet helicopters. This is the turning point of the game. As you clear that screen you are transported into 1980! The aircraft are more advanced than yours, so you have to play more tactically instead of just flying into them and shooting.

Finally, if you clear that screen you meet from the year 2001, Flying Saucers! Unfortunately I didn't clear that screen but I did valiantly!

This game in general has good graphics using sprites; the sound has a few effects but not too much to write home about. With the time



limits and other obstacles like the clouds it can lead to very interesting suicide runs.

S.L.P.

### Splat

\*\*\*  
Innovative Software  
CBM 64  
12/99

THIS IS A GAME FOR ALL potential members of the Strawberry Jam Preservation Society. The object of Splat, among other things, is to prevent the hero, Zippy, from being thoroughly pulverised or, more colloquially speaking, turned into strawberry jam.

To all intents and purposes, Splat is a variation on the age old theme of the maze game. Comparisons can be drawn with Pacman, the illustrious ancestor of all maze games, although it can

be a bit misleading. However, if the truth be known, it is just as addictive.

The main difference between Splat and other maze games is that the main character, Zippy which you control, is not being chased by an enemy, at least none that I've managed to discover. Zippy has plenty of freedom of movement within the boundaries of the maze which is much larger than the visible playing area and slides around unpredictably intent on splattering Zippy.

The purpose of the game is to explore the maze, scoring points by gobbling up grass, plums and invisible fingerweeps which have a habit of turning the tables on Zippy. There are seven skill levels in all and you have to race against the

timer keeping Zippy out of trouble long enough to reach the next level.

The main obstacle to avoid are the spikes which appear for the first time in level two, the water be-

cause Zippy can't swim and the sides of the playing area. There are also things called alternative Zippies which could be good, bad or just plain ugly but my failure to progress beyond level five has failed to reveal them.

The lack of any useful instructions doesn't help matters much either.

The great thing about Splat is that reasonable skill levels are quickly achieved and it has a perfect frustration factor which makes you keep plugging away at the high score.

One thing that might help, but I'm not quite sure should exist, is slow motion movement when the pause facility is on. Not only will it help calm your nerves but can also help you get through the difficult bits.

K.M.



Fancy having a go at  
writing your own  
arcade type game?  
William Fong has tried  
to give you some  
guidelines to start you  
on the way.

# YOUR OWN ARCADE GAME



HOW MANY TIMES HAVE you played a new computer game and after minutes (or sometimes hours!) of frustrated playing you are sure that you could have written a far better game yourself? It is really not as difficult as you might think provided you approach the task in the correct fashion — and you can get a great deal of enjoyment from the program writing itself, let alone when you can happily play with the finished product!

## The first move

Naturally your first move is to think of the basic idea for the game. Of course the idea should be an original one — try to break away from the same old mould of alien-zapping, maze-wandering and monster-killing! No doubt this is the most difficult part of writing a good game.

Now don't immediately rush off to your computer — well not yet anyway! Transfer your idea to paper — there's nothing worse than spending hours at the keyboard only to find that the game takes at a crucial time — this is a recipe to make you give up on the idea before you've given yourself a fighting chance. Think everything out before you get too carried away: what sort of characters will be involved? Are you going to use graphics and sound in a particularly unusual way? What will signify the ending of the game? Do you want a two-player facility? These and so many other factors will be vitally important to establish before you go

headlong into typing actual code.

## Throwing down the gauntlet

Bear in mind that the game should be challenging to be even the most skilful player. This does not mean that the game should be impossible to beat — people will easily lose interest in a game where they haven't got even

the faintest chance of winning! A way of getting round this problem could be the use of different levels of difficulty. Above all keep the concept simple — too many rules can be confusing, and they should remain the same throughout the game.

Most people prefer a joystick-operated game, but it is wise to keep all your options open. If you have

not got a joystick and are forced to use the keyboard, please give some thought to the choice of control keys so that the player does not have to perform super-human digital antics in order to fire a missile!

## On screen action

Your own arcade game 1. Having got your ideas and plans on paper, you can





touching the computer. A lot of thought is needed to make the game interesting and addictive — this is often the stage at which programmers don't apply too much care and the small but important touches are left out. In good games the screen is always changing and the graphics are usually moving. At this stage you can let your imagination run riot when thinking of the representation of the main character in the game and here you can use the IBM 48's graphics abilities to the full. But don't worry if you find it difficult — there are many graphics design packages around for the IBM 48.

Colour use is also very important and is probably best added after everything else has been completed. Generally colour should be used to minimise the player's frustration; critical

features and anything that's urgent should stand out clearly, whereas features that are not viewed often should be lightly coloured. Be careful with the colours of adjacent characters as the eye is easily fooled by contrast; try also to avoid filling the screen outside the playing area as this can be distracting and confusing. It is also often worth the trouble to try playing the game on a mono screen — different colours do not always distinguish themselves on a black and white TV.

### Listening in

Sound effects should be considered carefully; certainly with the 48's complicated SID chip excellent sound effects may be generated. Good use of sound can enhance a game and turn it into something

really special, but it is perhaps one of the most difficult things to perfect. It must suggest impending action to the beginner but not antagonise the advanced player; it is worth leaving in the option of turning the sound off. Trial and error is the only way to perfect different sounds for each character or piece of action — a long, low frequency sound suggests slow movement but you need a short distinct sound for more immediate action.

Try to make your sounds as pleasing as possible and try to avoid shifting from low to high frequencies too abruptly. If you are a little more expert in the musical field you could have a go at writing a little tune to go at the beginning of the game or running throughout it. Don't forget though that a catchy little tune heard

redundant times while you're trying to beat the computer can sometimes make you want to beat the computer instead!

### Other factors

Don't forget that throughout the writing of the wonderful game you will need to keep track of where the characters are at any one time — was the alien destroyed and therefore now out of the game, or have you merely blasted it into another position in the maze? There's surely nothing worse than having expertly destroyed a submarine only to find that minutes later it has mysteriously come back to life and suddenly blown your ship to smithereens!

A scoring strategy is purely a matter of personal preference, although it certainly appears that many people like a high-score record facility. You must also decide whether to have a time limit on the game — will you run out of fuel, missiles or oxygen?

### Game point

Well, that's all there is to it! I know it seems a totally daunting task, but rest assured (!) that the hardest game you will ever have to write will undoubtedly be your first. As you go through the routine for the first time you will gradually learn only little momentary routines, exciting explosion effects and more, and often you will feel that they fit in better to your Mark III version of the game which has all sorts of refinements added to the original. Throughout the process of writing your own game you must continually test each new stage you add on — who knows how that new routine will affect the part you have already perfected! It is also advisable to always keep back-up copies for those odd occasions when the whole thing crashes — you may not always have the stamina or memory to go back and start at the beginning again!



# STUCK?

Try a Course prescribed by Dr Watson

DR  
WATSON  
series

## BASIC Courses In Space For The Commodore 64 & VIC 20

An exciting course for all children from 7 to 77. Learn BASIC while enjoying this most exciting space adventure. Master the computer to escape the Alien Spaceship.

It combines a story which is compulsive reading for the young reader with a carefully structured approach to BASIC intricately interwoven into the text. Also, each BASIC command covered is given a separate, careful explanation in a special 'easy reference' section in the second half of the book.

The tape includes the programs on the ship's computer, and four computer aided learning (CAL) programs which teach the major BASIC commands.

"It should hold the interest of an under-12 year old."

- Home Computing Weekly



BOOK & TAPE

£9.95

## Beginner's BASIC For The Commodore 64

This project-based course guides you step-by-step through BASIC, developing games and utilities using progressively more complex BASIC commands. All the programs developed are provided on tape so that you can get a taste of them first, before working along with the detailed explanations given in the book.

Programs include three video ball games, a character generator utility, a sprite generator utility and a 'composition' music utility. You also get 'Honey-Aid!', a machine-code utility that adds 28 commands to Commodore 64 BASIC allowing easier programming, sound and graphics.



BOOK & TAPE

£10.50

"The Dr Watson series is rapidly becoming recognised as quality material."

"... a well thought out material... if this was the total content of the course, it would prove excellent value. However, there is more... much more."

"... the course consistently encourages you to learn more and more by making learning fun."

"When... manufacturers have been shown how an introductory manual should be written."

- Commodore Computing International

## Beginner's Assembly Language Courses For The Commodore 64, VIC 20 And PET

These courses introduce the real beginner to assembly language programming. No prior knowledge of assembly language is assumed and the aim is to ensure that every reader succeeds. Numerous examples illustrate the points while exercises along with solutions test the understanding.

The tape includes an assembler which assembles code anywhere in memory. Also on the tape is a binary/BOD/hexadecimal CAL tutor program which teaches about the various mathematical notations used in machine code programming.



BOOK & TAPE

£12.50

"There may be other ways of adding one and two together... But there isn't another way of learning how your computer does it."

- ICM

"The self-explaining VIC 20 user interested in programming should read this instruction set."

"The manual would be worth buying for the reference section alone."

"Don't miss it!"

- Mark How

"I have made more progress in a week than normally around for a year with other books and articles."

"A really super clear book..."

- The VIC 20 Month London User Group

"... this book is worth its weight in gold."

"If there was ever a good beginner's guide in this field, then this is it."

- Commodore Computing International

HONEYFOLD

HONEYFOLD SOFTWARE LTD

31 Standfoot House  
Bath Place  
High Street, Barnet  
London EN5 1ED  
Tel: 00-441 4128



Take to the air and fly  
with us in Concord.  
Fasten your seat belts  
tightly though and see  
if you can land the  
aircraft in William  
Long's simulation for  
the 64

# CONCORDE II



FOR MANY YEARS THE French and British always have been discussing how to speed up travel and at last they have designed a new Concorde: Concorde II. This should be far better than the original Concorde as it travels much faster and is more economical to run. Being a 'professional' pilot yourself you have been given the chance to have the first test flight. Your route is to Glasgow from Heathrow. The journey is not as easy as it seems as there are quite a few instruments to watch and the level you pick will have to depend on the weather conditions you will encounter. The flags, undercarriage, speed, altitude, deviation readout and artificial horizon are all important instruments to watch.

## How to fly

When you take off make sure your flags are down or the plane will not go up! When your flags are down your speed must not exceed 280 mph or you will damage your flags. Your undercarriage (jetties) must be taken up before the speed reaches 400mph or they will be damaged. However, the undercarriage must be down before the aircraft is below 500ft. If your flags are up the plane speed must not drop below 270mph.

The distance between the two airports is about 380 miles and because this is a real-time based simulation it will take a long time if you keep the speed at 500mph.

The aim of this test flight is to see how fast you can get to the other airport: speed approx 1700 mph.

The artificial horizon is an important instrument and it comes in very handy when visibility drops to zero. The deviation readout should be updated often as the pointer is in line with the approaching runway, so try and keep the pointer near zero. The map readout on the British Isles is useful as it indicates how far you are from Glasgow (if you know where Glasgow is).

When resetting the runway or any other unknown information the control tower will report to you. When going in to land the speed must be about 280 mph and the altitude less

than 50ft. The flags must be down and the undercarriage must also be down. When you are above the runway drop your speed to 180mph and altitude to about 20ft then you will get a perfect landing.

## Controls:

To turn push joystick in port two and turn as normal.

To increase altitude pull the joystick back.

To nose dive push the joystick forward and the plane will dive down.

To lower your undercarriage use function key J

To raise your undercarriage use F1

To lower flaps use F 5

To raise flaps use F 7

To drop altitude without the

plane diving use "A"  
To drop speed use "Y"  
To increase speed use the fire button on the joystick.

## Watch out

The runway is only 1.1 miles long so make sure you take off before you reach the end of the runway. When you see the other runway drop your altitude to about 30ft or you will fly over it and all effort would have been wasted.

The game may sound easy to complete but wait until you are in the cockpit. There are levels you choose to increase the difficulty, storms, engine failures, strong winds, and many other conditions. Good luck, and happy landings!

# CONCORDE II







Toiling deep within  
his murky cells,  
Runecaster opens the  
portal to Other  
Worlds . . .

# TALES FROM THE CRYPT

WHAT IS AN ADVENTURE program? Today we look on adventures as part of our everyday (computer) lives. They come in all shapes and sizes for all the different computers we can buy — where did they start? Many of us today, especially those reading this, own a personal home computer. Just a few years ago they did not exist — hard to believe, isn't it? Five or six years ago, the options were severely limited — PET, TRS-80s, build-your-own, a memory capacity standard of about 64K, some 16K or 32K perhaps, if you were rich. Those were the days when computers and computer talk were still the closely-guarded "hi-tech" secret of just a few people who would go to almost any length to protect, not knowledge, but bafflement and confusion to those who were not "signed on the line" members of that close-knit club.

## Gone, but . . .

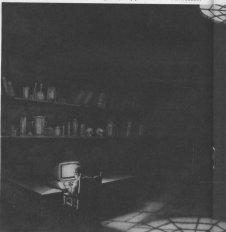
Fortunately those days are past, not long in time, but dead, dead, dead. The home computer has come of age — young but strong enough to walk on to over. But . . . one thing those maintenance freaks did have that we can thank them for being unable to keep to themselves was adventure, or more properly, Adventure. For, among the dim corridors of libraries, medical statistics, research into flu and that, was to be found (often illicitly), Adventure. The time was just right for its birth: Dungeons and Dragons was already covering more and more of its banner — a game (or way of life) where you, ordinary mortal, could enter a world so different yet so real, that your life could depend on you knowing the right spell or your being adept at

hearing the very fabric of a simple chest whispering to you that it contained a dozen poison arrows, primed and ready for action.

Into this environment came Adventure, written (I believe) by Messrs Crowther and Woods around 1976. What was it? Adventure was a new sort of game (like)

where you could enter an amazing world via your computer terminal — many of them did not even have a VDU. So that meant handling over some sort of "teletype" to see the answer to your simple instruction of GO SOUTH. This "world" within the computer was like no board game where you can see quite clearly

where you will "land" if you throw a 'W'. All you knew was the direction in which you could move — sometimes not even that! What was in that direction, you knew not. You had to explore your surroundings and gradually build up a map of them. I'm sure that one of the most important attractions is this freedom



to move where you choose, not bounded by the linear 'wall' imposed by the conventional game.

Back to Adventure. There were many passages to explore, logical puzzles to solve and treasure to find. Adventure taught us like no other game before or since. Research scientists and computer operators, designers and programmers spent (wasted?) hours trying to get past the Troll (unlike many of the successful arcade games which fade into history after months, Adventure has stood the test of time. Many programmers have tried to copy or improve upon it—a few have succeeded, but only a few.

We have moved to an age that expects pretty

pictures with our adventures; sometimes it is well-done and the pictures play a part in the adventure. Mostly the graphics that some reviewers rave about are only there to enable a product to be 'sold' to the public—some distributors (and much of what is in your local store is determined by the distributors) will not even take adventures unless they have 'those magic, 'willing' graphics!

Graphics take up memory that could otherwise be used to make the programs more interesting and present you with more of a challenge—significantly setting the scene to make you feel you are really there! The written word is very powerful, not just by

itself but in combination with your imagination. Mind you, you either have that sort of mind—or you don't! Many people (poor souls) cannot 'get into' Lord of the Rings by J. R. R. Tolkien; those that can are able to populate that world with beings so complex as to be real. The same comments may be made of Larry Niven's science fiction. How many of us have been disappointed at seeing an author's attempt to reproduce a scene from a book we love? And an author has at his disposal a far, far greater resolution than even the best monitor (with a BBC Model B in Mode 8 can even hope to attain. Graphics are fine but cannot for the foreseeable future replace the written word.

Adventure used the written word as an author does: descriptions of places and things were full and rich, not short-form 'you are in a cave NGE'. As the original was mainframe-based with megabytes of core store, it did take some time before it appeared on home computers (but one way and another, with the sophistication of text compression and the dedication of the programmers, you too can own a version of the first of its type, whatever it may be called! Versions have been around for the Spectrum and Macrom for some years. Now we also have them for the BBC and of course our special interest, the C64 64.

Two versions are produced by software houses that have a good reputation, so it is very much a question of 'you pay your money and take your choice'. The two programs are Colossal Adventure by Level 9 Computing, and Classic Adventure from Melbourne House. Both are based heavily on the original but in the case of Colossal Adventure, the number of locations has been increased by 70 and a completely new end-game has been added! Classic Adventure has a fast LOADING system (Parlopa) which cuts LOADING time to three minutes; on the review sample this

only worked one time in four, so both DO take some time to LOAD! Level 9 are hoping to have a fast LOADING version later in the year.

Both are not only—but what a great level 9 seems to have someone with a really imaginative mind writing their scripts! Even so, Classic Adventure is far more descriptive than the average run of text adventures. 'Classic' follows the mainframe more accurately when it comes to the mazes. Whereas I can refer to my much-copied, faded, almost unreadable map, culled from a maintenance version and escape (cheat!) easily from 'Classic', I have to actually work at it in 'Colossal'.

'Classic' has the odd spelling mistake and a few anomalies: UNLOCK GRATE—OK; OPEN GRATE—THINK IS NOTHING HERE WITH A LOCK. This is a little but does not spoil the overall adventure. GET 'A' AND 'B'—OK, but only 'A' is actually taken—neither game allows these multiple statements.

