

Computing *with the* AMSTRAD

The independent magazine for Amstrad computer users
Registered by Australia Post — Publication No. VBP7811

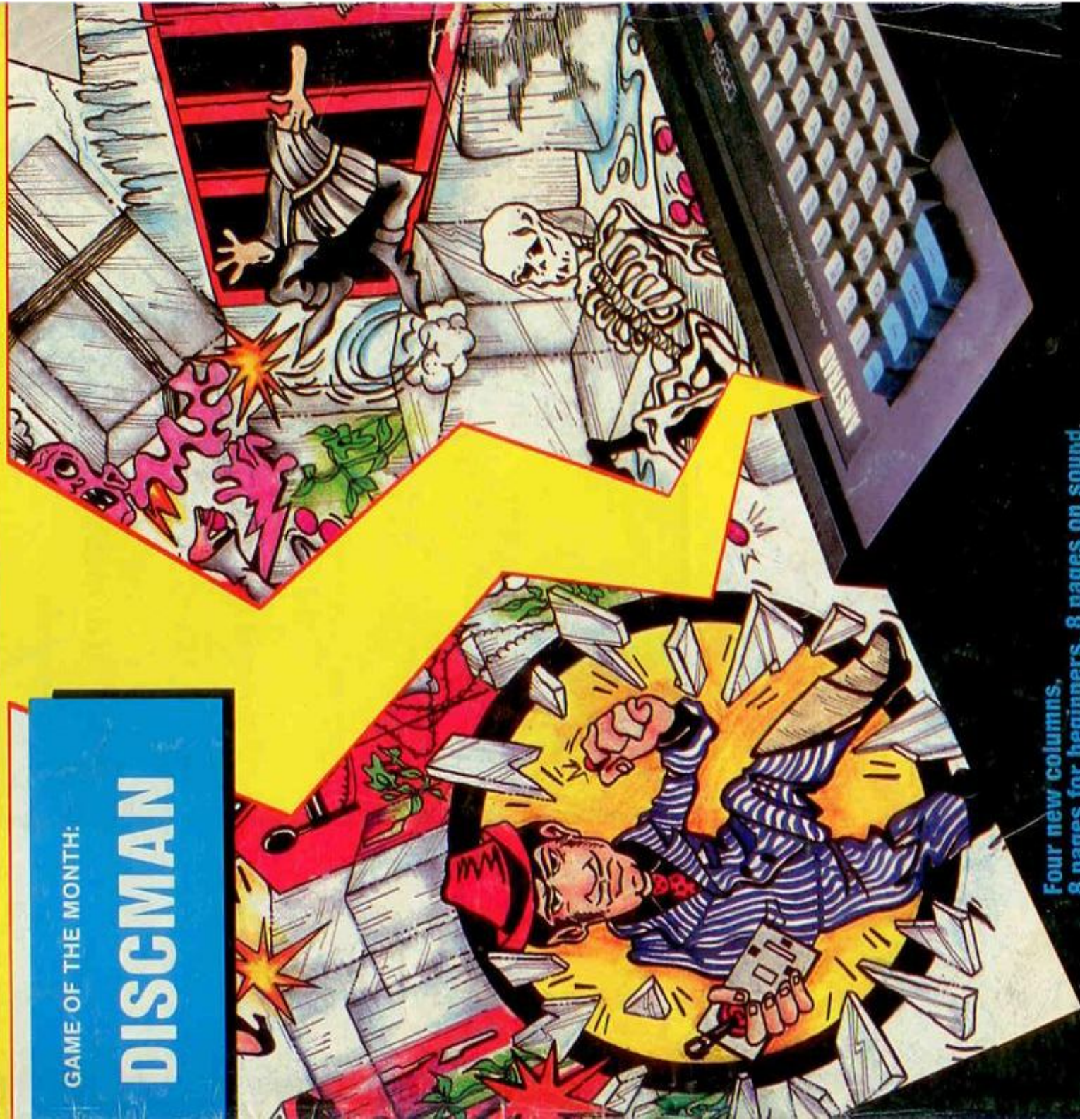
November, 1986

\$3.60

A Database Publication

GAME OF THE MONTH:

DISCMAN

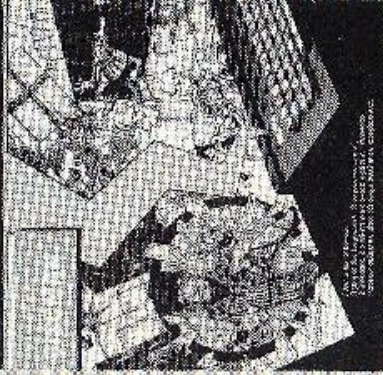


Four new columns,
8 pages for beginners, 8 pages on sound,
2 Utilities, 2 add-on disk drive options reviewed
regular columns plus 16 page business supplement.

**WORLD
SCOOP**
NEW Amstrad home computer
CWTA reveals

Computing with the AMSTRAD

Available in the
UK for £3.60



November 1986

Published monthly by Strategy
Software under license from
Database Publications Ltd.

Telephone [03] 560 4324

Telex AA 30625

Attn. ME 1609

Fax [03] 560 7482

Attn: Strategy Software

Box 5000, Glen Waverley

Victoria 3150

Annual Subscription:

Australia \$36

New Zealand & South Pacific

\$56 Air Mail

'Computing With The Amstrad' welcomes program listings and articles for publication. Material should be typed on computer printed, and preferably double spaced. Program listings should be accompanied by cassette tape, or disk. Please enclose a stamped addressed envelope or the return of material cannot be guaranteed. Contributions accepted for publication by Database Publications or its licensee will be on an all-rights basis.

©Database Publications Ltd and Strategy Software. No material may be reproduced in whole or part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles, listings or advertisements.

'Computing With The Amstrad' is an independent publication and neither Amstrad, Consumer Electronics plc, or Amsoft, or their distributors are responsible for any of the articles in this issue or for any of the opinions expressed.

CONT

Features

NEW 6 CP/M

First in a new series - what it's all about and how to make use of it.

16 Ready Reference

Have you got to grips with the WINDOW command yet? Confused by all those parameters? Let our handy, helpful guide make your windows clear.

26 Software Survey

From Knight Games to Gladiator, Pacific to Mermaid Madness - the latest software is evaluated here.

49 Postbag

All you need to know about Locomotive Basic and your Amstrad - we got the ball rolling with reprints of the first two of the series which appeared in the U.K. edition of COWTA. Fight pages packed with information.

52 U.K. News

All the latest and greatest hot off the FAX. Get up to date with information from the people who know.

Utilities

17 Fast Fill

Supercharge your CPC464 with this short but powerful routine to fill any enclosed shape - FAST!

59 Screen Clock

Don't waste a second - keep track of time with this machine code program to provide a screen clock.

Reviews

18 Forth

Fancy Forth as a second language? We compare two of the current market leaders.

58 Giltronics Disk Drive

If you're tired of paying \$10 for a three inch disk then the Giltronics 5 1/4 inch second drive could be the answer.

75 D'ktronics Silicon Disk

If you'd rather not pay for any disks at all then perhaps you should consider this alternative.

55 18 Commandments

Fancy trying your hand at getting your work of art published in CWTA? Here are 18 hints to show you how it's done.

57 Aleatoire

Puzzles you can put your Amstrad to work on - if only Maths at school had been this much fun!

NEW 64 Sound

We show you how to become a big noise in the far from silent world of the Amstrad micro. The first two parts of a series published in the U.K. edition of CWTA.

NEW 72 Public Domain

Shane Kelly joins the team - the Daylesford Dazzler tells us all about Public Domain software from the world of CP/M and makes you an offer you shouldn't refuse!

Games



20 Discman

Imprisoned on the planet Dartsma, the discman must survive the ghosts and deadly plants to provide the reactors with power.

61 Chipstow Stakes

Fancy a flutter? You'll have hours of fun with this exciting horse racing game for PCW owners.

THE BUSINESS END OF CWTA

34 Business Graphics

In need of a graphics package? DR Draw from Digital Research may be just what you're looking for.

37 Business News

Read all about it. How the Amstrad range is continuing to expand into the business world.

39 IBM

How the mainframe giant set the standards for the business micro market. And we ask: Do the clones match up?

42 Chess Review

We look at two more contenders in the PCW Chess stakes.

43 Application

Going from A to Z is not quite as simple as you think. How a professional indexer uses his PCW.

45 Mini Office II

This month we look at the Mini Office II database in action and show you how to manipulate your files.

47 Worstar Update

Pocket Wordstar has been customized for the PCW series. We see how effective the changes prove to be.

Apology

Due to lack of space, in spite of an increase in size to 80 pages, we were unable to bring you the promised financial software review - this will be published as soon as space allows.

The universal standard

It is now a year since Amstrad launched their disc drive for the CPC464 amid a blaze of publicity about it being the first home micro capable of running CP/M, and indeed "gave away" CP/M with the disc interface. And of course CP/M comes free with the CPC128 and the ill-starred CPC664.

Unfortunately, as few home users seem to know much about it — a state of affairs which Amstrad's disc drive manual has done little to improve — this fairly major advance for home computing has managed to pass largely unnoticed by the majority of owners.

This is really rather a shame, as I hope this series of articles will show. Over the next few months we will explore CP/M on the Amstrad in detail, starting with "What is it?" and progressing through the "How do I use it?" to "How does it work?"

It is only fair to warn here that several of the later articles will assume a reasonable working knowledge of Z80 machine code programming. If you've been following Mike Bibby's articles on machine code programming, you shouldn't find this too much of a problem.

In any case, those of you not wanting to delve too deeply into the darker arts will still benefit from a greater understanding of CP/M's structure, use and potential.

What is it?

So what do you get for your money when you enter CPM, other than the mysterious `A >` prompt? Well, typing in `DIR` performs a function similar to the `AmDOS` `CAT` command by listing the files present on the disc.

Any whose names end in a .COM extension are executable programs, and may be pulled into memory and run simply by typing their name (without the .COM on the end).

We'll discuss these in detail in a future article. For the moment, let's look at the most important program on the disc – the one which *doesn't* show up in the directory list – the control program itself. You see CP/M (Control Program for Microcomputers) is an operating system. This simply means that it is a program which carries out two main functions:

Now you can explore the full potential of CP/M on your Amstrad with COLIN FOSTER's comprehensive series

- It lets you "talk" to the machine – you can call programs into memory from disc and execute them, look at text files, manipulate the I/O (Input/Output) devices connected to your computer – printer, modem and so on – and generally keep control of things. The part of CP/M which does this job is called the CCP (Console Command Processor) and will be discussed in more detail in the next article.

- It frees programs from the extremely complex and tricky business of communicating directly

with the hardware themselves by providing specific service routines which programs can call to handle input and output functions from and to discs, printer, screen and keyboard. There are two "levels" to the part of CP/M which looks after this - the BDOS and the BIOS.

BDO'S

The BDOS (Basic Disc Operating System) provides relatively simple to use but powerful high level routines

PROCEDURE FOR COPYING DISCS ON A SINGLE DRIVE SYSTEM

WRITE PROTECT master disc
WRITE ENABLE blank disc
Insert master disc into drive

```

cpm
A>discopy <ENTER>

```

Put master disc into drive,
side 1 up, press <ENTER>

UNTIL copying complete

Put master disc into drive,
side 2 up, press <ENTER>

REPEAT

UNTIL copying complete.

to open and close files on disc, read from and write data to them, input and output characters from and to the keyboard and screen – called the console in CP/M-speak.

BIOS

The BIOS (Basic Input/Output System) provides the low level routines to actually perform the nitty-gritty operations involved in handling the hardware.

CP/M's real usefulness lies not so much in any great qualities of power or flexibility – there are better operating systems around – but in the fact that it is universal. CP/M-80 version 2.2, to give it its full Sunday name, has become the standard operating system for hundreds of different 8080 and Z80-based computers, all of which have wildly different amounts of memory, disc types and sizes, I/O ports, and so on.

However the service routines provided by the BDOS and BIOS, and the command facilities available from

the CCP, are all specified absolutely by Digital Research, the people who wrote CP/M, and perform exactly the same functions on all CP/M computers.

This means that if a program does all its I/O using these "approved" routines, then it can be transferred to any other computer running CP/M regardless of hardware differences, and will find present all the routines it needs to run properly.

So standard is this system in fact that the CCP and BDOS which we have running on our Amstrads are essentially the same programs as run on every other CP/M system in the world.

As the BDOS itself uses the BIOS to implement its own higher level functions, only the BIOS part of CP/M is machine specific, in other words, different for every computer.

Thus Amstrad have supplied their BIOS in the disc interface ROM, while the BDOS and CCP come on the system tracks of the Digital Research distribution disc and must be copied on to your working discs before you can use them.

What all this means is that there is a vast amount of software available, both commercial and public-domain

– free – most of which will run without alteration on the Amstrad. The reason why some won't is simply that the CPC464 doesn't have as much memory available to CP/M as most machines, and some programs can't cope with this.

Disc copying

Before we go any further, you should make a copy of your distribution disc, if you have not already done so. This is important to protect you from losing your only copy should your dog decide to eat it.

The easiest way to do so at this stage is to use the DISCCOPY utility, or COPYDISC, if you have two disc drives. The procedure is detailed in the panel.

I suggest that you make sure your source disc is write protected before you start. This prevents an embarrassing accident if you get the discs mixed up somewhere along the line.

The User Manual shows how to do this for the two different types of discs which are available.

Next month we will look at how CP/M organises the Amstrad's memory and see what commands are available from the CCP.

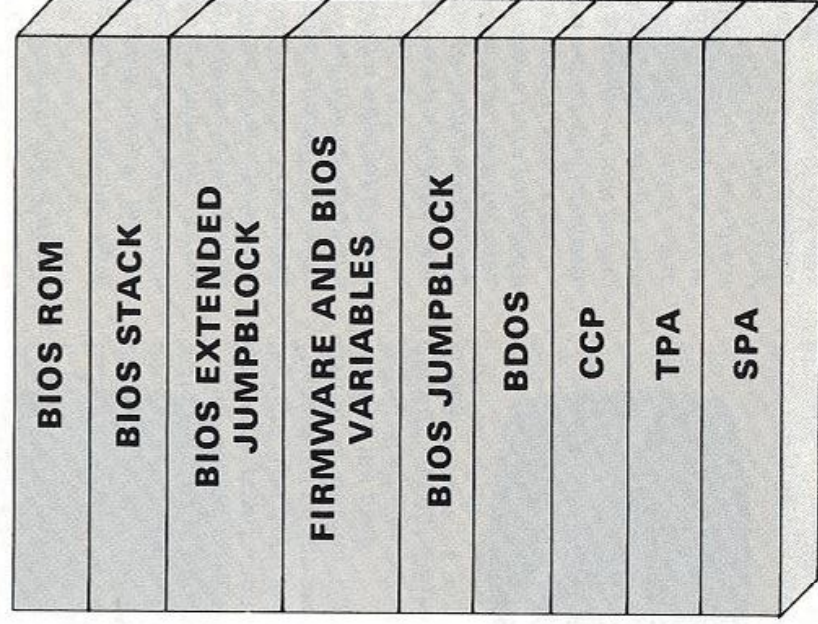
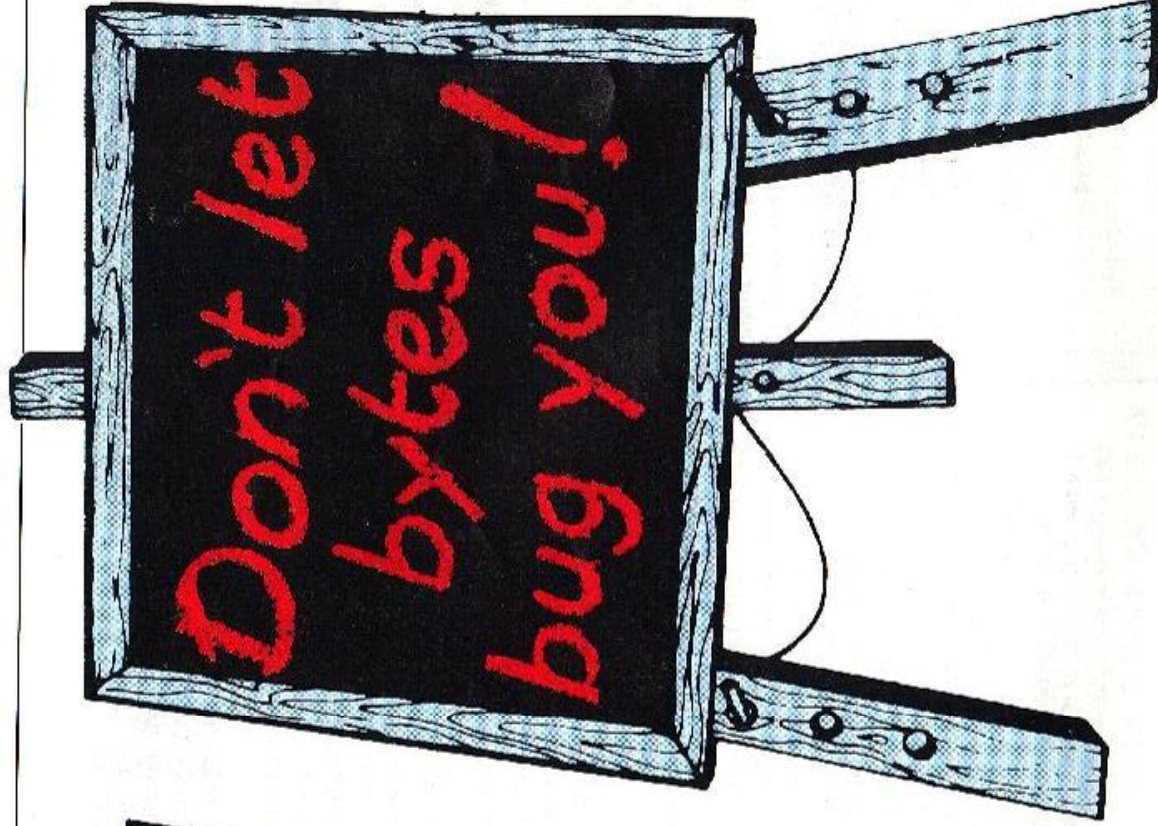


Figure 1: Amstrad CP/M memory map

Put copy disc into drive,
side 1 up, press <ENTER>

Put copy disc into drive,
side 2 up, press <ENTER>

Programming is easier than you think, and doubly so if you follow MIKE BIBBY's crystal clear guide through the micro jungle



I DON'T know who you are. You might be a wife whose CPC464-owning husband is away at work, or a father who is trying to come to terms with his daughter's Christmas present.

Alternatively you might be a teacher who has just been "computerised". Whoever you are, the the fact that you are reading this article tells me your guilty secret — you want to be able to program the Amstrad CPC464.

But how to begin? You must have noticed that some people take to computing like ducks to water, or an output port to an interface as they would say. Words like byte, strings and user-defined characters flow freely from their lips. They pass parameters and handle interrupts with ease, then get their hands on and peek and poke in a way that beggars belief!

You, I take it, are not like that. You are not a computer "natural". But you would dearly like to be. Well fear not, this series of articles is for you, and it was written by one of your kind.

I, too, have sat at the keyboard, staring at the cursor, without having any idea of what to do next (or even knowing it was called a cursor). I also know what it's like to have someone explain to me in the "simplest possible terms" and still find it way above my head. Yet I now program reasonably well . . . and so can you. Read on!

LET'S assume for a start that you are seated in front of your micro, which is already plugged in (the User Instructions book is pretty good on this.) Right — that's the end of our assumptions.

Turn on your monitor — the switch is a simple push button on the front at the right. You won't see a lot happen! Then switch on the micro itself — the switch is on the right-hand side towards the back.

The first sign of life from your CPC464 should be a message similar to this:



As the message says, you've got a 64k colour computer. The "64k" is a measure of the micro's memory. In

computing terms having more memory means, very roughly, being able to type more into the computer before it's full up.

Basic, as we shall see later, is the name of the language in which you give instructions to the CPC464. The remainder of the message tells you who markets the micro — Amstrad — and who wrote the Basic — Locomotive. The "Ready" means just that — the micro is ready for you to type in some information. We call it the prompt.

At the end of these initial messages — under the "R" of Ready — you should see a solid square on the screen. This is called the cursor, and it too indicates that the micro is ready for some information from you.

So let's give it some. Type in two or three letters — just the alphabet for the moment, please. It should be apparent that the cursor indicates the position at which the next character will appear on the screen.

Let's take a closer look at the keyboard. Fundamentally it is a

standard typewriter keyboard surrounded by several additional keys.

However, unlike the standard typewriter, not only do the numbers appear on the top row of keys as usual, but also on a separate "numeric keypad" to the right of the main keyboard.

Notice that the keyboard has separate keys for the letter O and the number 0. The O, as you'll see if you try it, always appears with a diagonal line across it – whether you type it on the main keyboard or the numeric keypad.

You must keep O and 0 entirely separate. I guarantee that a lot of your early mistakes in programs will be caused by typing O instead of 0!

On the same lines, notice that there are special keys for 1. Don't use the lower case "l" for one, as you have to on some typewriters – the micro won't appreciate it!

Above the numeric keypad there's a cluster of five keys – one labelled COPY, surrounded by four keys with arrows. These are simply dealt with – we're going to ignore them for the present.

Going back to the main keyboard, you'll see that as well as the letters and numbers ("alphanumerics" as the jargon has it) there are other keys, neatly picked out in colour.

Some, such as Caps Lock and Shift, you'll be familiar with from ordinary typewriters. Others, such as Ctrl and Esc, should be new to you.

Let's introduce a convention to make life easier. If I want you to press the Shift key, I'll ask you to type

[Shift]

If, on the other hand, I ask you to type

Shift

I want you to type S followed by h, followed by i and so on. So if I want you to press a single key, I'll put the name of that key in square brackets – otherwise type each character out separately.

Now Enter is quite an important key – it's the blue one at the right of the main keyboard. It's also at the bottom right-hand corner of the numeric keypad.

We use Enter in a similar manner to the Return key on an electric typewriter, to ensure that the typing continues on a new line.

It's far more important than that,

You can't hurt the Amstrad by accidentally mistyping something – so feel free . . .

though. Enter not only gives you a new line, it also sends the message you have typed on that line into the computer to be acted upon.

If you've been following so far, you should have a screen looking something like this:



If, however, you've been idling until now, type a few letters, then press [Enter]. Odds on, you'll get a message back from the computer saying:



Don't worry about this message – you can't hurt the micro by accidentally mistyping something, so feel free to experiment.

All that *Syntax error* means is that the computer doesn't understand the words you've just sent it. You see, it needs to be talked to in its own language, called Basic.

However learning Basic isn't like learning a genuinely foreign language. Basic is very similar to English, but it only uses a small set of selected English words in order to make things simpler for the computer. These selected words are called keywords.

This, by the way, is why I said it was "odds on" you'd get the *Syntax error* message when you pressed Enter. You might, by chance have hit

on a Basic keyword.

For example, you can mark the end of a Basic program with the keyword *end*. The people who designed Basic could have chosen the word *finish* to do this.

Let's see if there's any difference in the words. Type *end* and press [Enter]. Then type *finish* and press [Enter]. The screen will look like this:



Notice the difference? The CPC464 accepts *end*, but not *finish*.

Admittedly, *end* doesn't accomplish very much. After all, you haven't anything in there to end, have you? But at least the micro didn't hurl the message *Syntax error* at you as it did with *finish*.

This is because *end* is a Basic keyword, while *finish* isn't.

Notice something about the *Syntax error* message – it has a capital S. So far everything we've typed in has been in lower case – at least in theory. (Remember, I said stick to the letters of the alphabet!)

I bet you've already discovered that if you hold Shift down while you're typing the letters appear in upper case. If not, try it now. It's what you'd expect, if you've ever used a typewriter.

Again it will come as no surprise to see that if you press the



key with Shift, you get the "&", while un-Shifted you get the "6". If you press Caps Lock and release

it you'll find that the letters of the alphabet 'come out' in upper case automatically. Also if a key has two characters (or legends as they are known)...you get the upper one on the screen.

So, with the



key, you'll get "&" with Caps Lock on.

This upper case output will continue until you press Caps Lock again, then you'll go back to normal. Press it again and you're back into upper case. Once more and it's normal again. (This way of jumping between one "state" and another is called "toggling" in computer jargon.)

What do you suppose would happen if you typed in *END* instead of *end*? Try it and see. One of the nicest things about using a computer is that you don't have to speculate. If you find yourself asking "What would happen if?" or "I wonder if this would work?" you can go right ahead and try it.

As I said, you can't hurt the micro from the keyboard, so go ahead and experiment. Believe me, it's by far the best way to learn about programming.

Now we were wondering how the micro would react to *END* instead of *end*. Right then, just type in:

END [Enter]

(remembering that [Enter] means press the Enter key, don't type in *E,n,t*...!). As you'll see, the micro doesn't throw it out.

In fact the Amstrad doesn't mind if you type in your keywords in upper or lower case (unlike some other, more pedantic micros that demand keywords in upper case only.)

Another thing you've probably realised is that the keys have an auto-repeat facility. This means that having pressed a key, if you keep it down the letters repeatedly print themselves out automatically.

Press Enter to get on a new line,

ignoring any *Syntax error* messages. Using auto-repeat, put several letters on a line but don't press Enter. Right, you should now have a line of characters with the cursor hanging on the end of them.

Press the key marked Del — you'll see that the character on the left of the cursor is "devoured" by it.

This is one way of getting rid of, or deleting, characters (hence the Del). If you keep Del pressed down it will auto-repeat and gobble up the whole line, making a plaintive beeping sound when there's nothing left for it to eat!

You can use Del in this way to correct typing errors — just delete back to the mistake and retype.

Doing this sort of computer correcting is known as editing. There are more sophisticated ways of doing it that we'll be covering, involving the Ctrl, Copy and arrow keys. We'll leave dealing with these for another time.

By now, with all this experimenting, you'll have probably filled up a screenful of text and seen the scrolling action demonstrated. If not, press [Enter] several times in succession.

As you'll soon see, scrolling is when the top of the screen rolls up to allow more typing at the bottom.

(If you're wondering why I didn't say just hold Enter down to get the auto-repeat — it's because Enter doesn't auto-repeat.)

If you've had the patience to keep on pressing Enter until everything disappeared off the top of the screen ("scrolled off" is the jargon) you'll have a blank screen with the cursor at the bottom. There are easier ways of clearing the screen, however. Type in

some garbage and then follow up with:

cls [Enter]

The screen should clear, and you'll be left with the ready prompt and the cursor at the top of the screen. If you are not convinced about the CPC464 accepting keywords in both upper and lower case, try:

CLS [Enter]

Both are equally effective.

Well, this has been just a brief examination of the keyboard. There's lots more to cover, including the Esc and Ctrl keys. For the moment though, we'll change tack — after all, it's a computer, so let's get it to compute!

Don't worry, though — this isn't going to turn into a mathematical treatise. After a brief but necessary foray into simple sums this series is thoroughly non-mathematical.

Before we start, let me give you a warning. The computer will do exactly as you tell it, but only what you tell it. It's a very literal machine and in this respect is like my daughter on a particularly mischievous day. When asked to put on her pyjamas for bed she did exactly as she was told.

Of course, I hadn't asked her to take off her clothes first, had I? You can imagine the results.

Similar things happen with the computer. Say we wanted the micro to calculate $2 + 2$. Not only do we want it to do the sum, we also want it to tell us the answer when it's finished.

We instruct the Amstrad to write things on the screen with the Basic keyword *print*. This is a relic from the days when the computer's output, as

If you keep Del pressed down it will gobble up the whole line, making a plaintive beeping sound when there's nothing left for it to eat!

it's called, was actually printed out on paper rather than a screen as it is now.

So, to see the answer to $2 + 2$, type:

print 2+2 [Enter]

Note that you don't need the "=" sign as you do on a calculator. [Enter] takes care of that. Before continuing with this article, why not try a few simple addition sums? Also, try using *PRINT* instead of *print*.

Just as the micro does not allow you to use O or o for 0, so it does not let you use x or X for multiply. You must use the * symbol instead. For example try:

PRINT 4*3 [Enter]

Minus is straightforward. You'll find it sharing a key with =. Divide, however, is not ÷ but an oblique stroke /. For example $12 \div 4$ becomes:

PRINT 12/4 [Enter]

Though this may seem a little odd at first, you have met it when dealing with fractions: $3 \div 4$ is equivalent to the fraction $\frac{3}{4}$. Try:

PRINT 3/4 [Enter]

From now on I am going to assume you accept that before the CPC464 can act on your instructions they must be sent to it by [Enter]. I will, therefore, omit [Enter] from my examples. Make sure you don't!

Before experimenting with further sums of your own devising, I'd like you to try the following sequence:

PRINT 2+8-3

PRINT 4*8/2

PRINT 4*8+2

PRINT 4*(8+2)

If you think carefully about the results you'll see that the computer interprets sequences of sums in the order you learned at school. You do whatever is inside brackets first, multiplication and division next, then finally addition and subtraction.

Now try:

PRINT 2/3

PRINT 1000*1000*1000

PRINT 1/100/100/100/100/100

If you've managed this correctly, you should get:



The point to note here is that the micro works to a limit of accuracy. For example, $2/3$ is not exactly 0.666666667. The error is well under a millionth, though. Still, it must be borne in mind.