'Colossal' has screen colour that do not irritate my television but nevertheless are quite readable.

### Choose one. . .

Whichever you choose, do get one of them, so much has been written over the years about Adventure that although you may not realise it, you will have subconsciously picked up some threads of the plot. When you start playing you may well occasionally get the feeling that you have been here before.

Adventure games give you the opportunity of exploring another world, place or time. Part of their attraction is this feeling of freedom of decision amplified by our own imaginations. It is said that one picture is worth a thousand words, but when that one picture appears within the framework of an adventure game, it is my opinion that it destroys the one thing that adventure games give us—the total freedom for my imagination to run wild!



**Modems seem to be the 'in thing' these days for computer freaks, but how many people know what they can do? Simon Rockman introduces them and their uses to Commodore owners.**

A MODEM IS ONE OF THE most exciting black boxes that can be hooked up to a computer. Think of it as a telephone for your computer. The device plugs into the back of a computer using either the edge connector or the cartridge port. On the back of the modem is a lead with the new style telephone jack. This plugs into the BT approved hole in the wall just like a Mickey Mouse telephone. From then on the world is your oyster, but as with a telephone it is no use if you've got no-one to ring. There are quite a few major services to dial into.

#### Main contenders

The major database used by home computer owners is PRISTEL. This covers the whole country, but most people only have to make a local call to log in. To use PRISTEL you have to pay a standing charge of at least five pounds a quarter and this allows you to log into the general areas of the system. You are given a ten digit ID and a four digit password, the ten digits are fixed and can only be changed by PRISTEL themselves, the second one is like a combination lock and can be changed by the user. Both of these numbers should be kept secret and for added security the second one should be changed regularly. Prestel provides most of the information found in a general magazine — news, sports and holiday information on a specific topic than

# INTRODUCING MODEMS

the system provides it may be possible to join a Closed User Group (CUG). These usually cost extra to join and are particularly popular with travel agents, farmers and retiree owners. The section aimed at micro owners is called "PRISTEL Microcomputing" and incorporates the association of computer clubs, Viewfax and Micronet.

#### Micronet

Micronet is run like a daily newspaper, it is kept up to date with news daily, often being the best way to keep track of what is happening in the world of computers (apart from buying magazines produced by yours truly SD). There are special technical enquiry facilities and programs to download, some of them quite good, such as "The Hobbit". At the moment there is very little specifically for the Commodore 64 user. However, with the increasing number of 64 modems available, the force of Commodore owners on Prestel should soon begin to challenge the Spectrum and BBC strongholds. The Micronet letters page is very addictive; Prestel is a two-way system and this makes the most of that. The micromouse section of Micronet is a kind of gossip column, spouting rumour and statistics. Viewfax is a less formal version of Micronet offering many similar services but with a flavour of its own. Access to Prestel Microcomputing costs an extra eight pounds a quarter. One way an information provider (IP) can charge you money for information is by putting a

price on a page. When you look at the charged page the amount in the top right hand corner is added to your bill. This is the way a lot of IP's make their money.

On the whole, Prestel Microcomputing can be used very cheaply — the average bill for Prestel, including the Microcomputing section plus the odd





page charge is around £15 a quarter, about the same sort of price as a daily newspaper.

## Into business

There are many business-oriented networks; most of them require an eighty column display and so are not really suitable. They also tend to be rather expensive. The main system, which will be available to Commodore 64 owners is British Telecom Gold. This now to fame when it was broke into on the BBC television program-

me, but claims to be much safer now. Gold is a messaging system; it can be used to send long letters and programs to other users; you can store much information as you like but pay for the amount of memory this takes up. The system is only really useful if you log in regularly or want to send information that is difficult to read over the phone, like legal documents or programs.

## Latest bulletin

You don't have to have a huge mainframe to run a database, there are many

micros doing a similar job. These are called bulletin boards (BBs). They usually run on a Tandy computer and are free. Most BBs use a slow rate of data transfer and are not compatible with the system used by Prestel; this means that only modems which are capable of changing the speed at which they operate (known as a baud rate) can talk to these systems. With the increasing popularity of Prestel more and more BBs are switching to the system that Prestel uses. The advantage of a BB is that whatever you send in is put up instantly, everyone is an information provider. The disadvantage is that only one person can use the system at a time. The popular BBs can be very difficult to get through to, they are usually engaged. BBs are only local and it is necessary to dial the site where the computer is set up.

## Getting switched on

Bigger computers usually go through a switching system; the main one is PS5. This covers most of the country and is a kind of motorway for data: you dial a local number on the grid (a node), enter an identification number, a password and who you want to talk to and the system gets you through. This service is pretty cheap, data is charged for per packet (a number of bytes) and is much more economical than calling direct long distance, especially when dealing with other countries. PS5 requires the person you want to talk to to also be on the system.

## CompuNet

The most exciting database from the Commodore 64 owner's point of view is 'CompuNet' specially set up by Commodore in conjunction with a company called ADP. To use CompuNet it is necessary to buy Commodore's own modems.

Commodore have been spectacularly successful in

the US with their two modems for the VIC and 64. They hope to repeat this in Europe by turning the telephone into the computer's most useful peripheral. There are major differences between the American standards for modems and European ones so it was decided to build a new one from scratch in the UK. The job of designing the hardware and writing the Viewdata software was given to 92 in Watford. This has been around for a fair while now and has an established user base. With the modem comes special software to allow you to use CompuNet.

When you log into CompuNet for the first time all the extra software that you need is sent to you by the system. This includes a routine to allow you to talk to other users direct. CompuNet is designed as a cross between Prestel and a BB. All users are information providers but they can charge for the information they provide. There is a system of menus and you have the ability to communicate directly with other users and there are plans for a multi-player game like the American 'Mega wars' or Data 'MUD' space and adventure games which allow you to play against other players in real time. Each Commodore modem is unique; it contains a code number (like the registration on a car) which tells the system who is calling. This adds a great amount of security to the system and it is hoped that holidays, full home-banking and betting will be sold through CompuNet.

Commodore have built 3000 modems, they have the parts for 7000 and aim to sell 40,000 by Christmas. One person I spoke to said "We will make Micronet look silly". These plans are certainly ambitious, other computers will be allowed in after a while (Apple are known to be interested), but it is the Commodore 64 owner who will get the crack of the whip. Watch future issues of Your Commodore for details.



Take those earplugs  
out and flex those  
fingers — William  
Fong has written a  
helpful music  
program for those  
learning to play a  
musical instrument.

**MAKING MUSIC ON THE Commodore 64** using Commodore BASIC is a long and tiring process. I have written a program which will allow you to play any tune you wish and have it played back at your chosen speed and instrument.

When you RUN the program the screen will display the table def and a summary of the instructions. You do not play the notes as you would on a piano but you have to press the notes'

# MAKING MUSIC

names, and when you do the notes will be displayed on the scales in their correct places. I have done it this way because you will not be learning if you proceed any old notes; I had to go through this stage when learning the piano. For those who do not play any instruments here is a summary of the notes playable:

MIDDLE C-D-E-F-G-A-B-C-D-E

## Instructions

(\*) This will play back the

music you have just created.  
(CBM + PLAY KEY) This will play the sharp of the note depressed.

(\*) Stops music from continuing play back.

(CLR/HOME) This will clear all the variables giving you a fresh start.

(-) This will delete a note.

(1-9) The numbers from one to nine will play the playback at different speeds, one being the fastest.

(R) This will play a one beat rest and display a rest on the scale.

(H) This will display all the

data needed to produce the tune you have created. This is very useful in writing music for your own program is speeded up.

(FUNCTION KEYS) These will choose the different instruments.

(SHIFT WITH NOTE) This will give you a note an octave higher. N.B. The highest note is high E.

This is only a very short program for what it does but it will be very helpful for the beginner who's learning to play any instrument.

When the whole scale is full





Having problems?  
Want to moan? Like  
to praise? This is your  
chance to put pen to  
paper and air your  
views or state your  
case. Just write to us  
and see if we can  
help.

**Dear Sir,**  
I am a beginner to computers, having bought a Commodore 64, with which I have no complaints. However, I do have a problem with programming. The problem is to do with the symbols which are contained in so many programs for the 64. I find it hard to distinguish the symbols from the printed sheet and I have very little understanding of these symbols and their exact meaning. I would be grateful if you could give me some tips on making the symbols clearer and also the titles of any books which will explain the symbols a little more clearly than the Commodore 64 Handbook. Yours faithfully,  
David Forrest  
Reddishware

**We answer,**  
The symbols can be confusing, especially when getting confused with ASCII (ASCII, screen codes and BASIC codes). To help with this, the best book to buy would be the Programmer's Reference Guide which will help you with all aspects of programming on the C64 64. The only other method is to persevere and deliberately use the symbols in your programs so that you will finally learn them and understand them.

# INPUT

**Dear Sir,**  
I do not understand how to use machine code on my VIC 20. Please could you show me a small machine code program where I can hit Return after every line and type RETURN at the end. In books I see long lists of letters, symbols, numbers and spaces which, when entered, give the message SNAFU! ERROR.  
Yours faithfully  
Janis Okasha  
Cambridge

**We answer,**  
Here is a short machine code program with a Basic loader:

```
10 P=300
20 read a: if a=1 then 30
30 poke p,a: p=p+1: goto 20
50 vrs 838: end
100 data 160,0,160,0,254,0
120 data 50: rem change this to 16 for expanded vic
140 data 254,0
160 data 21: rem change this to 17 for expanded vic
180 data 254,0
200 data 194: rem change this to 148 for expanded vic
220 data 254,0
240 data 151: rem change this to 149 for expanded vic
260 data 232,268,341,368,388,338,96,-1
```

**Dear Sir,**  
I have a Commodore VIC 20 and would like to expand it to its maximum. I already have two 1K and a 2K RAM Packs and a Stack 4 slot switchable mother board.  
1) How can I use the second 2K RAM Pack to expand beyond the 1K that I got with one RAM Pack to place 2) Do you think that since there are few if any programs for a 32K VIC that I am wasting my time and money in buying extra, and

3) should I sell one of the 1K RAM Packs and buy a 32K RAM Pack faithfully,  
A Rust  
Kingston on Thames

**We answer,**  
Do not buy the 32K ram pack as you do not need it and also because plug-ins are extra like adventure cartridges, machine code monitors and so on cannot be used with it. To make full use of your packs, open them up and inside will be banks of switches or solder blocks depending on how old they are. The pins are numbered from 1 - 32. On one pack, make pins 16 and 31 and on the other pack make pins 12 and 11.

**Dear Sir,**  
I have a VIC 20 on which I have been writing programs since last June. In the program I have written, I usually use user-defined graphics. Now I have a problem! Last Christmas, I have received a 32K RAM pack which now gives me 19.5K of usable memory. The problem is that I can't use my user-defined graphics in the way that I used to.  
eg.  
10 POK=32:POKE 96,19  
20 FOR T=142 TO 146:  
READ A:POKE A,1457  
30 POK=1469,25  
40 DATA etc. . .  
which defines four characters starting at screen code 14.

On the paper which comes with the RAM pack it states that "it is not possible to move the VIC screen or character set into external memory". Can you help me by purchasing a program which allows you to design a

character set on the internal memory with a 16K RAM Pack fitted and putting the UDCs on the screen? Yours faithfully,  
Wayne Deuschamp  
Northolt

**We answer,**  
If you type the following:  
10 poke 848,30: poke 431:1:  
poke 44,32  
20 poke 35,0: poke 56,96:  
poke 24-49,0  
30 print chr(147): new

This means that your program will now work and all your existing graphics software will also work.

**Dear Sir,**  
I have just purchased a C64 64 which I spend many happy hours with. However I could not get my hands on a C64 tape recorder, so I bought an Altai tape deck. This works fine on loading programs which I have typed into the computer. But on some software which I have bought such as all Amstrad software and Mastertronics' Vegas (which do not run. On screen it reads OUT OF MEMORY or ERROR in 70.

Is it possible to connect this deck in the tape deck or have I wasted my money? Yours faithfully  
John Adams  
Belisle

**We answer,**  
The most likely problem with this is that your recorder has no pause facility. When the computer has finished loading from cassettes, it automatically pauses it. For protection

# OUTPUT

purpose, a lot of software houses store their products in several parts on the tape that when loading the game, it loads one part then another. In the meantime though it has expected the tape recorder to have stopped and if it has not stopped then it has probably continued on past the next program header for what ever start marker these companies use on their tapes — move protection). So only part of the program will load.

Dear Sir, I want to ask you if the VIC 20 games are stable on the CBM 64; can you send me a list of the games that are stable? I have read that some people were going to expand their VIC 20 up to 12K, but I have also read in the *Beginners' Handbook* that the VIC 20 is 5K and expandable to 25K not to 12K? Can you clarify the situation? Yours faithfully,  
Aine Aegan  
London W7

**We answer,**  
No VIC 20 games other than simple non-graphic games will run on the CBM 64. You can expand your VIC up to 38K with 21.5K available from Basic. You cannot expand the CBM 64 and this machine has 16K of RAM.

Dear Sir, Having owned a CBM 64 for some months now, I wonder if you could help with a number of problems I have encountered: 1) How can I get the computer variable to 2 decimal places? 2) The operator's manual says that you can define strings to 256 characters. When I try entering this, I get a "Syntax error" if I press the Return key after I have typed more than two lines of text. 3) Is there a simple way of using the  $\delta$  function keys in a program written in BASIC, or can they only be defined in machine code? Yours faithfully,  
Lesley Knight  
West Sussex

**We answer,**  
Rounding numbers to two decimal places is a common

problem but there is an easy solution. Firstly we need a function to round off a number. This means that 1.555 becomes 1.55 and 1.555 becomes 1.56. Here is the function (define it near the start of your program):

```
10 define r(x)=int(x*100+.5)/100
```

Now here is a subroutine which will format numbers to two decimal places (adding a "-" sign at the start if it is negative). Note that the format string that this subroutine returns (a\$) is always the same length (18 characters) so that a list of numbers will always be neat.

```
100 sub r2(x)
110 if x<0 then a$=" "
120 if x<0 then a$=" "
130 if x<0 then a$=" "
140 if x<0 then a$=" "
```

You can only define strings to 255 characters. If you are typing a string into the computer you must remember that when the computer inputs from the screen, it only inputs two adjoining lines of 40 characters each. So if you were typing a long string into a line then when you had finished the computer would look at the last two lines and not find a line number and would therefore produce format error. The way to create long strings is by concatenation — adding strings together. Each of the function keys has an ASCII value so if you wrote a program using the GET command and pressed one of these keys, the ASCII value of that key would be returned. The keys are not definable from BASIC.

Dear Sir, I own a Commodore VIC 20 and have recently received a Commodore 1341 disc drive with which I seem to be having some problems. After I have run some of my programs I've found that when I give the drive a command it comes up with "Device not present error"; but the drive is connected and is switched on. To get the drive working again I

have to Run Stop and Restore the computer and then give the command: OPEN 13,15,"\*" (a wave star for the drive). It must be something in the program that somehow switches the drive off. Could you please tell me what it is and how to overcome it? Also can you expand the buffer so that you have more than 30 characters — if so, how? Any help would be greatly appreciated.  
Yours faithfully,  
Stuart Fosong  
Victoria, Australia

**We answer,**  
To solve your problems with the drive, you must send the

appear — coincidentally this is the same as chr(10). If you poke a screen location with 1 on it will appear chr(10) a reverse for control 'a' and this would send your printer in to enhanced mode. This subroutine can be converted to run on any of the Commodore machines printing to a Commodore dot matrix printer. The listing here is for a 4012:

```
10 n0=40:n1=25:rem number of columns: number of rows
20 s=12756:rem memory location of start of screen
30 open4,4
40 for i=1 to n0-1: a$=""
for j=1 to n0-1
50 a=peek(screen+i+j): b=a/60
if a and 128 then a=a&chr(16)
127: b=1
70 if a<12 then a=" " or 64:
goto 90
80 if a=85 then a=" " or 128
90 a$=a$chr(a): if b then a$=a$chr(16)
100 next j: print4,a$: next i
close 4
```

As you can see, the program is quite straight forward and can be easily coded into machine code if required.

Dear Sir, I have a Commodore 64 computer and also the EP-22 typewriter/printer from Brother, but I have tried without success to get the appropriate interfacing cable. I shall be very grateful if you could kindly supply me with any information on such cables. Yours faithfully,  
J Lee  
Manchester

**We answer,**  
The interface and cable for the Brother EP22 printer is available from:

Scan Products  
41, River Lane  
Garnwood  
King's Lynn  
Norfolk  
PE30 4AD Tel: (094575) 581

It costs £15.50 and is suitable for the CBM 64 and for the VIC 20.

**You could well drive  
yourself completely  
round the bend with  
this exciting game for  
the unexpanded VIC  
20 from Andrew  
Laprock.**

# TRACK KING

ARE YOU SITTING COMFORTABLY? Is your seat belt firmly fastened and your nerves steady? Then you can begin! Use your skill as a driver to reach the end of the race track and will be alive! You have to choose the correct times to accelerate and decelerate or you could end up crashing into the barriers or into the back of another car.

## Key to success

The game is played under the direction of the keyboard: U slows down the car, H moves the car to the left, J moves the car to the right and M increases your speed. By keeping the U key pressed, you will slow down more each time you move and so if you want to

slow down by only the slightest fraction, press U once and then take your finger off the key. The same principle applies while speeding up using the M key.

Every time the level changes, the track becomes longer making it harder for you to reach the end!

## The details

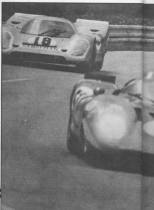
The game is made up of three programs. Program 1 covers mainly of PRINT statements although some FORs are involved. Program 2 defines the graphics. Program 3 contains the rinky-dinky stuff and the variables and line functions are given.

## Line explanation

LINE	FUNCTION
1-4	Initialise variables, first screen
9-12	Scroll screen in and out.
13	Set screen ready for game.
14	Print sides of track.
15-16	Variables.
20-28	Main section of program.
40-42	Crash routine.
43	Used with Line 41
50-61	Player reaches end: Prints what level he is at.
1000	Sound for every move.
2000	Used for crash graphics.
3000-3006	Sound for ambulance when crash occurs.

## Variable explanation

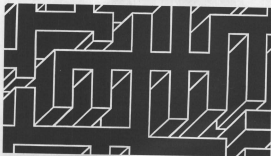
VARIABLE	FUNCTION
SC	Score
H0	Highest score
LC	Level
TC	Finishing distance
TA	Distance travelled
Y	Your car
SP	3070
J	Ignored
ST	3070: sound, etc
PI	Check for key pressed
L and M	Car that move towards you
K	Checks your car.





You have to cope with mazes (rather than snakes) with the ladders in this great game from Andrew Laycock.

# LADDER MAZE



A COMPASS MAY BE OF some help in this game, although I very much doubt it, and I'm not sure that a ball of string would add much to your chances of success! You must guide your man (could his name

be Vic by any chance?) through the maze and up the ladders.

The U key moves your man up the ladders, the H key moves him to the left and the J key moves him to the right. Ratings are given

on the time it takes you to reach the top so use your skills to get the best rating you can and see if you can rise to the challenge!

## To explain...

Program 1 is an

instructions program made up almost totally of PEEK statements.

Program 2 can be broken down as follows:

## Program 2 explanation

LINE	USE	3000-3010	Screen between games.
0	Starts at 1000	4000	Sets Y5
1-9	PRINTS maze lines and first ladder. Sets some variables. Main part of game movements, etc.	6000-6005	Ratings.
19-22	"You made it". Player reaches top.	The variables used are: A and X	
90	Reprint ladders. Choose new position for ladder.	OP	Used to set first screen: X = ladders.
100-110	Read data for graphics. Set screens	Z	Used to make man fall.
1000-1006	When player reaches top, send him back down.	M	Sets first ladder.
2000-2004		T5	Man
		L	Time
		H	Checks for key being pressed
			Check if new ladder is needed
		N1	Used in PEEK for man.
		M5	Substitutes for T5





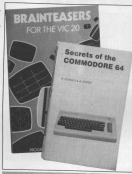
We have a look at  
some of the books  
sitting on our  
shelves.

# REFERENCE LIBRARY

**Book Title:**  
Secrets of the Com-  
modore 64  
**Author:**  
P. Collins and A. Cross  
**Publisher:**  
Bernard Babani Ltd.  
**Price:**  
£1.95

THIS POCKET-SIZED manual professes to be yet another beginner's guide to the Commodore 64. Although the book will win no prizes for originality, for those of you who haven't the time or the inclination to plough your way through some of the more weighty volumes on the market, it might prove invaluable as a brief yet informative introduction to the Commodore 64.

The book commences by informing us, for the umpteenth time in the history of computer literature, how the Commodore 64's memory is organised and of the proportion of that memory which is available to the Commodore 64 from BASIC. Some knowledge of the fundamental concepts of the Commodore 64 is obviously presumed since the authors ignore the simpler facets of the BASIC programming language by launching straight into such relative complexities as the all-important random numbers and means of generating them, the PEEK and POKE statements, the GET statement and input routines. The reader is also enlightened on character graphics, sprites and high resolution graphics. Chapters on sound, in the realm of both sound effects and music creation, and machine code programs and the use of machine code statements to extend BASIC, conclude this book.



On the whole, the book is user-friendly with chapters divided into short sections of succinct, jargon-free explanations elucidated by several diagrams and code-listings in easy-to-read bold type.

**Book Title:**  
Brain teasers for the VIC  
20  
**Author:**  
Genevieve Ludinski  
**Publisher:**  
Phoenix Publishing  
Associates  
**Price:**  
£5.95

GENEVEVE LUDINSKI, AN experienced programmer and technical author with her own software company

specialising in educational programs, has produced this book of twenty-two brain teasers for those VIC 20 users whose idea of fun is to tear their hair out and bite their nails while testing their mental agility. Many of the programs even contain an IQ rating at the end of the program, assessing the speed and dexterity with which you solve the problem in hand.

The brain teasers utilize the machine's music and graphics capabilities. Ms. Ludinski explains her puzzles briefly and intelligibly and the code for the brain teasers is listed in a clear and readable form.

A fairly wide selection of puzzles are offered ranging from run-of-the-mill

magazine games (Word Search) and maze games (Wire Maze and A-Mazing (not that one again!)) to musical puzzles (Sains to Sinners — a musical guessing game) and the obligatory space adventure, Close Encounters. The Francis Drake Adventure Game, the program rated by Ms. Ludinski as her star performer, lets the reader step back in history and into the shoes of Sir Francis Drake as he sailed off in pursuit of the North West Passage. You also have the chance to be on either side of the law. For the villains amongst you, the book offers Safe Cracker, a simple guessing game complete with police store of an incriminated gun; the cops hot on post mail and Western Adventure which allows you to live the life of lawless on the run from the sheriff after robbing a bank, in true western style. Or put on your detective hat for Detective on an Agatha Christie style Who Dunit.

So whether you see yourself as cop or robber, musician or explorer, this book provides the reader and VIC 20 user with puzzles to tax the brain while stepping into the shoes of his alter ego; the less adventurous amongst you can stick to Ms. Ludinski's number and word puzzles.

**Book Title:**  
Commodore 64 Sound  
& Graphics  
**Author:**  
Peter Falconer  
**Publisher:**  
Melbourne House  
Publishers

THE COMMODORE 64'S sophisticated sound and graphic capabilities, covered briefly in the

Commodore 64 manual, are expounded in this book by Peter Falconer. The features available are explained and the user is involved in the design and coding of a real application. The game used for the purpose of illustration throughout the book is written in both BASIC and machine code, thus enabling the user to exploit the full potential of the Commodore's sound and graphic features. But don't fret if you're yet to discover the delights of machine code since Mr. Falconer presents his machine code routines in three ways: as assembly language code, as BASIC programs illustrating the same idea (where possible) and as DATA statements for inclusion in BASIC programs. The author has organized a lot of useful information into a small book and has made this information easy to follow by presenting it in small blocks of text interspersed with examples of code. The book is divided into two sections, the first on low resolution graphics (ie, character graphics) and the second, smaller section, on hi mapped graphics. As per usual, the user's ability to follow the code is hampered by the illegible Commodore graphic output.

The book starts with a simple game program written in BASIC. This game is enhanced throughout the book by the addition of graphics, sound and more professional presentation. These features are covered chapter by chapter. The mysteries of sprites, music and interrupts are explained. Sound on the Commodore (voices, waveforms, volume, for example) is explored and the more complex facilities of the Commodore 64 such as scrolling, sprites and high resolution graphics are investigated.

Having pieced the game together chapter by chapter, the user should be left with the know-how to use the Commodore 64's graphic and sound features when developing his own games and applications.



**Book Title:**  
Business Systems on the Commodore 64

**Author:**  
Susan Curran and Margaret Norman  
**Publisher:**  
Granada  
**Price:**  
£6.95

**THIS BOOK, WRITTEN BY** Susan Curran, a full-time writer specialising in computers and their applications, and Margaret Norman, a freelance writer and computer programmer, provides a practical and

informative introduction to the Commodore 64 as a small business computer. It shows how the Commodore 64, low priced yet well established and highly reliable is crucial factor when considering hardware for business applications, can be employed to run business application packages.

The book starts with a general explanation of the Commodore 64, assessing why it is a good machine for the small business and painting a broad picture of the business software

available. It does not profess to be a maintenance guide or programming guide nor is it merely a lengthy list of business applications available for use on the Commodore 64. Each chapter commences with some background information into the type of application under discussion. The text is clearly illustrated with computer printouts and screen displays. The reader is told how transition from manual accounting, stock-taking, or whatever the business application in question may be, to computerisation may complement his business. The types of software available for a particular application are then discussed.

The applications included are computerised accounts, stock control and other money-oriented programs, spreadsheets, databases and filing systems, word processing programs and a final chapter on applications excluded from the list.

To conclude, "Business Systems on the Commodore 64", provides a clear and concise introduction to the different types of business applications which may be run on the Commodore 64 and explains how they may be integrated into the small business.

**Book Title:**  
The Penguin Book of VIC 20 Games  
**Author:**  
Paul Copeland  
**Publisher:**  
Penguin  
**Price:**  
£2.95

PAUL COPELAND'S BOOK of VIC 20 games, at £2.95, is good value for money. The book contains an assortment of adventures, arcade-type games and board games. The author's musical background is reflected in the sophisticated musical routines in some of the games, Life, A4-Musician and Music Sequencer, for example. As well as ten games, the book includes chapters on utility programs and creating your own

## business applications for the commodore 64

techniques and subroutines for business users

James Hall



games. The author claims to have fully exploited the colour, sound and graphics potential of the VIC 20.

The first two games, You Draw and Noughts and Crosses, may be adapted for use on disc based systems. All the games are introduced by a brief synopsis of what should happen if you enter the program code correctly and by descriptions of the program structure. The game listings conclude the chapters.

The games vary from the commonplace — Noughts and Crosses — to the adventurous such as Red Alert where, as commander-in-chief of a space ship you are sent on a secret mission into outer space to place various space modules in quadrants of space.

The entrepreneurs amongst you may wish to test your skills at Oil Rig whereby, as director of an oil mining company, you have to find as many oil fields as possible (up to a maximum of ten), if your fortune lies in gambling rather than business, step back to the Las Vegas of 1965 where your target is to break the bank before it breaks you.

Less philistine readers may wish to take their creative talents to the small screen with You Draw or Music Sequencer.

A set of handy utility routines and a group of scenarios to help you create your own games can be found at the end of the book.

**Book Title:**  
Mastering Machine Code on your Commodore 64

**Author:**  
Mark Greenfields  
**Publisher:**  
Interface Publications  
**Price:**  
£7.95

THIS GLOSSY LITTLE volume, written by the author of "Mastering the Commodore 64", allows the Commodore user, already conversant in BASIC, to create more efficient, faster and professional programs

through the medium of machine code — in this case, the 6502/6510 Assembly language, available for use on the Commodore 64.

The book comprises three sections and numerous informative appendices. Section one commences with a listing and explanation of a 6510 assembler/disassembler/monitor, SUPER-MON. Since all the programs in the book are listed in mnemonic format, an assembler is needed to enter them. This section continues with explanations of every 6510 Assembly language command and every programming mode of the 6510 chip. The tutorial is interspersed with copious and lucid examples of Assembly language programs.

The second section shows the user, having now achieved some proficiency in Assembly language programming, how to capitalise on his newly acquired skills by putting them to practical use. The use of Assembly language in

scrolling (both with plink and character), sprites, music scales and tunes on one and three channels, interrupts, raster scan graphics, high resolution graphics and, finally, adding commands to the BASIC language, are covered. As with the first section, Mr Greenfields' text is clarified by appropriate examples and listings. Furthermore, he inevitably coaches his explanations in lay-man's terms, thus maintaining his back-cover promise to produce a book for the Assembly language beginner.

The third and final section covers the ROM routines inside the Commodore 64 and instructs the user on how best to apply them to their programs.

The book ends with a generous spread of appendices (12 in all) many of which appear to have been reprinted from the Commodore 64 manual (ASCII values for characters and control codes).

**Book Title:**  
Business Applications for the Commodore 64

**Author:**  
James Hall  
**Publisher:**  
Sunshine Books  
**Price:**  
£5.95

FOR THOSE COMMODORE 64 users who wish to use their computer as a business machine without turning to the off-the-shelf business applications market, this book provides a useful introduction to designing and writing bespoke business software. In his introduction, Mr. Hall expresses the amount of time which may be saved in changing from a manual to a computerised system, as deduced from his own experience. Moreover, he states that there is no need to invest in expensive peripherals to run business programs on the Commodore 64 apart from the use of a printer for word processing (although he does, later on, stress the advantage of a disc drive in providing greater memory capacity and faster file access for business applications). The inexperienced programmer need not fret since these programs are accompanied by step-by-step documentation and illustrations where necessary.

The book introduces the reader to business program design with a synopsis of computers and file handling. The applications covered by the book are constructed from a library of subroutines which can then be amended, added to and tailored for the user's individual needs. A useful final chapter allows the user to enter a routine to replace dot matrix graphic symbols with letters and also includes a utility program.

So, if your Commodore 64 is employed as a business machine and you are brave enough to exploit the advantages of creating your own business applications, this attractive book is a wise investment.

This great series introduces you to the delights of programming games for the VIC in BASIC. Bryn Phillips takes us through the steps.

# VIC GAMES PROGRAMMING

THIS IS THE FIRST OF A FIVE part series of BASIC games programming for the VIC 20. The series is primarily intended for newcomers to games programming, but there might well be a few useful tips for seasoned programmers.

Many people are put off writing action games in BASIC because they think that it is too slow, and have read somewhere or other that any good game must be written in Machine Code. That's true up to a point. A badly written, full screen version of Space Invaders written in BASIC could be really painful. Semi-conscious Aliens jolking down the screen being panned off one by one with your laser

in between cups of coffee would not be exactly inspiring. On the other hand a well written Lunar Lander or Shooting Gallery game could rival many of the commercial products. In fact it would be better, because it would be YOUR game with your own amazing graphics and sound effects, and that is what it's all about.

In this series I will attempt to show you some of the techniques which you can use to write effective games programs in BASIC, and if you follow it through and use your imagination you should even be able to do something about that Space Invaders game!

## Screen display

This month the emphasis is on the screen display, and this can be one of the most

Listing 1	
10	PRINT"Q"
20	PRINT"*****"
30	PRINT" +"
40	PRINT" +"
50	PRINT" +"
60	PRINT" +*****"
70	PRINT" +"
80	PRINT" +"
90	PRINT" +"
100	PRINT" +*****"
110	PRINT" +"
120	PRINT" +"
130	PRINT" +"
140	PRINT" +*****"
150	PRINT" +"
160	PRINT" +"
170	PRINT" +"
180	PRINT" +*****"
190	PRINT" +"
200	PRINT" +"
210	PRINT" +"
220	PRINT"*****"
230	PRINT"MEMORY FREE"PRC(0)
240	DOTQ240

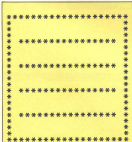


Figure 1.

enjoyable aspects of writing a program. It's a good place to start because speed doesn't matter — the action doesn't start until you've set up your play area. The key point is to get it looking good, and that takes planning. One way of doing this is to draw it out on paper first. Graph paper is OK, but don't forget your final screen display will be rectangular. The only problem with using paper is that it is time consuming and discourages experimentation. Once you've got something which looks reasonable you tend to stick with it, and you're immediately off to a bad start. It's much better to use the screen to experiment, and the utility program "SCREEN DISASSEMBLE" included in

this article will help you to do this. It's worth noting at this point that the VIC 20 has an extremely well thought out graphics set built into it. Unfortunately there is a great temptation to forget this and always use your own user defined graphics (we'll be covering this in Part 4). But don't ignore it — even if you use your own graphic characters you can always incorporate some of the standard graphics into your own set. When you see this program pay particular attention to the colours. Often an otherwise good program is ruined by murky graphics. This is because some of the character and colour combinations just don't mix very well, to the extent that some are virtually undecipherable. It

series from computer to computer and your TV can also make a difference. Play also with the graphics, make some shapes, design a screen — EXPERIMENT!

## The program

So now you've got a screen design and want to put it in your program. There are two options in BASIC on the VIC 20, PRINTing or POKing. Let's look at a simple screen display, such as the one in fig. 11.

You can either PRINT it or POKe it onto the screen. PRINTing is the simpler option as you simply write a number of PRINT statements containing the design. This is shown in Listing 1.

In contrast POKing a design onto the screen takes a bit more thought. You have to define your screen co-ordinates and use the appropriate POKe formulae. The simplest method is to regard the screen in terms of X,Y co-ordinates and define the bottom left hand corner as 0,0. To POKe a character into the screen use:

POKE P1\*X-22\*Y,CH

and to POKe a colour use:

POKE P1\*X-22\*Y,CL

Where P1=6554  
P2=36864

Listing 2 gives an example of how the screen design can be POKed onto the screen.