Similarly with especially large or small numbers. The computer saves space by storing them using a scientific notation called Exponent format. Here, for example, instead of printing out the answer to

1000*1000*1000

as you might expect:

1000000000

it prints out the result as 1E9.

For E, which stands for Exponent, you should read "multiplied by 10 to the power of". For example, 1E9 means "1 multiplied by 10 to the power of 9" which, if your maths is up to it, gives you the correct answer.

Similarly, the answer for:

1/100/100/100/100/100

was returned as 1E-10 which reads as "1 multiplied by 10 to the power of -10" which is 0.0000000001, the correct answer.

If you don't follow all this, don't worry. I've only covered it in this article to warn you about odd looking results to your sums which might pop up and confuse you.

Now let's try to get the computer to print out some words. Let's get it to print out Hello.

If you cast your mind back to your schooldays (and for some of us that's an awful long throw), you'll remember that when someone says something you surround what that person says with quotation marks (or quotes for short), such as, He said, "Hello".

In Basic, of course, we don't say words, we *PRINT* them, but we still

surround them by quotes. We omit, however, the comma and full stop Try:

PRINT "Hello"

The micro should print out:

Hello

Notice that the quotes are not printed. So, to get the CPC464 to print out a message on its screen, we just use *PRINT* or *print* followed by the message surrounded by quotes.

The message in quotes is called a string, or a string literal. "String" is because the micro considers them just to be a string of letters, one after the other. "Literal" is because the micro prints out literally, or exactly, what is between the quotes. So:

PRINT "Hello"

PRINT " Hello"

PRINT " Hello"

give different outputs since in each different numbers of spaces precede the Hello.

Actually strings do not have to be words. They can be any combination of symbols, including numbers. Just keep them in quotes. Try the following:

PRINT "4*3"

PRINT 4*3

This should convince you that the computer does print out strings – that is, what's between the quotes – literally.

When the calculation is in quotes the computer simply echoes the sum on the screen. When the calculation is not in quotes, it prints out the answer.

Experiment with printing out various messages on the screen. How long can you make them?

At the moment, the micro is responding to our commands as soon as we send them by pressing [Enter], but in a calculation or task requiring several steps this can be rather tedious.

It would be more satisfactory to give the computer a whole series of instructions that it could get on with, rather than spoon-feed it step by step.

Such a sequence of instructions is called a program, and we shall begin writing programs in next month's instalment.

Now for the next step Write your own programs

Second in
MIKE BIBBY'S
helpful series
for beginners

THIS month we are going to begin writing our own programs. Nothing spectacular mind, but enough to give a quiet glow of satisfaction. Firstly, let's discuss what we did last month.

We learned that to "talk" to the micro we had to speak to it in a language it already understood, called Basic. We also learned how to get the Amstrad to do sums for us and to print out messages, or strings, as they are known.

One Basic word we used quite frequently was PRINT, which instructs the micro to write or print things out on the screen. For instance, to do the sum 4+4 we typed

PRINT 4+4 [Enter]

where [Enter] means you should press the key marked ENTER. This sends the message we have typed into the micro. Hopefully it then responds by printing the correct answer, 8.

Notice something — there's a space between PRINT and the first 4. Try

doing the sum without that space. That is:

PRINT4+4 [Enter]

You'll find you get a "Syntax error" message. You see, the command PRINT is known as a keyword — Basic keeps it to itself and treats it specially. If you like, it's already in the micro's vocabulary.

However, to recognise you mean PRINT to be used as a keyword, you need to have a space after it — and before it if it's in the middle of a line. If not, the micro gets confused, as above. After all, it's never heard of PRINT4.

The technical term for the space marking the end of the PRINT is a delimiter — it shows the limit of the word PRINT.

While you're at it, try

print 4+4 [Enter]

You still get 8.

This may not surprise you, but it should — most other micros would not recognise the lower case "print". The CPC464 does.

Just as we can do addition, so we can do subtraction, multiplication and division — the symbols for which are -, * and / respectively. Notice, multiplication is *, not X. Also, divide isn't ÷, it's /. Although the Amstrad will print the more traditional ÷ symbol, it doesn't know that it means division; you have to use / for that.

We also learned last month that to print out messages, we had to surround them with quotes — as we do when we write what someone is actually saying.

So, to print out the message GOOD MORNING on the screen we type:

**PRINT "GOOD MORNING"
[Enter]**

which, as you'll see if you try it, causes

GOOD MORNING
to be written on the screen.

Of course, on the CPC464 we have lower case, or small letters, as well so we could have entered the above as **print "good morning" [Enter]**

This time:

good morning

will appear on the screen.

Notice the difference between the last two – the first was all upper case, the second all lower case.

In both instances the Amstrad recognised the command "print", proving once more that the Amstrad doesn't mind its keywords being in big or small letters – you can even mix them, as in Print.

It's different with the message in quotes, though – the first time the resulting print-out was in upper case, the second in lower case. You see, the Amstrad prints out exactly what's between the quotes: capitals and lower case letters are shown exactly as they are, so the two messages vary.

By the way, you don't need the space between the end of PRINT and the beginning quote marks of the message. This is because the quotes act as delimiters – that is, they mark the ends of keywords – just as spaces do.

In this series of articles we'll tend to keep to capital letters for our keywords so they stand out more clearly.

So far we have given the micro one instruction at a time, which it carried out immediately after we pressed Enter (assuming we'd typed it correctly).

Sometimes, though, we want to give the micro a series of instructions and then tell it to carry them out. For instance, suppose we want the message:

**PROGRAMMING
IS
EASY**

to appear on the screen. With our step-by-step method, we would have used:

```
PRINT "PROGRAMMING"  
[Enter]  
PRINT "IS" [Enter]  
PRINT "EASY" [Enter]
```

But, as you'll see if you try it, this doesn't produce the required effect, since each successive instruction spoils the layout.

We need to give the micro the instructions so that it:

1. Prints out PROGRAMMING
2. Prints out IS
3. Prints out EASY

in sequence, without stopping to ask us what to do next. Such a sequence

of instructions is called a program. Notice also that the sequence is numbered – after all, the micro needs to know the order to carry them out in.

Now let's write a program to print out

**PROGRAMMING
IS
EASY**

We were on the right lines with the first attempt, but this time, let's try numbering our instructions as we enter them.

First of all, we shall enter Mode 1, a text-only mode with:

MODE 1 [Enter]

Now type:

NEW [Enter]

New is a Basic keyword that clears out the micro's memory. If you don't do this the program you are typing in might get jumbled up with a previous one – you'll see more clearly how this can happen later.

You probably think that you haven't got a program in at the

**'The Amstrad
prints out exactly
what's between
the quotes'**

moment, but use NEW anyway, because it is possible that you might have entered a line or two by chance.

Then type:

**10 PRINT "PROGRAMMING"
[Enter]**

Notice two things:

- The first instruction is number 10, not number 1. In computing we tend to number our instructions in steps of ten for reasons that will become blindingly obvious later. We call the number of an instruction its line number.

- The micro didn't immediately carry out the instruction – it didn't print out PROGRAMMING after we pressed Enter. This is because of the line number. It tells the micro that what follows isn't to be done immediately but is to be remembered for later as it is just one in a series of instructions. I'll prove that the micro actually does

remember it in a moment.

Now type:

**20 PRINT "IS" [Enter]
30 PRINT "EASY" [Enter]**

What I'm going to ask you to do next should test your faith in me! Clear the screen by typing:

CLS [Enter]

All your typing should have disappeared, but don't worry – your work hasn't been wasted. Because of the line numbers, the micro has kept a list of your instructions in its memory. To see the list, type:

LIST [Enter]

and your program should reappear. We'll call it Program 1:

```
10 PRINT "PROGRAMMING"  
20 PRINT "IS"  
30 PRINT "EASY"
```

Program 1

If you've typed your keywords in lower case, you'll notice when you LIST them, they appear in upper case. This can be quite useful in looking for mistakes – you know all your keywords should be in capitals. If they're not, you've done something wrong!

An important point coming up now. We have entered a program (a numbered sequence of Basic instructions) into the Amstrad CPC464's memory and have got the micro to display those instructions with LIST. We have not, however, told the micro to do these instructions.

It's like having written a shopping list – you still have to go to the shops and turn your list into reality.

So to get the micro to actually do, or as we say, run the program in its memory, we type:

RUN [Enter]

and, if we've typed it in properly, we should see printed out:

**PROGRAMMING
IS
EASY**

If you've managed it, congratulations on running your first program.

If not, don't worry, it's probably some simple error. List your program and look for the mistake. You might actually have a message telling you that there is an error in a particular line.

What we're about to do next, although it assumes that you have been successful so far, will in fact

show you how to correct your mistakes.

Now let's try to alter our program so that it prints out:

PROGRAMMING IS SIMPLE

If you look back at the first program you will see that you need to alter line 30.

Changing line 30 couldn't be simpler – just type in the new line 30, remembering to start with the line number 30, then press Enter. The latest version will replace the old version in the micro's memory.

To demonstrate this, type:

30 PRINT "SIMPLE" [Enter]

and then:

LIST [Enter]

You should obtain Program II, which is:

```
10 PRINT "PROGRAMMING"  
20 PRINT "IS"  
30 PRINT "SIMPLE"
```

Program II

An examination of this listing should reveal that the new version of line 30 has indeed replaced the old one. (Notice also that we didn't give LIST a line number – we wanted the CPC464 to do it immediately.)

As a final proof that our amendment has been accepted, type:

RUN [Enter]

You should now get the revised message.

You can use this technique to correct mistakes in your programs. For example, if you accidentally typed line 10 as:

10 PINT "PROGRAMMING"

then, when you tried to run it you would get the message "Syntax error in 10". (Note that you don't receive this message when you first enter the line, only when you try to run it.)

To rectify such mistakes, simply retype the correct version of line 10 and press Enter to send it into the micro. The correct version will replace the faulty one.

There are more sophisticated ways of correcting, or editing, a line, but they can wait for a while. For the moment we shall simply retype the line, with its line number, and press Enter.

Of course, if you notice a mistake while you are entering a line, use the delete key to erase it, then continue

typing from that point.

So far I have given you just two programs to run. However, using those as models, you can print out virtually any message you want on the screen.

Just use line numbers in increments of 10, each line printing out part of the message you want out on the screen, by enclosing it in quotes after PRINT.

An important point about this series is that I'm going to give you lots of example programs to type in. Virtually all of them have two things in common:

- *They make vital teaching points (otherwise they wouldn't be there in the first place).*
- *The output – that is, what appears on the screen – is trivial in content and in many cases there are far easier ways of doing it.*

Programming is a skill like driving – you can only improve by doing it, not reading about it. Please carry out the examples, however simple or obvious they may seem to you.

Also, and this is far more important, I want you to go beyond the programs – try to alter, adapt and extend them, just to see what happens.

Adopt an experimental approach and a healthy scepticism for my pronouncements. If you are wondering whether something will work, go ahead and try it – you can't hurt the micro from the keyboard, so let your imagination run riot.

You'll learn far more from your own examples than you will by merely echoing mine. And the good thing is that you get such prompt feedback from a micro. If what you write isn't acceptable you'll soon get an error message.

So what I'd like you to do now is to spend a good time writing simple "message" programs for the micro to run. For some reason, in my experience in computing classes the messages tend to become quite scurrilous. There's one thing I've never been too sure of – is it slander or libel when it appears on a VDU?

Remember, type NEW before each new program, and use line numbers for each instruction. It's also good policy to LIST your program before you RUN it, just to make sure that all is as you intend.

Now suppose we wanted to alter

Program II so that it printed out the message:

PROGRAMMING IS RATHER SIMPLE

We need a line in there between 20 and 30 to print out "RATHER". Well, 25 is a number between 20 and 30, so let's try:

25 PRINT "RATHER" [Enter]

If you list it you'll see that the program has now become Program III:

```
10 PRINT "PROGRAMMING"  
20 PRINT "IS"  
25 PRINT "RATHER"  
30 PRINT "SIMPLE"
```

Program III

So line 25 has "crept in" between 20 and 30. Even though we entered it out of order, the Amstrad stores it in memory in its correct numerical position. Try running the program as final confirmation.

This ability to insert lines into programs is the reason our line numbers go up in steps of 10 when we are writing programs – it leaves us plenty of spare line numbers in between for when we are patching them up.

Now enter Program IV:

```
10 CLS  
20 PRINT "COMPUTING"  
30 PRINT "WITH THE"  
40 PRINT "AMSTRAD"
```

Program IV

remembering to press Enter after typing each line.

Now LIST it. Is there a phantom line 25 in there?

If so, you didn't type NEW after the last program – the lines 10, 20 and 30 of the latest program have replaced those lines in the old program. But as the new program doesn't have a line 25, the old one remains to ruin your program.

The moral is to use NEW before entering a new program.

If you have got an unwanted line 25, don't worry – you can easily get rid of it by typing:

25 [Enter]

This will delete the line since you replace the old line 25 with a new line which contains nothing – which the micro then "forgets".

This method holds good for

deleting any line from a program – simply type out the line number, then press Enter.

Program IV contains the keyword CLS. And this, as you shall see when you run the program, clears the screen.

Now let's try to print out our message with blank lines between. We can use a line containing just PRINT to obtain a blank line, so Program V should do the trick.

```
10 CLS
15 PRINT
20 PRINT "COMPUTING"
25 PRINT
30 PRINT "WITH THE"
35 PRINT
40 PRINT "AMSTRAD"
45 PRINT
```

Program V

Now try Program VI:

```
10 CLS
20 PRINT "HELLO";
30 PRINT "OUT";
40 PRINT "THERE"
```

Program VI

The output you will get is:

HELLOOUTTHERE

that is, each successive string is printed after the preceding one. The semicolon stops the next string being printed on a new line, "gluing" it to the end of the previous string printed.

Notice that, since there are no spaces inside the strings, none appear between the words when they are printed out together.

Try to get the message to appear legibly by rewriting the program with appropriate spaces in the strings. Also notice that you can obtain the same output, far more simply, with Program VII:

```
10 CLS
20 PRINT "HELLO OUT THERE"
```

Program VII

However, as I said above, the programs I present to you are for making teaching points, which does not necessarily imply showing you the most efficient methods.

Experiment with joining up the output of successive PRINT statements with the use of the semicolon until you feel confident about it.

And now for something com-

Compulsively introspective micros are not useful machines!

pletely different. Try running Program VIII:

```
10 PRINT "I"
20 PRINT "FEEL"
30 PRINT "DIZZY"
40 GOTO 10
```

Program VIII

I think the effect is pretty impressive.

So far all our programs have merely copied back onto the screen what you have typed in. This program shows how, with the addition of one line (line 40), you can obtain a huge increase in the amount of output.

It is this ability, to repeat a simple operation rapidly, that gives the Amstrad much of its power.

If things are happening a little too fast for you, you can temporarily halt proceedings by pressing Esc once. If you then press the Space Bar or any other key things will continue at their normal rate.

If on the other hand, you press Esc once more, the program will stop running completely, and the message "Break" will appear on the screen.

What is happening is that the micro follows lines 10, 20 and 30 and prints out:

```
I (line 10)
FEEL (line 20)
DIZZY (line 30)
```

followed by three blank lines due to the apostrophes. It then encounters line 40, which tells it to go back to line 10. It duly does so and prints out:

```
I (line 10)
FEEL (line 20)
```

and so on until it reaches line 40, when it goes back to line 10 and so on ad infinitum. Notice that when the screen is full, it scrolls up to make more room.

Now the name for such a condition in a program, where you keep on repeating lines of code (as the program lines are known), is a loop.

We say here that we are in an unconditional loop because we haven't given the program any

conditions for it to cease repeating itself.

This is bad programming practice – compulsively introspective micros are not useful machines!

To stop such unconditional loops, you have to interrupt them from "outside" by either pressing Esc (Escape), or pressing down (and keeping down) Ctrl followed by Shift followed by Esc. The latter causes what's known as a reset.

Of the two, Esc is to be preferred – you can compare it to stopping a car with the brakes. Using a reset is more akin to stopping your car by driving into a wall.

It won't actually damage your micro but it will cause you to return to the state of the micro on switch on, which is Mode 1, and with no program in when you LIST.

Esc on the other hand actually lets you continue if you wish, by simply pressing any key. The program will then continue. If not, press Esc again. The program will stop, but it's still there. Always use Esc if you can.

If you want to have some fun with an unconditional loop, try printing out repeatedly an arrow composed of asterisks such as:

```
*
***
*****
*****
*****
***
***
***
***
***
```

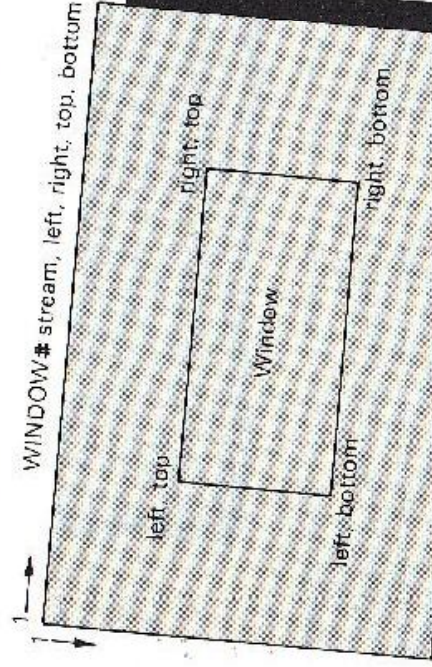
which will scroll upwards off the screen.

Finally, apart from its being an unconditional loop, which is always naughty, can you see what else is going wrong with this program?

```
10 CLS
20 PRINT "THIS IS"
30 PRINT "VERY SILLY"
40 GOTO 10
```

Program IX

● *NEXT MONTH we'll use variables to give our programs even more power.*

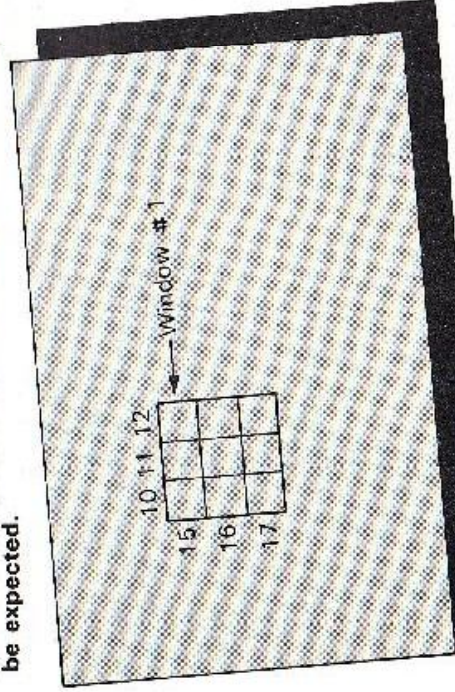


Window coordinates

Mode	stream	left	right	top	bottom
0	0-7	1-20	1-20	1-25	1-25
1	0-7	1-40	1-40	1-25	1-25
2	0-7	1-80	1-80	1-25	1-25

Window parameter ranges

Beware the following trap: WINDOW#1, 10, 12, 15, 17 gives a window 3 x 3 characters wide not 2 x 2 as might be expected.



Each window is referred to by its stream number.

INPUT
CLS
PRINT
LOCATE
PAPER
PEN
TAG

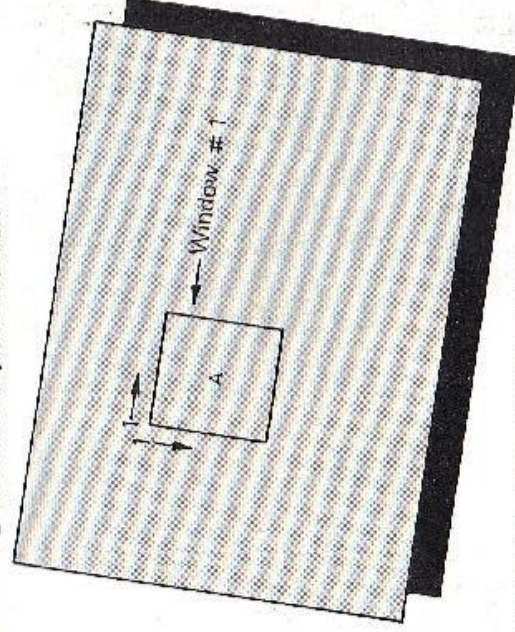
All these can use
stream to specify
a window.

Window#0 is SPECIAL:

1. It's the default window. If you don't put in a stream number or try to write to a window you haven't defined, it assumes window #0.
2. It fills the whole screen (unless you change this).
3. The system messages all use window #0.

Get the facts at your fingertips with the second of our ready reference charts

The LOCATE command works for each window in exactly the same way as it does for the whole screen measuring from the top left corner.



LOCATE #1, 2, 2: PRINT#1 "A"

Each window acts as a miniature screen, working independently. They all scroll text upwards when the window is full, the top line disappearing.

WINDOW SWAP stream, stream
This exchanges the specified text windows. For example:
WINDOW #3, 5, 7, 9, 12: WINDOW SWAP 0, 3
means that all system messages will appear in the smaller window.

In the September 1986 issue of *Computing with the Amstrad* Chris Ratcliffe showed us a routine to fill any shape, no matter how intricate, with any colour. This is for CPC464 owners of course – 6128 owners have their own built-in fill command.

Unfortunately the CPC464 fill routine is rather slow. The problem is that it's a legal fill using the Amstrad approved and documented firmware calls.

This Fast Fill routine doesn't use the firmware, but simply peeks and pokes the screen memory.

The result is that the code is only half as long yet it runs up to 12 times faster. To fill a shape first move to the start position and:

CALL &9E00,pen

The fill stops when it hits its own colour, so if you wanted a red flower draw the outline in red and fill it with red.

You can't draw a shape in one colour and fill with another – they both have to be the same. That's not a bug in the routine, it's the way the CPC6128 fill works so I've merely kept it the same.

There is one restriction though. Make sure that there aren't any gaps in the outline or the fill will leak out. It will colour the rest of the Amstrad's memory given half a chance.

When Fast Fill is run HIMEM is moved down to &9E00 to make room for the code and workspace. The routine is less than a page long and is followed by 1k of workspace.

You can use the workspace from &9D00 onwards as a scratchpad, but remember that Fast Fill will use it when called, so don't store anything there that you want to keep. ■

```

10 REM Fast Fill
20 REM By R.A.Waddilove
30 REM (c)Computing with the Amstrad
40 REM CALL &9E00,colour
50 MEMORY &9DFF
60 address=&9E00
70 FOR i=1 TO 21
80 sum=0:READ code$,check$
90 FOR j=1 TO 21 STEP 2
100 byte=VAL("&"+MID$(code$,j,2))
110 POKE address,byte
120 sum=sum+byte:address=address+1
130 NEXT

```

```

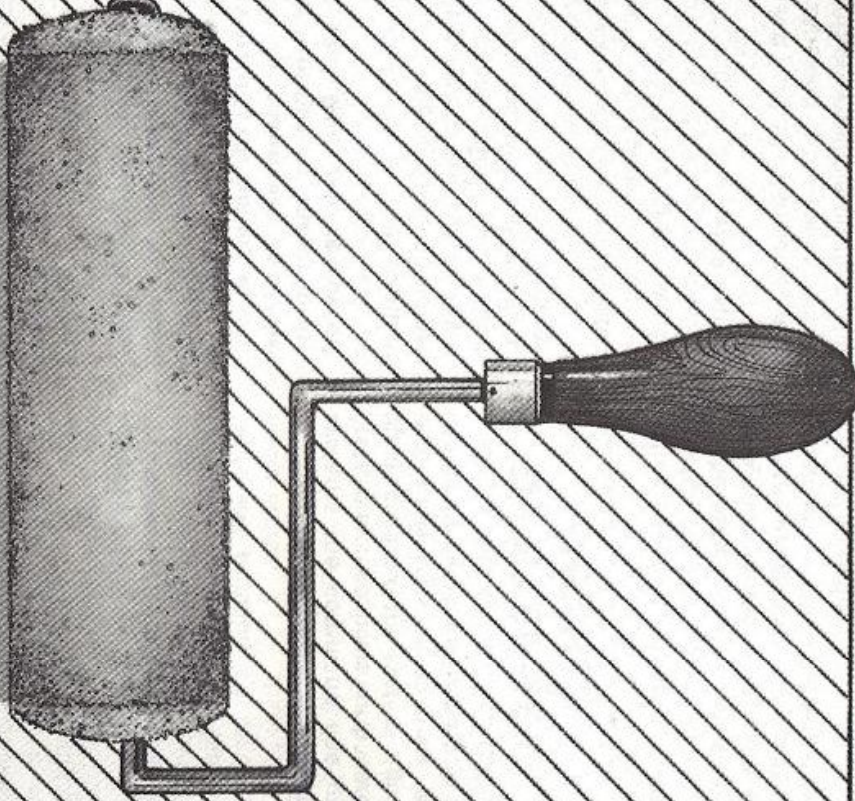
140 IF sum<>VAL("&"+check$) THEN PRIN
T "Error in line ";150+i*10
150 NEXT
160 DATA 3DC0D0460078CDEB878CD,643
170 DATA 2C8C32BA9E21040022479E,39E
180 DATA 2E0022C59ECD688CDF08B,679
190 DATA D08E00C8C0C68BCD11BC30,678
200 DATA 08C83ACB18C83B18062004,33B
210 DATA C83ACB18C83CC01DCD1D8C,580
220 DATA F3CDB86E11000021E19E19,4DE
230 DATA 13131313C892ED53479E5E,42C
240 DATA 2356237EEB22959E32989E,4C2
250 DATA 4F7CD60867FEC030041180,4C3

```

```

260 DATA 3F19CD869E2A959E7CC600,520
270 DATA 67D2819E1150C0193A989E,502
280 DATA CDB59E2A959E3A989E0730,524
290 DATA 0128CD859E2100003E000F,28A
300 DATA 300123CD859EED5BC59E13,532
310 DATA 131313C8922A479EA7ED52,48B
320 DATA C2469EFC94F7E61473E00,55D
330 DATA A180C8477EB1A980771100,578
340 DATA 00131313C892ED53C59E,44C
350 DATA DD21E19EDD19DD7500DD74,616
360 DATA 01DD7102C9000000000000,21A

```



By
ROLAND WADDILOVE

**FAST
FILL**

Two ways to get to grips with Forth on your Amstrad

SEVERAL alternative languages are now available to Amstrad users and among the most prominent is Forth. Versions of this extremely versatile language are available from a number of major publishers and two are reviewed here.

Both conform to the popular fig-Forth standard and as such almost all Forth programs will run on them without modification. In addition each version has extra commands to cater for specific Amstrad features such as sound and graphics.

These extensions are implemented in different ways depending on the system, and Forth programs which use them may require some modification to run on another Amstrad version.

Abersoft Forth from Amsoft comes in both tape and disc versions, with a CP/M version available on disc. The cassette version comes in a distinctive Amsoft package with a loose-leaf instruction manual contained in a small ring binder and a twin cassette tray containing the single Forth cassette.

The manual is not intended as a teach yourself guide to Forth and states this explicitly in the introduction. However it does devote a few pages to explaining the basics of the language to beginners, and these are illustrated with appropriate examples.

The latter half of the manual consists of a complete glossary of fig-Forth and all the Amstrad extensions are detailed fully in a

separate chapter.

The review copy contained some minor misprints and one or two pages were not printed at all, but these faults will probably be rectified in future issues.

In common with most Forth systems the command line interpreter only recognises Forth words in upper case but the system signs on with Caps Lock on so there is no danger of laboriously typing in long

BY STEPHEN DEVINE

definitions only to have them all rejected when you press Enter.

The basic system supports all the usual looping and control structures of Forth and also includes a full CASE statement which will be familiar to users of Pascal.

Equivalents are provided for most Basic commands, allowing full use of sound and graphics, and text windows can be set up and used as in Basic.

Printer output is catered for by the LINK command which causes all screen output to be echoed to the printer, but beware—if this command is used with no printer connected the system will hang up and the only way out is to reset the computer.

Programs can be entered directly from the command line but to make full use of the system you must first type your definitions in to screens before compiling them. This allows

partly completed programs to be edited, saved to or loaded from cassette.

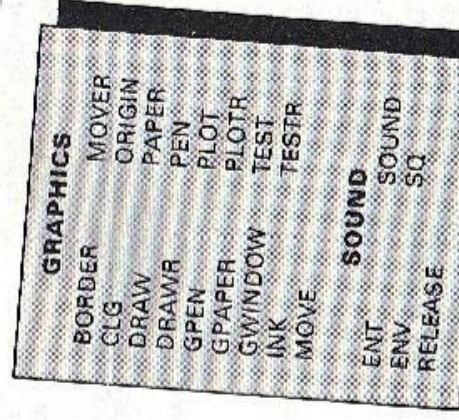
Each screen occupies 1k of memory organised as 16 lines of 64 characters. Eleven such screens are available to the user and although they are used one at a time when entering programs the entire block of 11k must be saved or loaded each time. Since this can take a considerable time a choice of write speeds is available using the word SPEED-WRITE as in Basic.