At first sight PRINTing is better as it's much faster.

Wait a minute though, speed doesn't matter here. What about memory? If you look at the free memory displayed you will see that Listing 1 only left 2929 Bytes free, whereas Listing 2 left 3262 Bytes free, a saving of 333 Bytes, and that could be crucial later on. If you examine Listing 2 you will

see that it could be tightened up even further by using multiple statement lines, GOSUBs etc, and this is shown in Listing 3 where a further 88 Bytes have been saved, a total saving of 375 over Listing 1.

If you're writing games programs on an unexpanded VIC 20, tight programming is essential. Admittedly the listings don't look so good, the program can be difficult to follow and debug, but you leave yourself enough memory over for the game and with any luck will have enough left over for all those trills that make it look professional — High score record, menu options, keyboard/joystick capability and of course the sound effects. As you will see in a future article tight programming also means speedier action, and it really is worth taking the time to plan ahead, and to practice squeezing everything down as far as you can.

## Random thoughts

Finally let's have a look at the RND function. Often it can be used to very good effect in screen design and in a multi-level game can be used to add variety in the game program. You can use it to change character

Listing 1

```

10 PRINT"Q"
20 P1=6164:P2=36864
30 FOR% =1 TO 20
40 FORY=1 TO 13 STEP 20
50 POKEP1+% - 22*Y,42
60 POKEP2+% - 22*Y,6
70 NEXT Y
80 NEXT X
90 FOR% =3 TO 18
100 FORY=5 TO 18 STEP 4
110 POKEP1+% - 22*Y,42
120 POKEP2+% - 22*Y,6
130 NEXT Y
140 NEXT X
150 FOR% =4 TO 20
160 FOR% =1 TO 20 STEP 15
170 POKEP1+% - 22*Y,42
180 POKEP2+% - 22*Y,6
190 NEXT X
200 NEXT Y
210 PRINT"MEMORY FREE"PRE(8)
220 GOTO 220

```

Listing 2

```

10 PRINT"Q"
20 P1=6164:P2=36864
30 FOR% =1 TO 20:PORY%=1 TO 13 STEP 4:GOSUB100:HEXTY, X
40 FORY=1 TO 21:POR% =1 TO 20 STEP 19:GOSUB100:HEXTX, Y
50 FOR% =2 TO 20:POR% =3 TO 19 STEP 17:POKEP1+% - 22*Y,32:HEXTX, Y
60 PRINT"MEMORY FREE"PRE(8)
70 GOTO 70
100 POKEP1+% - 22*Y,42:POKEP2+% - 22*Y,6:RETURN

```

Listing 3

```

10 PRINT"Q"
20 P1=6164:P2=36864
30 CH=169:CL=8
40 FOR% =1 TO 20:PORY%=1 TO 13 STEP 20:GOSUB100:HEXTY, X
50 FORY=1 TO 21:POR% =1 TO 20 STEP 19:GOSUB100:HEXTX, Y
60 CH=162:CL=2
90 GOSUB90:GOSUB100:GOSUB90:GOSUB110:GOTO 90
90 X=INT(RND(1)*10+2):Y=INT(RND(1)*41+2):GOSUB100:RETURN
100 POKEP1+% - 22*Y,CH:POKEP2+% - 22*Y,CL:RETURN
110 POKEP1+% - 22*Y,32:RETURN

```



## Screen Designer Program

```

10 DIM SCREEN DESIGNER FOR ANY VIC 20
20 DIM DIMS (255,255)
30 REM
40 REM MEMORY FULL ERROR
50 REM
60 SCREEN=0
70 DIM C(255)
80 DIM S(255)
90 DIM T(255)
100 DIM P(255)
110 DIM B(255)
120 DIM R(255)
130 DIM G(255)
140 DIM BL(255)
150 DIM BR(255)
160 DIM BG(255)
170 DIM BB(255)
180 DIM B1(255)
190 DIM B2(255)
200 DIM B3(255)
210 DIM B4(255)
220 DIM B5(255)
230 DIM B6(255)
240 DIM B7(255)
250 DIM B8(255)
260 DIM B9(255)
270 DIM B10(255)
280 DIM B11(255)
290 DIM B12(255)
300 DIM B13(255)
310 DIM B14(255)
320 DIM B15(255)
330 DIM B16(255)
340 DIM B17(255)
350 DIM B18(255)
360 DIM B19(255)
370 DIM B20(255)
380 DIM B21(255)
390 DIM B22(255)
400 DIM B23(255)
410 DIM B24(255)
420 DIM B25(255)
430 DIM B26(255)
440 DIM B27(255)
450 DIM B28(255)
460 DIM B29(255)
470 DIM B30(255)
480 DIM B31(255)
490 DIM B32(255)
500 DIM B33(255)
510 DIM B34(255)
520 DIM B35(255)
530 DIM B36(255)
540 DIM B37(255)
550 DIM B38(255)
560 DIM B39(255)
570 DIM B40(255)
580 DIM B41(255)
590 DIM B42(255)
600 DIM B43(255)
610 DIM B44(255)
620 DIM B45(255)
630 DIM B46(255)
640 DIM B47(255)
650 DIM B48(255)
660 DIM B49(255)
670 DIM B50(255)
680 DIM B51(255)
690 DIM B52(255)
700 DIM B53(255)
710 DIM B54(255)
720 DIM B55(255)
730 DIM B56(255)
740 DIM B57(255)
750 DIM B58(255)
760 DIM B59(255)
770 DIM B60(255)
780 DIM B61(255)
790 DIM B62(255)
800 DIM B63(255)
810 DIM B64(255)
820 DIM B65(255)
830 DIM B66(255)
840 DIM B67(255)
850 DIM B68(255)
860 DIM B69(255)
870 DIM B70(255)
880 DIM B71(255)
890 DIM B72(255)
900 DIM B73(255)
910 DIM B74(255)
920 DIM B75(255)
930 DIM B76(255)
940 DIM B77(255)
950 DIM B78(255)
960 DIM B79(255)
970 DIM B80(255)
980 DIM B81(255)
990 DIM B82(255)

```

positions, characters and colours, and is particularly useful in maze designs. A simple example is shown in Listing 4.

Type in the program and RUN it. The program simply draws a black boundary around the screen, and randomly FORKs red squares onto the screen. Now comes the important part. Sit back and watch the screen. At this point it is possible that you will just see squares appearing and disappearing. If this is the case you need to try harder. Relax, and look again. It could be the start of a monster storm scenario, or a devilish hyper-galactic map, or moles appearing and disappearing in your garden! Remember, we'll be able to make those squares look like anything, and you can change the ratio of squares to blanks, mix colours, add extra characters etc. It could be the basis of a good game. We'll have another look at it next month.

**Next time...**

In the next part of this series we'll be looking at getting the characters moving, using programmed commands, and keyboard or joystick interaction. In other words we'll get it all moving.

**Screen designer**

This program will allow you to experiment with screen designs. On running the program you will be presented with a blank screen containing a flashing cursor which you can move around the screen using the cursor controls. If you press a key then the appropriate character will be entered on the screen elsewhere the cursor goes. You can use the full VIC character set, including reversed characters. If you want a reversed character press CTRL/PA5, and the next character entered from the keyboard will be reversed. You can change the colour of the character by pressing F1, followed by the appropriate colour key, and similarly change the border and screen colours by using F3 and F5 respectively.

Just as a change from the usual type of book review, we thought we'd show you exactly the sort of thing you'll be getting for your money.

# COMMODORE 64 EXPOSED

THIS ARTICLE IS AN extract from the Graphics chapter of Melbourne House's *Commodore 64 Exposed* by Bruce Bayley, which we are reprinting with their kind permission. This should give you, not just a lot of useful information on sprites, but an idea of the high quality of this book. *Commodore 64 Exposed* costs \$6.95 in paperback, and contains 195 pages.

Melbourne House are concentrating heavily on the Commodore 64 this autumn. They have just published *Commodore 64 Sound and Graphics*, and at the end of September *Commodore 64 Machine Language for the Absolute Beginner* will be available. Both books will also be priced at \$6.95.

Melbourne House books can be found in all good computer bookshops, or they can be contacted directly at Castle Yard House, Castle Yard, Richmond, TW9 1DT.

## Sprites

A sprite is a form of user-defined character that is controlled by a powerful video chip called the 6566. Up to 8 sprites can be displayed at a time automatically. More sprites can be displayed using raster interrupt techniques. Sprites have the following advantages over user-defined characters:

1. Pixel by pixel movement in any direction.
2. The 24 by 23 pixel sprite shape can be moved as though it were a single character.



3. Magnification (2X) in both horizontal and vertical directions.
4. Independent high-res/multicolour mode.
5. Selectable sprite to background overlay priority.
6. Sprite to sprite collision detection.
7. Sprite to background collision detection.

A sprite is larger than a character, therefore more data is needed to define the shape of a sprite. A sprite is 24 pixels (3 bytes) wide and 23 pixels high which gives a total of  $3 \times 23 = 69$  bytes of data to define the shape of a single sprite. Even though a single sprite is made up of so

much data, the video chip moves the sprite as if it were a single character.

## Sprite Pointers

The 64 byte blocks of data that define the shape of each sprite can be placed in any 64 byte multiple of unused memory. In order to tell the video chip where in memory each sprite-shape block is located, eight sprite pointers are provided.

The shape of a sprite may be changed by adjusting the sprite pointer allocated to that sprite to point to a different block of sprite-shape data. Using this

method a single sprite may be animated by quickly changing the sprite's pointer to switch through a series of shapes provided for that sprite's animation (e.g. an explosion). Switching the pointer rather than switching between sprites leaves the other sprites free for other uses.

The sprite pointers are the last 8 bytes of unused screen memory (2040 - 2047). If you move screen memory, the pointers will move with it (but not their contents). You must remember when writing up your sprite pointers that the pointer must point to the first byte within the sprite and that the value in the sprite pointer is the actual memory location of the sprite over 64. Therefore, the following formula applies:

Location = Sprite pointer \* 64

Also if you are not using video bank # 0 (default bank) then you must also add bank number \* 16384 to the location. If you haven't switched video banks that don't worry.

Two important points to remember when choosing where to put your sprite data in memory are 1, its location must be a multiple of 64, and 2, check the memory map to make sure that you are only using spare memory.

## Turning Sprites On

For a sprite to be displayed to the screen, it must be turned on. The memory location where the video chip gets its information on which sprites should be





turned on and which should be turned off is location \$1268. The 8 bits within byte \$1268 are labeled from right to left 0—7. Therefore, if we label our sprites from 0—7 then we easily determine which sprites should be on and which should be off by the value contained in byte \$1268. The way that the on/off status of each sprite is determined is as follows:

A 1 in the bit corresponding to the sprite determines that the sprite should be displayed (turned on) and a 0 determines that the sprite should not be displayed (turned off).  
eg. 7 6 5 4 3 2 1 0  
1 1 0 1 1 1 1 = 215

Therefore the statement POKE \$1268, 215 would supply the video chip with the following information: Sprites 7,6,4,3,1 and 0 are to be turned on. Sprites 5 and 2 are to be turned off.

To turn on a single sprite without affecting the others, use the following statement:

```
POKE $1269, PEIK ($1269)
OR ($N)
```

where \$N is the sprite number (0—7).

To turn off a single sprite without affecting the others, use the following statement:

```
POKE $1268, PEIK ($1268)
AND ($N - 255)
```

## Sprite Colour

High resolution (single colour) sprites can be any one of the 16 colours. The colour of each sprite 0—7 should be POKE'd into their respective colour registers, memory location \$1262—\$1268 (see video register map). Each pixel turned on within the sprite will be displayed in the colour determined by the sprite's colour register. Each pixel turned off will be displayed in the colour behind the sprite (i.e. it is transparent).

## Multicolour Sprites

In multicolour mode, it is possible to have four different colours in each sprite. Though, as with multicoloured characters, multicoloured sprites are

only half the resolution of single coloured sprites (ie. pixels must be displayed in pairs). The following table gives the colours determined by each bit-pair combination.

Bit pair	Resultant Colour
00	Transparent (clear colour)
01	Sprite multicolour register #0 (location \$1262)
10	Sprite-colour register #0 (location \$1266)
11	Sprite multicolour register #1 (location \$1268)

The register that holds information on which sprites are multicoloured and which sprites are not is mapped to location \$1276.

To set a sprite to multicolour, use the following statement:  
POKE \$1276, PEIK (\$1276)
OR (\$255)

where \$N is the sprite number (0—7).

To switch a sprite out of multicolour mode, use the following statement:  
POKE \$1276, PEIK (\$1276)
AND (\$N - 255)

## Expanding Sprites

Sprites can be expanded vertically, horizontally or both. A sprite is expanded by putting 2 pixels in place of 1 and 2 blanks in place of 1 in the direction of expansion thus giving a 2x expansion. To expand a sprite horizontally, the corresponding bit in location \$1277 must be set to 1. To reduce the sprite, the bit must be set to 0. Vertical expansion is done in the same way using location \$1277. The POKE statements to control expansion and reduction of sprites are as follows:

```
Horizontal expansion
POKE $1277, PEIK ($1277)
OR ($N)
```

```
Horizontal reduction
POKE $1277, PEIK ($1277)
AND ($N - 255)
```

```
Vertical expansion
POKE $1275, PEIK ($1275)
OR ($N)
```

```
Vertical reduction
POKE $1275, PEIK ($1275)
AND ($N - 255)
```

where \$N is the sprite number from 0—7.

## Sprite Movement

Sprites are moved around the display by changing the values in each sprite's horizontal and vertical position registers. These

registers are mapped to memory location \$1248 to \$1263 and a most-significant-bit (MSB) register at location \$1264. The MSB register is used to rectify the problem of horizontal screen width.

The MSB register works as follows. In order to gain pixel movement, the horizontal position register needs to be able to hold values from 0 to 255 (screen width). A single register can only hold values from 0 to 255; therefore we need at least one more bit to handle values up to 259. An extra bit (9th bit) would allow us control over positions 0 to 251. This is the purpose of the MSB register. The bits in the MSB register correspond to the sprite number (ie bit 0 for sprite 0, bit 1 for sprite 1, etc.). A register map of all sprite positioning registers is given overleaf. Note that horizontal positions 24 and 244 are the left and right boundaries of the screen. Sprites continue to move outside this range but cannot be seen.

It's about time we had a look at one of these sprites.

Study the first program and its comments. Type it in and run it.

Run the program; you should see a square sprite float across the screen.

To expand the sprite in the horizontal and vertical directions before moving, add the following line:  
105 POKE VIDREG + 24,1 : POKE VIDREG + 24,1

and run the program again. The second program allows you to use the cursor keys to draw a sprite by editing DATA statements. Type RUN 1, then use the cursor keys to move around

the DATA statements. Use the shift-Q character to signify a pixel-ON and a full-stop to signify a pixel-OFF. When you have finished drawing your sprite, move the cursor to the top of the screen, then keep hitting the RETURN key until you have entered all of the DATA statements. Now type RUN, and the program will generate the sprites and the DATA statements needed to generate that sprite. To use these DATA statements, use the same method as you used on the last set of DATA statements.

## Sprite Display Priorities

Sprite priority determines if the sprite should appear in front or behind another background. If the background is another sprite, then the priority is fixed by the sprite's sprite number. Sprite 0 has the highest priority, sprite 1 has the next priority, and so on, up to sprite 7. For example, if sprite 0 and sprite 7 are positioned so that they cross each other, sprite 0 will be in front of sprite 7, though you would be able to see sprite 7 through sprite 0 (unless of course sprite 0 was a completely filled square). Sprite to background priority is more flexible in the way that each sprite can be set with priority above or below the background. The sprite to background priorities are controlled by the sprite priority register (memory location \$1273). A 1 in the bit number corresponding to the sprite number will set that sprite with a lower priority than the background. A 0 in this bit position will give the sprite a higher priority than the background. By moving sprites back and forth over other objects, at the same time changing the sprite-background priorities, it is possible to make it look as if the sprites are moving in front and behind the object thus creating a three dimensional effect.

The third program overlays 8 sprites to demonstrate sprite priority.



## Sprite Collisions

Sprite collisions are detected by the computer and collision information is stored in location 53278 for sprite to sprite collisions and location 53279 for sprite to another background collision. The bit set to 1 in each of these registers corresponds to the sprite involved in the collision. So, for example, if you wish to have objects that should not cause a collision (eg. a cloud) then they should be coloured by using a bit pair 01 (multicolour register # 0).

value into a variable. Also, programs that use the sprite collision registers should include in their initialisation a PEEK of each of these registers to clear them of previous collision data.

Note: A bit pair 01 in a multicoloured mode will not be detected in a sprite to background collision, even though it can be seen on the screen. So, for example, if you wish to have objects that should not cause a collision (eg. a cloud) then they should be coloured by using a bit pair 01 (multicolour register # 0).

Location	Use of Register
53286	Sprite 0 X position
53289	Sprite 0 Y position
53290	Sprite 1 X position
53291	Sprite 1 Y position
53292	Sprite 2 X position
53293	Sprite 2 Y position
53294	Sprite 3 X position
53295	Sprite 3 Y position
53296	Sprite 4 X position
53297	Sprite 4 Y position
53298	Sprite 5 X position
53299	Sprite 5 Y position
53300	Sprite 6 X position
53301	Sprite 6 Y position
53302	Sprite 7 X position
53303	Sprite 7 Y position
53304	Sprite (8-7)MSB register

## VIC-II Register Map

Reg.	0	1	2	3	4	5	6	7	8	9
0	Sprite 0 X									
1	Sprite 0 Y									
2	Sprite 1 X									
3	Sprite 1 Y									
4	Sprite 2 X									
5	Sprite 2 Y									
6	Sprite 3 X									
7	Sprite 3 Y									
8	Sprite 4 X									
9	Sprite 4 Y									
10	Sprite 5 X									
11	Sprite 5 Y									
12	Sprite 6 X									
13	Sprite 6 Y									
14	Sprite 7 X									
15	Sprite 7 Y									
16	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	MSB of 8-sprite memory bank
17	Sprite MSB	Extended colour	Background	Colour enable	Screen height			Vertical scroll		
18	Colour register									
19	to sprite palette									
20	to sprite palette									
21	Sprite 0 X	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Sprite enable
22	Sprite 1	Sprite 0	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	MSB colour's scroll
23	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Sprite 0 - enabled
24	MSB									
25										
26	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Colour register
27	Sprite 1	Sprite 0	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Colour register
28	Sprite 2	Sprite 1	Sprite 3	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Sprite 0	Background priority
29	Sprite 3	Sprite 2	Sprite 4	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Sprite 0	Sprite 1	Sprite multicolour select
30	Sprite 4	Sprite 3	Sprite 5	Sprite 6	Sprite 7	Sprite 8	Sprite 0	Sprite 1	Sprite 2	Sprite 0 - enabled
31	Sprite 5	Sprite 4	Sprite 6	Sprite 7	Sprite 8	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite to sprite collision
32	Sprite 6	Sprite 5	Sprite 7	Sprite 8	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite to background collision
33	Sprite 7	Sprite 6	Sprite 8	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	
34	Sprite 8	Sprite 7	Sprite 0	Sprite 1	Sprite 2	Sprite 3	Sprite 4	Sprite 5	Sprite 6	
35	Background Colour 0									
36	Background Colour 1									
37	Background Colour 2									
38	Background Colour 3									
39	Sprite multicolour = 0									
40	Sprite multicolour = 1									
41	Sprite 0 Colour									
42	Sprite 1 Colour									
43	Sprite 2 Colour									
44	Sprite 3 Colour									
45	Sprite 4 Colour									
46	Sprite 5 Colour									
47	Sprite 6 Colour									
48	Sprite 7 Colour									
49	Sprite 8 Colour									

# 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  
Argus 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 computers/memory size which your program runs on without conversion or job \_\_\_\_\_

Does your game need or use (provide) \_\_\_\_\_

Yes

No

Have you sent our game to another magazine \_\_\_\_\_

Yes

No

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

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

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

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



Argus Press Software Publications      Sept/Oct 1986      £2.99

# 64 TAPE COMPUTING

For all CBM 64 Users

● **3 Original Games**

Arcade, Adventure and Strategy – more fun, more value – only from Argus!

● **UTILITIES**

Generators, delights, galore – improved graphics, sounds, disassemblers – it's all here!

● **On screen reviews of newly launched games**

News and views of the wonderful, whacky world of home computing



## 64 Tape Computing adds a new dimension to your micro!

Run this Argus 64 tape and you'll soon see why it's Britain's top selling tape magazine. Each issue gives you a variety of exciting and challenging games to play.

reviews of other newly released software plus valuable utilities enabling you to write your own programmes and games.

Stretch your imagination and skills with 64 Tape Computing – available every other month from WH Smith, Menzies and other leading stores.

(You'll see them advertised on TV from September!)

**Get your copy today!**



Argus Tape Magazines produced by  
**ARGUS PRESS SOFTWARE**

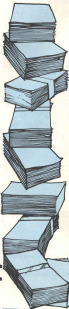
1 Golden Square, London W1B 3AB  
Telephone: 01 492 0066



Dave Crisp helps out in a retail outlet and he helps here with relative files, which seem to cause a lot of bother.

# RELATIVE FILES

## RELATIVE FILES



JUDGING BY THE AMOUNT of telephone calls, enquiries and questions I receive it would seem that one thing which causes a lot of problems to owners of the 1541 disc drive is relative files. Except for a few things the manual is more or less useless. Using these two programs you should at least be able to get BASIC relative files working and once you have got the idea it is easy to start to manipulate files and file sizes etc.

### Relatively speaking...

Think of a relative file as a large array. Instead of the array being in RAM though it is stored in numbered order on the disc. As with an array it is easy to find a particular record if you know the number. If the number is not known it is possible to search the disc record by record.

The two programs are very basic. They have simple screen displays, there is very

little error checking, and virtually no string handling. Once you have got the idea of what is going on then you can go back to the manual, which should then make more sense and tidy up the programs.

When creating the file for the first time, I recommend that you use a blank disc. This will ensure that all record numbers are available. Initially create a file of 254 characters/record and have about 100 records. It is worth noting that it is

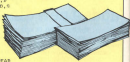
### Program 1 Listing

```

READY.

1 REM00000 PROGRAM 1000000
2 PRINT "M"
3 INPUT "*****HOW MANY CHARACTERS/RECORD?":RC
4 INPUT "*****HOW MANY RECORDS MAX.500":RM
10 OPEN "S,B,1":OPEN2 "B,2,":OPEN3 "FILE,L,":CHR$(RC)
20 FOR N=1 TO RM
30 N=N+1
40 FN=INT (RM/254):FN=INT (RM/254)
45 PRINT FN = "FN", FN = "FN"
50 PRINT LC, "P":CHR$(2):CHR$(L):CHR$(RC):CHR$(RM)
45 PRINT LC, "C":CHR$(1)
45 INPUT LC, "A,B,C,D,B"
50 PRINT LC, "A,B,C,D,B"
250 NEXT N
260 CLOSE1:CLOSE2

```



```

READY.

SEARCHING FOR RECORDS
LOADING

READY.

```



not possible to replace a relative file using "RE. The only way to remove a relative file is by the scratch command.

### Get running

When you run the first program you will notice that an error "RECORD NOT PRESENT" appears. When creating a file this is quite normal and can be ignored.

What does the program do? PROGRAM 1. LINE 5 is the number of characters/record. To start with stick to 24 (this is the maximum possible).

LINE 6 is the number of

records you want to create initially. Stick to about 100 for now.

LINE 10 opens the file and gets it a name. The J, part tells the disc that it is a relative file and the CHR\$(J) tells the disc the record length.

LINE 40 is an easy way to split a record into a two byte location. NULL the command AND.

LINE 58. The "P" stands for POSITION and this tells the disc drive to be ready to locate the read/write head at the following track/sector. The CHR\$(J) in this line are A: the channel being used (2 in this case) B: This is the low byte of the track/sector

address. C: this is the high byte of the address. D: this is location within the record. After running the program and watching the display you should see what is happening.

PROGRAM 2. This is split into two parts, Read and Write. The screen display is simple and needs more string handling routines but these are easy to write in later.

#### VARIABLES USED

RM = 1 Maximum No. of records.

RH = Record number.

RL = High byte of record number.

RL = low byte of record number.

### Program 1 Listing

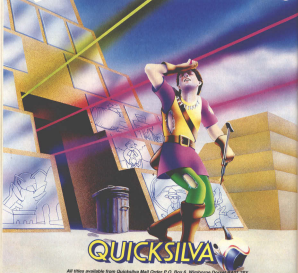
```

READY,
1 PRINT "Q":RM=44499000RM=24444
2 DIMA(4444444):REC=0
3 FORI=55200,0:FORI=50000,0:PRINT "Q"
4 INPUT "RECORD NO. READ OR WRITE R/W?":Q
5 IF Q="R" THEN
6   I=55200
7   I=50000
8   OPENA:Q,I:OPENB:Q,I:"O:REL FILE"
9   INPUT "ENTER NEW RECORD E" :I=I+1
10  INPUT "ENTER MESSAGE 1":I=I+1
11  INPUT "ENTER MESSAGE 2":I=I+1
12  INPUT "ENTER MESSAGE 3":I=I+1
13  INPUT "ENTER MESSAGE 4":I=I+1
14  R=INT (RM/256):C=INT (RM AND 255)
15  PRINT I5,"P"CHR$(I):CHR$(R):CHR$(C):CHR$(I)
16  C=","
17  PRINT I2,OR(I):24OR(I):24OR(I):24OR(I):24OR(I)
18  INPUT I5,A,B,C,B:PRINT "ERROR IS = TO "I4)
19  PRINT "PRESS ANY KEY TO CONT."
20  GET:GOTO 17END
21 CLOSE:CLOSE I:GOTO 0
1000 OPENA:Q,I:OPENB:Q,I:"O:REL FILE"
1010 INPUT "RECORD NO.":I=I+1
1020 R=INT (RM/256):C=INT (RM AND 255)
1030 PRINT I5,"P"CHR$(I):CHR$(R):CHR$(C):CHR$(I)
1040 INPUT I2,OR(I):24OR(I):24OR(I):24OR(I)
1050 INPUT I5,A,B,C,B:IF A=50 THEN PRINT "RECORD NOT PRESENT TRY AGAIN" :GOTO 1000
1060 PRINT "RECORD E "I5: " = "
1070 PRINT "1" I5 "10" I10 PRINT I2 "1" I10 I20 PRINT I3 "1" I10 I30 PRINT I4 "1" I10 I40 "1" I10 I4)
1075 PRINT "RECORD E "I5: " = "
1076 PRINT "PRESS ANY KEY TO CONT."
1080 GET:GOTO 17END
1090 CLOSE:CLOSE I:GOTO 0
READY,

```



# QUICKSILVA'D!



## QUICKSILVA

All titles available from Quicksilver Mail Order, P.O. Box 8, Winborne-Dorset SS21 7PF.

SAVO TRADER



Developed by  
MIND GAMES

1970'S  
SUBVERSIVE GAMES



Developed by  
THE GAMES WORKSHOP

WARREN



Developed by  
THE GAMES WORKSHOP

TRUCKER



Developed by  
MIND GAMES

TRUCK



Developed by  
MIND GAMES

WARRIOR



Developed by  
MIND GAMES

WED



Developed by  
MIND GAMES

WET ATLAS



Developed by  
MIND GAMES

**GED**  
GAMES EDUCATIONAL DEVELOPMENT

AVAILABLE  
SOON



WARRIOR



WED

WARRIOR

WED

SEND £1.10 FOR CATALOGUE AND FREE  
GAMES. CHECK FOR THE BEST  
DEALS. ORDER TODAY. MAX. 25% OFF  
ALL GAMES. **ARE YOU GEDDED THE  
GAME? ORDER THE  
GAME! ORDER THE**



# DATA STATEMENTS

## Great reading

TWO WELL-KNOWN AND established book publishers Corp and Addison-Wesley, have joined together to produce a series of computer books and in October they are due to release three books for Commodore owners; all written by Brett Hale, the titles are *Arcade Games For Your VIC 20*, *Arcade Games For Your Commodore 64* and *More Arcade Games For Your Commodore 64*. To keep your eyes on those book shelves!

## Buying Powers

Purchase ledger is a new business program available for the Commodore 64 from Kemp Limited. It is a high quality ledger accounting program for recording and analysing purchases and includes nominal analysis and will account for VAT. Ten menu options are available; the program can handle up to 100 creditor accounts and up to 30 nominal accounts and can deal with up to 640 entries each month. It has a full range of printout and can also be used with Commodore interfaces. It is a disc and tape compatible and can be transferred to disc without difficulty. The retail price is £27.95 on cassette and £29.95 on disc and comes with a 16 page manual. The program can be obtained from good computer shops or from Kemp Limited by mail order. The price includes VAT and postage and packing within Europe. Further information can be obtained by contacting Kemp Limited at 48 Maxwell Hill, London N16 9PN, tel: 01-444-5495.

## New stuff from PSS

MIDWAY, FOR THE Commodore 64, is a diversion from other PSS products, being a strategic war game based on the World War Two clash between Japan and the U.S. The program has been written by a 'wargamer', who has ensured that the Japanese strategy, (the player takes the role of the U.S.), is the same as that in the real battle. There are three levels, making it suitable for beginners or experts. Midway will be available on disc or cassette at £12.95 and £9.95 respectively.

## Commodore 16

THE NEW COMMODORE 16 is the ideal introduction to home computing, offering a powerful 128K RAM, full-size professional keyboard, 121 colours for high-quality graphics and sophisticated sound capabilities. Designed with the first-time user in mind, the Commodore 16 will be sold as a complete 'Starter Pack' comprising: computer, cassette unit, introduction to BASIC Part 1 and four recreational programs — everything needed by the beginner — for £129.99. "The Commodore 16 is the perfect point of entry for anyone interested in serious home computing", said Howard Stanforth, LK General Manager of Commodore Business Machines. "We've packaged it in a Starter Pack because the VIC-20 has already proved that that's what the public wants and we believe that in this form it offers the best deal

## Getting Stacked

STACK COMPUTERS HAVE dropped the price of their programmers aid cartridge for the Commodore 64. Their cartridges which provide everything from extensions to BASIC, through fast tape operation and a fast assembler, range in price from £25.00 upwards. When combined with their unique 4 slot motherboard, which allows use of programmers aid and games cartridges, without powering down, Stack give the standard features of the

64 a very powerful boost.

For those of us with smaller budgets, the Stack 100 range now includes a cassette based complex. Priced at £14.95 inc. VAT, the end-user can now achieve most professional results with his own BASIC programs.

More details are available from Stack Computer Services Ltd, 296-298 Derby Road, Bootle, Liverpool L20 6LN.



## Spritely, spritely

GO-SPRITE IS A VERSATILE, easy-to-use sprite editor which makes full use of the extensive facilities on the popular Commodore 64, available from Microsoft.

Among the many features of this Microsoft machine stretcher is the fact that all program facilities can be operated by joystick alone. The Program also features light pen and keyboard control options.

Go-Sprite gives the user ease of use with icon driven commands — on-screen symbols give one-touch commands for all program features, simple animation of up to 32 sprites with 255 frames, easy handling of multi-coloured sprites, overlap of up to seven layers, user definable keyboard, sprite data files on disc and tape with data display option, and sprite editor.

Go-Sprite was written for Microsoft by Bright Green Software.

# DATA STATEMENTS

## Great Eight

ACTIVISION IS NOW producing games cassettes for the Commodore 64 home computer. Five titles off the production line are Beamrider, Decathlon, H.E.R.O., Zanj, Toy Bizarr, River Raid, Pitfall I and Pitfall II. Of these titles, Beamrider, Decathlon, River Raid and Pitfall I are new versions especially adapted and enhanced for the Commodore 64, of Activision's top-selling 1983

titles. Pitfall II and H.E.R.O. are Commodore 64 versions of Spring 1984 releases which are already high in the charts, and Zanj and Toy Bizarr are brand new Summer 1984 titles which are being launched simultaneously for the Commodore 64 and other systems.

Zanj is claimed to introduce the software entertainment industry's first mystical game of intuition. Begin at the source and feel your way through the unknown, correcting path after path



and you've mastered the mystery and your senses. Zanj, designed for Activision by Matthew Hubbard.

In Toy Bizarr, it's midnight in the toy factory. Peaceful Quiet. All of a sudden the balloon valves open. Camps of innocent toys begin leaping from level to level taking over the

top shop. Can you stop them before they capture you.

Toy Bizarr, designed for Activision by Mark Turnell.

All these new Activision cassettes for the Commodore 64 retail at \$19.99 including VAT and are available from video games and home computer software outlets everywhere.

## Cortek rides again

FOLLOWING THE SUCCESS of 'Cortek & the Microchips', Commodore have launched an educational package for the 64, 'Cortek & the Kryptobytes'. In the new program, space hero Cortek, together with his Microchips, again demonstrates a new and unusual way of learning to program in Commodore BASIC.

The first program has already been translated into at least five languages and 'Kryptobytes' is designed by the same team of three schoolteachers from Southern England. It is accompanied by a full colour story book which contains a 'light file' to teach additional programming skills and programs to type into the computer. An acetate sheet 'map' is used as an overlay in the book to help the student create his own designs and make it easier to POKE them onto the screen. The scenario of the story centres on the planet Jura where the Kryptobytes have established a data centre to store information they have

acquired from other planets. However, a mysterious force begins to extract the information. Cortek and his Microchips are called to the rescue and, with you, have to try to neutralise the force before the safety of the whole universe is jeopardised! Cortek and the Kryptobytes teaches students the construction of programs using the ROM and LIST commands; the storage of data and the use of the READ command; the application of FOR and NEXT, both as a delay and in nested form for loops. In addition practice is given in simple program editing; the use of GET and INPUT, as well as a host of other programming features which take the student to an advanced level.