To facilitate the use of screens a comprehensive editor is included which can operate in one of three modes. The first is a simple line editor which allows you to manipulate lines of text within a screen. In this way whole lines can be inserted and deleted or copied to other parts of the screen as required.

The second mode allows you to list a screen and move around it using the cursor control keys. You can enter or remove text anywhere on the screen, but it is remarkably slow at making major changes. Since it is possible to make a mess of your text in this mode you can choose to abort the edit and return the screen to normal.

A third mode of operation is provided to enable screens to be searched for specified words which can then be deleted or altered as required. This mode also allows entire screens to be moved about in memory.

Since Forth is an extensible language, allowing you to add your own words to the basic dictionary, experienced programmers often prefer to define the simpler and more



Amstrad extensions - Abersoft Forth

	Abersoft	Kuma	
		Slow	Fast
BM1	2	2	2
BM2	11	15	12
BM3	15	20	16
BM4	22	31	22
BM5	15	19	15
BM6	22	31	22
BM7	15	20	16
BM8	19	26	19
BM9	32	26	19
BM10	38	51	37
BM11	25	33	25
BM12	39	35	25
BM13	27	39	27
BM14	11	14	10
BM15	46	50	45
Ave.	22.6	27.5	20.8

PCW benchmark timings in seconds

often used words in a program directly in machine code. Unfortunately most versions of Forth come with little or no facilities to enable you to do so.

However this cannot be said of Abersoft Forth, since the basic system includes a full Z80 assembler as part of its vocabulary. Since this is necessarily Forth-like in its structure it does take a while to get used to, but the effort is worthwhile since it is a powerful assembler, allowing most Forth loop structures to be used in machine code.

The Kuma Forth package also comes in both tape and disc versions. The first comes in a small cardboard box containing a single cassette—in a library case, unlike the Abersoft version—and an instruction manual in the form of a chunky 120 page pocket-sized booklet.

This is a far more comprehensive manual than the Amsoft one. In addition to a complete fig-Forth glossary and full details of all the extra Amstrad commands, there is a detailed guide to programming in Forth which is helpful for beginners.

Although there were a number of misprints in this manual none of them appeared to cause any serious difficulties in understanding.

As with the Abersoft version, the Kuma interpreter only recognises capital letters and you must press Caps Lock before entering any commands. (Could it be that Kuma are unaware that Caps Lock can be

switched on from within a program?)

All the standard fig-Forth words are recognised as well as a variety of loop structures and, again, a full CASE statement is also included. Extensions are also provided to enable sound, graphics and most other Amstrad features to be implemented.

Most of these extensions are not provided as part of the basic vocabulary. Instead they reside on tape, just after the Forth itself, and must be loaded into memory and compiled.

At first this may appear to be a nuisance, but in fact it illustrates just how versatile Forth can be and is the proper method of adding extensions to the language.

With this system you are provided with a compact version of Forth which contains all the necessary building blocks to enable you to construct a comprehensive library of extensions which you can then use, or not, as desired.

This is far more efficient than having a vast mega-Forth system which contains every possible feature you are ever likely to need, since no single application is going to use them all. Those that remain unused take up valuable space and will often slow down the execution of a program.

Kuma has made good use of this facility by providing features which would not normally be available on a stand-alone system. These include extensions for string handling, arrays and floating point arithmetic.

On a standard Amstrad 16 screens are available for creating programs, but this figure can be altered if more or less memory is available.

Screens can be saved to tape individually and are automatically loaded back into memory as required. But this method requires a tape to be formatted by writing a number of blank screens to it before it can be used.

The editor provided is similar to the Abersoft one, with the line editor being almost identical. The full-screen editor operates mainly in overwrite mode—where anything you type replaces the character under the cursor—but there is also a simple insert mode where text can be inserted one character at a time.

You can move freely from one

screen to another without having to leave the editor first, but any changes made are permanent—there is no abandon facility as with Abersoft.

No assembler is provided, which means that any machine code has to be entered byte by byte. This is a pity since one could easily have been included as an extension on tape.

Having said this, the system does provide useful facilities for machine code buffs, allowing all Z80 registers to be examined and modified. Their values can then be passed directly to or from firmware routines.

Both versions provide the user with a full fig-Forth system with ample additional commands to make good use of the Amstrad's special features.

If you are operating solely from tape and will make use of sound and graphics in short, easy-to-write programs, or if you really need a full assembler, then Abersoft Forth will probably be sufficient.

Otherwise the Kuma version, with its better manual and more versatile method of operation, will enable you to really get to grips with Forth on the Amstrad.

GRAPHICS

ASKCUR	LINER
BORDER	MOVEA
CINK	MOVER
GCLR	PAPER
GETHEIGHT	PEN
GETORG	PLOTA
GETPAPER	PLOTB
GETPEN	SETORG
GETWIDTH	SETPAPER
GINIT	SETPEN
GRAPHICS	TESTA
GRESET	TESTR
INK	WWWHEIGHT
LINEA	WWWIDTH

SOUND

AENV	NPD
AMP-ENV	PITCH
CH	PLAY
CHECK	QUEUE
CONT	RELEASE
DUR	RENDEVOUS
EMPTY	RESET
HOLD	SOUND
HOLDIT	TENV
IAMP	TONE-ENV
MUSIC	TPD
NOTEARRAY	WAIT

Amstrad extensions—Kuma Forth

THE Discman has been imprisoned in the power house of the planet Dartma. This is an endless series of rooms designed to hold the numerous energy generators needed to provide the planet with its fuel.

The rooms are built from ice blocks placed in such a way as to break the area up into a maze of passages.

You play the role of the Discman and move him around the maze using the A, Z, K and L keys or a joystick.

Your task is to transfer the flashing power pills that appear at the bottom of each screen to each of the eight reactors.

To make your task that little more difficult the place is littered with deadly plants which must be avoided at all costs. There are also the ghosts of previous prisoners who seem bent on seeing you off.

They are particularly dangerous because of their ability to walk through walls, and they can only be killed by being crushed by an ice block. To move a block, position the Discman next to it and kick using the Spacebar or fire button.

You can only leave a room when all the power pills have been relocated and at least five ghosts have been killed.



DISCMAN'S

```

10 REM Discman's Revenge
20 REM By J.L.Lau
30 REM Computing with the Amstrad
40 SYMBOL AFTER 33
50 SYMBOL 38,66,153,86,68,153,86,68,65
60 SYMBOL 145,8,24,24,24,24,24,24,24
70 SYMBOL 33,255,129,129,129,129,129,129,255
80 SYMBOL 64,24,68,126,78,126,255,255
90 SYMBOL 91,62,254,247,31,31,255,1
100 SYMBOL 93,62,127,239,248,248,25
110 SYMBOL 5,126,62
120 SYMBOL 123,188,183,183,255,223,255
130 SYMBOL 125,62,126,255,223,255
140 SYMBOL 182,188
150 SYMBOL 169,255,129,129,129,129,129
160 SYMBOL 178,255,24,24,24,24,2
170 SYMBOL 171,171,171,171,171,171,171
180 DEFINT a=1

```

```

110 ENV 1,188,2,5:ENT 1,188,-2,1:ENV
2,58,1,3:ENT 2,58,-2,2:ENT -3,58,30,2
120 ENT 4,239,-2,18:ENV 4,23,4,1:ENV
5,38,2,2:ENV 6,15,-1,4
130 DIM a(20,24),a(2),b(2),d(2),u(28),24)
140 a$="AMSTRAD",6010 1530
150 REM ***** Initialise *****
160 FOR i=1 TO 25:LOCATE 1,25:PRINT C
CHR$(160):SOUND 1,f+7,1,5:NEXT i:MODE 0:
LI=3:GI=0:PI=0
170 FOR f=1 TO 28:FOR g=4 TO 23:a(f,g)
=-0:NEXT:NEXT
180 REM ***** set screen *****
190 ON SCREEN GOSUB 2570,2760,2920,30
80,3250
200 FOR i=3 TO 17:STEP 2:a(i,3)=4:NEXT
i
210 GOSUB 228:GOTO 280
220 LOCATE 1,1:OPEN 18:PRINT STRING$(2

```

```

8,CHR$(143)):LOCATE 1,2:PRINT STRING
(28,CHR$(143)):LOCATE 1,3:PRINT STRI
NG$(28,CHR$(143))
230 PRINT CHR$(22)+CHR$(1):LOCATE 1,
:PEN 9:PRINT STRING$(20,CHR$(169)):L
OCATE 1,2:PRINT STRING$(20,CHR$(178))
240 FOR f=1 TO 28:LOCATE f,3:PRINT
CHR$(169):NEXT
250 PRINT CHR$(22)+CHR$(0):FOR f=3 TO
18:STEP 2:PEN 4:LOCATE f,3:PRINT CH
R$(142):NEXT
260 RETURN
270 REM ***** print screen *****
280 PRINT CHR$(22)+CHR$(1)
290 FOR f=1 TO 28:FOR g=4 TO 23:IF a
(f,g)=2 THEN PEN 2:LOCATE f,g:PRINT C
HR$(143):PEN 5:LOCATE f,g:PRINT CHR$(
33):GOTO 310
300 IF a(f,g)=1 THEN PEN 1:LOCATE f,g
:PRINT CHR$(38):PEN 4:LOCATE f,g:PRIN

```




Game of the Month

REVENGE

By DAK
LONG LAU

VARIABLES

m() Screen map.
a() Ghosts' x coordinates.
b() Ghosts' y coordinates.
dr() Ghosts' direction.
n\$ Highest scorer's name.
li Lives.
gk Number of ghosts killed.
pl Number of pills transferred.
screen Screen number.
x,y Discman's coordinates.
sc Score.
hi Hi score.
pu Block's direction.
dp Discman's direction.
in Random number.
ch% Character on screen.
du Flag.
p1,p2 Block's coordinates.

PROGRAM STRUCTURE

40-170 Initialisation.
180-260 Read screen map data.
270-310 Construct screen.
320-560 Main loop.
570-890 Discman's movement.
900-1520 Move ghost.
1530-1670 Title.
1680-2120 Ice block movement and checks.
2130-2210 Reset ghosts.
2220-2280 Deduct life.
2290-2320 Kill ghosts.
2330-2350 Dead discman.
2360-2490 Enter hi score.
2500-2560 Complete screen.
2570-3450 Different screens.
3460-3550 Game over.

```
T CHR$(145)
310 NEXT: NEXT
320 REM ***** start game *****
330 PRINT CHR$(22)+CHR$(0):dp=1:LOCAT
E 10,23:PEN 14:PRINT CHR$(231):m(10,2
3)=3:ch%=0
340 LOCATE 3,1:PEN 2:PRINT"GHOSTS KIL
LED 0"
350 PEN 12:LOCATE 1,25:PRINT"HI":LOCA
TE 9,25:PRINT "SC":LOCATE 1,24:PEN 1:
PRINT STRING$(20,"-")
360 LOCATE 4,25:PEN 3:PRINT hi:LOCATE
12,25:PRINT sc:LOCATE 2,2
370 pu=0:dp=1:PEN 12:LOCATE 17,25:PRI
NT STRING$(11,"J")
380 a(1)=2:b(1)=10:a(2)=19:b(2)=10:dr
(1)=2:dr(2)=4
390 x=10:y=22:LOCATE x,y:PEN 1:PRINT"
":LOCATE a(1),b(1):PEN 8:PRINT"0":LO
CATE a(2),b(2):PEN 8:PRINT"0"
```

```
400 IF INKEY(36)=0 OR JOY(0) AND 8 TH
EN 60SUB 570
410 ON pu 60SUB 1730,1850,1960,2080
420 IF INKEY(37)=0 OR JOY(0) AND 4 TH
EN 60SUB 660:60TO 450
430 IF INKEY(69)=0 OR JOY(0) AND 1 TH
EN 60SUB 740:60TO 450
440 IF INKEY(71)=0 OR JOY(0) AND 2 TH
EN 60SUB 820
450 ON pu 60SUB 1730,1850,1960,2080
460 IF (INKEY(47)=0 OR JOY(0) AND 16)
AND pu=0 THEN ON dp 60SUB 1680,1770,1
900,2010
470 in=INT(RND*6):p=1:ON dr(p) 60SUB
900,1070,1260,1400
480 IF INT(RND*350)=2 THEN h=INT(RND*
20)+1:i=INT(RND*20)+4:IF m(h,i)=1 OR
(h=x AND y=i) THEN 490 ELSE SOUND 1,2
50,60,5,5:m(h,i)=2:LOCATE h,i:PEN 2:P
RINT CHR$(143):PRINT CHR$(22)+CHR$(1)
```

```
:PEN 5:LOCATE h,i:PRINT CHR$(233):PRI
NT CHR$(22)+CHR$(0)
490 IF INKEY(36)=0 OR JOY(0) AND 8 TH
EN 60SUB 570
500 ON pu 60SUB 1730,1850,1960,2080
510 IF INKEY(37)=0 OR JOY(0) AND 4 TH
EN 60SUB 660:60TO 540
520 IF INKEY(69)=0 OR JOY(0) AND 1 TH
EN 60SUB 740:60TO 540
530 IF INKEY(71)=0 OR JOY(0) AND 2 TH
EN 60SUB 820
540 ON pu 60SUB 1730,1850,1960,2080
550 in=INT(RND*5):p=2:ON dr(p) 60SUB
900,1070,1260,1400
560 60TO 400
570 REM ***** pac right *****
580 LOCATE x,y:PEN 1:PRINT"J":dp=2
590 IF x=20 THEN RETURN
600 IF m(x+1,y)=2 THEN RETURN
610 IF m(x+1,y)=1 THEN 2230
```



```

620 IF x+1=a(1) AND y=b(1) THEN 2230
630 IF x+1=a(2) AND y=b(2) THEN 2230
640 IF x+1=18 AND y=23 AND do=1 THEN
SOUND 4,286,45,0,4;INK 1,24,2;do=0
650 x=x+1;LOCATE x-1,y;PRINT "J";RETI
RN
660 REM ***** pac left *****
670 LOCATE x,y;PEN 1;PRINT "L";dp=4;IF
x=1 THEN RETURN
680 IF m(x-1,y)=2 THEN RETURN
690 IF m(x-1,y)=1 THEN 2230
700 IF x-1=a(1) AND y=b(1) THEN 2230
710 IF x-1=a(2) AND y=b(2) THEN 2230
720 IF x-1=18 AND y=23 AND do=1 THEN
SOUND 4,286,45,0,4;INK 1,24,2;do=0
730 x=x-1;LOCATE x,y;PRINT "I";RETURN
740 REM ***** man up *****
750 LOCATE x,y;PEN 1;PRINT "U";dp=1
760 IF m(x,y-1)=4 AND do=0 THEN SOUND
1,164,150,1,2,2;PEN 1;LOCATE x,y-1;
PRINT CHR$(23);LOCATE 10,23;PRINT CH
R$(23);INK 1,24;do=1;p1=p1+PEN 3;m
(x,y-1)=0;sc=sc+10;LOCATE 12,25;PRINT
sc;IF p1=8 AND qf>4 THEN 2500
770 IF y=4 OR m(x,y-1)=2 THEN RETURN
780 IF x=a(1) AND y=b(1) THEN 2220
790 IF x=a(2) AND y=b(2) THEN 2220
800 IF m(x,y-1)=1 THEN 2220
810 y=y-1;LOCATE x,y-1;PRINT " ";LOCAT
E x,y;PEN 1;PRINT " ";RETURN
820 REM ***** can down *****
830 LOCATE x,y;PEN 1;PRINT "D";dp=3;IF
y=23 THEN RETURN
840 IF m(x,y+1)=2 THEN RETURN
850 IF m(x,y+1)=1 THEN 2220
860 IF x=a(1) AND y=b(1) THEN 2220
870 IF x=a(2) AND y=b(2) THEN 2220
880 IF x=10 AND y+1=23 AND do=1 THEN
SOUND 4,286,45,0,4;INK 1,24,2;do=0
890 y=y+1;LOCATE x,y-1;PRINT " ";LOCAT
E x,y;PEN 1;PRINT " ";RETURN
900 REM ***** ghost up *****
910 IF b(p)=4 OR a(p)=1 THEN 1000
920 IF m(a(p),b(p)-1)=1 THEN 1000
930 IF a(p)=x AND b(p)-1=y THEN 2220
940 IF a(p)=p1 AND b(p)-1=p2 AND pu<>
0 THEN LOCATE a(p),b(p)-1;PRINT " ";m
(a(p),b(p)-1)=0;GOTO 2300
950 LOCATE a(p),b(p)
960 IF m(a(p),b(p))=2 THEN PEN 2;PRI
NT CHR$(143);PRINT CHR$(22)+CHR$(1);PE
N 5;LOCATE a(p),b(p);PRINT CHR$(23);
PRINT CHR$(22)+CHR$(p);GOTO 980
970 PRINT " "
980 b(p)=b(p)-1;LOCATE a(p),b(p);PEN
0;PRINT "B";RETURN
990 REM ***** turn left or right ***
1000 IF a(p)>1 THEN 1020
1010 IF a(p)+1;b(p)=1 THEN d(a(p)=3

```



```

x,y-1:SOUND 4,0,65,0,6,0,5:PEN 2:PRINT CHR$(171):FOR f=1 TO 40:NEXT:LOCATE
E x,y-1:m(x,y-1)=0:PRINT" ":RETURN
1710 SOUND 1,140,10,7,0,0,8:pu=1:p1=x
:p2=y-1:m(p1,p2)=0
1720 REM ***** push block up *****
1730 IF m(p1,p2-1)=2 OR m(p1,p2-1)=1
OR p2=4 THEN pu=0:m(p1,p2)=2:LOCATE p
1,p2:PEN 2:PRINT"#":PRINT CHR$(22)+CHR$(11):PEN 5:LOCATE p1,p2:PRINT CHR$(22)+CHR$(23):PRINT CHR$(22)+CHR$(0):p1=0:RETURN
N
1740 IF p1=a(1) AND p2-1=b(1) THEN a=
1:GOTO 2140
1750 IF p1=a(2) AND p2-1=b(2) THEN a=
2:GOTO 2140
1760 p2=p2-1:LOCATE p1,p2+1:PRINT" ":
LOCATE p1,p2:PEN 2:PRINT"#":RETURN
1770 REM ***** push r. *****
1780 IF x=20 THEN RETURN
1790 IF m(x+1,y)<2 THEN RETURN
1800 IF x=19 THEN LOCATE x+1,y:SOUND
4,0,65,0,6,0,5:PEN 2:PRINT CHR$(171):
FOR f=1 TO 40:NEXT:p1=0:LOCATE x+1,y:
m(x+1,y)=0:PRINT" ":RETURN
1810 IF m(x+2,y)=2 OR (x+2=10 AND y=
3) THEN LOCATE x+1,y:SOUND 4,0,65,0,6
,0,5:PEN 2:PRINT CHR$(171):FOR f=1 TO
40:NEXT:p1=0:LOCATE x+1,y:m(x+1,y)=0
:PRINT" ":RETURN
1820 IF m(x+2,y)=1 THEN RETURN
1830 SOUND 1,140,10,7,0,0,8:pu=2:p1=x
+1:p2=y:m(p1,p2)=0
1840 REM ***** push block right ****
1850 IF p1=20 THEN pu=0:m(p1,p2)=2:L0
CATE p1,p2:PEN 2:PRINT"#":PRINT CHR$(
22)+CHR$(11):PEN 5:LOCATE p1,p2:PRINT
CHR$(233):PRINT CHR$(22)+CHR$(0):p1=0
:RETURN
1860 IF m(p1+1,p2)=2 OR m(p1+1,p2)=1
OR (p1+1=10 AND p2=23) THEN pu=0:m(p1
,p2)=2:LOCATE p1,p2:PEN 2:PRINT"#":PR
INT CHR$(22)+CHR$(11):PEN 5:LOCATE p1,
p2:PRINT CHR$(233):PRINT CHR$(22)+CHR
$(0):p1=0:RETURN
1870 IF p1+1=a(1) AND p2=b(1) THEN a=
1:GOTO 2150
1880 IF p1+1=a(2) AND p2=b(2) THEN a=
2:GOTO 2150
1890 p1=p1+1:LOCATE p1-1,p2:PRINT" ":
LOCATE p1,p2:PEN 2:PRINT"#":RETURN
1900 REM ***** push d. *****
1910 IF m(x,y+1)<2 THEN RETURN
1920 IF m(x,y+2)=2 OR y=22 OR (x=10 A
ND y+2=23) THEN LOCATE x,y+1:SOUND 4,
0,65,0,6,0,5:PEN 2:PRINT CHR$(171):FO
R f=1 TO 40:NEXT:p1=0:LOCATE x,y+1:m(
x,y+1)=0:PRINT" ":RETURN
1930 IF m(x,y+2)=1 THEN RETURN

```



```

1940 SOUND 1,140,10,7,0,0,8:pu=3:p1=x
:p2=y+1:m(p1,p2)=0
1950 REM ***** push black down ****
1960 IF p2=23 THEN pu=0:m(p1,p2)=2:L0
CATE p1,p2:PEN 2:PRINT"#":PRINT CHR$(
22)+CHR$(11):PEN 5:LOCATE p1,p2:PRINT
CHR$(233):PRINT CHR$(22)+CHR$(0):p1=0
:RETURN
1970 IF m(p1,p2+1)=2 OR m(p1,p2+1)=1
OR (p1=10 AND p2+1=23) THEN pu=0:m(p1
,p2)=2:LOCATE p1,p2:PEN 2:PRINT"#":PR
INT CHR$(22)+CHR$(11):PEN 5:LOCATE p1,
p2:PRINT CHR$(233):PRINT CHR$(22)+CHR
$(0):p1=0:RETURN
1980 IF p1=a(1) AND p2+1=b(1) THEN a=
1:GOTO 2160
1990 IF p1=a(2) AND p2+1=b(2) THEN a=
2:GOTO 2160
2000 p2=p2+1:LOCATE p1,p2-1:PRINT" ":
LOCATE p1,p2:PEN 2:PRINT"#":RETURN
2010 REM ***** push l. *****
2020 IF m(x-1,y)<2 THEN RETURN
2030 IF x=2 THEN LOCATE x-1,y:SOUND 4
,0,65,0,6,0,5:PEN 2:PRINT CHR$(171):F
OR f=1 TO 40:NEXT:LOCATE x-1,y:m(x-1,
y)=0:p1=0:PRINT" ":RETURN
2040 IF m(x-2,y)=2 OR m(x-2,y)=1 OR (
x-2=10 AND y=23) THEN LOCATE x-1,y:SOU
ND 4,0,65,0,6,0,5:PEN 2:PRINT CHR$(17
1):FOR f=1 TO 40:NEXT:LOCATE x-1,y:m(
x-1,y)=0:PRINT" ":p1=0:RETURN
2050 IF m(x-1,y)=1 THEN RETURN
2060 SOUND 1,140,10,7,0,0,8:pu=4:p1=x
-1:p2=y:m(p1,p2)=0
2070 REM ***** push block left ****
2080 IF p1=1 THEN pu=0:m(p1,p2)=2:LOC
ATE p1,p2:PEN 2:PRINT"#":PRINT CHR$(2
2)+CHR$(11):PEN 5:LOCATE p1,p2:PRINT C
HR$(233):PRINT CHR$(22)+CHR$(0):REUR
N
2090 IF m(p1-1,p2)=2 OR m(p1-1,p2)=1

```

```

OR (p1-1=10 AND p2=23) THEN pu=0:m(p1
,p2)=2:LOCATE p1,p2:PEN 2:PRINT"#":PR
INT CHR$(22)+CHR$(11):PEN 5:LOCATE p1,
p2:PRINT CHR$(233):PRINT CHR$(22)+CHR
$(0):p1=0:RETURN
2100 IF p1-1=a(1) AND p2=b(1) THEN a=
1:GOTO 2170
2110 IF p1-1=a(2) AND p2=b(2) THEN a=
2:GOTO 2170
2120 p1=p1-1:LOCATE p1+1,p2:PRINT" ":
LOCATE p1,p2:PEN 2:PRINT"#":RETURN
2130 REM ***** reset ghosts *****
2140 SOUND 4,0,65,0,6,0,5:pu=0:m(p1,p
2)=0:LOCATE p1,p2:PRINT" ":LOCATE p1,
p2-1:PEN 2:PRINT CHR$(171):FOR f=1 TO
40:NEXT:LOCATE p1,p2-1:PRINT" ":GOTO
2180
2150 SOUND 4,0,65,0,6,0,5:pu=0:m(p1,p
2)=0:LOCATE p1,p2:PRINT" ":LOCATE p1+
1,p2:PEN 2:PRINT CHR$(171):FOR f=1 TO
40:NEXT:LOCATE p1+1,p2:PRINT" ":GOTO
2180
2160 SOUND 4,0,65,0,6,0,5:pu=0:m(p1,p
2)=0:LOCATE p1,p2:PRINT" ":LOCATE p1,
p2+1:PEN 2:PRINT CHR$(171):FOR f=1 TO
40:NEXT:LOCATE p1,p2+1:PRINT" ":GOTO
2180
2170 SOUND 4,0,65,0,6,0,5:pu=0:m(p1,p
2)=0:LOCATE p1,p2:PRINT" ":LOCATE p1-
1,p2:PEN 2:PRINT CHR$(171):FOR f=1 TO
40:NEXT:LOCATE p1-1,p2:PRINT" "
2180 a(a)=20:b(a)=4:p1=0
2190 gk=gk+1:LOCATE 16,1:PEN 1:PRINT
gk:sc=sc+5:PEN 3:LOCATE 12,25:PRINT s
c
2200 IF p1=0 AND gk>4 THEN 2500
2210 RETURN
2220 REM ***** deduct life *****
2230 SOUND 2,100,190,7,0,3:INK 2,20,1
5:INK 8,8,2:INK 0,0,3:PRINT CHR$(22)+
CHR$(1):PEN 5:IF pu<>0 THEN LOCATE p1

```



```

221:PRINT CHR$(233)
2240:PRINT CHR$(22)+CHR$(0):a1=0
2250:FOR f=1 TO 2:LOCATE a(f),b(f):IF
  m(f),b(f)=0 THEN PRINT " ";GOTO 22
70
2260:IF a(f),b(f)=2 THEN:PRINT CHR$(22)+CHR$(11):LOCATE a(f),b(f):
  PRINT CHR$(143)+PEN 5:LOCATE a(f),b(f)
  ):PRINT CHR$(233):PRINT CHR$(22)+CHR$(
  10) ELSE:PRINT CHR$(231)
2270:NEXT f:FOR f=1 TO 2499:NEXT:INK 2,
20:INK 8,10:INK 0,3:LOCATE x,y:PRINT "
2280:Y=Y-1:LOCATE 17,25:PRINT "
  IF Y=0 THEN 2330 ELSE 370
2290:REM *****ghost *****
2300:SOUND 4,0,65,0,0,5:LOCATE a(0),
  b(0):PEN 2:PRINT CHR$(17):p1=0:oc=0
  :qa=a(0):qb=b(0):IF p1 THEN a(1)=20:
  b(1)=4 ELSE a(2)=20:b(2)=4
2310:FOR f=1 TO 30:NEXT
2320:LOCATE qa,qb:PRINT " ":(qf=qk+1:LO
  DATE 16,1:PEN 1:PRINT qk:sc=sc+5:PEN
  3:LOCATE 12,25:PRINT sc:GOTO 2200
2330:REM *****pac died *****
2340:FOR f=1 TO 25:LOCATE 1,1:PRINT C
  HR$(11):SOUND 1,f+2,2:NEXT
2350:IF sc>hi THEN 2360 ELSE 1540
2360:REM *****enter name *****
2370:FOR f=1 TO 100:k=INKEY$:NEXT
2380:CLS:n$="":PEN 1:PRINT:PRINT:PRIN
  T " ENTER YOUR NAME"
2390:PEN 4:LOCATE 7,8:PRINT STRINGS(8
  ,CHR$(208))
2400:LOCATE 1,18:PEN 3:PRINT " Your s
  core is:"sc
2410:FOR f=6:WHILE f(1):f=f+1
2420:IF INKEY$(18)=B THEN 2490
2430:IF INKEY$(8)=B THEN 2430
2440:IF f$=CHR$(127) AND f>7 THEN f=f
  -1:LOCATE f,7:PRINT " ":(f$=LEFT$(f$,
  LEN(f$)-1):SOUND 1,30,5,4:f=f-1:GOTO
  2480
2450:IF f$=CHR$(127) THEN f=6:GOTO 24
  80
2460:PEN 2:LOCATE f,7:PRINT UPPER$(f$
  )
2470:SOUND 1,f*5,5,4:m$=f$+j$
2480:END
2490:IF UPPER$(n$)=sc:GOTO 1540
2500:REM *****screen finished*****
2510:SOUND 1,284,100,5,5,5:INK 0,3,1:
  FOR f=1 TO 6908:NEXT:INK 0,3:FOR f=1
  TO 25:LOCATE 1,1:PRINT CHR$(11):SOUND
  1,f*5,1,4:NEXT:MODE 1
2520:LOCATE 11,10:PRINT " *** Well don
  e, ***"
2530:screen=screen+1:IF screen=6 THEN
  3470