Cortek and the Kryptobytes is designed for 10 to 13 year olds, although it is suitable for younger children with parental assistance, while older children and even parents will find it informative and great fun to use.

## Pets can microwave

COMMODORE PET USERS can now communicate with the Microwriter — the portable hand-held word processor with a unique and extremely simple to use keyboard of just six keys. Microcomputer Services, an approved Microwriting Centre, has developed the software program 'Speakeasy', which allows two-way transfer of text between PETs and Microwriter.

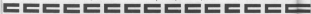
Now PET users can transfer text to their data discs for storage, merging of files or for printing out a convenient format. Documents can also be retrieved from the PET and entered into the Microwriter's memory for reference, updating or amendment. The Microwriter can also be used in a networked environment.

An interface lead, enabling communication between the PET, which has 800 connectors, and the Microwriter's in-built RS232 is available from Micro-Computer Services.

Due to Microwriter's total portability, it is

especially suited for any situation where text needs to be written away from the office — whilst travelling or for meeting notes, or even at home out of office hours. Within the office, Microwriting provides an invaluable method of producing written text without recourse to dictation. Neither do notes need to be handwritten and later transcribed by a typist. The Microwriter user has complete control over content, format and amendments, with material being written whenever and wherever convenient. Hardcopy printout can also be obtained directly from the Microwriter linked to a printer.

The 'Speakeasy' program is available from Micro-Computer Services, priced at £140 for two way communications, and £95 for one way. The 'link' will also work with all Commodore models, and details are available from: Microwriter Ltd., 31 Southampton Row, London WC1. Tel: 01-831-6601.



## Sages, Supersoft

SUPERSOFT PRODUCE AN enormous range of programs and accessories for the Commodore 64 micro, for example...

Available for the CBM 64 is a truly 3-dimensional spreadsheet program, Basicalc 3. This offers a variety of features over those included in the rest of the Basicalc range, including up to 999 rows and 388 columns, 3-dimensional formulas and user-defined functions, add or subtract sheets, fast F10 command, unique VU window function, link to Easy Script or Visamate and also to Chartpak 64, simple bar charts, sample files and utilities and variable-column width. Basicalc works with all CBM printers, is available on disc only and costs £75 including VAT.

Still on the serious side of computing, Supersoft have taken on the marketing of Arco Applications' utility, Master, for the CBM 64. They have cut the price to just £69 including VAT, less than half the previous price.

Rather than produce yet another flight simulator for the Commodore 64 SUPERSOFT have gone one better with the release of INTERDICTOR PILOT, a space flight simulator! Written by a serving RAF officer, INTERDICTOR PILOT comes with a comprehensive 48-page manual which provides an insight into the latest technological developments including travel at the speed of light! In simulator mode the trainee pilot can find his way around the controls, and take part in simulated dogfights with alien craft. Many hours of simulator training are necessary before newly commissioned sub-lieutenants take part in real life sorties. INTERDICTOR PILOT is available from SUPERSOFT by post, or through dealers, at £79.95 (including VAT). A disc version is also available at £79.95.

Why not find out more about Supersoft's range of products by contacting them at Winchester House, Canning Road, Walthamstow, Harrow, Middlesex HA3 7SL. Tel: 01-881-7185.

## Getting Educated

GET READY TO READ, THE first in a comprehensive suite of educational programs, featuring **B** the Bear, is now available from Commodore for use on the Commodore 64. Written by Dr. Richard Riding, Lecturer in Educational Psychology at the University of Birmingham, and Mrs Lillian Simmons, Headmistress at Moors Mount Nursery and First School in Redditch, Wores, the program is available on cassette or disc and is accompanied by a **B** the Bear book and a parents and teachers' manual.

Other programs in this series will include Get Ready to Think and Get Ready for Numbers. The Get Ready programs will be followed by Start to... and Continue to... for the same subjects but at increasingly advanced levels. The Get Ready programs are designed to introduce children of three-to-five years of age to the initial

stages of reading. The **A4** size, full colour "**B** the Bear" book is divided into four levels, each containing a different story about **B**, and is designed to be read to the child in order to prepare him for the learning activities of the program, all of which relate to the Bear's adventures. The book also contains simple pictures to match with letters of the alphabet, a guide to the learning of letters of the alphabet, and join-the-dots and coloring pages. A smiling **B** appears on the screen when the child has made the correct decision, and a frowning **B** appears when the child is incorrect.

The manual, which includes a set of progress charts, instructs the parent or teacher on how to prepare for teaching, how to use the programs, how to guide the child through the program, and how to grade their performance. "Get Ready to Read" is available from Commodore at £12.99

## Bly Gum!

AFTER PACMAN, NOW welcome Plaque-man, hero of Tooth Invaders, one of Commodore's recent programs for the Commodore 64 home computer. Tooth Invaders demands enough computer game-ownership to satisfy the most practiced young tyro, whilst at the same time offering a worthwhile reminder on dental care — reassurance for Mum and Dad that computer games can have a useful message. As his name suggests, Plaque-man's mission is to eliminate all the plaque on tooth surfaces. He is guided by joystick and must collect a toothbrush and charge it with fluoride toothpaste to clean the teeth. He must then use dental floss for the gaps in between to complete the cleaning process.

But Plaque-man's enemy, DR. the Plaque Count, gets ever more cunning and tooth decay sets in

spontaneously, leaving only seconds for remedial treatment. The standard screen shows set of eight teeth — four up, four down — and there is a zoom screen to give a close up view of each tooth as Plaque-man goes to work. Plaque-man has three lives in each game and there are nine skill levels, plus a range of anti-plaque tricks any dentist would be proud of. When all the teeth are completely clean a fluoride rinse cloud descends and the game moves onto the next skill level.

The action is accompanied by music — for triumph or disaster, as appropriate — and there are special sound effects such as brushing noises and "bleep alert" warning pips when plaque building becomes critical on untreated teeth. Available on cartridge, Tooth Invaders is priced at £9.99.

## FEDERAL INTER STARBANK PATROL FORCE



PILOT'S INSTRUCTION MANUAL

## INTERDICTOR MK III

# DATA STATEMENTS

## Software Support

HANDIC SOFTWARE ARE obviously firm believers in the potential of the Commodore range of computers used in both the home and in business. Fairly recently they have produced a range of new programs specifically designed to boost the commercial applications potential of the 64, including such titles as CALC RESULT, DIARY 64, MON 64 (machine code monitor on cartridge), RII, 64 (for such duties as activating burglar alarms, locks, telephones, central

heating, lighting, etc.), STAT 64 (for statistics and graphics displays), GRAP 64 (for studying complicated mathematical functions by their graphs), SUPERBOX 64 (an expansion board and parallel IEE for the 64), PIT-SWITCH and VIC SWITCH (a multi-user system for the VIC 20 and the 64).

Handic do in fact have a catalogue of all their products, so why not contact them and see what they have to offer? They are based at 5 Albert Road, Crowthorne, Berkshire RG71 7LT.

## Getting Yourself

### Re-arranged

Based in Horsham Anagram Systems produce a range of Commodore software aimed at the small to medium sized company. The latest in their range of business accounting packages is Cash Book 64. Based on standard double entry book keeping practice, the system enables the user to maintain accounts to Trial Balance by the simple medium of recording Sales/Income, Purchases/Expenses, Assets and Liabilities. The Trial Balance may be produced by month end or year to date. The system will also record monthly budget figures and gives a budget to actual performance comparison. From a mass of only seven prompts, Cash Book 64 builds a complete accounting record including VAT summary, Audit Trail, Budget Summary and Performance as well as Trial Balance and Cash/Bank book. Cash book 64 costs £75 including VAT.

Going up-market somewhat, is the Integrated Accounting System running on Commodore 700 and 800 series micro which has now been enhanced to include Sales Order Processing and full optional Factoring. The IAS is based on standard double entry book keeping practice and comprises Sales, Purchase and Nominal Ledgers with Integrated Stock Control, full Purchase Order Processing and full Sales Order Processing. The full package allows some 200 book keeping operations to be conducted by means of just over 50 screen prompted functions which make it an ideal system for first time users or companies who do not have access to trained accountancy staff. Version 4 with Sales Order Processing and Factoring costs £1999 plus VAT.

Anagram Systems are at 80A Queen Street, Horsham, West Sussex.

## Back To School

ASK HAVE ADAPTED SEVEN of their highly successful range of mathematics and literacy programs for the BBC Model B Micro to run on the Commodore 64. Available on cartridge at £8.99 is the Autumn 6 Number Painter for the 7-14 year age group. It is an arcade style game with 12 levels of difficulty to test your mental arithmetic. Three cassettes at £13.99 comprise Words, Words, Words; it links 10 different words together and the aim is to improve the spelling and vocabulary of children (7-10) who are learning to

read. Number Chaser develops estimation skills through an exciting arcade game. This should be around at Christmas and is £9.99 as a cartridge.

Also available at the moment is Facemaker (building faces identical to a) at £4.99, Number Puzzler (thoughts and crosses with a mathematical bent) at £4.99, Hide and Seek (for improving reading skills and practice short term memory retention) at £3.99 and Let's Count (to teach young children to count) at £3.99.

ASK specialise in educational software and the products are marketed by Commodore themselves.

## Price War

THEY HAVE ANNOUNCED an entire range of price swings on many of their best selling games for the Commodore 64 and VIC 20. In some cases, Audlogic has slashed prices by more than half. The price reductions include:

Commodore 64 software: Motor Mania (£8.99 to £3.99), Renaissance (£8.99 to £3.95), Grandmaster (Cassette version) (£17.95 to £8.99) and Grandmaster (disc version) (£17.95 to £11.99).

VIC 20 Software: Borneo (£7.99 to £3.95) and Grandmaster (on cassette only) (£17.95 to £8.99).

Audlogic has launched MONZO for the Commodore 64 featuring brilliant full-screen graphics and some of the most devious and vicious monsters ever devised.

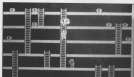
MONZO puts the player

in the role of a workman whose job it is to collect boxes from different levels of the screen. The various levels are connected by ladders, of which several are available giving the player a choice of strategy. Unfortunately, the monsters which guard the boxes can also climb.

A unique factor of MONZO is in the monster's ability to react to the game with intelligence. The monsters do not just wander about the screen but actually try to trap the player on a particular level.

Once the player has successfully collected all the boxes, he progresses to the next level with more valuable boxes and more, even meaner monsters.

MONZO is available on cassette at £3.99 from Audlogic or via the nationwide dealer network.



## CompuNet

COMPUNET IS THE NAME of a new service which will allow home computer owners to buy software, use financial and information services and even do everyday shopping direct from their own living rooms. They will be able to do all this using a simple telephone connection. The key item will be a small box called a Modem, which plugs into the computer and a telephone socket and allows the ordinary home computer to reach and communicate with the COMPUNET service. Commodore have developed the Modem especially for COMPUNET and an estimated 500,000 Commodore 64 owners, will have a chance to link up by the end of this year. COMPUNET will offer several key benefits that establish it as a premier service. Firstly, there is convenience. Users need only switch on and enter the name of 32 COMPUNET telephone numbers spread around the country, the modem's auto-dialler and software, contained in its built-in memory, will complete the connection in seconds. And once linked to the system, the software design means the user only ever has to use two or three keys to select and display material.



A second key advantage, and a remarkable development in itself, is the ability to offer reliable 2-way transfer of information and programmes, by cutting out the adverse effects of poor telephone lines. Thirdly, uploading as well as downloading is available. That is to say software and information can be transferred in both directions between the home computer and COMPUNET service. COMPUNET will offer quality software products at discount prices in an area called Software Park. Special features in the Modem and COMPUNET service prevent the software being run on any computer other than the one to which it was downloaded. By eliminating the risk of illegal

copying the software protection facility makes the COMPUNET tele-software service more attractive to suppliers of popular products. Another facility expected to interest computer clubs and hobbyists is an area of the service called 'The Jungle'. Subscribers can upload software and information they have created themselves directly into the jungle. Other subscribers can then download it on a free or chargeable basis. Seller's earnings, less a commission are credited to their COMPUNET account and once a quarter the balance is credited or debited to their Bank Account using a Direct Debitting Scheme. Other facilities include an electronic mailbox service,

Commodore product information and alternative software which will allow the Modem to link with viewdata services.

The Modem costs £79.99 and, as an introductory offer, the first year's subscription to COMPUNET will be free to purchasers. Subscribers have free access in evenings and at weekends and can also use many parts of the service without charge. Later in the year, after an introductory period to familiarise COMPUNET to use and service provider requirements, further mailing and information services will be brought on-stream. These will include catalogue shopping, home financial and insurance services, educational products and information facilities for the domestic environment. COMPUNET has been developed and introduced by Commodore Business Machines (UK) Limited and ADP Network Services Limited. COMPUNET will be operated on ADP computers and users will be able to link in via a local call to the ADP Network in twelve locations throughout the UK.

Note: for more information on modems, see the relevant article in this issue. Watch future issues of Your Commodore for reviews of modems themselves.

## Tandata modems

TANDATA MARKETING have two 'smart' modems, the TM110 and TM120 together with Micropak for the Commodore PET and VIC 20. The TM 110 operates on 1200/75 baud 'V21' full duplex mode and is able to store and access data in its own CMOS RAM. It features an eight-telephone number store allowing up to 16 digits per number with a 32-digit reference comment for each number for ease of location and auto dial of stored numbers with auto recall of unobtained numbers. Up to eight separate ID numbers and

passwords each with up to 16 alpha or numeric characters can be permanently and securely stored in the modem's own memory. Also the sometimes lengthy log-on procedure is reduced to a single key function.

The TM 120 multi-rate modem includes all the features of the TM110 and in addition of 1200/75 baud full duplex, also offers 75/1200 baud full duplex and 1200/1200 full duplex to allow two micros to 'chat'. Both modems allow access to Prestel and its 380,000 plus pages of information, electronic mail

and telex facilities, so Telecom Gold, Micronet and Prestel compatible private viewdata screens. The TM120 costs £165 plus VAT. Software to allow a micro to act as a viewdata terminal is now available for the entire range of Commodore micros. For the Commodore 64, software is available both on disc (£37 plus VAT) and in ROM (£42 plus VAT) and gives full Prestel emulation with colour display and the inclusion of an offline message editor. The micropak includes a plug-in card and comprehensive instruction manual. Soft-

ware for the PET range is available on disc giving full Prestel emulation except for Keypad, Flash, Double Height and Separated Graphics. Displays in mono and off-line message editor (included) the pack comes with a plug-in card and a graphics chip, for the VIC 20, the software is on cassette with Prestel emulation as for the PET and costs £27 plus VAT.

More information is available from Tandata Marketing, Albert Road North, Malvern, Worcestershire tel: (05645) 68421.

# DATA STATEMENTS

## New concept

THE CONCEPT KEYBOARD is an original data input device offering many advantages over other keyboard systems. The user or programmer has complete flexibility in simply assigning the fixed codes generated from the CONCEPT keyboard matrix to characters, words, shapes, objects etc. set out on an overlay, thereby giving the user the most efficient keyboard layout for a particular application. The underlying principle is a touch sensitive input selection area-sensitive over the entire matrix area — each touch call

producing its own unique output code.

Three versions of the CONCEPT keyboard are available:

The A1 unit is 227 x 315 x 25 mm with a 16 x 8 matrix giving 128 touch cells each cell measuring 18 x 18 mm. The two A2 units are 315 x 435 x 25 mm. The A3-128 has a 16 x 8 matrix each cell measuring 24 x 30 mm. With a 16 x 16 matrix the A3-256 has 256 touch cells each 24 x 15 mm.

The codes output from the CONCEPT keyboard are incrementing binary, corresponding at cell 0 with Hex7 to HexF (F4) at cell 127 (255) compatible with the majority of common code systems — ASCII — ISO



CONCEPT KEYBOARD

— IBM/PC for example. Both positive and negative strobes are available, making the interfacing to any computer system very simple. 2 cell rollover with 4 cell lockout is standard on all CONCEPT keyboards. The standard output is 8 bit parallel, but 4 serial output options are available — Passive optically isolated 20mA current loop — DA 8530C/834 — 8542 — 8543. Each is supplied with a switch selectable baud

rate generator, covering the range 90 to 5,600 bauds. The power requirement being very low is usually taken from the host computer...5 volts at 20mA.

The CONCEPT keyboard is easily interfaced into any computer system and a range of ready made-up interface leads are available for the majority of popular micro-computers including Commodore.

Prices start at £69.00 for the A4 CONCEPT keyboard and £149.00 for the A3-256.

## ORC Attack

CREATIVE SPARKS HAS introduced a Commodore 64 version of its highly rated game *Orc Attack* which is already available for Spectrum and Atari computers.

In *Orc Attack*, you must defend your castle against the rampaging hordes of Orcs and their lethal crossbows. To defend yourself, you have a broadsword, rocks and your ultimate weapon, vat of boiling oil. *Orc Attack*

features four screens of action with excellent sound and original graphics, and is available on cassette for £7.95.