```

```

11,11,11,13,11,1,14,20,14
2800:DATA 10,15,11,15,19,16,19,9,20
  ,12,20,4,21,17,21,7,22,14,22
2900:RETURN
2910:REM *****screen 3 *****
2920:REM *****set blocks *****
2930:FOR f=5 TO 9:m(f,5)=2:m(f+7,5)=2
  :m(f+4,7)=2:m(f+4,7)=2:m(f+11,7)=2
2940:m(f-1,10)=2:m(f+8,10)=2:m(f-3,22
  )=2:m(f+10,22)=2
2950:NEXT
2960:FOR f=1 TO 4:m(f,12)=2:m(f+16,12
  )=2:m(f+4,15)=2:m(f+8,15)=2:m(f+12,15
  )=2
2970:m(f+4,20)=2:m(f+12,20)=2:NEXT
2980:FOR f=15 TO 18:m(2,f)=2:m(18,f)=2
  :m(13,f)=2:m(19,f)=2
2990:m(5,f+2)=2:m(16,f+2)=2:NEXT
3000:RESTORE 3010:FOR f=1 TO 21:READ
  a,b:m(a,b)=2:NEXT
3010:DATA 8,7,4,11,17,11,7,12,8,12,8,
  12,7,13,12,12,13,12,14,12,14,13,1,19,
  20,7,2,19,19,19,20,19,7,22,7,23,14,22
  ,14,23
3020:REM ***set stinging plants***
3030:RESTORE 3050:FOR f=1 TO 24:READ
  a,b:m(a,b)=1:NEXT
3040:RETURN
3050:DATA 2,5,4,5,10,5,11,5,17,5,19,5
  ,7,8,14,8,4,9,17,9,10,10,11,10
3060:DATA 10,14,11,14,4,16,17,16,9,17
  ,12,17,10,19,11,19,3,20,18,20,6,23,15
  ,23
3070:REM *****screen 4 *****
3080:REM *****set blocks *****
3090:FOR f=3 TO 7:m(f,4)=2:m(f+11,4)=
  2:m(f+2,6)=2:m(f+9,6)=2:m(f+6,9)=2
3100:m(f-2,11)=2:m(f+3,11)=2:m(f+2,1
  5)=2:m(f+9,15)=2:m(f+2,17)=2:m(f+9,17
  )=2
3110:m(f-2,21)=2:m(f+13,21)=2:m(f-1,2
  3)=2:m(f+2,23)=2:NEXT
3120:FOR f=10 TO 11:m(f,4)=2:m(f,17)=
  2:m(f-2,19)=2:m(f+2,19)=2
3130:m(f-3,23)=2:m(f+3,23)=2:NEXT
3140:FOR f=4 TO 9:m(2,f)=2:m(19,f)=2:
  m(5,f+2)=2:m(16,f+2)=2:m(3,f+9)=2
3150:m(18,f+9)=2:NEXT
3160:FOR f=9 TO 11:m(8,f)=2:m(13,f)=2
  :m(5,f+4)=2:m(16,f+4)=2
3170:m(5,f+9)=2:m(16,f+9)=2:m(18,f+10
  )=2:m(13,f+11)=2:NEXT
3180:REM ***set stinging plants***
3190:RESTORE 3210:FOR f=1 TO 34:READ
  a,b:m(a,b)=1:NEXT
3200:RETURN
3210:DATA 8,5,9,5,12,5,13,5,1,6,4,6,1
  ,7,5,20,6,10,8,11,9,3,9,18,9
3220:DATA 6,10,7,10,14,10,15,10,10,11

```



```

1,11,19,13,12,13,1,14,20,14
3230 DATA 10,15,11,15,2,17,19,17,10,1
8,11,18,7,20,14,20,9,21,12,21,6,23,15
,23
3240 REM ***** screen 5 *****
3250 REM ***** set blocks *****
3260 FOR f=2 TO 5:m(f,5)=2:m(f+7,5)=2
:m(f+14,5)=2:m(f+2,7)=2:m(f+12,7)=2
3270 m(f-1,11)=2:m(f+15,11)=2:m(f+4,1
5)=2:m(f+10,15)=2:m(f+5,17)=2:m(f+9,1
7)=2
3280 m(f-1,21)=2:m(f+15,21)=2:m(f+3,2
3)=2:m(f+11,23)=2:m(f+10,9)=2
3290 m(f+5,20)=2:m(f+9,20)=2
3300 NEXT
3310 FOR f=2 TO 4:m(f,9)=2:m(f+4,9)=2
:m(f+7,9)=2:m(f+15,9)=2
3320 m(f+6,11)=2:m(f+9,11)=2:m(f+1,13
)=2:m(f+4,13)=2:m(f+13,13)=2:m(f+14,1
3)=2
3330 m(f-1,17)=2:m(f+16,17)=2:m(f,23)
=2:m(f+15,23)=2:m(f+11,13)=2:NEXT
3340 FOR f=5 TO 7:m(2,f)=2:m(7,f)=2:m
(14,f)=2:m(19,f)=2
3350 m(6,f+4)=2:m(15,f+4)=2:m(1,f+8)=

```

```

2:m(20,f+8)=2:m(3,f+9)=2:m(18,f+9)=2
3360 m(5,f+10)=2:m(16,f+10)=2:m(5,f+1
4)=2:m(16,f+14)=2:NEXT
3370 RESTORE 3380:FOR f=1 TO 6:READ a
,b:m(a,b)=2:NEXT
3380 DATA 10,7,11,7,4,8,17,8,1,12,20,
12
3390 REM *** set stinging plants ***
3400 RESTORE 3410:FOR f=1 TO 41:READ
a,b:m(a,b)=1:NEXT
3410 DATA 4,4,6,4,8,4,10,4,12,4,14,4,
16,4,1,5,20,5,9,7,12,7,2,8,19,8
3420 DATA 5,10,10,10,11,10,16,10,7,12
,14,12,2,13,9,13,12,13,19,13,4,18,17,
18
3430 DATA 1,19,3,19,7,19,8,19,10,19,1
1,19,13,19,14,19,18,19,20,19
3440 DATA 6,21,15,21,8,22,13,22,4,23,
17,23
3450 RETURN
3460 REM *** end of game ***
3470 MODE 1:PEN 1:LOCATE 16,10:PRINT"
EXCELLENT!";PRINT:PRINT" You have
completed the whole game."
3480 PRINT:PEN 2:PRINT"

```

Score



Give your fingers a rest ...

All the listings from this month's issue are available on cassette.

FULL AMSTRAD RANGE OVER 220 SOFTWARE TITLES IN STOCK AT ALL TIMES!

Mail order and
Telephone specialists

Computer Oasis

Telephone: (09) 385 1885

- AMSTRAD DUST COVERS
- DISK WIZARD
- GOLDMARK T.I.E.S.
- MONITOR EXTENSION LEADS
- DISCMASTER
- SPEEDTRANS

Dealer inquiries welcome.

tech-soff

Computer Wholesale
Telephone: (09) 385 1765

324 STIRLING HIGHWAY, CLAREMONT, WA 6010

Reel Time Audio Tutor for Locoscript

IT may at first seem rather a strange idea to use an audio tape for teaching what is normally an entirely visual activity like word processing, but in fact a talking book can offer considerable advantages over printed material.

True, a cassette tape cannot replace a reference manual, but it is the ideal medium for step-by-step tutorial sessions, mostly because you can keep your eyes on the screen or the keyboard while listening to instructions and taking appropriate action.

Of course much depends on the quality of the tuition, so let me say right away that this is one of the best audio com-

Tutor with flair

puter courses I have come across, both in presentation and in the selection and organisation of material.

The course is aimed at the novice, and therefore assumes absolutely no knowledge of either the PCW or LocoScript.

You are taken from loading paper into the printer, and using the Direct Printing mode – certainly the best way of introducing both hardware and software – via Cut and Paste, Find and Exchange, Set and Clear, block operations, copying files and so forth, to an overview of templates and layouts.

Obviously the full details of LocoScript's many features cannot be covered in just two hours of spoken tutorial. However all the basics are there.

At the end of the course a

beginner should be able to generate professional-looking hard copy, and be fairly competent in disc and file management, none of which can be guaranteed after a new user has ploughed for the first time through the official LocoScript documentation.

The tutor has neither the pompous seriousness of Radio 3 English nor a childish, falsely chummy humour which in the end only insults the intelligence of the learner.

In the Reel Time course everything is explained with admirable clarity, and the material has been carefully paced to maintain interest without the risk of saturation.

There is neither too much recapping, nor too much emphasis on introducing new concepts as quickly as possible. The course comes on two

AMSTRAD PCW 8256 & 8512

Word Processing with

LOCOSCRIPT

REEL TIME

tapes, with full instructions and a LocoScript command reference card.

The price is far from exorbitant for such a polished and effective product, and indeed a bargain if you consider the work that has clearly gone into it.

Gabriel Jacobs

Batman

"HOLY smoke Batman, arcade games on the PCW".

"Yes Boy Wonder, it's not just a word processor you know". I always knew it was possible, I just wondered how long it would take.

Well now it's finally happened, and you can take a break from accounts/payroll/spreadsheets to play zap 'em shoot 'em ups and arcade adventures on your PCW – when the boss isn't around.

The caped crusader's latest adventure, originally written for the CPC range of micros, has been converted for the PCW. If you're wondering whether a word processor which runs CP/M could ever produce a good arcade game there's no need to worry.

Naturally the sound is somewhat limited since the PCW isn't capable of the range notes or effects available on the CPC micros. What sound there is good, and certainly adds atmosphere to the game.

Now on to the game itself. Gotham City's super villains have gained entry to the

batcave, and have hatched a fiendish plot to rid the city of the caped crusader.

The batmobile has been sliced into small pieces and hidden deep within the labyrinth of the batcave, and to make things more difficult many booby traps and devilish monsters have been placed within the cave and must be avoided at all costs.

There's no time to waste. Mild-mannered multi-millionaire Bruce Wayne dashes into his study, jumps on to the batpole and slides down to the batcave, donning his suit on the way.

This is where you take over. You start by dropping into the batcave, and you must find the pieces of the batcar and glue it back together with super batglue.

The screen display shows a superbly detailed 3D view of the room and its contents. The graphics are state of the art and couldn't be better.

You can walk in front, behind and around objects, jump on top of them and even kick them around. Beware though, some are deadly and

you'll disappear in a puff of smoke if you try.

Initially all you can do is wander around the many rooms in the batcave. It's best to forget about the batmobile to start with and map the batcave. This is essential since it's very easy to get lost.

You'll find several objects that will help you in your quest. On entering certain rooms a small Batman floats down from the top of the screen. These are useful, so if you see one run over and give it a kick.

Some provide you with super batenergy enabling you to run at lightning speed for a limited time and is useful for dodging monsters.

Others provide you with protection against danger such as falling on spikes, bumping into exploding rocks and bites from vicious dogs.

In one of the rooms you'll find your batboots which enable you to jump.

Your batbackpack is useful for storing objects, a batthruster will enable you to steer when falling and an anti-gravity belt will slow your rate of descent. Some rooms look



fairly straightforward at first sight, but can be quite tricky. There are sliding conveyor belts which whisk you off in the wrong direction – usually to a horrible untimely death.

Batman is a superb multi-screen arcade game. The many locations and puzzles will keep you occupied for weeks.

I did miss the Riddler, Joker, Penguin and friends, along with the Zap! Kapow! Thunk! of the original TV series, but even so it's an excellent game.

If you want a change from word processing I can thoroughly recommend Batman.

Roland Waddilove

Sound	4
Graphics	10
Playability	10
Value for money	10
Overall	9

INTERNATIONAL KARATE

THIS is a typical martial arts game, the theme having been used to good effect before.

The loading screen displays the two opponents set against a montage of scenes. Also in the top left hand corner is a number which decreases as the tape loads, giving you an idea when loading will be complete.

You take charge of one of the characters, the computer or an opponent the other, and there

Eastern combat

are 16 different movements controlled by either joystick or keyboard. They include high kicks, flying kicks, jumps and a variety of punches and blocks.

Points or half-points are awarded not only on whether you successfully strike your opponent but also on the way in which you execute your moves.

If there is no outright winner after 60 seconds the judge either asks you to fight again or will award the round to the player with the highest score. The fight sequences are

set against different backdrops including Sidney harbour with its famous Opera House, Copa-Cobana beach and Sugar Loaf mountain.

The backdrop graphics are colourful and effective, the characters easy to control and their movement smooth and precise.

The sound effects are typical but the incidental music that plays all through the game will soon have you turning down the volume control.

Even though International

KNIGHT GAMES

JUST as the software houses were running short of martial arts to computerise, English Software has come up with a new slant.

Using ye olde worlde combat as inspiration they have put together Knight Games, a compendium of eight different ways to knock your medieval opponent senseless.

The events are divided

equally between the two sides of the cassette and English Software has provided space on the cassette inlay to record the tape counter readings which correspond to each event.

The program consists of two swordfighting games, quarterstaff, pikestaff, archery, crossbow, ball and chain and axeman.

The games can be played by one or two players and control is via keyboard or joystick.

Unlike many of the Kung Fu games a description or dia-

Alternative combat

gram of each individual move is not supplied. Most must therefore be approached using a furious joystick-wagging technique until individual moves become apparent.

The first swordfight takes place between two armoured knights on the battlements of an English castle.

The game is played for points, the winner being the one to score the most points before the time limit expires.

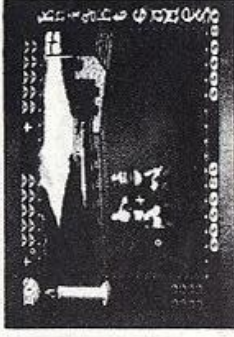
All the games are fought using similar tactics, with only the weapons and backgrounds



Karate is not that much different in playing style from many others the backdrops do give it a slightly different feel.

Ian Duerden

Sound	6
Graphics	8
Playability	7
Value for money	8
Overall	8



changing.

Knight Games provides a big, bold, colourful, alternative to the many oriental offerings on the market.

Jon Revis

Sound	8
Graphics	9
Playability	7
Value for money	8
Overall	8

Beat the baron

undercover to try and smash the drug ring and will help you all they can.

Whenever a member of the Nexus group appears on the screen a digitised photograph of his or her face is displayed in one of two windows.

By using one of the many blue personnel computer terminals you can identify these members. Most of them have special skills which will be of use to you.

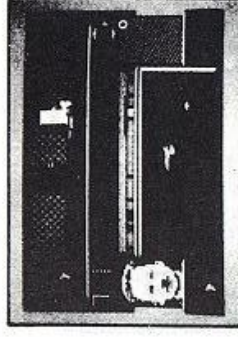
If you wish to track down one person in particular then the terminals can be used to activate a homing device which indicates the direction

in which the agent can be found.

Apart from the digitised picture windows there is a large one at the top of the screen in which most of the action takes place.

You are an extremely athletic reporter who can execute somersaults, forward rolls and a wide range of karate moves at the push of a button.

By locating specific members of the Nexus group you can arm yourself with a machine gun and grenades. Initially the bad guys kept on getting up after I had shot them - that was until a Nexus



member told me to jump on them while they were on the floor.

This is a pretty good example of an arcade/adventure spy game - plenty of action, plus problems to solve.

Steve Brook

Sound	7
Graphics	7
Playability	7
Value for money	7
Overall	7

NEXUS

IN Nexus you are a newspaper reporter whose friend has been kidnapped by an evil drugs baron in South America.

The game is divided into two parts. First you must locate and free your friend, and then piece together 128 snippets of information, edit them into an article, and send them to your newspaper.

Members of the Nexus organisation are working

Gladiator

AS the Roman legions swept through Europe you were one of the many innocent people captured and sold as slaves.

Being a sporting man your master sent you to train as a gladiator and the only chance you have to regain your freedom is to become the Emperor's champion.

The price of your freedom is 32,400 coins. To earn them you must defeat 14 other gladiators and gamble some of your earnings on the outcome

Roman rumpus

of other fights.

You can select either a one or two player game and use of keyboard or joystick. A demo fight is available.

You can kit yourself out with three of a formidable array of 45 shields, swords, axes, spears, tridents, nets and daggers.

The preliminaries dispensed with, you enter the arena where the emperor looks down from the royal box. The gladiators are large and colourful, though rather chunky.

A total of 16 offensive and

defensive moves can be executed. The gladiators' movements would have been greatly improved had they been more exaggerated. I often found it difficult to distinguish between the different moves during a fight.

Armed with an axe and a net I defeated my first two opponents without laying a finger on the joystick. They just battered themselves to death on the axe.

From this point onwards they got a little cleverer. My third opponent entered the arena with identical weapons

Nick Faldo plays the Open

THIS golf simulation uses an accurate scrolling map of the Royal St. George's course at Sandwich to test your skill.

It cleverly incorporates all the hazards of the real game including rough, bunkers, water, out of bounds, sea, roads and pathways.

The opening menu allows you to select either 18 holes, outward or inward nine holes, or practice any of the holes.

The display is split horizontally, the top half showing a

bird's eye view of the hole being played, while the bottom half contains a selection of icons which can be chosen with a pointer controlled by either joystick or keyboard.

Using these you can control club size and the strength and direction of the shot.

Once satisfied with your selection you move the arrow to the play shot icon – a teed golf ball. Alongside, the front view of a beautifully animated figure goes through a better golf swing than I could ever achieve and strikes the ball.

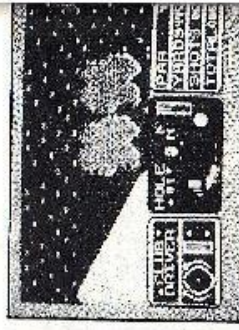
As the ball disappears off screen to the right you can see its progress simulated on the

map in the top half of the screen.

The ball, which was originally represented by a couple of pixels, grows in size as it climbs, and shrinks again as it reaches the fairway – or in my case, more often than not, the bunker.

My only criticism is a minor one. I wish there was some indication of the maximum and minimum distances that can be achieved with each club – a lot of my earlier games were spoilt by over or under hit shots.

The package comes with a hole-by-hole description of the Royal St. George's course,



with measurements from te to bunkers and so on.

The graphics were extremely well done and make excellent use of colour.

This game is a real challenge for anyone who fancies armchair golf.

Alan McLachlan

Sound	3
Graphics	9
Playability	9
Value for money	10
Overall	9

Trashman

I NEVER realised that a binman's life could be so hectic until I played Trashman from Virgin Games.

You apply to the local council for the vacant post of trashman and after your interview are taken on for a probationary period.

On day one you must collect and empty five bins from Montague Road. This must be completed before your 300 bonus points have

ticked away to zero.

The screen displays an aerial view of both the road, the houses and the gardens on either side as you guide Trashman up the garden paths.

Having picked up a bin you will find that Trashman's progress is noticeably slower due to the weight.

At the majority of houses you will be asked by the householder to perform some service for which you will be rewarded with bonus points.

Be careful not to step on the

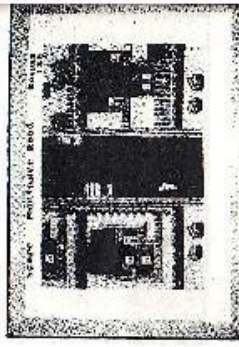
Garbage it's not

grass while they are in the doorway or they will slam the door and you'll get no bonus points from the encounter.

After the first day you are given a new street to work on, involving more bins, more traffic and maniac cyclists who race along the pavement.

Should you fail to finish within the time limit you get two more attempts. If you are killed by a passing car you lose your job immediately. The logic is impeccable.

Trashman requires logical thought to pick the quickest



route to take, combined with the agility of a Frogger expert to negotiate the busy roads. This game is definitely not garbage.

James Riddell

Sound	6
Graphics	8
Playability	8
Value for money	8
Overall	8

Jack The Nipper

JACK the Nipper, bored with the wimps in his town, decides to liven the place up. So with peashooter in hand he sets out on a journey of chaos.

The aim of this game is to attain 100 per cent on the naughtymeter displayed at the bottom of the screen by wreaking havoc in your role as Jack, in the way most appropriate to your surroundings.

Jack has five lives, losing one each time his nappy

Smashing time!

rashometer reaches 100 per cent. At this point one of the five smiling Jack icons on the screen is replaced with a frowning one complete with a sore bottom.

Contact with any other character on the screen, or moving too close to a fire-place, will increase your nappy rash level.

The final piece of information displayed on the screen is the contents of your two pockets.

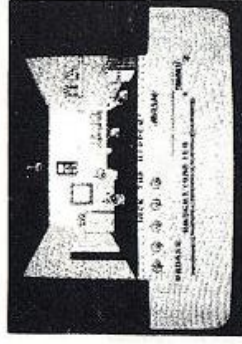
Your trusty peashooter can be fired left or right by moving the joystick in the required

direction and pressing the fire button. You have an unlimited supply of ammunition and are allowed to shoot at anyone or anything.

A word of warning: When firing at people or animals be prepared to run for it.

Buildings include a computer shop, china shop, laundrette, police station, museum, and many more. Some contain objects which enable you to cause havoc elsewhere.

Other objects are just there to be smashed, I had a great time in the china shop! Once



again Gremlin Graphics has produced a graphically excellent and highly entertaining game. The perfect opportunity for naughty children to vent their anger.

Carol Barrow

Sound	8
Graphics	8
Playability	8
Value for money	8
Overall	8

Pacific

SOMEWHERE deep beneath the Pacific lies the lost city of Atlantis, its fabulous treasures waiting to be claimed by an explorer as fearless as yourself.

Your diver - you have four - is armed with an underwater pistol and 32 rounds of ammunition. When he is down to his last eight shots a little yellow stripe on his back begins to shorten.

Further stocks of ammunition can be found in underwater galleries concealing

Dive into danger

a magic lamp which will provide you with an extra three lives. I've seen the lamp, but don't ask me how to get to it.

Leaving the safety of your diving bell you begin the long descent to the ocean bed.

The ocean is at first bright blue and featureless except for venomous fish swimming around.

As you reach the bottom of the screen the display is re-drawn revealing the next stretch of ocean.

The diver can be moved by joystick or keyboard in any direction. However you will soon realise that he can only exit the screen to the left, right,

or bottom. There is no way back up.

As you descend further the background turns black, littered with stunning fluorescent coral structures.

One very important feature of the display is your air supply meter. As your reserves become depleted you had better locate one of the many air pumps to be found on the reef.

One of the most useful items you will encounter is a keg of gunpowder. This can be placed over a section of floor and detonated by firing at it. You must remember to run to an adjacent screen before



the keg explodes or you'll go up with it.

Pacific is not only one of the most colourful games on the market it is also one of the most fiendishly clever.

Jon Revis

Sound	8
Graphics	9
Playability	8
Value for money	9
Overall	9

Classical trio

Oops! command to take back your last move if you make a mistake.

While the games can be played separately, the intention is that players should play the games in sequence and, on completing all three adventures, earn the title of Supreme Adventurer.

Colossal Adventure is based on the first adventure ever written, Colossal Caves. It is quite a lot larger than the original and, in my opinion, much improved.

Your task is to search an underground cave complex and

collect all the treasure by solving all the puzzles. Easily said, harder done.

Adventure Quest has you seeking out and killing the Demon Lord Agaliarept in his dark tower.

You have to find the four Stones of the Elements to destroy the Demon Lord, and this forms most of the basis of the quest. Once again, don't expect to finish this in weeks.

Dungeon Adventure takes up where Adventure Quest left off. The Demon Lord is dead and this time you are a rogue who sets off for the dark tower



intent on collecting the fabulous wealth rumoured to lie in the tower's vaults.

All three games are rightly called classics.

Paul Gardener

Presentation	10
Atmosphere	10
Frustration factor	10
Value for money	10
Overall	10

Jewels of Darkness

JEWELS of Darkness is an updated compilation of three of the best adventures written for any computer. All three, originally released separately without graphics.

For this compilation graphics have been added and the command parser has been improved. You can save positions to RAM, and use the

Blackstar

AS you awaken from a sleep troubled by strange dreams and visions you find yourself in a luxurious room furnished in silver and glass.

A woman, tall and willowy with hair like spun silver is speaking to you. She carries easily an air of authority and wisdom.

She speaks again: "...finally when you locate the orb you must return it to me. You may keep any mortal treasures you find after I have cleansed them of their evil".

Darkness enfolds you until suddenly you find yourself in the sunlight on a valley road close to Castle Blackstar, ready to begin your quest.

Castle of adventure

Having found the castle you are faced with your first problem, how to get in. The drawbridge is up and any attempt to cross the moat is doomed to failure.

A quick search of the castle surrounds should reveal two objects that can be used to lower the drawbridge.

Your next obstacle is the portcullis. This shouldn't prove too much of a problem to an ex-mountaineer like you.

Once inside the castle you will find a bewildering number of objects, which may or may not have a use.

A search of the upper levels reveals several useful objects, and if you take a close look, a bedside adornment that will find you taking a trip across the water. Descending to the

ground floor will lead you to one of the most useful objects of all, a lamp.

If you look at it closely you will see that you cannot use it yet.

The answer was outside the castle — well, you can't expect to always get it right first time, can you?

This is the first PCW adventure that I have enjoyed playing. I have found over 100 locations, with many problems still to overcome, so I would estimate that there are about 200 locations to explore altogether.

The command parser handles most inputs well and the game has several useful commands available such as FREEZE and UNFREEZE to save your position in memory



and to then restore it — handy you contemplate doing something risky.

This is an extremely good adventure. I recommend wholeheartedly to anyone looking for something different on their PCW.

Paul Gardener

Presentation	5
Atmosphere	7
Frustration factor	8
Value for money	7
Overall	7

Fairlight

HOT on the heels of Ocean's Batman is Fairlight by The Edge. Like Batman, it is a massive 3D arcade adventure with many locations, characters and puzzles to solve.

The history books describe the land of Fairlight as a beautiful place, peopled by a fair race with mighty kings and queens.

Many years have passed since that golden age and the once great leaders have grown weaker and there have been many wars.

There is only one remnant of that once great age. Castle Avars, which once belonged to King Avars who ruled when the land was still full of magic.

In 3,000 years on one has ever gained entry to this castle, and myths and legends abound. Some say that within these walls the tall ancestors of Fairlight's people still live.

One day while out walking you are drawn by some

strange power to Ogri's Wood, a place described by the elders as evil and dangerous, an area to be avoided.

Deep within the wood you stumble across Ogri herself, a woman-like creature, and fall in a daze. You awake in a cave to find an old man standing over you.

He leads you outside to the foot of a great wall where there is a small hole. You squeeze through and find yourself within the walls of the great castle Avars. The hole closes up behind you.

There he tells you that the figure you see before you is merely a projection of himself and that he has been imprisoned in the tower for many years.

There is only one way of releasing him and that is to find the magic Book of Light and take it to him in the tower. There he will help you escape from the castle.

So the scene is set, and you are ready to start your adventure.

You'll need to take care: I

Accent on action

don't know how many locations there are, but I was soon lost in the great castle, so it's best to go slowly and draw a map.

This could be tricky, since there are several levels. Rooms may have steps which lead up or down to another level. Many of the doors are open, but some are locked and often guarded.

To gain entry you must find the key and do battle with the guard. If you win you may unlock the door and enter.

You'll find stools, books, barrels, bread, chicken, scrolls, keys and potions.

The food and drink give you energy, the scroll seems to have magical powers of transportation, the keys open doors and the stools are useful for standing on.

There may be others, but even after hours of intensive playing I found I'd hardly scratched the surface of this marvellous game.

The 3D graphics found in Fairlight are almost obligatory these days and no self



respecting software company would be without at least one such title in its range.

The Edge use what they call the 3D Worldmaker technique which adds "real world" features such as size, weight momentum and so on to all the objects. Some characters even have limited intelligence.

Fairlight is an excellent game with superb graphics. Its many locations and puzzles will keep you occupied for weeks, if not months and the game represents good value for money.

I can thoroughly recommend it for all PCW arcade addicts.

Roland Waddilove

Sound	2
Graphics	10
Playability	10
Value for money	10
Overall	9

RESCUE ON FRACTALUS

THE intergalactic war against the Jaggies has been going badly ever since they got a foothold on Fractalus. Our space pilots have suffered heavy losses and many have crash landed on the planet's surface.

It is up to you to take the controls of a nifty space craft and rescue them.

The mothership drops you into the atmosphere to begin your mission. Fractalus is an

Cracking mission

incredibly mountainous planet, so be prepared for some pretty fancy flying.

The screen displays cockpit instruments and the outside view. The graphics are nothing spectacular, but this does not detract from the game's playability.

The ship's long range scanner will detect the signal being emitted by any stranded pilot. You home in on him, wait for the scanner to flash, then hit the button to land your ship.

Rescuing pilots also tops up your energy banks as the men

salvage the energy cells from their crashed ships.

Flying the ship couldn't be simpler. Speed and altitude are relatively unimportant and a clever little display shows you how far each wingtip is away from the nearest peak.

All you have to do is steer, which allows you time to concentrate on zapping the enemy gun emplacements and flying saucers.

There are 16 skill levels and you can start on any of them. The skill level determines the number of alien guns and

MERMAID MADNESS

IF you've ever wondered what an 112-year-old mermaid does for a living when she's not lounging about the rocks combing her hair Mermaid Madness from Electric Dreams will let you into the secret.

You play Myrtle, who needs a husband. She stands at the candy stall on the pier, dreaming of Mr Right when along comes Gormless Gordon, a diver of little repute,

wearing his scuba gear.

Myrtle gives Gordon a toothless smile and, Gordon promptly dives off the pier.

In a bid to escape from her amorous clutches he gets entangled in an old wreck and becomes trapped. His only hope of survival is to be rescued by Myrtle.

As she follows Gordon into the deep blue sea her legs turn into a tail to help her to evade creatures abounding off the coast.

As Myrtle swims past rocky coves, wrecks and Egyptian ruins she has to watch out for

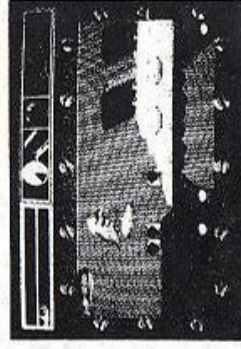
Music charms, but..

sharks, jelly fish, sea horses and rock lobsters. Contact with these creatures deplete her energy.

Above the screen is a bottle of stout whose level decreases when Myrtle loses energy, a heart which beats faster the nearer you are to Gordon, and a gauge showing how much air Gordon has left.

Also displayed are the score, high score and the objects you are carrying.

The graphics are colourful, and there are enough puzzles to keep most players happy, but the game is let down by



Myrtle's indecisive movements, which can only be controlled with the joystick. The best part of the game is the incidental music.

Ian Duerden

Sound	9
Graphics	7
Playability	8
Value for money	7
Overall	7

POWER PLAY

I HAD always believed that the gods whiled away the hours making volcanoes erupt or thunderbolting unbelievers. Well I was wrong. Zeus and Co's favourite pastime was general knowledge quizzes.

Powerplay, designed by Zeus to determine who was the wisest of the gods, can be played by between two and four gods.

Each controls four warriors

which come in four different forms: Hercules, Cyclops, Minotaur and Satyr, in ascending order of strength.

As the game begins you each have three Hercules and a Cyclops. The aim is to be sole survivor in head-to-head challenges.

The action takes place on a courtyard in the gardens of the Temple of Apollo, the floor paved in five different colours of tile.

Four represent the question groups - general knowledge, sport and leisure, science and technology, and history and geography. The fifth colour is a

teleport tile which will whisk you to a different part of the courtyard.

Every turn you decide which player to move and in which direction. The question category is displayed, followed by a scroll containing both question and a choice of four answers.

The desired answer is selected by moving either joystick or cursor keys. If you are correct you can move your warrior to an adjacent square.

Should the square be occupied by an opposing warrior a challenge ensues.

The question and answer



routine is then performed again, and the first person to give the correct answer wins. The winner mutates to the next highest lifeform.

The game is supplied with four separate files of questions.

Carol Barrow

Sound	7
Graphics	8
Playability	8
Value for money	8
Overall	8

THIS month Analysis takes a look at a short but impressive graphics program, exploring how the ORIGIN, MOVE and DRAW commands work.

Amstrad Analysis!

By Trevor Roberts

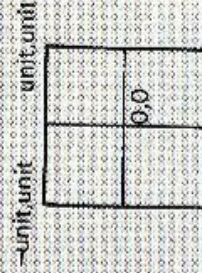
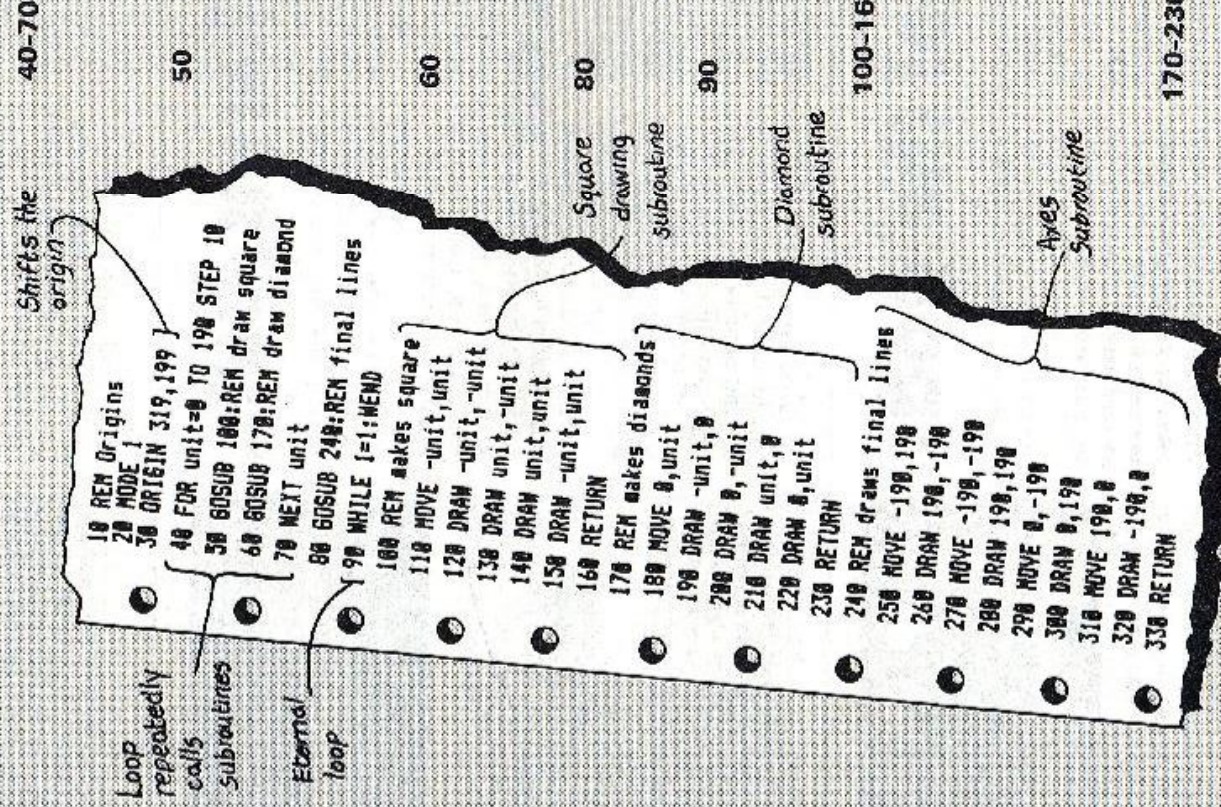
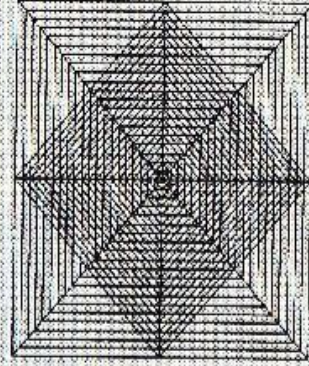


Figure 1: Square coordinates

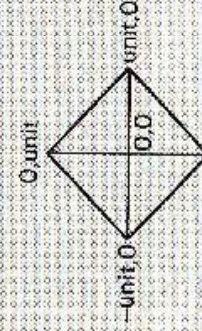


Figure 2: Diamond coordinates

This holds the title of the program. Everything on the line that follows the REM is ignored by the micro.

Clears the screen and puts the Amstrad into Mode 1.

The ORIGIN command is used to reposition the graphics origin (the point that the DRAW and MOVE commands treat as 0,0). Normally this is at the bottom left of the screen, but here the ORIGIN command shifts it to the centre.

These lines form a FOR ... NEXT loop with loop control variable unit. The STEP 10 means that unit varies from 0 to 190 in increments of 10 each time round the loop.

Calls the subroutine that draws the squares. Notice that each time round the loop the value of unit is different. Since the subroutine uses the value of unit to calculate where to draw the lines, this means that the subroutine will produce squares of differing sizes each time it is called.

Another subroutine call, this time one that has the Amstrad drawing diamonds. Again as unit has a different value each time round the loop, so the diamond drawn differs in size.

When the loop has finished repeating the program drops out of it and comes to the final subroutine call. This draws the connecting lines.

This WHILE ... WEND loop is endless. Since !=1 is always true, the loop just cycles over and over. It is included to stop "Ready" appearing on screen and spoiling the display.

These lines form the subroutine that draws each square. Figure 1 shows how the coordinates of the squares are related to each other in terms of the variable unit and the origin 0,0. As unit increases in value so the sides of the squares will increase in length. If you find that confusing, try working out the coordinates of the corners of the squares, as the value of unit goes from 0, to 10, to 20 and so on to 190.

This is the second subroutine. Its job is to draw diamonds. How big each diamond will be depends on the value of unit when the subroutine is called. Figure 2 shows the basic diamond coordinates.

This last subroutine is called when the FOR ... NEXT loop has finished drawing all the squares and diamonds. It just draws in the final axes and diagonals.

BUSINESS COMPUTING

WITH THE AMSTAR



Opening a
window on
vertical
software

How the new PC is slashing prices. The PC family - this is how it all began. Graphics on the PCW: DR Draw in action. Indexing made easy, thanks to Locoscript. Pocket Wordstar: latest update evaluated. Mini Office II and 2 Chess programs for the PCW, reviewed.

I OFTEN have to present my technical ideas supported by view-graphs. LocoScript gave me the means to prepare high quality text but what I wanted was a means of drawing illustrations to accompany the text of various technical and business documents.

Impressed by the PCW's high resolution screen and the high quality graphics function of the printer, I set out to find a way of producing simple block diagrams or flowcharts.

My search started by scanning the ads in *Computing with the Amstrad* and ended after a few minutes when I had established that the only general purpose drawing package advertised was DR Draw - a CP/M+ package developed by Digital Research, the architects of CP/M itself.

Having very little factual information I decided to risk ordering the package on the strength of its pedigree alone. I have not been disappointed.

Now not only have I the means of presenting pictures worth thousands of words to a standard consistent with my text but I can also prepare material for View Foils for use on an overhead projector using type fonts which are much easier to read and which create a much better impression than normal typewriter styles.

DR Draw allows pictures to be built up using a number of fundamental design elements such as polygons, circles and text sequences (see Figure 1). These are laid out and drawn on a "drawing board" by using a cross-hair cursor and two keys which the handbook calls PICK and DONE, the Spacebar and the Return key respectively.

The pictures may be displayed on the screen, printed on the standard PCW printer or output to other specialist devices such as a plotter. Stored as disc files with the Filetype .PIX, they can be recalled and modified at will.

DR Draw was not written specifically for the PCW series and one area where this shows is the cursor control mechanism. It is tedious to move the cross-hairs around the drawing board using only the North, South, East, West arrow keys which, even with the auto-repeat facility takes five seconds to move the cursor across a screen

Add another dimension to your Amstrad

MIKE CROWE makes pictures tell his story with DR Draw

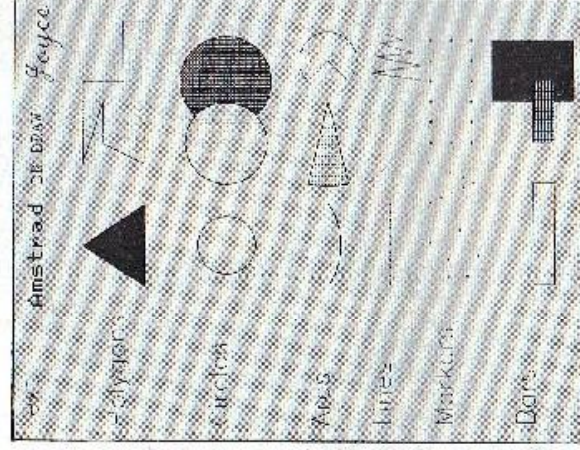


Figure 1: The seven types of design elements

drawing board has a 30 x 20 grid of dots superimposed on it.

You might expect that the cross-hairs have to be positioned over the grid points with pixel accuracy if symmetrical shape is required. Fortunately, a SNAP function is provided. This means that when SNAP is on (the usual situation) the position used by the PICK key is not the cursor position itself but the nearest grid point to it.

For intricate drawing SNAP can be turned off and the cursor may be precisely adjusted by pressing the Shift or Shift-lock keys, which reduce the movement to single pixel steps.

In a similar way further design elements can be added to generate pictures of great complexity, though DR Draw limits the number of elements to 200. Elements may be arcs, sectors, circles, bars (which are rectangles with sides parallel to the edges of the paper), lines, markers (points displayed as single characters), or text.

An element can be modified after

diagonal. The software clearly makes provision for mouse or light pen control and anyone expecting to use DR Draw extensively should budget for such extra hardware. However with a little patience the software is quite usable on a basic PCW.

Suppose you wanted to draw the arrow shown in Figure 11, which is, of course, a polygon. First you select the Add Polygon function and then you are invited to "PICK points, press DONE when finished".

The cursor is positioned at each of the vertices of the arrow in turn and PICK pressed. When the last vertex has been entered you press DONE and the design element is complete.

A drawing grid makes alignment of the design elements simple. When you start a new drawing the blank

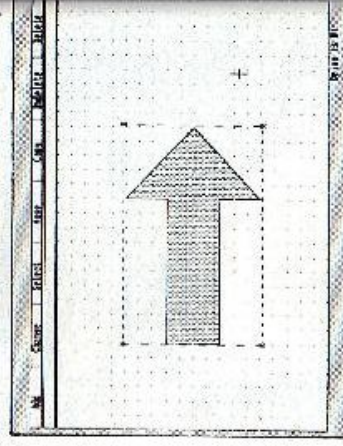


Figure 11: A diagram being produced using DR Draw. Note the main edit menu on the top line of the screen, the cross hair cursor and the drawing grid.

has been drawn. It can be moved keeping its current shape and size, and it can be stretched or shrunk in the horizontal and vertical dimensions. It can be deleted altogether (but I have often been grateful that DR had the forethought to provide an Undeleted function to recover the last deleted element).

If a drawing has a number of identical symbols the first one may be copied again and again in different positions.

The current element may also be presented in different styles. For example enclosed shapes like circles, sectors, polygons and bars may be shaded in one of 12 patterns.

Linear elements such as lines and arcs may be drawn with a variety of

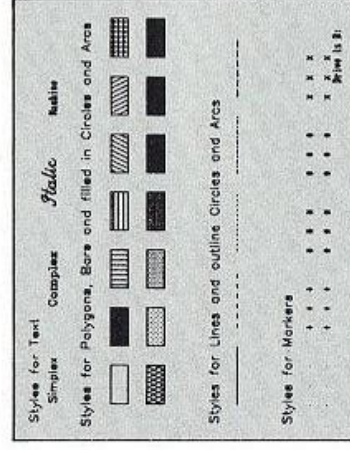


Figure III: Available styles

broken lines as well as just a continuous one.

Text may be presented in one of four fonts – machine font (standard PCW characters), simplex, complex or italic. The last three are selected from a library of eight fonts provided with the software – three of which are almost useless as they have no numerals. Figure III shows the choice of shading, lines and the three default text fonts.

The package is menu-driven so an inexperienced user will find it easy. However as proficiency grows, the speed of operation becomes limited by the speed at which the menu tree can be traversed.

The top line of the screen shows the current main menu, and when appropriate the second line shows a sub-menu. To select an item from the menu, you position the cursor under the chosen item and use the PICK key. Figure IV shows the logical map

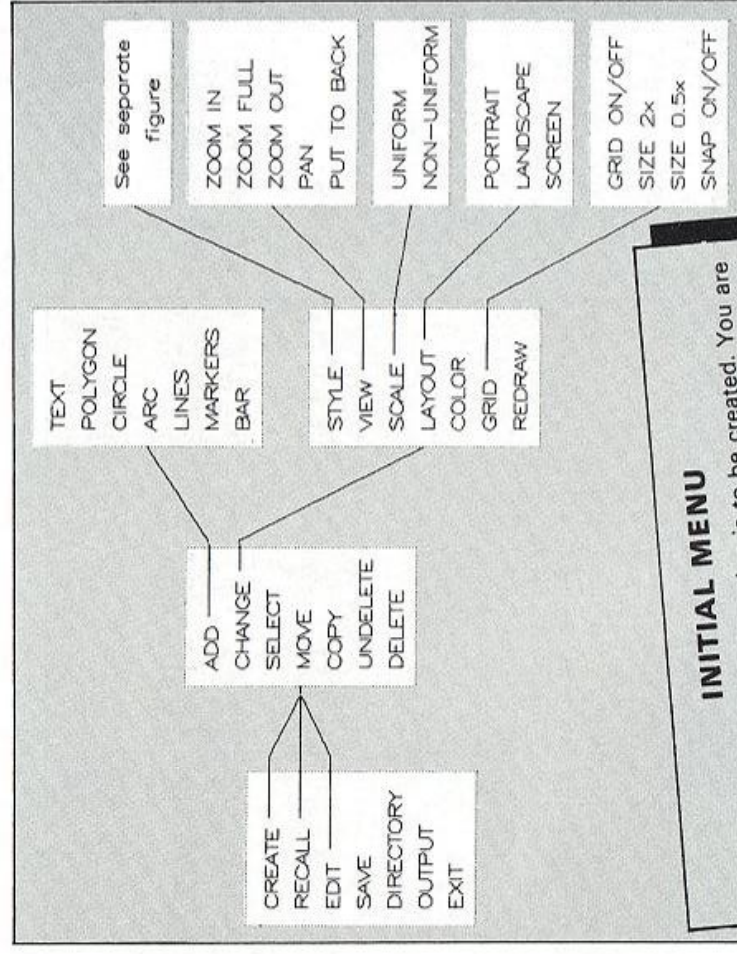


Figure IV: The menu tree

INITIAL MENU

CREATE

A new drawing is to be created. You are asked to specify its name, which will be subsequently used as the default for its .PIX filename, and then you are presented with a blank drawing board with a grid, and the Edit Menu.

RECALL

Allows a specified existing drawing (.PIX file) to be recovered from the disc to allow it to be edited. The original picture is set up on the drawing board and the Edit Menu set up.

EDIT

Sets up the current drawing for further editing.

SAVE

Writes the current drawing to disc as a .PIX file for subsequent recall or output.

DIRECTORY

Displays the names of all .PIX files on the disc.

OUTPUT

Causes the drawing contained in a specified .PIX file to be displayed on the screen or to be printed. It also allows for output to other devices, such as a plotter.

EXIT

Returns to the CP/M operating system.

EDIT MENU

Allows one of the design elements shown in Figure I to be added to the drawing.

Allows the shape, size and style of the current element to be altered and also enables several changes to be made to the way the whole picture is presented. Figure III shows the styles available.

Designates any design element as the current element so that editing operations can be carried out upon it. The cursor is used to point to the element to be selected.

MOVE

Moves the current element to another part of the picture indicated by the cursor.

COPY

Copies the current element to another part of the picture indicated by the cursor.

DELETE

Removes the current element from the picture.

UNDELETE

Reinstates the last design element removed by DELETE.

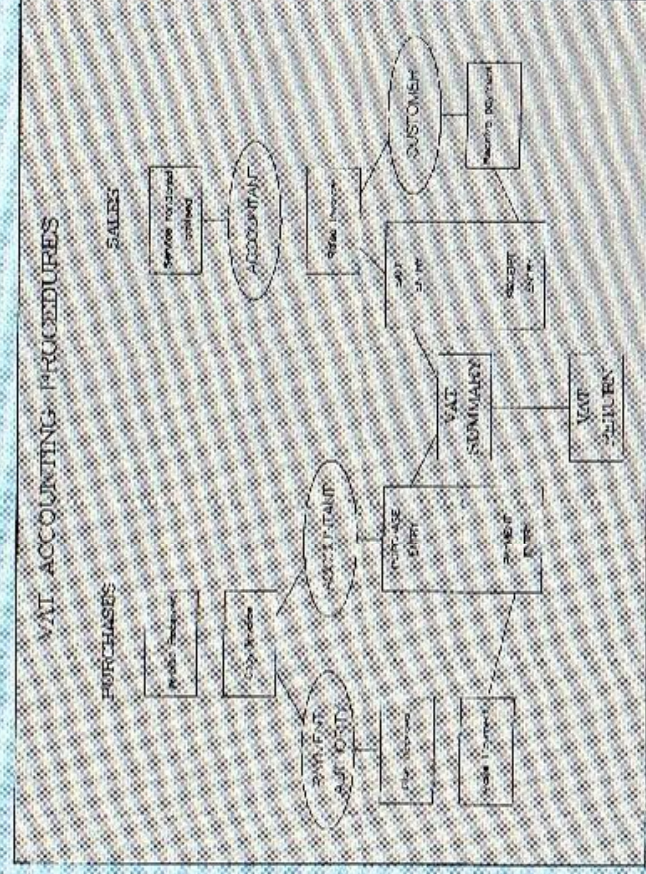


Figure V: A diagram produced by DR Draw on a PCW

of the menu tree.

My main criticisms concern speed of operations. The user interface is a bit clumsy and it can be frustratingly slow to produce a drawing. Also the software is implemented as nine separate overlays and, even using the memory disc for holding these, a disc operation occurs every menu step.

The menu tree does not seem to be optimised for repeated operations, such as happen in practice. A mouse would help, but it would be even better if DR took a leaf out of LocoScript's book and allowed for short key sequences to bypass the menu.

Some of the drawing processes are time-consuming. For example, it takes nearly 20 minutes just to print Figure V which takes up 14k of disc space. Even to redraw this picture on the screen or to change the pitch of the grid can take minutes rather than seconds. This is because the operations are processor intensive complex tasks and the Amstrad CPU is an overworked Z80. Professional CAD drawing systems employ very powerful specialised chips for the drawing function.

A single disc system makes heavy weather of running DR Draw with a lot of disc swapping and flipping. But when the files are sensibly divided between the discs with just the three chosen font files and the current picture file (xxxxxx.PIX) on the working disc on drive A: and with the nine overlay files and the device driver

files on disc M: at least the swapping is limited to the loading phase of the programs.

This can be streamlined by including a copy of CP/M+ on the disc and by using a PROFILE.SUB file to give a load-and-go capability.

There are only three functions absent which I feel ought to be provided in DR Draw. I wish the program had a means of moving, copying or changing composite elements as is the case with more advanced drawing programs. For example, if you need to move a box containing text you must move the box and the text separately.

Also it would be good if text could be automatically centralised in boxes in a single operation. In addition it is often useful to be able to draw lines of different thicknesses, but DR Draw only provides one thickness. Perhaps this is a weakness of GSX (Graphics System Extension) supplied with the PCW rather than of this particular package.

On the positive side, DR Draw comes with an excellent generic handbook which serves as a primer, tutorial and reference manual, supplemented by an adequate installation guide for the PCW. To make a working copy of the software from the release disc is a lengthy but straightforward process despite what seems 100 disc swaps.

DR has clearly set out to make the operation of the program as simple as possible, even if a little long-winded at times. I drew my first DR Draw

picture within an hour of opening the package.

The software had every indication of being well battle-proven and I have encountered no bugs. In particular error conditions, usually caused by operator finger trouble, result in clear messages. It is hard to ruin a drawing which is the result of hours of work by a single careless action.

You can judge for yourself the quality of the diagrams produced here. I am delighted with it.

To summarise my experience, DR Draw adds another dimension to you Amstrad by allowing you to produce professional draughtsman-quality diagrams. It is not the fastest draw in the west and you will usually be able to produce a simple picture more quickly using pen and paper.

However, DR Draw really beats the pen and paper when the diagram needs amendment. The PCW's high-resolution monitor and graphics-quality printer are well matched to the software, which is easy to use, even a little tedious without a mouse.

For my money, at under £50, DR Draw represents excellent value for those PCW users among you who need to produce professional standard diagrams.

Product: DR Draw

Price:

Supplier: Digital Research, c/o Bell Technical Services, 13 Mount Road, Hanworth, Feltham, Middlesex TW13 6JG. Tel: 01-898 9664

Amstrad's arrival sends PC software prices tumbling

AMSTRAD's arrival in the PC market has precipitated the most dramatic price cuts ever by leading business software houses.

Eager to take advantage of the new boom market, they have been falling over one another in their eagerness to exploit the huge potential.

First off the mark was MAP Computer Systems which slashed the cost of its bestselling business packages ninefold.

This meant that overnight its critically acclaimed accounts software prices tumbled from around £1200 to just £149.

"If we had tried this a year ago, we would have been out of business in a couple of days", says Andris Ernstsons, MAP's sales manager.

"Without Amstrad being involved, none of this would have been possible".

MAP in fact jumped the gun on its rivals in the Amstrad market by three months.

That goes back to the time when the company decided to produce PCW versions of its packages at greatly reduced prices.

And it has certainly paid dividends, with turnover more than doubling in the subsequent period.

Digital Research is another major software house which has sufficient faith in the potential of the Amstrad PC for it to announce significant price cuts on its established IBM packages.

It now intends to offer three packages — all of which previously

sold for between £250 and £300 — for just £49.99.

"We are adapting to market conditions", says Frank Iveson of Digital Research. "It is cost effective to produce at one sixth of the price when the market is to increase more than a hundred fold".

Meanwhile Amstrad itself is reporting to have been negotiating licensing deals with other software houses to ensure a plentiful supply of low cost software for the PC.

If there is one thing on which the

business software houses courting the Amstrad market agree it is that the competition will be intense.

"At first, you will see a lot of junk products around", says MAP's Andris Ernstsons, "but when things start to settle down there will be a big shake out and the companies offering inferior products will disappear.

"In our specific area of accounts I anticipate there will eventually only be five software houses left. And you can be quite sure that we will be one of them".

Cheaper US comms

A DEAL struck between UK on-line database MicroLink and its American counterpart Mnemonics will make Amstrad machines even more effective business tools.

The transatlantic computer link that has resulted opens up a new way of communicating with the States.

It means that Amstrad business users who subscribe to MicroLink will be able to send messages via their office micros to the USA for approximately half the cost of an equivalent telephone call. They will

also be able to take advantage of a vast array of additional services offered by the American giant.

Stock market watchers will be able to access instant information from the Dow Jones service.

There are even special interest groups available for professionals such as doctors, dentists, lawyers and writers.

And it also becomes possible for British businessmen to arrange on-line conferences with their opposite numbers in America.

8256 sparks software exports boom

says the rich pickings came when those who had bought their 8256 primarily as a word processor began to realise its potential as a business tool.

"Things really started to take off when users began to say to themselves: 'Wait a minute, I can also do my accounts, stock control and payroll on the 8256'".

He believes that the system's unprecedented low price has had an even greater affect on sales abroad than at home, arguing that there had previously been such little demand for British software because micros had

priced themselves out of the foreign market.

"Australia and Ireland are typical examples, and this is reflected in software exports to these countries which have increased out of all proportion since the release of the 8256".

Grenyer is convinced that the foundation for today's boom was laid when the 8256 was launched in the UK ahead of rival countries. "We had a head start in answering the call for suitable software and were deter-

THE PCW 8256's international success has proved the catalyst of an export boom for the British software industry. Demand has hit record levels not only in Europe but also in Australasia and the US.

Insiders believe that sales are flourishing abroad for largely the same reasons they have been doing so in the UK for the last year.

However because of the staggered release of the 8256 on the home and foreign markets it is only now that software houses are reaping the rewards.

Director of Camsoft Roger Grenyer

mined to make the most of this opportunity", he said. "We succeeded in producing quality, cost effective goods that have yet to be equalled by any other European country."

"This was made easier as our industry is not labour intensive and we have no union/management conflicts. By the time the 8256 was released internationally we knew consumers requirements and had proven products at hand".

Exporting has been a viable proposition because English is such an international language. Like its fellow companies, Camsoft began selling to English-speaking countries.

It has achieved record sales in Europe this year and has made an encouraging breakthrough in Australia where it has just received 200 orders. However it has become obvious that to take full advantage of the foreign market software houses are going to have to translate their programs.

Indeed Camsoft has just completed a Spanish version of its popular database for sale initially in Spain, and potentially throughout Latin America. Grenyer admits that exercise had not been without its difficulties: "Translations have proved a far more intricate, time consuming and costly job than we anticipated".

The accounts package illustrates some of the obstacles that can be faced in exporting: "At present it is not commercially feasible to prepare it for abroad. In addition to the considerable translation costs it would have to be virtually rewritten to accommodate each country's different practices", says Grenyer.

Another company to benefit from the rush for British software is Digital Research, currently exporting on average 3,500 units a month, with French and Spanish translations being prepared of its DR Graph and DR Draw.

Caxton's assistant sales manager Paul Askham reports "a notable increase in the activity of our overseas team. Exports have grown to five per cent of total sales and this figure is steadily picking up".

In accordance with other software producers, Caxton believes that the US market is the ultimate one to conquer. However while the potential awards are phenomenal, so is the

initial investment needed to ensure a realistic chance of making any headway.

Askham feels there is no prospect of Caxton competing in the US in the immediate future. "After all, most of our products originated from there, and they already have their own efficient distribution networks".

He believes the best bet is to exploit those markets which have a dearth of quality software rather than one which is saturated with it.

Sandpiper is a company which has won itself a distribution deal in the US for its accounts package. American director Peter Mackreth admits that contacts have made it easier, but still believes that most British houses have taken the wrong approach to this most competitive of markets.

"Too much money has been thrown indiscriminately at US marketing firms. Alan Sugar got it right when he went direct to Sears and came away with a lucrative deal", he said.