Also available from Creative Sparks for the Commodore 64 are two other games. *Black Hawk & Shurp*, *Black Hawk* is a game of action and strategy, featuring the world's deadliest bar laser aircraft, and *Shurp* is a comedy, cave dwelling character with the biggest appetite ever.

Creative Sparks games are distributed in the UK through THORN EMI Software Distributors and will be available in all principal retail outlets. Some products will also be available mail order.

Further information is available from THORN EMI Computer Software, Creative Sparks, Thornton House, 296 Farnborough Road, Farnborough, Hants, Tel: (0332) 543333

## Definitely not rubbish

CERTAINLY A GAME TO have least respectability to rank in the program from New Generation Software called *Trashmen*. Originally for the Spectrum, it is currently being marketed in the USA for the Commodore 64 under the title of — wait for it — *Garbage Gobbler*. The only change is the name — the game itself is the same as that which was the subject of a recent competition for journalists, won by Computing Today's Editor, Peter Green. He came first playing the game on the Spectrum and won a weekend in Paris; he's just sitting waiting now for the chance to travel Stateside and throw down the gauntlet to our colonial cousin!

## Mole in a hole

Wanted: *Misty Mole* is a new game from Grenlin Graphics for the Commodore 64 featuring a new micro character, *Misty Mole*. His adversary is the awesome *Arthur* (who'd have guessed?), safely ensconced in his fortified castle and protected by a bunch of living pickets. *Misty* tries to defeat *Arthur*

by sneaking into the pit and snatching coal and as a final effort, steals the secret ballor papers and vortasting scroll. *Arthur* and his living pickets have a variety of means at their disposal to prevent *Misty* from succeeding. I personally haven't won the game in action, and I'm all for innovative games that break away from the usual alien killing, but the idea seems to me to be in bad taste —

the fact that this game was written by a miner's son only causes me more worry about the current mining dispute, not to mention the direction of computer games, regardless of how good they may be technically. The game itself is priced at £7.95 and for every game sold, Grenlin Graphics will donate 5p to the Miners' Welfare Fund. Can we not opt out of this political levy?





**Business software can be expensive and doesn't always cater to all your needs. This series by Grahame Davies will give you an insight into how to write your own programs to suit your requirements.**

# DOING IT YOURSELF

THIS SERIES WILL ENABLE even the most experienced BASIC programmer to tackle the problem of storing and retrieving data for personal applications. It is based on the CBM 64 connected to a 1241 disc drive but all the techniques and logic used will be applicable to all Commodore machines, even if you only want to write games. The programming principals and discipline set out in this series will be of great use.

This series will not include a complete listing of a record keeping system but will provide utilities and techniques as well as explanations on how to successfully use disc files (and when to use each type). All the code has been deliberately left in BASIC but the routines may be converted into machine code if required. I suggest you read the articles and then practice some of the logic and ideas perhaps writing little test programs. Having satisfied yourself that you understand the principle (or at least have got your routines to work) then have a go at writing your own program. Just one other thing to remember — this series is designed for you to be able to write several systems, not just one, so if you find some of the explanations complicated or long winded do not be put off as you will really benefit in the end.





## Setting out the programs

The first things to note are some of the do's and don'ts of BASIC programming. Firstly, bear in mind when writing that not only have you got to debug your program, but you may discover a bug in several months time, so make the code readable. Also remember that having got it working, you may want to modify it to store more data and if it is well written, this will be far easier to do. Do not try to put too many commands on one line. Having said that, it is unnecessary to put just one command on a line but with a little thought we can soon see some obvious guide lines. Here are a few examples:

```
10 rem your routine is here
20 rem and here
100 c = c + 1
200 if c < 5 then 10
```

In this case, we alter the variable 'c' and want to immediately do a test on it to know whether to end or not. Having put the test on a new line, we have effectively separated the two. If we come back to this code and add a line 140 for example, although the code may still work the test has become further away from the increment making it more difficult to follow. Here is another:

```
100 for i = 1 to 10
200 if i < 5 or i = 6 then 200
240 rem your routine is here
300 next
```

In this example we want to go round the loop ten times but ignoring numbers five and six. The test wants to be on the same line as the loop, not loop because there is no reason to have any code between them. Doing this will make the code safer and less prone to bugs if something is added.

Putting so many statements on one line saves memory and marginally speeds up the code but these advantages are insignificant and are far

outweighed by the disadvantages — do not try to be memory efficient in this way.

We will now examine the REM command. This command is an essential programming tool but must be used sensibly:

```
100 a = 0: rem initialise a to zero
```

This is hopeless! The REM says exactly the same as the code. Much better would be:

```
100 a = 0: rem number of times
```

This tells us what 'a' is being used for. The code itself tells us that it is being set to zero, so why put it in a REM? Experience will show when

now let's decide what variable names to use. I advise having temporary variables which can be used to pass data to a subroutine and to be received back from subroutines. Also for common FOR...NEXT loops, try to use the same variables and avoid using any variable that you use in a subroutine. This is a common problem which causes FOR...NEXT loops to suddenly or never end!

For all the new flagsets, that is to say all new variables that are specific to the program being written and not in common subroutines, use two letters (if more) because this is more descriptive and easier to follow. 'AM' for amount is obviously quite clear but

then amend your program so that it is neat and easy to follow. In this way the benefits will become more obvious.

## Getting framed

If your first program is successful and you want to write more, you will need a program frame containing your favourite tried and tested routines. I will call this program a NET. In your net you will need subroutines for screen handling, disk handling, error messages and all the little things that you only want to write once. After a while you may add to them or improve them making them more elaborate. By keeping these in a net you



to use REMs and what to put in them. For now, just remember that as you write a bit of code you understand it fully but in two months' time it will be harder to follow, so use some REMs telling you what variables are expected for the routine and what variables are returned. REMs do use memory and marginally slow the program — ignore these facts and use as many as you like. Try to format them neatly on the screen — leave spaces if necessary so that they may be easily read:

```
100 a = 0: rem          exchange rate
200 a = 0: rem          number of things
300 for i = 1 to 100: rem wait loop
```

'99' for amount is clearer. Using more than two letters makes the code even clearer ('AMOUNT' for amount) but is unnecessary and has some disadvantages — it uses memory, it leads to more syntax errors as BASIC may discover a reserved word such as 'ON' in 'MONTAINS' and in any case BASIC only stores the first two characters of the variable name so the variables 'AMOUNT' and 'AMEND' are the same.

When you have completed your program, go back and read the above,

will have easy access to them and the subroutines will be in the same place in every program making each program easier to understand. As your routines get more advanced you will find them becoming like a set of advanced BASIC commands and you will really be writing up a tailor-made high level language (well higher than BASIC anyway).

The line numbers you use are a matter of personal preference but there are still some things to remember: do not start at line zero or one (or ten etc) — you can have line numbers up to 65535 so make good use of them. Any routines listed in this series will have line numbers between 1000 and 2000 but you may relocate them (changing all the relevant jumps etc) where you wish.

Commodore have  
 launched a new  
 business machine with  
 a full 128K of RAM  
 and some interesting  
 features. Will it appeal  
 to businessmen?  
 Simon Dismore  
 investigates.

# IS BIGGER BETTER?

COMMODORE'S TOP-OF-the-range machine, the 8296-D, is an interesting mixture of the old and the new. Underneath the "milked ice-cream" casing is a system which is fully compatible with the earlier products in the 8000 range, but carries a full 128K of RAM, organised in blocks of 32K.

"Very nice", we thought, is our evaluation machine was unloaded from the container lorry (Commodore have not skimped on protective packaging). Our first question was "what can we do with all this memory?" — and here the answers were a little disappointing.

You might wish to use a computer either for developing your own



## Memory blocks

Notes to

Bit 7: ENABLE EXPANSION MEMORY 0 disabled (default on powerup) 1 enabled	Bit 2: SELECT MEMORY Addresses 8000 - 8FFFF 0 block 0 1 block 1
Bit 6: I/O PEAK THROUGH Addresses 8000 - 8FFFF 0 disabled 1 enabled	Bit 1: WRITE PROTECT EXPANSION RAM Addresses 8C000 - 8FFFF 0 not protected 1 protected (I/O is not protected if peak through is enabled)
Bit 5: SCREEN PEAK THROUGH 0 disabled 1 enabled	Bit 0: WRITE PROTECT EXPANSION RAM Addresses 80000 - 80FFF 0 not protected 1 protected (screen is not protected if peak through is enabled)
Bit 3: SELECT MEMORY Addresses 8C000 - 8FFFF 0 block 2 1 block 3	COM reserve bit 4 for future applications



EXP. 8296 MEMORY MAP (KB)		
0	MEMORY BANK 0	0
100	MEMORY BANK 1	4000
200		8000
EXPANSION RAM (16000 B to 1)		
MEM...		
500	EXPAN	8000
600	EXPAN	16000
700	EXPAN	16000
800	EXPAN	16000
EXPANSION RAM (16000 B to 1)		
MEM...		
100	EXPAN	16000
200	EXPAN	16000
300	EXPAN	16000
400	EXPAN	16000
500	EXPAN	16000

Figure 1b

programs, or for running other peoples' programs, 99% of purchasers fall into the latter category, and we were surprised to find that there was no software available that took advantage of the extra memory. One might expect that products might be in short supply for what is a relatively new machine — but the complete absence of appropriate programs is quite baffling. What can your friendly Commodore dealer demonstrate except the existing range of software, none of which was written with the 8296 in mind!

First in the pipeline, said Commodore, would be Hardik Software's CalcResult spreadsheet program. CalcResult for the 8296 is still a few weeks away from completion, but the authors very kindly offered to share their experiences with us, even to the extent of getting their programming manager to call us from Sweden.

They tell that the extra memory was "useful but not essential" for their sort of program. The benefit is that larger spreadsheets with more data can be manipulated in the memory of the system, while maintaining compatibility with data on earlier machines.



Once CalcResult was up and running, their word processing package might follow. An interesting possibility would be the use of the expansion RAM as a "clipboard" to hold results while programs were being changed — the first step towards an integrated environment of the sort offered by AppleWorks and Lotus 1-2-3.

Hardik also pointed out that they had to go into the machine with a soldering iron to strap the appropriate internal jumpers so that they could select between expansion memory and firmware under software control (once strapped, a USER call can be used to enable and disable any resident EPROMs or ROMs). Your dealer could probably do this for you if necessary. We would be interested to hear any other experiences in this area — perhaps Commodore should add switches which can be used to configure the system with less trouble.

Because the expansion memory is configured in fixed 16K blocks you would have to write your own memory allocation routines before attempting to access the memory for any sophisticated application (the clipboard idea would probably just use the

memory as an ultra-fast disk drive). If you are converting a word processing or spreadsheet package, you will probably have existing routines which would work with little modification. Otherwise, you should be careful to design your allocation routines which read or permanently in the lower 12K main memory at any early stage of development. One final word of advice from Hardik: interrupts are tricky to programme, jumpers and interrupts aside, selecting memory blocks is easily

performed by POKEing your selection into the memory control register at \$7F00 (see Figure 1).

The 8296 also gives you four "pages" of video memory, instead of the usual one; this could be very useful, and is much easier to program. Commodore suggests that you might use the additional screens for help screens and menus, though once again strapping might be required if you have fitted extra EPROMs.

The alternate screens are selected through a two

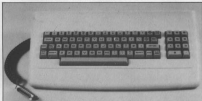


line POKE sequence. Unfortunately, the PRINT statement in the resident basic (4.0) appears to work with only the default screen (Commodore number the screens from 1 to 4 in their documentation, but you will probably prefer to use a range from 0 to 3 for ease of programming — in either case, the default screen is the lowest number). This presumably means that you have to use the POKE statement to put characters onto any of the alternate screens. Unless we missed something in Commodore's (as that wonderful, as usual) documentation, this seems to be a major stumbling block to productive use of the extra screen memory.

Be that as it may, we did feel that the multi-page screens were worth the small amount of RAM invested in them. We have reproduced some simple routines which we wrote for screen handling. Figure 2 shows the POKE sequence for selecting between the four screens, while Figures 3 and 4 illustrate a full screen copy and a partial copy respectively. We learned (to our cost) that the 8296 does NOT take kindly to errors in POKE statements, so we have shown some ways in which errors can be detected and trapped (see Figure 5).

The most exciting application for multiple screens is almost certainly "windowing". You can try this with our partial screen program. The 8296 clears screen 0 on power-up but leaves the other screens with random values, so you would probably wish to clear them to blanks using a looping POKE 32 space). The alternate screens are NOT cleared when you press reset, so any data you put in there is safe from everything except a power out.

The resident BASIC is very slow (over 10 seconds to copy a screenful even with all remarks removed and multiple statements on a line), so you would almost certainly want to re-code any windowing routines



into machine code once you have them working properly. Other amusing experiments include reversing rows and columns and simple text editing using the alternate screens as buffers.

If you are already a Commodore user and own a set of 8050 disk drives you might find the 8296 in its disk-less version was an interesting machine. It certainly doesn't seem

worth replacing the 8096 with an 8296 just to get the benefits of additional memory for which no software is yet available.

If you don't already have a machine, should you consider the 8296-D? Well... Commodore have produced a respectable but not exactly exciting product, which couldn't be regarded as 'state of the art' in any respect.

**Disk:** Single-sided, single-

density 8050-compatible disks offer approximately 900K on each built-in drive, a figure which compares well with most other 5 1/4" disks. Maximum file size under DOS 4.0 is 198K, and there is room for up to 224 directory entries, both of which should be satisfactory for most purposes.

**CPU:** The 8296 certainly cannot compare with advanced 16-bit products

## Screen handling

ROUTINE TO COPY PARTIAL OUR SCREEN TO ANOTHER  
 (APPLIC. FOR "WINDOWING" SOFTWARE)

```

1300 rem Subroutine to copy a partial screen block
1310 rem On entry, variables select source 'sr' and destination 'td' screens
1320 rem variables 'x1' 'y1' 'x2' 'y2' give source coordinates
1330 rem variables 'td' 'td' give destination starting point
1340 rem variables 'sr' 'td' as in lines 1120,1130
1350 rem Loop 'x1' etc to ensure top left, bottom right...
1360 goto 14 if x1=0 then x1=x2: y1=0
1370 goto 14 if y1=0 then y1=y2: y2=0
1380 rem Variable 'x1' etc...
1390 if (x1=0) or (x1=79) or (x2=0) or (x2=79) then chr$(149)?: goto 1400
1400 if (y1=0) or (y1=19) or (y2=0) or (y2=19) then chr$(149)?: goto 1400
1410 rem Calculation of 'td' 'td' to ensure no illegal point
1420 if (x2=0) or (x2=79) or (y2=0) or (y2=19) then chr$(149)?: goto 1400
1430 if (x1>=x2) or (y1>=y2) or (x1<0) or (x1>79) then chr$(149)?: goto 1400
1440 rem Calculate parameters 'sr' and 'td' as defined in lines 1140,1150
1450 chr$(149)?: print$(0);: print$(0);: rem Absolute source start address
1460 chr$(149)?: print$(0);: print$(0);: rem Absolute destination start address
1470 rem Now copy line for line
1480 for y1=0 to (y2-y1)
1490 for x1=0 to (x2-x1)
1500 poke(x1+1),peek(x1+1)
1510 next x1
1520 address=0: address=0
1530 next y1
1540 rem Copy completed
1550 return
  
```





like the 8000 series, it runs slowly and has a limited addressing range. You might start to worry about its limitations after a while — it depends whether you expect that your applications will grow substantially within the lifetime of your purchase.

**Screens:** Contrast and character set are both good, and you can tilt and twist the screen as required. However, we did find that the pronounced curvature of the screen and the high-contrast phosphor took a bit of getting used to. This is simply a matter of individual preference, rather than a serious criticism.

The review machine tended to whine like an old monochrome TV — something which your dealer would presumably adjust before delivering the system.

**Keyboard:** The keyboard has a very 'springy' feel which would probably irritate touch typists at first. Keys must be depressed fully before a character is sent, so there is little risk of mistyping. We did feel that Commodore should have provided a full four-key cursor pad, rather than using shifts to generate cursor left and up/down opposites, given that this product is obviously intended for the serious business market.

We also thought that there should be an Inter-key on the numeric keypad (generating a carriage return character). There is certainly room for these features on the detachable keyboard unit, which is one of the largest on the market.

As we go to print, Commodore are preparing plans for promoting the 8296-D from September

onwards. The price of the system will be £1698 (excluding VAT for the standard two-drive configuration). This price will include three 'bundled' software packages covering the main applications areas. These consist of SuperScript, a word processor and spelling checker; The Manager, a database filing package; and CalcResult, a spreadsheet program. Bought separately, these would cost about £308. This makes the 8296-D reasonable value for money at the moment — though we are a little concerned that it could lose ground to some of the more up-to-date 16-bit products which will appear over the next six to nine months.

Handic's upgraded CalcResult will also appear in September, in a version which is compatible with all 8000 series machines

(for further details contact their extremely helpful team on 0144-778800).

As to availability, Commodore tell us that 250 dealers throughout the country will be stocking the 8296. No new peripherals are being launched for the machine, but three compatible printers are available. Two are tractor feed dot matrix types with 60 cps/80 columns and 120 cps/132 columns respectively, while the third is a daisy-wheel tractor feed printer working at 40 cps/136 columns. If you want the diskless version it will cost you £880 (excluding VAT).

Finally, a message to Software Houses who are planning to use the expansion memory or the paged screens. Please drop us a line and tell us what you are up to, so that we can follow the progress of the 8296 in future issues!

## Error Trapping

COMMON ERROR TRAP ROUTINE TO DISPLAY MESSAGE ON PAGE 0

```

9090 rrr Error trap with message in variable 'err'
9091 rrr Display diagnostic
9092 print "*** ILLEGAL"jerr" ***"
9093 rrr Select page 0 for viewing
9094 sp=0; goto 1300
9095 rrr

```

Our fearless reviewer,  
Dave Crisp, gives you  
his general view of  
Superbase for the  
Commodore 64.

# SUPERBASE OVERVIEW

ON MY COMMODORE 64 I use the SILICON OFFICE and on my 55-64 I use SUPERBASE. If you have never used programs like the above before, here is a brief description of what they are used for.

In their basic form they are best described as a pre-written database program; their power however comes when you want to cross reference information from one file to another, input new information and to generate a REPORT compiled from information from one or more files. Imagine for example three drawers of a filing cabinet. Each drawer contains a set of files,

eg. Drawer 1 - Names and addresses

Drawer 2 - Personal file (salary, car etc.)

Drawer 3 - Work record.

If the company wanted to send a standard letter to all employees who satisfy a particular set of conditions it would normally be a long and difficult job checking and cross checking information from the three drawers. With a program of the above type however a small sub-program could be written to examine all files in turn and create a list of all the people who meet the requirements. The list could then be used to personalise the stock letter which would have been created on a word processor. That is one use of a program like the above. Other uses are mailing lists, patient records, invoicing etc.

A program which is used to fulfil a specialised function through SUPERBASE and programs like it is called an APPLICATION

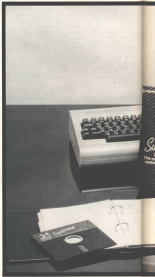
and it is often possible to buy an applications package to run with the database management system.

## Superbase

I will not try and describe how to use Superbase as it is a very comprehensive program. I will just give an overview of it and a summary of how I have been able to use it. I had already learned to use the SILICON OFFICE and so when I first came to use SUPERBASE I was pleased to see that there were many similarities. Within a couple of weeks I had produced a stock control application linked to an electronic till drawer which could cope with about 700 stock lines and produce daily sales lists, low stock items, price lists, etc, etc. SUPERBASE can be used NAKED, by this I mean that it can be used as a menu driven database with all commands entered direct. By using the menu commands you soon get the idea of manipulating data in files and it is then a simple job of linking these direct commands together to learn a program.

## With discs

In use with a single disc drive it is possible to have up to 15 separate data files/disc and also any programs that you may have produced. The number of records in each file is in theory unlimited but it is of course restricted by the 170K restriction of the 55-64 drive. It is possible and also easy to use two drives and using this system I have not yet found any limitations. With two



drives it is easy to make backup copies of your data. Initially I found I was not bothering to take copies of my data but the dreaded day came when with the help of a split cup of coffee about three weeks' worth of data had gone. One of the interesting aspects of SUPERBASE is that it will link up to EASYSCRIPT and so personalised letters and documents are reasonably easy to create. I say

reasonably easy to create because this is not a program that you would be able to just load and run — you would need to spend quite a few hours learning to link and manipulate files and to program effectively.

#### Basically speaking...

The language used by SUPERBASE is a mixture of BASIC plus some of its own commands. Its own

commands can be entered in an abbreviated form or in a sentence, eg. to create a list of all records from a particular file you can write into a program 'display all records from file name' and to create a specialised list you can simply say 'find list where (town) = (town name)'.

This is the easy way to input commands but it is effective and once you have the idea you can input commands in an abbreviated form. The abbreviations are logical and are still meaningful when you go back to a program some time later.

There are built in pages called help screens which you can call up and these pages give you the syntax and use of most of the important commands available. On the whole it is user friendly program which is only limited in use by your imagination. Some of the important features of file manipulation are listed below with a brief explanation of their function.

1. **BATCH.** This allows the whole file to be updated automatically for instance if the VAT rate changed it would be possible to update each record with a simple list of commands.
2. **FILE SIZE.** It is possible to increase the length of a record without losing any data. This is a very useful feature and one which should be available on any good database.
3. **FIND LIST.** This allows you to create a specialised list of things which satisfy a given set of conditions. These can be as many conditions as you want and the created list can be used and destroyed or kept for future use.
4. **DATE.** This function is very useful. It enables you to set up conditions that must be met by a given date. Very useful on invoice/statement type applications.

It is possible if you have written an application with SUPERBASE to PROTECT it so that the program cannot be listed and also to pro-

tect parts of the program to restrict unauthorised use, a feature which I have found useful to restrict access of personal details while allowing basic name address type information to be readily accessible. Screen colours can be changed easily and sound can be used to draw attention to important inputs. There is a built in software printer interface and so you should be able to use most of the common printers and switching from printer to screen is simply itself if when writing a program you want to see the effects of a report on screen rather than on the printer it is just a matter of replacing the word PRINT with DISPLAY in the program. This can save pumping out reams of paper during debugging.

#### Great stuff

This is a program that I would not like to be without. It is versatile and probably the most useful of all my software. It costs about £100.00 but is considerably cheaper than its big brothers on larger machines. One thing that is important is when I had a problem with one of the programs PRECISION Software were very keen and quick to help. They do seem concerned with after sales service, something which is sadly lacking in many soft/hardware companies.

Dealers amongst you may be interested to know that there is a demonstration disc available which shows much of the potential of SUPERBASE.

If you are in business or run a club/society/have large collections/an untidy filing cabinet or just enjoy databases then this is the one.

# 64<sup>bit</sup>



# Doesn't your 64 deserve a good book?

To make full use of the your Commodore 64's powerful capabilities, you need the help of good books. You need SAMS books - they'll keep you and your micro busy for hours.

## Commodore 64 Programmer's Reference Guide

A bestselling book packed with professional tips to help you program your 64 successfully and creatively. Now at a new low price.

486 pages/ISBN 0 672 22054 5/£9.95

## Commodore 64 Graphics and Sounds

Timothy O'Riordan

A combination of book and software to help you master the 64's powerful graphics and sound capabilities, and use them in spectacular routines.

Book: 112 pages/ISBN 0 672 22070 7/£6.95

Book with software: ISBN 0 672 26180 3/  
£15.95 + VAT

## Commodore 64 BASIC Programs

Knight and Laddie

A well illustrated collection of fun and practical programs, each of which is explained in full so that you learn how it works as you use it.

Book: 178 pages/ISBN 0 672 22171 3/£7.95

Book with software: ISBN 0 672 26171 5/  
£13.50 + VAT

## Learn BASIC Programming in 14 days on your Commodore 64

Cl Schaefer

An easy to use and rapid method of learning how to program your 64, consisting of 14 chapters each of which can be covered in a day.

182 pages/ISBN 0 672 22079 5/£10.50

Available from all good bookshops.

SAMS books are distributed in the UK by Pitman Publishing, 128 Long Acre, London WC2E 6AN (Tel. 01 276 1330).

**Pitman**

## Superbase 96

Superbase 96 is a powerful database management system for the Commodore 64. It allows you to create, store, and retrieve data in a structured and organized manner. The system is easy to use and provides a wide range of features for data management.

## SuperScript II

SuperScript II is a word processing software package for the Commodore 64. It offers a variety of text editing and formatting options, making it a useful tool for writing and editing documents.

## Master

Master is a spreadsheet application for the Commodore 64. It enables users to create and manage spreadsheets, perform calculations, and generate reports. The software is designed for ease of use and flexibility.

\*\*\* VHS 12 DISKETTES \*\*\* CALL US TODAY \*\*\* 01 276 1330

ITEM NO.	DESCRIPTION	PRICE	ITEM NO.	DESCRIPTION	PRICE
101	Superbase 96	£12.95	104	Master	£12.95
102	SuperScript II	£12.95	105	Master	£12.95
103	Master	£12.95	106	Master	£12.95
104	Master	£12.95	107	Master	£12.95
105	Master	£12.95	108	Master	£12.95
106	Master	£12.95	109	Master	£12.95
107	Master	£12.95	110	Master	£12.95
108	Master	£12.95	111	Master	£12.95
109	Master	£12.95	112	Master	£12.95
110	Master	£12.95	113	Master	£12.95
111	Master	£12.95	114	Master	£12.95
112	Master	£12.95	115	Master	£12.95



128 LONG ACRE, LONDON WC2E 6AN, ENGLAND. TEL: 01 276 1330

# LIGHT TO THE POINT



SPECTRUM AND COMMODORE 64  
**£17.25**  
**DRAGON TANDY**  
**£11.50**  
Inclusive  
NO INTERFACE REQUIRED

Discover the exciting world of creating your own graphics on screen.  
The Trojan Light Pen will draw boxes, circles, lines, freehand pictures, arcs and free pictures with full erase facility.  
All in Hi-Res screens in any of 4 colours for the Dragon/Tandy, 8 colours for the Spectrum, and 16 colours for the Commodore 64.  
For educational or leisure use.

# TROJAN

Micro Computer Software & Accessories

Send cheque/P.O. to  
TROJAN PRODUCTS  
168, Dalrymple, Dunfermline, Fife KY12 7TF  
Tel: (0719) 205491





**\*\*\*\* COMMODORE 64 \*\*\*\***

**R.S.S. (RAPID BACKUP SYSTEM)**

CONVERT YOUR SINGLE AND MULTITASK TAPE PROGRAMS TO TUBES-LOAD. FASTER THAN THE COMMODORE (20) COPY. EACH CONVERTED PROGRAM WILL LOAD INDEPENDENTLY OF R.S.S. NO FORMS, NO EYE-CALLS, NO USER KNOWLEDGE REQUIRED. LOAD "THE ROBERT" IN 104 SECS. FOR PERSONAL USE ONLY. CASSETTE £7.95 INC.

**HYPERSAVE - 84**

A VERSATILE AND FLEXIBLE FAST-LOADING UTILITY. 8 ADDITIONAL BASIC COMMANDS. SAVE/LOAD/VERIFY UP TO 48K CONTIGUOUS RAM AT HYPER-SPEED.

CASSETTE £7.95 INC.

**BACK-UP - DELUXE**

A POWERFUL AND USER FRIENDLY SECURITY BACKUP UTILITY. LOAD/SAVE ALL PROGRAM PARTS CONTINUOUSLY. INCORPORATES A SOPHISTICATED HEADER-HEADER. FOR PERSONAL USE ONLY. CASSETTE £7.95 INC.

**PRO-SPRITE**

SPRITE LIBRARY. DESIGN, MANIPULATE, CATALOGUE MULTI AND SINGLE COLOUR SPRITES. CREATES DATA. JOYSTICK CONTROL. PACKED WITH FEATURES 100% MACHINE CODE. CASSETTE £7.95 INC. COMPLETE WITH YOUR FIRST SPRITE LIBRARY.

**SPECIAL PROMOTION - ANY TWO PROGRAMS £9.95**  
ALL FOUR £29.95 PLUS FREE! SECRETLY  
1/0 GUIDE WITH EVERY ORDER.  
CHECKLIST P.O. 26 SAE TO:

**DO&SOFT**  
**2 OAKMOOR AVENUE**  
**BLACKPOOL FY2 0EE**

\*\*\* FAST DISPATCH GUARANTEED \*\*\*\*

**BUNTAFT**

GREAT ADVENTURES, STUNNING LOW PRICES!

**KRALIA'S KINGDOM** - MEDIEVAL TEXT ADVENTURE FOR COMMODORE 64 AND HQ VIC 20 (118 2 186 EP000484)

★★★★★ FIVE STAR QUALITY -

HOME COMPUTING WEEKLY £4.98

**BAROONED** - SPACEAGE TEXT

ADVENTURE INC. SOUND EFFECTS.

HQ VIC 20 ONLY £4.98

"GOOD VALUE..." - MICRO ADVENTURER

6 A & E FOR COMPLETE LIST

1 CHIPPING HALL COTTAGE

CHIPPING WERTS, SO5 0PG



If an advertisement is wrong we're here to put it right.

If you see an advertisement in the press, in print, on posters or in the cinema which you find unacceptable, write to us at the address below.

The Advertising Standards Authority

AAA Ltd, Dept 3 Millers House, Torrington Place, London WC1E 7HN

**Supersoft 64**

IN PRINT AS OF 20 JANUARY 1984. THE AUTHOR HAS ADVISED THAT THE INFORMATION CONTAINED HEREIN IS CORRECT AS OF THE DATE OF THE LAST UPDATE. THE INFORMATION CONTAINED HEREIN IS SUBJECT TO CHANGE WITHOUT NOTICE. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY ERRORS OR OMISSIONS. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY DAMAGES, INCLUDING CONSEQUENTIAL DAMAGES, ARISING FROM THE USE OF THE INFORMATION CONTAINED HEREIN. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY DAMAGES, INCLUDING CONSEQUENTIAL DAMAGES, ARISING FROM THE USE OF THE INFORMATION CONTAINED HEREIN.

**Figaro 64**

IN PRINT AS OF 20 JANUARY 1984. THE AUTHOR HAS ADVISED THAT THE INFORMATION CONTAINED HEREIN IS CORRECT AS OF THE DATE OF THE LAST UPDATE. THE INFORMATION CONTAINED HEREIN IS SUBJECT TO CHANGE WITHOUT NOTICE. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY ERRORS OR OMISSIONS. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY DAMAGES, INCLUDING CONSEQUENTIAL DAMAGES, ARISING FROM THE USE OF THE INFORMATION CONTAINED HEREIN.

**Vizawrite 64**

IN PRINT AS OF 20 JANUARY 1984. THE AUTHOR HAS ADVISED THAT THE INFORMATION CONTAINED HEREIN IS CORRECT AS OF THE DATE OF THE LAST UPDATE. THE INFORMATION CONTAINED HEREIN IS SUBJECT TO CHANGE WITHOUT NOTICE. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY ERRORS OR OMISSIONS. THE AUTHOR ACCEPTS NO LIABILITY FOR ANY DAMAGES, INCLUDING CONSEQUENTIAL DAMAGES, ARISING FROM THE USE OF THE INFORMATION CONTAINED HEREIN.

COMMODORE	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060
1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081

SEARCH FOR THE BEST SOFTWARE DEAL... CALCO SOFTWARE

14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/10/83 8:00 PM, 14/10/83 9:00 PM, 14/10/83 10:00 PM, 14/10/83 11:00 PM, 14/10/83 12:00 AM, 14/10/83 1:00 AM, 14/10/83 2:00 AM, 14/10/83 3:00 AM, 14/10/83 4:00 AM, 14/10/83 5:00 AM, 14/10/83 6:00 AM, 14/10/83 7:00 AM, 14/10/83 8:00 AM, 14/10/83 9:00 AM, 14/10/83 10:00 AM, 14/10/83 11:00 AM, 14/10/83 12:00 PM, 14/10/83 1:00 PM, 14/10/83 2:00 PM, 14/10/83 3:00 PM, 14/10/83 4:00 PM, 14/10/83 5:00 PM, 14/10/83 6:00 PM, 14/10/83 7:00 PM, 14/1

# ANIROG

NEW  
TURBO 64  
GAMES

P.C. FUZZ



Aggro at closing time at the local. Getaway car screeches to a halt outside a bank. Ingenious methods employed by the Mafia to literally spirit away the loot from the High Street. Never fear - P.C. Fuzz is on patrol.

COMMODORE 64 £7.95

## ZARCA MISSION



**Also available on  
Disk at £9.95**

An action  
thriller by the  
COMMODORE

packed 3D scrolling arcade  
author of Space Pilot  
64 £7.95

TRADE ENQUIRIES: ANIROG SOFTWARE LTD, 29 WEST HILL, BARTFORD KENT (G322) 92513/9  
MAIL ORDER: 8 HIGH STREET, HORLEY, SURREY. 24 HOUR CREDIT CARD SALES: HORLEY (00304) 6060  
PAYMENT BY CHEQUE P.O. ACCESS/VISA. 50p POSTAGE & PACKAGING

# BUSICALC 3

**- the sophisticated spreadsheet !**

Easy to learn, easy to use - something that can't be said of many business programs. But it's true of all the programs in the BUSICALC series.

BUSICALC 3 can handle all sorts of jobs - budgets, expenditure analysis, stock lists, price lists, and product costing are just a few of the possibilities. Three-dimensional formulae automatically access data stored on disk, so that you can easily pull together information from several different sheets and summarise or manipulate it.

It's simple to transfer data to other programs such as Easy Script. And you can use virtually any printer with BUSICALC 3, whether dot matrix or daisy wheel, Commodore or non-Commodore.

For the CBM 64 and PET/CBM 4000 & 8000 series.

Available through dealers or from:

**Supersoft, Winchester House, Canning Road, Harrow HA3 7UJ**

**Phone 01-861 1166 for more details and a free catalogue.**