Mackreth makes an analogy with a job searcher: "Who is more likely to succeed—the person who goes to the unemployment agency or the one with the initiative to knock on the manager's door?"

Like most companies, Sandpiper refuses to set itself a target for overseas sales, feeling it futile to make predictions in an industry where conditions are so volatile.

Currently the biggest unanswered question for software suppliers is how will the Amstrad PC affect demand?

What is certain though is the new confidence pervading the industry and the shared conviction that British software houses will continue to increase their stake in the world market.

Eye opener for opticians

A PACKAGE worth keeping an eye out for—but only if you happen to be an optician—is soon to be launched by JRA of Bristol.

Known as the Mint Optician Patient Recall System, it sets out to increase the number of eye-test appointments.

It keeps comprehensive patient details including the last appointment date, next contact date, prescription,

results and special notes.

The system then generates the appointment schedule, producing addressed sticky labels, postcard standard pre-printed letters or merged with the integrated word processing package—individual letters.

Designed by opticians, it is to be priced at £99.99.

The program will be available both the PCW and PC.

Pockets in the chains

AGREEMENT has been reached for three High Street stores to sell MicroPro's Pocket WordStar and Pocket WordStar Deluxe. These low-cost word processing packages for the Amstrad will now be available in all major branches of W.H. Smith, Boots and Lasky's.

All the other Pocket products including Software Classics' Pocket CalcStar, Pocket InfoStar and Pocket SuperSort—will also be available in the same stores. W.H. Smith has included three free Hotline calls with each one bought, so novices can have the benefit of professional help.

Pocket WordStar costs £49.95 and Pocket WordStar Deluxe £69.95.

Compilers

FULL Modula-2 and C program development systems are now available for the PCWs from In Touch. The FTL Modula-2 compiler has separate compilation, procedures as parameters, open arrays and co-processes.

It has its own assembler and linker and directly produces Z80 code in the form of a CP/M.COM system. Real numbers are supported with a 15 digit accuracy.

Included in the development system are library modules, library source code and a full screen editor which can edit multiple files. Price £49.95.

Features of the Ecosoft C compiler include an assembler, linker, more than 100 library functions with transcendental, full preprocessor and manuals. Price: £59.95.

THE first thing to know about IBM is that it is not called Big Blue for nothing. It dwarfs most other computer companies, having over 400,000 employees worldwide, more than 100,000 suppliers, and assets (at the end of 1985) greater than the unimaginable sum of \$50,000,000,000.

This is the company that created the PC standard, to which the Amstrad PC1512 is the latest micro to conform – and the one that will make by far the biggest impact on the UK marketplace.

The PC is, of course, only a cog in the giant IBM wheel, but it has steadily grown in importance since its creation, and has had a huge impact, not only on sales of personal computers but also on their design and performance, whether for good or for bad depending on which end of the telescope you're looking through.

The PC was unveiled in the United States early in 1982. It was not until March 1983 that it was officially launched in the UK, though machines were available here long before that, almost on the black market.

A PC with 64k of RAM and dual 160k floppy drives then cost around £2500, thus slightly more than the other 16 bit machine of the time selling in quantity in this country, the Sirius 1.

The Sirius offered reliability and hardware dealer support equal to that of IBM, and specifications superior to those of the PC.

In other words, things began as they were to continue, with the PC selling on IBM's solid quality brand name, but always lagging behind the competition. Aloof as Rolls Royce, it sanctified innovations pioneered by competitors only after a fitting time had elapsed.

Some people would maintain that IBM's cautious attitude has held back technological progress in the field of micros, others that it has stabilised the market, preventing it from moving too quickly for the consumer to cope, and providing the standardisation so badly needed.

By the time the machine had been launched in Britain scores of software packages were already available for it in the States, and dealers in this country were soon vying with each

Why Blue grew so Big

GABRIEL JACOBS looks at the IBM success story and considers the major part the PC has played

other to be the first to import the latest PC packages. These included now famous names such as Multiplan, VisiCalc, Lotus 1-2-3, WordStar, dBase II and III, and Framework.

It was the enormous amount of software of all varieties produced for the PC right from the start which ensured that it was destined to go from strength to strength. By mid-1984 the fight for standards between IBM and manufacturers of other personal computers was in reality over.

It has to be said that a few manufacturers – Apple is the obvious example – are to this day still battling valiantly against what have for some time seemed impossible odds. But IBM standards will no doubt maintain their dominance in the marketplace, given the inextricably linked facts that the PC's already gigantic software base is continually growing, and that the machine has been universally accepted as the de facto industry norm – a vicious circle as far as the competition is concerned.

Many manufacturers have started out by vowing never to paralyse themselves by making their machines conform to the PC standard, but in the end they have nearly all been obliged to fall into step.

As the PC was buttoning up the market between 1983 and 1985, hardware prices were generally falling, but until very recently IBM's cuts did not keep pace with those of other manufacturers, and the most dramatic slashes have been seen not in the PC itself, but in add-ons and clones.

To get an idea of the fall in the cost

of hardware add-ons, for example, consider that when the PC first appeared, you had to pay around £500 for a 256k RAM expansion board. By the end of 1984 it could be bought for about £250. The official IBM 256k board today costs £100, and you can pick up a compatible 512k board for under £50.

The absolute base-level IBM PC has changed little since it first appeared, and by modern standards is therefore a pretty underpowered machine. It still comes with only 64k of RAM, one floppy disc drive of 360k and a monochrome monitor. But few people own such a rudimentary model. Most of the advanced PC software packages require at least 128k of RAM, many need 256k, and some will run properly only with dual drives, 640k of RAM, a graphics card and a colour monitor.

So, the most common entry level PC is a 256k version with twin 360k drives and a monochrome display. At the time of writing the recommended price of this configuration is £1,988, but it is possible to buy it for as little as about £1,250 or for as much as twice that amount, depending on the kind of dealer support accompanying it.

Most PC users, however, accept that in performance and value for money the PC does not at present measure up to the competition, cloned or not.

The machine is relatively slow, the monochrome screen display is far from impressive when compared to machines in the same price bracket,

From Page 39

and where other manufacturers include as standard a number of features which IBM call optional extras, as well as bundling in masses of free software, the PC arrives on your desk as naked as the day it was born.

Well, not quite. You can at least boot up PC-DOS, the operating system which, because it is supplied free with the machine, has become the most widely used 16 bit system. It is often known simply as DOS, and is a version of Microsoft's MS-DOS. The PC will also run under the rival CP/M-86, which differs little from the CP/M familiar to many Amstrad owners. The other major PC system is Unix, a powerful multi-user multitasking operating system.

One new software package for the IBM PC appears every day on average. The most common practice in the world of 16 bit software is to write first for the PC, then to

configure the program for other machines if possible.

If a piece of software is available for a micro, someone somewhere will have produced it, or something similar, for the IBM PC. The only exception to this is in the world of games for home computers, and even there you can find a wide choice of quality products written specifically for the PC, from flight simulators to adventure programs.

However the machine's main attraction for software houses is in professional applications. The range of serious software is so great that it is a major undertaking these days simply to choose a word processor, a spreadsheet, a database, an accounting package, a programming language implementation, a virtual drive, a graphics pack, a file transfer utility, or even (I'm tempted to write) a program for underwater chicken sexing.

If you want to use your PC as a terminal to a mainframe (not

necessarily an IBM mainframe) or network several PCs, for example you will have to allow plenty of time to evaluate the numerous options.

The choice of hardware accessories is just as awesome as the range of available software. Fixed and removable hard discs, plug-in-and-go printers and plotters, print buffers, memory expansions, graphics cards, graphics tablets, graphics monitors, light pens, multi-function boards, extra ports, specialised keyboard, floor-standing enclosures ... you name it and it's available, either with the coveted seal of approval in the IBM catalogue, or much cheaper elsewhere.

But the PC has had an even greater impact on the hardware market in the way it has spawned compatibles, some of which were already appearing in this country even before the original was officially available.

IBM has spent what to most firms would be vast sums of money pursuing manufacturers - often successfully - whom they believe have infringed their patents and copyrights. But this has not held back the tide, nor the ingenuity of engineers to produce ever more compatible compatibles which will nevertheless not bring them immediately to court.

One of the results of all this is that it has given rise to many different levels of so-called compatibility. As a rule of thumb, you can take the word compatible to mean one of three things, which I set out below with only the tip of my tongue in cheek.

- On its own, compatible means fair.
- Compatible means fair and compatible. Many of the major PC software packages will run with few problems, and some hardware add-ons will work.

- Fully compatible means 90 per cent compatible. Most packages will run reasonably successfully, though some of their lesser-used features may cause difficulties. Many hardware add-ons will need no modifications.

- 100 per cent compatible means 95 per cent compatible. Most users will never know the difference, except that some programs will run faster. A number of clones sell successfully partly because of their speed, though a processor running faster than that of the PC can cause problems under certain circumstances. Care will still therefore be needed in buying

BASIC IBM PC CHARACTERISTICS

Processor	Intel 8088 running at 4.77 MHz
Supplied operating system	PC-DOS (a version of MS-DOS)
RAM	64k expandable to 640k
Drives	One or two double-sided 5.25in floppies of 360k capacity
Expansion slots	5, of which 3 are spare
Monochrome monitor	80 columns, 25 lines, 720 x 350 pixels
Colour monitor	40 or 80 columns, 25 lines, 16 colours, 640 x 200 pixels maximum
Keyboard	83 keys, including 10 programmable function keys and 12 further function keys. Numeric pad.

The PC XT (Extended Technology) has a minimum of 128k RAM and the system unit includes a 10 or 20 mb hard disc. The motherboard differs slightly from that of the standard PC, but to all intents and purposes the XT is the same machine.

The PC AT (Advanced Technology), however, with its 20 or 30 mb hard disc, has its own operating system, DOS 3, and is really a different machine. It is about three times faster than the standard PC, thanks to its Intel 80286 processor. Not all software packages written for the PC will run on it.

Other versions of the PC include a portable, or rather a luggable, and a number of specialised workstations. A lap-held portable, the PC Convertible, will be available soon.

software in order to be sure that it will run precisely as intended (or that any differences are immaterial), and certain hardware add-ons should be tested on the compatible in question, unless the manufacturer of the add-on specifically names it as being a suitable recipient, and even then...

One curious fact about the whole problem of incompatibility is that IBM itself is in some ways the worst offender. The PC AT (Advanced Technology) sounds as if it ought to be a PC upgrade, and in some ways it is. But it uses a different processor, its own operating system (DOS 3), and is only partly compatible with the base-line machine.

Again, early in 1984 IBM launched the PC Junior and it turned out to be far less compatible with the PC than any clone, which ironically was probably the reason for its rapid demise (it was abandoned before ever being marketed in the UK). What is more, there are now half a dozen different versions of PC-DOS and, across the range of IBM personal computers, five different so-called standard keyboards, not to mention numerous points of incompatibility between various IBM, or IBM-supported, networks.

The American satirist Ogden Nash once declared that it does not matter if a husband and wife are incompatible as long as he has the income and she's pottable.

Ignoring the sexist nature of this remark, there is probably something to be learned from it in the realm of the 16 bit personal computer, where there's a deceptively stable marriage between IBM with the money, sometimes bumbling its way along, and other manufacturers producing attractive PC clones, or at least dual-standard compatibles.

It is not surprising, therefore, that the PC far outstrips all other personal computers when it comes to sales figures, despite all IBM's complaints about unfair competition from the clone manufacturers. At the time of writing, the PC XT is the best-selling 16 bit personal computer in Britain (over 15 per cent of market share), followed by the PC AT (12 per cent), and the basic PC (10 per cent).

It is true that number four in the charts is a blatant clone, the Olivetti M24 (8 per cent), but ACT and Apple, whose models share the next six

FIVE SELECTED VERY COMPATIBLE COMPATIBLES - A QUICK GUIDE

Comparison based on a configuration of 256k RAM (though in the case of the Olivetti and the Ericsson the minimum RAM is 128k), twin 360k drives and monochrome monitor.

	IBM PC	Commodore PC	Compaq	Ericsson PC	Olivetti M24	Zenith PC
RRP (£)	1,988	1,199	1,995	1,667	2,165	1,795
Intel processor	8088	8088	8086	8088	8086	8088
Clock speed (MHz)	4.77	4.77	4.77 or 8.0	4.77	8.0	4.77 or 8.0
Output ports supplied (P = Parallel S = Serial)	P	P & S	P	P & S	P & S	P & S
IBM keyboard configuration	No	Yes	Yes	Yes	Yes	No

Most compatibles (particularly those priced higher than the PC itself, and including the five mentioned above) offer various extras as standard, and are capable of running straight MS-DOS as well as PC-DOS.

It is possible to buy compatibles at prices very much below those quoted here, since the competition is fierce, and they tend to be

heavily discounted. An Olivetti M24, for example, with the RAM upgraded to 256k, twin drives and monochrome display can be found for as little as about £1,400. If you want to buy a machine produced in the Far East, you can get a 640k twin-drive model for around the £700 mark, and Osborne's new compatible costs only £495 in its single-drive version.

places, can manage only about 25 per cent between them. What is more, according to a recent survey it seems that dealers are reluctant to sell very low-priced compatibles, and not one has ever appeared in the top ten for sales figures either in Britain, Europe as a whole, or the States.

Despite the fact that IBM, as a matter of policy, never discusses either future products or possible future price changes, there have been strong rumours this summer of massive price cuts in the States which could well have reached us by the time you have bought this magazine, and which, it is whispered, could bring the cost of a heavily discounted PC to within the price bracket of the cheapest compatibles.

Precisely what effect that will have on the clone manufacturers is anybody's guess, since most of them

claim that their profit margins have been cut almost to the bone. If the genuine article costs little more than an imitation, even for a short while, we could see a sudden increase in the already large chunk of the market held by the PC.

One final word: It seems that the Amstrad PCW, until now classed by most computer market research companies in the dedicated word processor category rather than as a personal computer, will soon be joining the big 16 bit boys for survey purposes.

We shall see how that affects published market share percentages, though the view of many pundits is that the IBM PC and the Amstrad PCW simply do not share the same sector of the market. That may be true, but then, of course, there's that other Amstrad machine...

3D CLOCK
CHESSCOLOSSUS
4 CHESS

NOW - CHESS ON THE PCW

3D CLOCK is an impressive new program written specifically for the PCW, and Colossus has earned itself the reputation of being the most powerful home computer chess program around.

The obvious first point of comparison in reviewing these two packages was to pit them directly against each other. Various levels of play were tried, and the results were unequivocal — Colossus won nine games out of the ten played.

Quite a performance. But does it mean that Colossus must be a better choice than 3D Clock? Not necessarily.

In the first place screen presentation can be a crucial consideration in computer chess, and here 3D Clock claws back what it loses in power to Colossus.

It offers superb graphics — in fact as good as that vogue-setting Sinclair QL display often left on the screen in computer stores.

Secondly 3D Clock gave its rival a good run for its money in most of the games. Colossus' record against a human former county player — me as a matter of fact, though that was a long time ago — was only slightly better than 3D Clock. The scores were 5-3 and 3-3 respectively.

What is certain is that both these programs will offer a fair challenge to the majority of club players, without an interminable wait between each computer move.

Both offer all the standard facilities now expected in chess software: self-play, forced computer moves, solving mating problems — unfortunately only of the chess variety — reverse board-orientation, save and load games from disc and so on.

Colossus has a wider range

of features but some players may well feel that 3D Clock has all they will need.

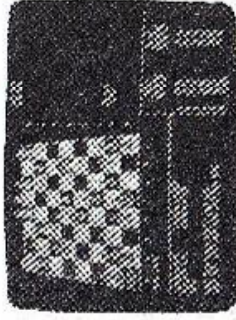
The 3D Clock display consists of a large board with beautifully drawn shaded pieces and an animated analog chess clock, complete with buttons and sweeping second hands, all giving an amazingly life-like effect.

But it is not just a matter of appreciating the computer wizardry that went into its development — it really does recreate a match-play atmosphere.

One of the problems of a 3D board is that it can be difficult to distinguish pieces when they overlap on the display, and I occasionally found myself checking the tops of other free pieces to be certain that I had not made a mistake.

However the problem has been reduced in 3D Clock by clear outlines and a good board angle.

The screen also shows digital read-outs of elapsed move time and total time, and various pieces of information concerning the program's activity, such as the number of positions it has analysed and the depth of its look-ahead



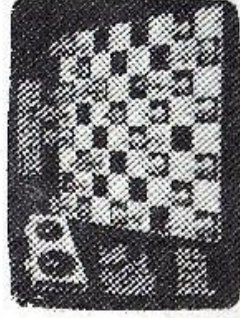
search measured in ply — as in playwood — a move by one side being one ply. It also shows the move it predicts you will make — which you can use as a hint, an assessment based on pieces and position of how the program thinks you are doing and so forth.

As with all modern chess

programs the level of play is set on a sliding scale by how much thinking time the computer is allowed — in this case between two seconds and about 16 minutes.

During this time it will systematically search deeper and deeper, while displaying the best line it has so far found, until its time is up.

Colossus matches all the screen information given in 3D Clock, but its display is not of the same standard. The 3D



board is smaller, the pieces are less well drawn and confusing, and overlap sometimes causes serious problems of piece recognition.

Fortunately the 2D option has been kept, and produces a more readable and playable display.

When it comes to board control, however, Colossus wins hands down. Whereas in 3D Clock all moves have to be entered in alphanumeric form — d1-e3 and so on Colossus makes full use of the cursor keys, allowing you to keep an eye on the chess board instead of carefully looking at the keyboard for fear of making a mistake with the coordinates.

Another major plus for Colossus, and no doubt one of the reasons why it plays a stronger game, is that it uses its opponent's time to carry on thinking.

It assumes that its opponent will make the move it has predicted — as with 3D Clock this can be displayed as a hint — and calculates a

response accordingly.

Of course if the opponent decides on a different move the extra thinking time has been wasted, but I found practice that Colossus was predicting nearly 50 per cent of my moves.

Other features not found in 3D Clock include the ability to single-step backwards or forwards through a game and disable the computer moves and thus provide an electronic chess board for two players, illegal moves and positions are still trapped.

Unlike 3D Clock, Colossus can disable the openings book (which incidentally is far more extensive than that of 3D Clock), provide a continuous display of the last seven moves, adjust the mechanism controlling the extent to which the computer will play for win or a draw and print out a symbolic representation of the current position instead of just a screen dump.

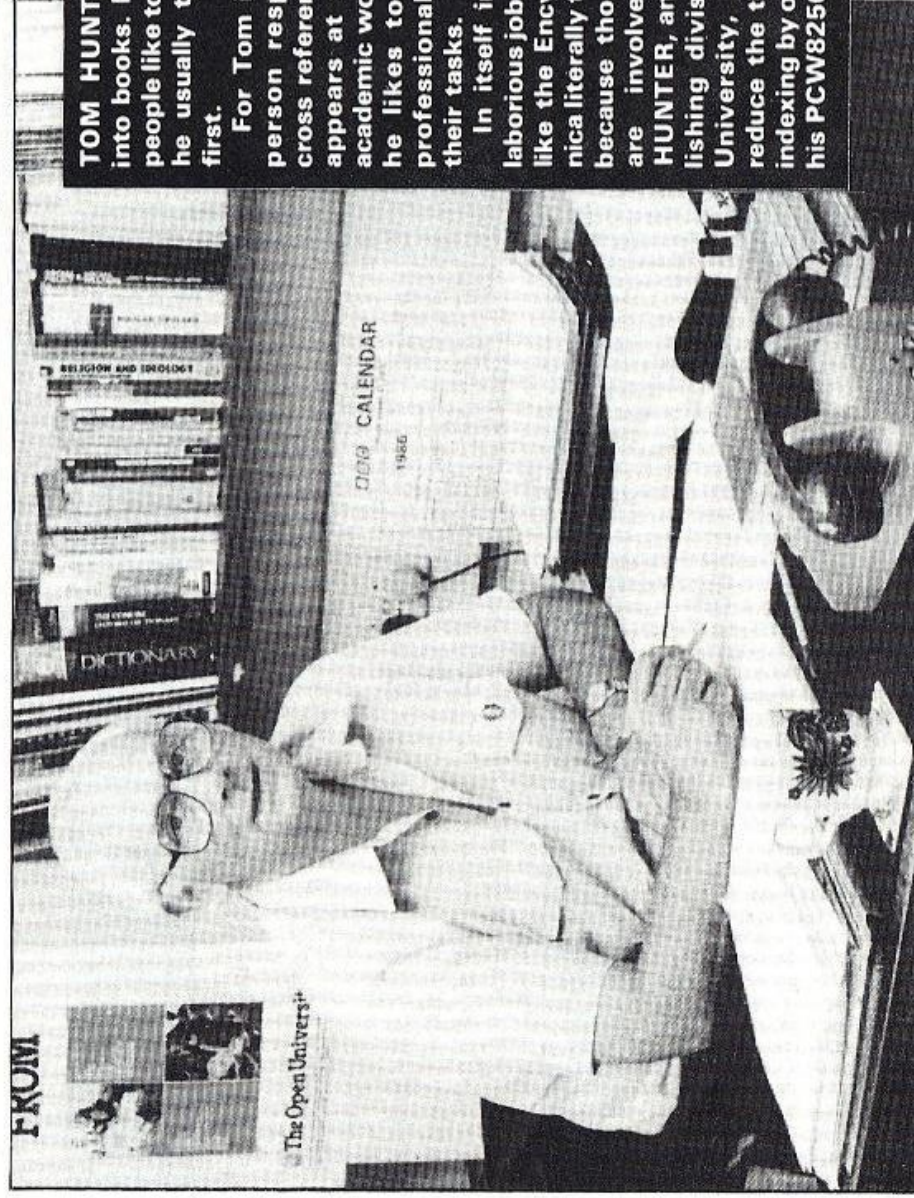
3D Clock comes with just a folded piece of A5 paper Colossus with a dense packed 20 page instruction manual.

True, the author of Colossus has more to explain, but his manual does contain a measure of redundant information. In contrast the 3D Clock instruction sheet is all lean meat and no fat, and crystal clear. Many technical writers would do well to take a leaf out of this book.

If you want to sit looking at a magnificent screen display get the feeling of playing a real match and have a reasonably good game of chess plump for 3D Clock.

If you want a stronger game, easier control of the pieces and are not too worried about good 3D graphics Colossus will be a better bet.

Gabriel Jacob



TOM HUNTER is very much into books. But whereas most people like to start at page one, he usually turns to the back first.

For Tom is an indexer, the person responsible for the cross reference section which appears at the end of most academic works. So, naturally, he likes to see how other professionals have handled their tasks.

In itself indexing can be a laborious job. Reference works like the Encyclopaedia Britannica literally take years to index because thousands of entries are involved. Today **TOM HUNTER**, an editor in the publishing division of the Open University, has managed to reduce the time he spends on indexing by one third, thanks to his PCW8256. This is his story.

Look up 'easy' in the index and find an 8256

MY main reason for buying a PCW 8256 was to help me in my work as an indexer. I wanted something more than just an electronic typewriter – which I already had anyway – I wanted a reasonable word processor with the capability of being up-graded to a more sophisticated machine at some time in the future.

And I didn't want to spend more than a few hundred pounds in the first place, otherwise all my potential earnings would be tied up in hardware for some years to come.

So that you can understand how I thought the PCW would be able to help me with my indexing work, I'll sketch in briefly my routine for compiling an index. The indexer must read through the proofs and select the entries or headings recording them in some easily manipulable form, written in longhand. In simple terms, a heading is a subject, say, economics,

or an individual discussed in the text. Each heading may have one or more sub-headings and these can be further divided into sub-sub-headings, each helping the reader to more easily identify a particular subject in the text and to quickly find it.

I use 5in x 8in record cards to enter, by hand, the various headings for the index. These are then sorted into word-by-word or letter-by-letter order and edited. The entries are then typed up on A4 sheets of paper. A 200 page book can generate 600 or more record cards which, together with 25 letter dividers, can very quickly become unwieldy. Add a set of page proofs and you've got a fairly hefty package to lug around.

Compiling an index usually throws up a number of queries which require to be answered, and you have to add such bits and pieces as the birth and death dates of important historical

personages. As few indexers can boast a sufficiently comprehensive library at home where answers to all the various questions can be found, this means paying a visit to a main branch library and totting your record cards with you.

But even when you are working at home several hundred record cards on the loose can quickly become a logistical problem. Where do you keep them tidy? How can they be manipulated quickly and easily? Keeping them reasonably tidy is probably the easier of the two problems to solve. A lot of indexers file their cards in show boxes – 5in x 8in cards need the largest of the shoe boxes, perhaps even a boot box.

But the manipulation problem grows with the number of cards that you use. I once used 1,000 record



From Page 43

cards to index a 300 page book, so I've nothing but admiration for indexers who have indexed much longer tomes using nothing but cards.

For me, this is where the word processor comes in. Now I use record cards only for the first 200 or so entries, or about half the length of the book, then I key these into my PCW and print out a draft version with double-line spacing. It's much easier to work with 25 or so A4 pages than 600 cards.

I usually start keying the index into the Amstrad using a fresh data disc each time so that I have as much memory available as possible. I transfer a suitable template across from my start of the day disc via the PCW's memory drive. At this stage, alphabetisation, whether word-by-word or letter-by-letter, is not an important criterion and I'm not too concerned about typing errors, all of which can be corrected later.

Basically an index is a list, and an index could be produced by a computer using a sophisticated software package. But it wouldn't be a very useful reference tool because it would lack the skills of interpretation and cross-referencing that only an indexer can give it.

However so far as the PCW is concerned an index is a list — often of fairly short lines with a lot of tabbing for sub-headings under main headings. When you have a longer than usual sub-heading entry to key in and it entails more than one line you will find that one of LocoScript's unfortunate shortcomings. You can only tab after a hard return and not after a soft return or wordwrap. Which means I have to send the cursor back to the end of the previous line and press the Return key. Figure 1 shows you what happens if you use a soft Return and Figure 2 a hard Return.

My wife helps me to key in the entries, which allows me to do other little jobs — research for example — which I wouldn't be able to do at this stage. When all the entries have been keyed in I produce a draft printout which is checked against the record

Armed Forces' Revolutionary Council (AFC) 16, 29, 43, 51, 76, 80,
92-7, 100 8, 113, 122, 140, 157
arrests
for armed robbery 17, 35, 49, 72
for housebreaking 7, 59
Association of Recognized Professional Bodies 117

Figure 1: The result of a soft Return

Armed Forces' Revolutionary Council (AFC) 16, 29, 43, 51, 76, 80,
92-7, 100-8, 113, 122, 140, 157
arrests 17, 35, 49, 72
for armed robbery 17, 35, 49, 72
for housebreaking 7, 59
Association of Recognized Professional Bodies 117

Figure 2: The result of a hard Return

cards for accuracy. Spelling and errors in alphabetisation are also checked at this stage.

The remainder of the book is now indexed, the entries being written on the draft printout in coloured inks for legibility. Once this part of the job has been completed both corrections and new copy are keyed into the PCW. LocoScript's Cut and Paste feature is a tremendous help in manipulating entries into the correct alphabetical order.

Another way in which the word processor has a tremendous advantage over most typewriters is that words which should appear in italics in the finished index can actually be printed in *italics* on the 8256's printer. Keying in the instruction to italicise the relevant words or figures is just as easy for the typist as underlining.

A last-quick visual check on the VDU, and the final version of the index is printed out. At present I use single sheets for the printout rather than continuous stationery, mainly because I never seem to be able to burst continuous stationery without leaving a ragged edge, which I don't

think lends a good appearance to job.

Unfortunately, this does take me time because each sheet has to be in by hand, while continuous stationery needs less supervision, as time becomes more of a constraint. I may have to look at the problem again in the not too distant future.

I do have one other problem: continuous stationery needs more space for the pad of paper to be positioned next to the printer. Space is at a premium and the lead between printer and computer is not all too long, although of course I am aware that an extension lead is available as an accessory.

My plans for the future? I am looking at the possibility of buying a spelling checker which would be an asset in both indexing and free-lance writing and an indexing package such as Macrex that would automatically do many of the things I have present to do by hand while editing the record cards.

Whichever way I decide to move, I am certain the PCW will prove its an ever more useful tool.

Mini Office II database in action

In the July issue of CWTA, after a brief introduction comparing Mini Office II with other all-in-one systems, I described the creation of the Database structure and the entry of the records themselves.

Now we'll look at the facilities provided for manipulating the records once they are on file by looking at a cricket club's cashbook.

The first of these requirements is to be able to list these records, either in whole or in part. The record we created is represented in Figure 1.

In order to have a sensible number of records to work with I have created 10 records on a file called CRICKET. These are listed below. Since we are assuming the treasurer will only need to sort or select the records by the fields marked with a *, I requested a listing which showed only these five fields plus the date.

between each field. If in doubt as to what the eventual listing will look like, you should select Hardware Options from the main database menu followed by toggling Output to the Screen.

Once satisfied with the design, you should toggle back to your printer type.

Assuming you enter these or similar records on to the file the following examples of sorting and selecting will match the explanations below. However, first I will discuss the marking of records. The aim of this feature is to split the file that is held in memory into two, where Group 1 is those records which are marked - in other words identified by a * in the top right of the screen when they are presented by the Edit Record option - and Group 2 is the unmarked records.

Selecting records for marking can be done either directly from the Edit Record option by pressing M or with the Search

or separately printed with Print Marked Records. The former permits the construction of a series of subfiles, each with identical structures but representing different situations. An example of this would be to search for all records which had:

Field 2 Transaction Type = MEMFEE

and save the records which "passed" this on to a separate file called MEMBERS.

After selecting Clearing All Markers, in other words restoring the file CRICKET to its state before the marking of records of Transaction Type = MEMFEE, a search could be made for all records which had:

Field 2 Transaction Type = TEAS

This new set of marked records could then be saved on to yet a third file called TEAS. Figure II shows the result.

Searching or selection is not limited to just a single field, since any number of criteria may be chosen, for example:

Field 2 Transaction Type = MEMFEE

and

Field 2 Reference = CASH

would select those records which represent the money received for membership fees in cash rather than by cheque.

The final point to note when using the search facilities is that Mini Office II is case specific. This means that if you had input some records using MEMFEE, others using Memfee and some using memfee, then a request to search on Field 2 using MEMFEE would only mark the first group. You would not have selected all the membership fee records.

This is a clear proof of the requirement to make a note of the valid entries which may be made in any search field, as mentioned in the July /August edition.

Sort	Select	Field	Length
*		Date	3 characters
*	*	Transaction type	8 characters
		Name	16 characters
	*	Purpose	20 characters
		Quantity	3 characters
		Unit price	3 characters
		Amount received	0 characters+
		Quantity	3 characters
		Unit price	3 characters
*	*	Amount paid	0 characters+
		Reference	4 characters
		Notes	30 characters
		Total	94 characters

Figure 1: Record structure

A point to remember when printing is that in calculating whether the record, or the portion of it that is required, will be folded on to two or more lines, you should allocate a couple of spaces

Data option.

Records which have been marked may be separately saved on to a new file by using the Save Marked Records option

Two basic types of calculation may be carried out. The easiest to understand is the production of totals. Using the marked records we selected representing the 'Tea Money', this would be carried out by selecting Calculate from the main database menu to enter the calculator, followed by Total Marked Records.

Once the correct field to total has been selected, in this case Unit Price, this would produce the total amount of money the club received for the teas it provided.

The other enables the database to be used in a "What if?" type of operation. If the tea records were selected or marked, one could calculate what the revenue would have been had the price been 15¢ higher. This would be done by adding:

Field 9 - Amount Received by 15

and then requesting the total as before.

Such calculations are performed by selecting Calculation from the database main menu, followed by Alter Marked Records, Selecting Field 9, Selecting '+', 15 followed by Totalling as already described.

The last pair of commands regarding marked records can be particularly useful. The first, Toggle All Markers, is used to remove the markers from the currently marked records while simultaneously placing markers on all those records which previously had none - the selected state of all the records is reversed.

11/05/86	EMPT/PC	S. SURRIDGE	CRICKET BATS	CRICK
12/05/86	DISCOUNT	RECEIVED	GROUND MAINT.	GRND
09/05/86	SALARY	WATERS B.J.	GROUND MAINT.	GRND
09/05/86	SALARY	WATERS B.J.	GROUND MAINT.	GRND
16/05/86	MEMBER	GREGORY D.F.	PLAYING MEMBER	CAS
14/05/86	MEMBER	LEWIS G.D.	PLAYING MEMBER	CHG
14/05/86	MEMBER	RIPPOWS G.M.	PLAYING MEMBER	CAS
25/04/86	MEMBER	PETERS G.R.M.	PLAYING MEMBER	CAS
25/04/86	MEMBER	LAURENCE E.B.	RETIRED MEMBER	CAS
25/05/86	TEAS	1st XI / 3rd XI	VS APPLESTROP	CAS
16/05/86	TEAS	3rd XI	VS SWINGLEBURY	CAS

Figure II: Records in file CRICKET

This is particularly useful when you wish to exclude some records on the file from future treatment. Assuming we wish to total all payments received with the exception of those made for teas, this would be carried out as follows.

Once again we would search through the file for:

Field 2 Transaction Type = TEAS

If we then Toggle Markers, every record except for those which the search found would be marked. We could then do our calculation or saving as required.

The second command prints the records we have marked. This is done by selecting Print Marked Records from the main database menu. The teas records shown were printed off in this way.

As explained last month, with our record size of 94 characters we are restricted to 196 records on our CRICKET file. The

same is true of TEAS, MEMBERS any other we may wish to create. With a creative use of the Save Marked Records option one can easily overcome this limitation.

The main file will always start off CRICKET, but this can then be split that a separate file represents all postings made in a single month. 196 postings limit should not be a problem even at the height of the season. These files could be named CRICKET101 through to CRICKET196 for January through December respectively.

As for subfiles such as MEMBERS providing the separate classifications do not represent a significant percentage of the total number of records held, once again 196 records should be sufficient.

Next time we will look at the Data again, but with the specific application of producing mailing lists.

Date	Transaction Type	Name	Purpose	Quantity
Unit Price	Amount	Receive	Ref	
25/05/86	TEAS	1st XI / 3rd XI	VS APPLESTROP	23
16/05/86	TEAS	CASH 2 TEAS + 1 SCORER		
16/05/86	TEAS	1st XI	VS SWINGLEBURY	12
16/05/86	TEAS	CASH		

Figure III: Results of searching for TEAS

CUMANA'S new update of Pocket WordStar for the PCW will not be hailed as a giant leap for mankind, but it is a step in the right direction for those who want the speed and portability of WordStar – and access to its range of add-ons – combined with some of the goodies that the PCW printer offers so effortlessly under LocoScript.

The update is still the antiquated WordStar Version 3 – no bold print or underline on screen, no intelligent paragraph reform – but it comes much better installed for the PCW. In particular it no longer thinks it is talking to an Epson MX80 printer, something which occasionally used to send the PCW print head into convulsions.

There is now a choice of 10 or 12 pitch print, wide – 10 pitch double, condensed – 17 pitch, and italic script, all of which can be happily and fairly easily mixed. The program will also handle subscripts and superscripts using half-height characters, underline, bold, double-strike and strike-out – see Figure 1.

Line height and character width dot commands are not implemented, but this will be of little consequence to most users. Draft/NLQ modes are controlled via the normal CP/M menu brought up on the bottom line of the screen with the Ptr key.

The default printer installation is for 10 rather than 12 pitch. This can be changed using Install – WordStar's standard installation program which is supplied on the reverse side of the system disc – but unfortunately the full Install documentation is not included in the package. Hence Figure 1, where there is some advice for those who like fiddling with parameters to get precisely the customisation they want.

Printing aside, the other major improvement in the update is that

SPECIAL PRINTING CAPABILITIES INCLUDE:

- * Boldface, Double Strike, Underline
- * Strikeout SUPERSCRIPT SUBSCRIPT
- * Italic Condensed print Wide print
- * and in combination -- **WordStar**

Figure 1: Example printout using Pocket WordStar

Pocket WordStar update for PCW

**Answer to a user's prayer?
GABRIEL JACOBS investigates**

SET24X80.COM is no longer required. The new version takes full advantage of the PCW's big screen, and with the Help level set low and the ruler line masked it gives 29 lines of text.

With right-justification off the right margin can be safely set at 88 without + signs constantly appearing, though a right margin of 80 is about the maximum for a reasonable page layout in 12 pitch with A4 paper.

The supplied auto-boot sequence transfers all the WordStar files, including Mailmerge, to Drive M – it all fits, even on an 8256 with no RAM upgrade – and there is just enough room on the system disc for CP/M Plus to be put on it. So the whole process of bringing up WordStar from scratch can be carried out in a single operation.

In addition to all that a sensible Setkeys file has been included in the boot-up sequence. This configures the function keys to generate some common commands, and the dedicated word processing keys to perform more or less the functions assigned to them under LocoScript.

Of course you can alter the Setkeys file to suit your tastes if you are not happy with the configuration supplied. I should say, however, that I am still struggling with the international character set and other squiggles in the high Ascii range.

WordStar does not like Ascii

characters above 128, and it produces strange results both on-screen and at the printer when it tries to access them.

If all you require is, say, a French character set this can be achieved without too much bother. You simply copy LANGUAGE.COM on to the system disc and put the command LANGUAGE 1 (French) into the PROFILE.SUB file.

The result will be that certain keys will generate French accents on the screen – see page 134 of the Amstrad CP/M manual for full details. Then to get these accents to the printer you need to include the following sequence in the printer initialisation string in Install:

**1bh 52h (access character set)
01h (French character set)**

Not all that difficult. However problems arise if you then want characters which are not part of the French character set – the pound sign, for example – or if you want the UK character set as standard but occasionally require continental accents.

For a pound sign you can use an upper case L with a minus sign through it as in Figure 1. You can always put accents in by hand, but this is far from satisfactory.

I'm working on the problem, and I'm sure I shall eventually get to the bottom of it. In the meantime I should be glad to hear from anyone with a foolproof solution.

In my experience the problem of foreign accents and other special characters is more widespread than software developers seem to imagine – and it says a great deal for the LocoScript development team that

their product takes all special characters in its stride.

But I suppose that such weaknesses will be dismissed as insignificant by the distributors of Pocket WordStar, and they may be right. After all WordStar may not be capable of the sophistication and flexibility of some of its younger competitors, including LocoScript, but it is well-proven and well-loved.

Therefore for some PCW users Cumana's new implementation could be the answer to a prayer. If you've been missing WordStar after turning your back on it for LocoScript it may be worth your while taking another look at it in its new Cumana clothes. If you've never tried it now is certainly a better time than before. ■

POCKET WORDSTAR

Customising Pocket WordStar

To have 12 pitch as standard with Pocket WordStar you need to enter the following codes into the printer initialisation sequence — option 'D' in the install printer customisation menu:

18h — flush buffer and

1bh 4dh — set to 12 pitch.

You should keep the user-defined functions as in the supplied WS.COM file, except for Function 3 — Control + PE — which should be set to:

12h — cancel condensed print.

14h — cancel enlarged print.

1bh 4dh — change to 12 pitch.

Thereafter Control + PA will revert to 10 pitch, and Control + PN to 12 pitch.

If you prefer proportional space to 12 pitch you will be disappointed. For straight text

proportional space is possible — by entering **1bh 70h 01h** into the printer initialisation string, instead of **1bh 4dh** — but then underline, double-strike and overtype go haywire. There may be a way out of this difficulty, but I haven't found it yet.

One final tip — to make the printer accept continuous or single-sheet stationery from the main WordStar menu, rather than having to keep pressing Clear and Exit as you have to with the installation as it stands, enter **1bh 63h** into the printer initialisation string, then set paper length as required, say 72.

Install will also handle top and bottom margins, and it's worth experimenting a bit until you come up with something that suits you.

Figure 11: Customising Pocket WordStar

NEW FROM PRIDE

SUPERSPRITES	(D)	\$36.50
SUPERSPRITES	(C)	\$27.50
PRINTER PAC II	(D)	\$36.50
PRINTER PAC II	(C)	\$27.50
* F.I.D.O.*	(D)	\$36.50
* AVAILABLE ON ROM		

CREATE FAST MOVING ANIMATED GAMES EASILY. MENU DRIVEN AND INCLUDES ANIMATION DEMO.

FOR USE WITH PERSON COMPATIBLE PRINTERS. PROPORTIONAL SLIDING TEXT DUMPS/ BUFFER OPTION.

FILE AND DISC ORGANISER. ORGANISE AND MANIPULATE YOUR FILES AND DISCS. A VERY USEFUL UTILITY.

PRIDE UTILITIES

ODDJOB	(D)	\$36.50 *
TRANSMAT	(D)	\$36.50 *
TRANSMAT	(C)	\$27.50
SPIRIT	(C)	\$27.50
TOMCAT	(C)	\$19.50
SYSTEM X	(D)	\$26.50 *
SYSTEM X	(C)	\$18.50
SCRIPTOR	(C)	\$17.50

* AVAILABLE ON ROM

DK'Tronics Chip Kit
256k RAM expansion kit for
Amstrad PCW 8256

\$145



Bankcard & Mastercard
Welcome

P.O. BOX 288, MORISSET
NSW 2264
(049) 732754

FROM K.D.S. ELECTRONICS

- * 8-BIT PRINTER PORT \$54.50
- * POWER CONTROLLER - 8 SWITCHING RELAYS : 5 AMPS AC OR DC UP TO 240 VOLTS. CONTROL LIGHTS FANS ETC. \$145.00
- * PRINTER 'T' SWITCH CONTROLS 2 PRINTERS BY EITHER SOFTWARE OR MANUAL CONTROL. \$78.00

PRICES MAY VARY AS STOCK IS REPLACED. WE ALWAYS REDUCE PRICES WHERE CIRCUMSTANCES ALLOW. ALL PRICES INC. P&P.

64k MEMORY
EXPANSION
FROM DK'TRONICS --
\$145.00

ESP LIGHT PENS
LSP 464 (cas) \$45.00
ESP 464 (disc) \$65.00
ESP 6128 (disc) \$65.00

KONIX SPEED KING

JOYSTICK STEEL SHAFTED
* SHAPED TO FIT YOUR
HAND
* INDUSTRIAL MICRO
SWITCHES --- \$35.00

Menace at Midnight

I am depressed to see that Amstrad has not learned from the 664 fiasco, and is intent on maintaining its customer relations at the same low level.

The immediate cause of this outburst is the unpleasant surprise which many other owners of the disc version of Lords of Midnight have also probably experienced by now.

If you save a situation on to the disc containing your copy of the game you will no longer have a game.

Nowhere in the booklet or the packaging does it warn of this, or even suggest that you should use a different disc.

The hideous truth is that the saved versions are simply stored on consecutive tracks, located on disc at tracks - (version number *2) and (version number *2+1).

This explains the 20 different saved versions we are permitted by the program - each disc has 40 tracks.

You must use a completely blank disc to store your saved versions, and never use it for anything else.

I eventually discovered that this information is actually documented - on the disc.

After finding that my saved situation had blown away one copy of the program I decided to investigate with a disc repair utility before turning the disc over and

trying again.

Track 16, sector 1 contains a segment of text entitled "Hi, hacker!" where the saving method is described.

It makes interesting reading, so why is it not in the manual?

It is ridiculous that Amstrad users, who have waited for ages for this magnificent game to appear, should now be expected to be adept at using a CP/M repair utility before we can find out how to play the thing safely.

I consider it irresponsible and hypocritical for Amsoft to release suicidal software such as Sorcery for which it is impossible to make a backup copy.

It will certainly provide ammunition for the argument against copy protection - thanks Amstrad.

It would be bad enough if this were the end of the story, alas it isn't.

Taking a directory of the disc reveals a small number of files.

All programs printed in this magazine are exact reproduction of listings taken from running programs which have been thoroughly tested.

However, on the rare occasions when mistakes occur corrections are published as a matter of urgency. Should you encounter error messages when you type in a program,

However a closer inspection of the disc reveals that they are all small loaders or such, and the bulk of the program is stored and accessed physically as sectors which will appear to the DOS to be unallocated and will be reported as spare to the user by programs such as STAT.

This is fine as far as Lords of Midnight is concerned. However it means that if you ever try to copy any other files on to the disc CP/M is almost certain to overwrite sections of the original program.

Neither of these problems would be more than annoying if Amsoft had bothered to permanently write-project the disc or include warnings in the packaging, or both.

Obviously all Amstrad users are presumed to be experts on CP/M, telepathic or both.

I hope Amsoft has enough spare copies for those people who have been silly enough to trust the game to know how to save itself safely, and who consequently now have a useless disc.

Nicol Froste

they will almost certainly be the result of your own typing mistakes.

Unfortunately we can no longer answer personal programming queries concerning these mistakes. Of course letters about suggested errors will be investigated without delay, but any replies found necessary will only appear in the mail pages.

Tracking down BASIC variables' location

I am interested in Basic programming, with a few machine code subroutines.

Generally I first write my program in Basic, and once the logic of it is sorted out I take sections of it which need speeding up and replace them one by one by a machine code routine to be called from Basic.

How can I, either in Basic or machine code, establish the memory locations where the Basic variables are stored, thus enabling me to manipulate them in the machine code subroutine - numeric, string and array variables?

Before obtaining by 6128, I used MBasic to achieve this goal.

As far as I can tell Locomotive Basic has not got such facilities. Is there a way out?

Leslie Fahidy

•@ is used to find the address at which a variable is stored, @score is the address at which score is stored.

In the case of string variables the address returned is the string descriptor.

The first byte is the length and the second and third bytes give the actual storage address of the string in lo byte, hi byte order.

By the way, Mallard Basic is compatible with MBasic and should achieve the same results.

The facts on discs

I hope you can comment on a disturbing piece of information gleaned from my local computer store.

I recently got a PCW8512 and subsequently went shopping for some 3in double density discs.

Having tried a number of dealers to no avail I came to a rather out-of-the-way establishment, and was told that Amstrad doesn't make them.

I queried the point and he claimed that all Amsoft 3in discs are actually of the same quality.

He went on to say that Amstrad buys the discs from a Japanese company, and then labels them single or double density as required.

If this outlandish information proves to be correct it means that Amstrad is charging customers roughly \$6 extra per disc for a label saying double density.

I have tried formatting and using single density discs as doubles, and at present all seem fine.

However cramming twice the amount of information on what is a single density disc seems dodgy to me.

Please could you help clear up this controversy?
Jon Wooler

•Your informant was wrong when you were told that Amstrad doesn't make 3in double density discs. It does - the catalog number is CF2DD.

All discs start off life as double sided double density, and if any batch fails the rigid quality control checks it is downgraded to single sided.

It is not unreasonable to assume the same applies in distinguishing double from single density.

Mallard Basic

I have a CPC6128 and am interested in writing programs for use on the PCW8256.

Apart from buying Mallard Basic I understand this may be possible if I use compatible Locomotive Basic commands and save by 6128 programs in Ascii format.

I would be grateful for your comments and a list of Locomotive Basic commands incompatible with Mallard Basic.

Geoffrey Rounce

•Mallard Basic on the PCW and Locomotive Basic on the CPC computers have a common core, but work in rather different environments.

So things like FOR statements, GOSUBs and so on are the same, but file handling, screen control and the like are different - but still obviously from the same stable.

We don't have a definitive list, but generally say that people should not have too much difficulty converting non-graphics programs.

Mallard itself - being completely machine independent - has no graphics. The advantage of this is that you can take a program from your PCW and run it on, say, an IBM PC without change.

The disadvantage is that CPC programs which use graphics heavily could be difficult to convert. We would recommend using GSX, the CPIM graphics add-on.

Here's how to get at GSX from Mallard on the 6128 - the PCW is almost identical, it's just the drivers which are different. The following information is for 6128 owners only.

In order to produce a copy of Basic with GSX installed take a blank, formatted disc and copy the following files:

MALLARD.COM from side A of the Mallard Basic master disc,
GENGRAF.COM from side 3 of the 6128 distribution disc, and GSX.SYS from side 3 of the 6128 distribution disc.

Then copy ASSIGN.SYS from side B of the Mallard Basic master disc, DD*PRL from sides 2 and 3 of the 6128 distribution disc and side B of the Mallard Basic master disc and finally GSX-
PREP.BAS from side B of the Mallard Basic master disc.

Note that the files on the Mallard distribution disc should be used instead of the

files with the same name the 6128 distribution disc

Rename the copy of Basic to GSXBASIC.COM and run GENGRAF to start GSX to this copy using:

GENGRAF GSXBASIC.COM MALLARD.COM

GENGRAF.COM GSX.SYS may now be erased.

Construct a subdirectory called ASSIGN.SYS file. ASSIGN.SYS you have copied gives device 1 Mode 2 screen and may be modified to reflect actual devices to be used.

It is wise to remove from ASSIGN.SYS file devices which you do intend to use. All the drivers used in ASSIGN.SYS must be present on your disc.

On the 6128 distribution disc these contain the extension PRL.

New versions of DDMODE0.PRL, DDMODE1.PRL and DDMODE2.PRL are on the Mallard Basic master disc.

These should be used in place of the similarly named files on the 6128 distribution disc. The new version allows area fill and graphical keyboard locator. They can be used with RSX to provide joystick to functions.

Start Basic from the prompt by:

GSXBASIC GSXPREP

which loads and runs with GSX installed and

the program GSXPREP which sets up the jump to GSX.

A Basic program which is to use GSX should contain the following, at some point before the first call of GSX:

```
DIM contrl%(6),ptsin%(128),
ptout%(12),intin%(128),int
out%(45)
GSX%=&H30
```

The arrays defined have the same name as those described in the GSX Programmer's Guide, though note that the parameter block - PB - is not explicitly declared.

To invoke a GSX function it is necessary to set suitable values into entries in the contrl% array and suitable parameters into the ptsin% and intin% arrays. GSX is then called by:

```
CALL GSX%(GSX%,GSX%,
,contrl%(1),intin%(1),ptsin%(
1),intout%(1),ptout%(1))
```

While it is perfectly possible to use different names for the arrays the form of this call must be retained. In particular please note that the arrays must be Integer.

Howard Fisher,
Locomotive Software

No half measures

After owning a CPC464 for nearly two years I decided to buy my first major add-on - an Amstrad DMP 2000 printer.

However I soon found out that I was unable to produce characters such as halves and quarters. I tried in vain for hours on end with no success. Can you help.

A. Atkinson

•Unfortunately you can't print halves or quarters. There are two problems.

Firstly the Amstrad's printer port is only 7 bits wide and the code for a half is 169 which is an eight bit number. It is therefore impossible to send it to the printer.

The second problem is with the printer - does it have a half in its character set?

We haven't got a DMP 2000 manual to check this, but it is unlikely, so even if you could send the code for a half or quarter it probably would not print as one.

Paging puzzles

I am a computer programmer and have been the proud owner of a 6128 since they were released. I am however puzzled by the paging system used by the computer.

I understand that many ROMs may be mapped on to the top area of RAM - C000 to FFFF hex - and that each one is given a unique number to identify it.

Calling the appropriate routine with this number enables or disables the required ROM, as is clearly explained in the 464 firmware manual.

Where I get unstuck is how the system uses its 64k of pages RAM. Is this mapped in the same way as ROM with an individual number and, if so, what are the relevant numbers to swap these blocks in and out of

memory?

I would like to be able to use this memory to hold machine code routines that are switched in and out as necessary.

Also what is the amount of memory that the disc ROM (7) needs for its own use, excluding the 2k buffer it needs when loading - in other words the difference between the lower and upper usable addresses when initializing ROM 7?

In exchange for this information I would like to detail a method of creating invisible lines in Basic code.

The first stage is to type in the code you want to make invisible on to the end of an existing, but short, line - for example:

```
10 TAG::: PRINT"hello"
```

The code is added with three colons, then two spaces before the code. Only one command per line is possible.

By peeking memory from 368 onwards a pattern of

three ones is looked for, and when found the first two ones are poked over the zeroes. This has the effect of causing a fake end of line marker for the Basic lister.

On listing the text after the colons will not be printed, but will execute. To remove the line either edit the line - which will destroy it altogether - or repoke the zeros with ones.

Howard G. Soper

•The CPC6128 has four banks of RAM, each 16k long. These are numbered 1 to 4 and are mapped into the address space between &4000 and &7FFF.

To select a bank load the A register with the bank number and call &BD5B.

Save the A register and flags. When you've finished restore the original RAM state by restoring AF and calling &BD5B again.

The routines below show how a screen may be stored in a bank of RAM and recalled at any time. In fact even Control + Shift + Escape will not destroy it.

```
*****
; CPC6128 Screen Save
*****
ORG &A000
.save
LD A,-1
CALL &BD5B
PUSH AF
LD DE,&4000
LD B,D
LD C,E
LD HL,&C000
LDIR
POP AF
JP &BD5B
.restore
LD A,-1
CALL &BD5B
PUSH AF
LD HL,&4000
LD B,H
LD C,L
LD DE,&C000
LDIR
POP AF
JP &BD5B
;select bank
;switch in
;save flags
;copy RAM
;to screen
;to screen
;get flags
;restore
END
```


SAGESOFT'S Adele Knox-
Roberts, technical support,
and Jim Beckworth, training
manager, giving a dram-
atised presentation of Sage
business software.

The demonstration was just one of a succession of non-stop business presentations which played daily to packed houses in the 150 seat theatre at the three day Amstrad Computer Show.

Within an hour of the doors opening on Friday, exhibitors reported unusually heavy demand from serious users, particularly those interested in the two PCW models. And a

Many came looking for goods to import to their own countries where demand is currently outstripping supply. As a result there were numerous reports of deals worth in excess of £100,000 being struck over the weekend.

One exhibitor confided to *Computing with the Amstrad*

Gareth Little of network specialists Northern Computer best summed up the show: "It's been at each of the Amstel Computer Shows and each has reflected best what has been happening in the market."

"This latest one exacts mirrored what has been happening of late with its shift in business.

"Not that I mean it was full of people in bowler hats wearing broties — although there were a few of those around — but there was a substantial number of what I took to be self-employed people or those with small businesses."

Richard White of Quaker International was yet another who noticed the swing to more serious user. "People were looking for products that make their machines more useful, rather than just entertain them", he said.

Not that the show is exclusively the domain of business market. Far from it. The number of new games the CPC range in fact outnumbered those launched at previous show.

"I took my 10-year-old along with me and left him to own devices while I looked some business software".

garage owner Paul Johnson. "Obviously he was quite happy, because after three hours I had to drag him away from some game or other."

"As far as I am concerned, it proved it was a great success catering for all ages and tastes."

A CRACKDOWN on sales of tape-to-tape copying devices was ordered by Database Exhibitions during the Amstrad Computer Show.

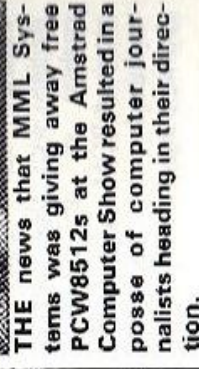
This followed reports that at least two companies had deliberately ignored a directive from the organisers that the products encouraging piracy should not be sold.

"We told them quite plainly that if they persisted, we would have to ask them to leave the show immediately", said Peter Brameld of Database.

in the light of the fact they had chosen to ignore our pre-show warning. But you can rest assured we will make certain it doesn't happen again."

Yet at least one of the companies that had slipped through the net claimed it had ignored the instructions from Database because of Amstrad's own attitude to copying.

"Here you have the firm that is offering twin tape decks with its hi fi equipment. And what more obvious purpose do they have apart from copying other tapes", asked the irate exhibitor.



It turned out the laugh was on them. For the "freebie" was a paper cut-out of the machine that they needed to assemble themselves.

Andrew Clarke, of MML, and Alex Poel, the wife of William Poel, head of New Star Software, show off the inkie offer.

AMSTRAD'S money-spinning PCW8256 has gained its second award in just two months, being named top in the consumer durables category at the 1986 Awards for Marketing.

This follows its success at the British Microcomputing Awards when it was judged the Best Home and Small Business Computer of the Year.

Alan Sugar, Amstrad's chairman, who was pipped by 60 nominations for individuals and organisations.

Computing with the Amazon - November

Compilers for 8256

LOW cost versions of Pro Pascal and Pro Fortran compilers have been launched for the PCW8256.

Prospero Software is making them available through Software Toolshop, the Dunstable-based utility software distributor, at £75 each.

manager for Prospero said: "Amstrad users can now have a try at programming using some of the best professional software available and at a minimal cost."

Both products are equipped with a number of extensions such as long integers, random-access file handling and single and double precision floating point numbers.

The Amstrad versions feature new-style packaging and A5 sized manuals.

WORDSTAR UPGRADE

AN improved version of Pocket WordStar has been released by MicroPro for the CPC6128, PCW8256 and 8512.

Pocket WordStar Deluxe has all the features of the original program plus SpellStar, for checking spelling, and a facility for creating a personal dictionary.

Price: £69.95.

Lifespan of PCW 8256/8512

Amstrad is likely to carry on manufacturing the PCW 8256 & 8512 for at least another two years. That was disclosed by Roland Perry, the company's technical boss, when questioned by a user at the Amstrad Computer Show.

"I think you have got to look at the investment put into the machine, not just in design but in marketing", he said. "I would

be surprised if any distinct product like the PCWs had a life of less than two to three years".

Asked which of the two models were the most popular, Perry revealed they were selling in almost identical quantities. The Amstrad boffin, while commenting on Arnor's new spelling checker for the PCW, leaked the news that the company has one of its

own in the pipeline.

"There is a general intention of having an official spelling checker which you will see later this year", he told the audience at the Amstrad theatre. (Just arrived in OZ - see p.78)

A question about the availability of all Amstrad's computers elicited this answer: "In general all our machines are in full production"



They're the micro beach boys

A SEASIDE cafe overlooking Anglesey, North Wales, is the unusual setting for the newest and probably the smallest computer club.

Retired amusement machine engineer Harry Trew, 62 years young, set up the club with his Amstrad PCW8256, in the beach pavilion and cafe at Llanfairfechan, near Conwy.

"About five years ago I became interested in a computer using it to do the safe accounts. Now I have retired I thought it a good idea to encourage youngsters and grown ups to use the safe as a base for swapping computer ideas and knowledge," said

Harry. The nearest club is at Colwyn Bay, about 15 miles away.

"We have two local experts able to help the youngsters. Some of them are studying A levels and use the machine for their homework".

When Computing with the Amstrad heard about the Welsh club we contacted Inter-Action's Community Computers UK's managing director, Molly Lowell.

She said: "We are more than happy to respond. We will send him two of our handbooks which will answer all his problems about setting up a computer club."

Focus on fun

UNARMED combat skills are vital factors in the new adventure game for the CPC range released by Activision.

The object in The Fifth Axis is to defeat the cyborgs in a race for artefacts from different periods of history.

It contains realistic graphics and multi-level challenges. The disc version costs £14.99, the cassette £9.99.

★
★
★

THE CPC version of Mercenary – Escape from Targ – is to be released next month by Novagen. It will cost £9.95.

★
★
★

GOBLINS and "horrible furry spiders" are promised in the new adventure game for the CPC range.

The Boggit, from CRL, has the ancient wizard Grandalf tricking Bimbo into a quest to find a dragon's secret cave.

The game is split in three parts and has over 100 locations. It will cost £7.95.

★
★
★

FLAME-throwers, grenades and rocket launchers have come into their own in the war game Green Beret from Imagine Software for the CPC range.

Players act as a war machine and have to infiltrate four of the enemy's defence installations.

The game can be played via keyboard or with joystick. The cassette version costs £8.95 and the disc £14.95.

★
★
★

A COMPUTER game version of the science fiction television series V is the latest offering from Ocean Software for CPC computers.

In the game the Visitors plan to destroy the Earth's population from a mother ship commanded by their leader Diana.

The game can be played by joystick or keyboard. The cassette costs £8.95 and the disc £14.95.

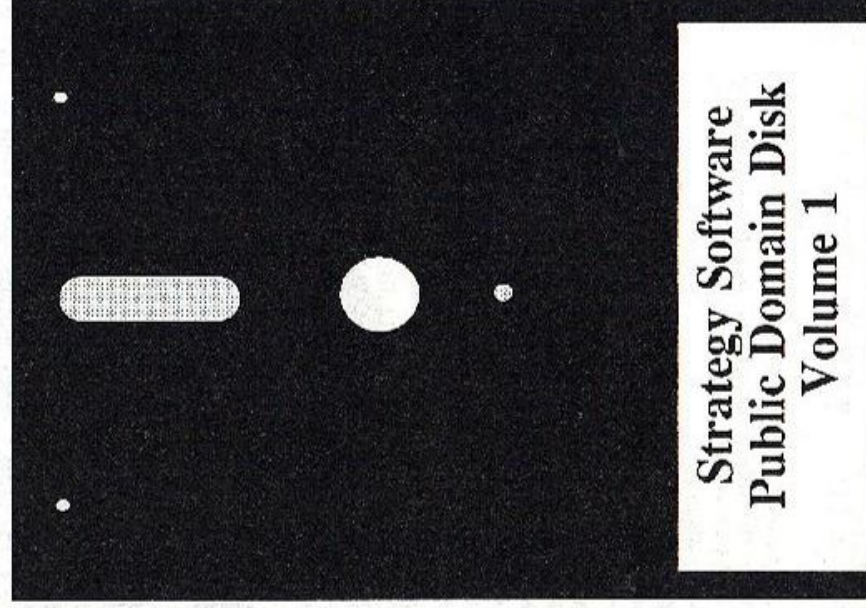
Public Domain Disk Volume 1

Only \$19.95

Price includes P&P

Send cheque or order using
Mastercard or Bankcard

Please quote
Catalog # 8001



Strategy Software
Box 5000
Glen Waverley
Vic. 3150

[03] 560 4324

Suitable for both CPC and PCW computers

CHECKS.COM/DOC
ALLOC.COM
COMPARE.COM
DISPLAY.COM/DOC
CRC.COM/DOC
DIR-DUMP.COM
DIRCHC.COM/DOC
EXAMPL.CHK/NAM/DOC
FILTER.COM/DOC
FINDED54.COM/DOC
MENU.ASM/COM/DOC
NSWP207.COM
PROTOCOL.DOC
SUPERDIR.COM/DOC
VLIST.COM/DOC
DU-77.COM/DOC
FIND.COM
LASM.COM/DOC
LISTT.COM/DOC
PGLST.COM/DOC
PRINTPRN.ASM

SORT.COM
WASH.COM

ZIP.COM

- A cheque book maintenance program [3.0 only]
- Disc utility to show allocation vector [2.2 only]
- compares two files and reports on # of bytes [3.0 and 2.2]
- allows formatted display of files [3.0 and 2.2]
- Compute cyclic redundancy check of file/disk [3.0 and 2.2]
- Dumps directory sectors on Dump format [2.2 only]
- Checks correctness of directory sectors [3.0 and 2.2]
- Example file for CHECKS.COM [see Checks]
- Removes all control characters except: tab,cr, and lf from ASCII files [3.0 and 2.2]
- checks entire disk and locks out bad sectors
- Displays disk contents in menu form and invokes programs by keypress [2.2 only]
- see article [3.0 and 2.2]
- Discussion of XMODEM protocols for comms programs [doc file]
- Rather like 3.0 DIR command [3.0 and 2.2]
- Varies the speed of a file typed on the screen [3.0 and 2.2]
- a disk disector of high caliber [2.2 only]
- Search for an ASCII string in a file [3.0 and 2.2]
- A linking version of ASM [3.0 and 2.2]
- Formatted output to printer [3.0 and 2.2]
- Changes an ASCII file to columnar format [3.0 and 2.2]
- Prints a CROWE assembler file in expanded form (included so non-programmers can see a ASM file) [unknown]
- Sorts an ASCII into ascending order [2.2 only]
- Older version of NSWP - include for interest to see how a file can progress and improve over time [3.0 and 2.2]
- A buffered file transfer utility that's really quick! [3.0 and 2.2]

WE already know how skilled and creative Amstrad users are, and we look forward to receiving your programs and articles for publication in future issues of *Computing with the Amstrad*. However before you send your masterpiece off to us there are one or two points that you ought to bear in mind to make all our lives easier. We call them the eighteen commandments...

The 18 Commandments

A guide to presenting your masterpiece for publication

WHILE not wanting to put programmers' creativity into a straightjacket we've found that life can be made a lot easier for the magazine, our readers and the programmers themselves if we stick to certain standards.

It has also occurred to us that it's no good our just knowing what we want, we have to tell you, our potential contributors. So here are our 18 commandments. Don't be too daunted by the list - it's mostly just commonsense and good programming practice.

- Send us your programs on tape. There's no point in just sending a listing and asking if we're interested. You can't expect us to evaluate a program from merely reading a listing. We may be good, but we're not that good! A cassette with the program on is a must.

We don't use two part programs in the magazine. Games in two files may look professional but they're the kiss of death as far as the magazine is concerned. Too much can go wrong when people type them in.

- Avoid variable names that lead to confusion such as *1* and *l*, *O* and *o* and try to use lowercase variable names as it makes life easier for the reader who's trying to type it in and debug his errors. Meaningful variable names help as well - *aliens* is far more understandable than *al*.

- Tell us what the program is supposed to do and refer to it by name. You'd be amazed at the number of programs we get where the author forgets to tell us what it is all about.

In any subsequent correspondence, reference to "my program" can cause problems by its vagueness. Okay, we'd have the program on record somewhere, but life would be a lot easier all round if its author were

less modest and admitted he was the genius behind "Mega-invaders".

- Label everything with both the program's name and your own name and address. Keep your own copy of it, too. So far the only existing copy of one particular classic game hasn't disappeared in the post - but there's no reason to run the risk of yours being the first.

If it's a game let us know how to "cheat" so we can test out the higher levels. We're getting on a bit here and our reactions aren't as good as they used to be. (Not that they were up to much when they were as good as they used to be...)

And an adventure-type game or whatnot should come with a map of the rooms and any other crib sheet you possess. Much as we'd like to, we just don't have time to guess the name of Rumpelstiltskin's brother, no matter how much we admire your ingenuity. (Anyway he works in our artroom.)

- Put more than one copy of the program on your tape, recorded at different speeds and on different sides. And if you want the cassette back let us have a stamped addressed envelope with the name of the program on it.

You won't appreciate this unless you've run a computer magazine, but please send each different program on a different cassette. If not, we just can't handle them. The rule is, one program per cassette - though recorded several times on it.

- Let us have a printed listing if possible. Screen dumps or off-screen photos are much appreciated, though

not vital. Diagrams are always of use. Often a point that's difficult to put into words becomes clear as crystal when you sketch it out.

- Give a description of the program, what it does, why you wrote it, and outline the way it works and its variables and subroutines.

If it's a game let us have a plot. You'll get an idea of the sort of thing we want by reading the introductions to one or two of our games.

Maybe you could also give a few ideas for its improvement or expansion. Even if you can't get your upgrades to work there's a good chance that someone among our very talented readers will.

Every subroutine must be titled clearly with a REM and should be referred to by it. Again, make the title meaningful. Also when you GOSUB use a REM to indicate which subroutine you're using. For example:

100 GOSUB 1000: REM Move man

1000 REM ** Move Man ******

1100 RETURN

At first this may seem to be far too much fuss, but it's not just for the readers' benefit. As your programs grow you'll find that such REMs more than repay the effort by allowing you to keep track of your work.

When you write out your list of subroutines (vital) try to do it in the form:

100 example Shows how we want...

200 delay Holds things up...

where the line numbers refer to the lines where the subroutine is defined. Again, this helps by making things

Contributions should be sent to: Features Editor, Computing with the Amstrad, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

clearer to our readers – and you!

We don't expect your program descriptions to be classics of English literature, but it does help if they make sense and are easy to follow. Try reading them out loud – you'd be amazed how much such a simple technique can improve your writing.

Also if you get stuck trying to put something into words use this trick: tell someone what it is you're trying to put into words – then write it down. Before you reject this hint, try it – more than one professional writer owes his career to it.

It is good practice to renumber your program, starting at 10 in increments of 10 – the standard default. This way a missing line stands out like a sore thumb.

● Make sure that the program actually works. Try it out on your friends for their criticism (painful though it may be). The acid test is to ask them to type it in. And – when you find yourself muttering through clenched teeth, "How could anyone be that stupid?" (the answer is "regularly") – cast out the mote in your own eye and alter your program to take account of the feedback.

It's not easy to do, as the all-too-frequent blood feuds among the editorial staff here testify, but it's worth it.

Instructions can make or break a game. Make sure that yours really do instruct. They should be complete and it helps if the spelling and grammar are correct. Apart from causing confusion, such errors also make programs look amateurish.

As well as misspellings, bad grammar, split words and general untidiness are all to be avoided.

Following even the simplest program can cause problems for the most experienced programmer – don't add to them unnecessarily. One major cause of having to return programs for modifications is the colour/monochrome dilemma.

You can develop a beautiful program making use of all the splendid colour the Amstrad is capable of, only to find that the action disappears in an impenetrable fog on a monochrome monitor – and vice-versa.

If possible try your program on both versions of the monitor. User groups are invaluable here as they are in all aspects of program develop-

ment.

● Please do put lots of nice explanatory REMs in your programs. A couple of REM statements with nothing after them at the beginning of the program gives us room to put in our messages without messing up all the line numbers you have referred to in your program description.

● Avoid having just a line number with a colon and nothing else. It may make the program look neater but we won't welcome letters asking what the missing line was. Remember, people will be spending hours typing

If you follow these rules when you submit a program you'll stand a better chance of having it published

your programs into their micros. Make their life easier if you can.

● Double space all your written matter. This means leaving a blank line between each line of text – it's vital from our point of view. Try to follow our style. We have our own ways of doing things. We talk about modes in general but Mode 1 in particular. We press the Enter key, not the ENTER key as you might expect.

Just look how we do it in the magazine. Our programs are Program I, Program II, and so on; our diagrams Figure I, Figure II.

● If you must use long multiple lines don't go over about 175 characters, as people always add spaces when they type them in, then complain the lines are too long.

● Always put in the right number of NEXTs – don't just use NEXT followed by a comma, as it causes a lot of confusion.

Similarly with REM. You can abbreviate it with an apostrophe, but it's exceptionally easy to miss – for

the sake of typing two extra characters you can save our readers hours of frustration.

● Please, when you send us your work, include a separate page telling us that it is your own work, it has been offered elsewhere and we have your permission to print it. If you don't, we'll have to return it.

● It's always nice if a program can have an alternative key or joystick option.

● One of the major causes of programs crashing is because user inputs something the programmer wasn't expecting. All right, idiot shouldn't type in -999 when you ask him his age, but believe me they will, out of sheer perversity particularly if the program is educational. There is something about CAL programs that brings out the devil in us all...

So try out all the unlikely options if you don't, some poor user will.

Actually it takes a lot of skill to do proof a program, as it's delicate known in the trade.

Often you're so involved in getting the program to work as it's supposed to that you just can't make the mental leap needed to see it as the passive malevolent reader does. So try it on your friends!

● Another irritation for a reader when he sees something like:

PRINT"

Exactly how many blanks is supposed to enter?

Use:

STRING\$(n,"")

for *n* spaces.

● Tell us who you are. We like to know your Christian name and it's interesting to know your age and profession. After all, we might reject your program, but if we knew you were a fellow fettler we'd have been able to send you Obscure's "Fettling fetlocks on the Amstrad CPC464" for review.

Also a telephone number – both home and work – with the correct STD code is really useful, and save a lot of time.

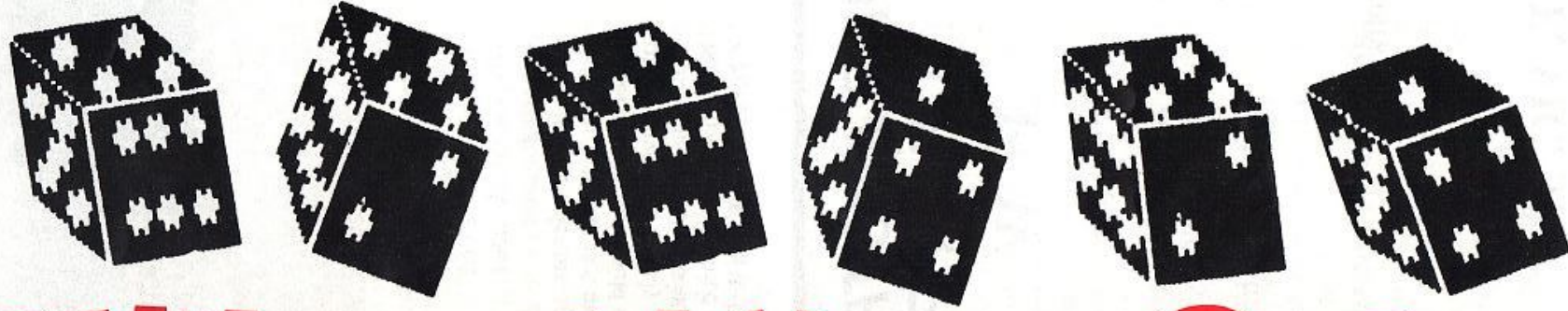
Thus endeth the 18 rules. If you follow these when you submit something to us you'll stand a much better chance of having it published. More importantly, you'll become a more professional programmer.

And the better you become the more satisfying it is.

RANDOM

By
Aleatoire

REFLECTIONS



THE chance of throwing any number with a die is one in six. What is the chance of throwing a six in three throws? If you think it's 50 per cent then you are a sucker and long may the Ace of Diamonds squirt cider in your ear.

Way back in 16th century Italy gambling with cards, dice and even chess was a major pastime and the more assiduous players knew from experience that you needed four throws to have a roughly even chance of throwing a six.

It was not, however, until about 100 years later, around 1654, that such probabilities were studied and correctly explained mathematically.

Probably the most famous problem was posed by Monsieur de Mere to the French mathematician Blaise Pascal.

M. de Mere knew and accepted that the odds of failing to throw a six are 5/6 and therefore the chance of failing in four throws is:

$$5 \times 5 \times 5 \times 5 / 6 \times 6 \times 6 \times 6 = 625 / 1296$$

Consequently the chance of success is 671/1296 as observed in practice, slightly better than evens.

From this de Mere argued that the chance of throwing 12 with two dice (the "Sonnez" from "Sonnez les cloches, le Diable est mort" whenever 12 was thrown in Backgammon) should also be better than evens for 24 throws (since there are 36 possible throws and $36 \times 4/6 = 24$).

However de Mere knew from experience that the chance was less than 50 per cent and told Pascal that this revealed a flaw in the mathematical approach. So here is your chance to try the problem and also convince the wife that you didn't just buy the Amstrad to play games.

All you have to do is use the RND function to generate two dice throws and then calculate the number of times the machine manages to throw Sonnez in 24 (or less) throws divided by the total number of trials $\times 100$ per cent.

The following program does this

experiment 100 times in just under 30 seconds and prints the percentage of success:

```
10 RANDOMIZE TIME
20 sonnez=0
30 FOR i=1 TO 100
40 FOR t=1 TO 24
50 d1=INT(RND*6+1)
60 d2=INT(RND*6+1)
70 IF d1+d2=12 THEN sonnez=sonnez+1
80 NEXT t
90 NEXT i
100 PRINT sonnez;"%"
```

If you try it you will find that 100 trials are not particularly conclusive and the mind boggles at how much dice throwing was required to detect the slight difference – certainly many thousands.

Pascal explained that the mathematical proof again depends on calculating the chance of FAILURE to throw the Sonnez in 24 throws, which is $(35/36)$ to the power 24 or:

```
10 c=1
20 FOR i=1 TO 24
30 c=c*35/36
40 NEXT i
50 PRINT c*100;"%"
```

This established the theory of random numbers.

A more modern, though equally famous problem, is that of a submarine captain who is on patrol with only one torpedo left. He expects to separately encounter 10 enemy ships of random size and wants to sink the largest.

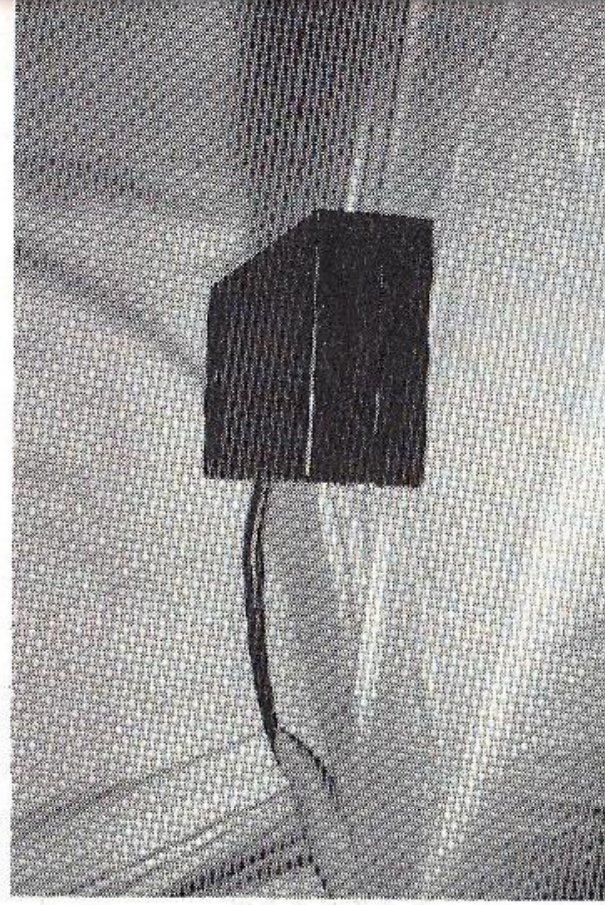
This is equivalent to choosing the largest of an unknown set of 10 random numbers when that number is called out and the correct strategy succeeds about once every three trials.

How does the captain choose?

I'll give the answer next month.

Safe and Simple

Giltronics have become known for their disk drive add-ons for Amstrad computers. They have recently released a whole series of disk drives for both CPC and PCW series. Pictured is the Single Sided, Single Density 169k drive. Also available are the 40 track, Double sided 360k drives and 80 track, double sided 700k drives. The latter two are provided with software mods to CP/M to allow use of the extra storage capacity. The single sided, single density drive has the same storage capacity as the Amstrad's drive but of course you'll save a fortune on disk prices. Unfortunately, as supplied, 5 1/4 disks have only one notch thereby allowing use of only one side of each disk. However, if you ask Giltronics nicely when you purchase your drive I'm sure they'd provide you with a 'notcher' to allow use of both sides of the disk.



We've been using the 40 track single sided drive for about two months now. It hasn't missed a beat and is so quiet in operation you'll wonder if it's turned on. Connection to a 664 or 6128 is as simple as plugging in a lead whereas the 464 requires an adaptor cable. The whole system is so simple to use it's hard to imagine anyone having any trouble. The drives are well constructed and very solid. Color is matt black matching the computer.

Prices are:

40 track, single sided \$329
40 track, double sided \$399
80 track, double sided \$499
464 adaptor cable \$ 9

prices include sales tax, software and delivery. Warranty is * months.

In conclusion, you could save the cost of the drive very quickly. The low unit being priced at about the difference between 4 boxes of three inch disks 4 boxes of 5 1/4 inch disks.

ADD-ON DRIVES FOR YOUR AMSTRAD CPC & PCW

Giltronics

AUSTRALIA

AMSTRAD

CPC

464, 664, 6128

40 TRACK SS \$329
40 TRACK DS \$399
80 TRACK DS \$499
464 CABLE \$ 9

PCW

8256,8512

40 TRACK SS \$349
80 TRACK DS \$535

Installation extra for PCW

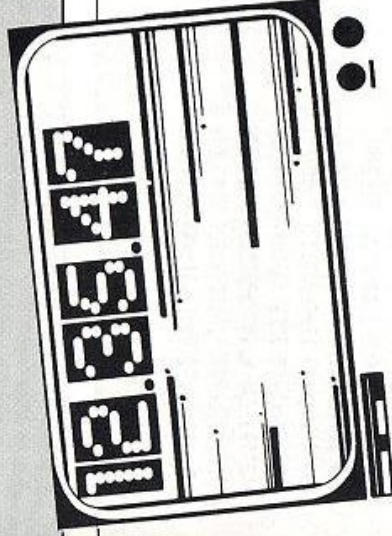
MAIL ORDER

SOFTWARE & HARDWARE

Prices include freight, software and warranty.

(03) 580 9839

54 EULINGA AVENUE, ASPENDALE, VIC. 3195



Clock on With your Amstrad

By **SIMON HOLDERNESS**

HAVE you ever been so engrossed in your programming that you've lost all track of time? I bet every programmer's had the annoying experience of missing an appointment in this way. Help's at hand though, with this machine code real time clock for the CPC series.

Most people know of the Amstrad's internal clock system, constantly updating itself every 1/300th of a second. It is very easy to access it from Basic by typing:

PRINT TIME

This will just give you the number of 1/300ths of a second which have passed since the computer was last reset. Although simple to use it wouldn't be very useful as a real time clock.

A better method would be to use Basic's EVERY routine to update our own clock every second. This would take the form:

```
10 EVERY 50,1 GOSUB 1000
```

where the clock routine is at line 1000 onwards. The 50 tells the computer to go to the routine every second and the 1 tells it to use timer 1. This works fine if you stay within a program which is being executed, but when you break out the clock stops updating itself.

The solution is to use a machine code routine which is called every second whatever the computer is doing at the time, be it running a program or waiting for a command to be input directly.

This program will display the clock in the top left hand corner of window #7 in the form:

HOURS:MINUTES:SECONDS

Once the code is poked into memory via Program I, or Program II if you have an assembler, the clock can be started with:

CALL &8000

This initialises a ticker event and

sets it calling the clock routine every second. Once this has been set going it can be stopped with:

CALL &8017

which takes the event off the ticker list.

Once the ticker event has been initialised you will see, after a second, the time 12:00:01. This will continue to update itself every second whatever you do.

This isn't a lot of good as it is, unless you started the clock at 12 o'clock, as it will not be displaying the correct time. However any clock worth its salt can be adjusted, and to change this one you must type:

```
POKE &80AA, hours (1-12)
POKE &80AB, minutes (0-59)
POKE &80AC, seconds (0-59)
```

The clock should now be displaying the correct time.

The routine works by loading the E register with the hour, obtained from location &80AA, L with the minutes stored at &80AB and H with the seconds from &80AC.

H, the seconds, is incremented first and tested to see if it's 60 and if it is then L, the minutes, is also incremented and H set to zero. If L is now 60 the hours in E are also incremented and the minutes in L reset to zero. Finally if the hours reach 13, E is set to 1 and the time can be printed.

The display routine first stores the current position of the text cursor, switches to window #7 and moves the cursor to the top left corner. The hours, minutes and seconds in the E,

L and H registers are printed and the cursor replaced.

The machine code routine is called every second so there is a slight slowing down of the Amstrad when it is doing other tasks. The clock itself does not lose much time, except when the computer is doing certain tasks like accessing the disc drive or more noticeably the tape recorder, since the Amstrad switches interrupts off. It is therefore necessary to set the clock again if it does lose a lot of time.

Problems can also occur if you are not working in the default stream #0, as the routine switches to this stream after it has displayed the time.

The machine code can be saved on its own by typing:

```
SAVE "CLOCK.BIN",B,&8000,&80C
```

Typing:

```
MEMORY &7FFF
LOAD "CLOCK.BIN"
CALL &8000
```

will load the code and call it. You could also have it change the clock to the correct time and set up window #7 so that the top left of the window is where you want the clock to be. For instance:

```
WINDOW #7,16,25,11,11
```

would place the clock in the centre of the screen in Mode 1. Window #0 could similarly be defined so that is out of the way of the clock.

Now there will be no excuse for missing meals or your favourite TV program because you didn't realise what the time was—because it will be there in front of your eyes!

Computing with the Austral - November 19

Chipstow Stakes

By

CHRISTOPHER BERRY

YOU are cordially invited to spend a day at the Chipstow Races. This luxuriant green racecourse is ideally situated at eye-level on any Amstrad PCW.

It is an all-weather course and the going is good. Although there's a fence and sometimes

horses fall we never have any injuries even though you'll see the occasional prostrate horse and unseated rider.

To add to the excitement we operate a Tote, but in order to prevent heavy gambling from spoiling the fun the amount you can bet is limited. If you win £100

or more you will be invited to the VIP stand to buy champagne.

The program contains an **OPTION RUN** command at line 90 which disables the effects of **Control+C** and **Control+S**. Do not enter this line until you have checked you have entered the program correctly.

PROGRAM STRUCTURE

100-150 Set up the screen display.
160-420 Set of five races.
230-280 Create the strings that make up the horses and jockeys.
430-810 Move the horses across the screen.
820-960 Draw the field.
970-1000 Place the horses on the starting line.
1010-1140 Start the race.
1150-1300 Rules.
1310-1360 Generate a random seed.
1380-1480 Input the player's name.
1490-1720 Display the horse names and the bets.
1730-1780 Names of the horses.
1790-2090 Calculate and display the results.
2110-2210 Generate the next random seed.

VARIABLES

punters%	The number of players.
punter\$(i)	The name of the player.
p	Identifies the player.
h	Identifies the horse number.
race%	The number of the race.
winnings%	The winnings of player p in race number
(p,race%)	race%
n	The random seed.

Full listing starts
on Page 62




```

10 REM *****
20 REM *****
30 REM * CHIPSTON STAKES *
40 REM * by Christopher "CB" Berry *
50 REM *
60 REM !Computing with the Amstrad
70 REM *
80 REM *****
90 OPTION RUN
100 REM * controlling the screen *
110 REM * type pos not POS
120 escape=CHR$(27)
130 home$=escape+"H"
140 cls$=escape+"C"+home$
150 DEF FNpos(x,y,text$)=escape+"Y"+CHR$(32-y)+CHR$(32+x)+text$
160 PRINT cls$;
170 race$="1
180 GOSUB 1150
190 GOSUB 1300
200 GOSUB 1450
210 REM * the parts of the horses and jockeys *
220 REM
230 t$=CHR$(45)+CHR$(61)+CHR$(30)+CHR$(47)+CHR$(126)
240 t2$=CHR$(45)+CHR$(61)+CHR$(30)+CHR$(45)+CHR$(223)
250 b$=CHR$(47)+CHR$(32)+CHR$(32)+CHR$(92)+CHR$(32)
260 b2$=CHR$(32)+CHR$(92)+CHR$(47)+CHR$(32)+CHR$(32)
270 o$=STRING$(5," ") REM * 1 space between inverted commas *
280 fallens$=CHR$(32)+CHR$(35)+CHR$(45)+CHR$(61)+CHR$(61)+CHR$(223)+CHR$(32)+CHR$(35)
290 GOSUB 820
300 GOSUB 430
310 GOSUB 1750
320 PRINT cls$;
330 PRINT "** MEETING OVER - ENTER 'R' FOR A NEW RACE MEETING - 'END' TO RETURN TO BASIC **"
340 PRINT:PRINT
350 FOR p=1 TO punter$
360 IF totalwinning$(p)>99 THEN PRINT "Congratulations ";punter$(p);
you are invited to buy champagne in the VIP stand! **:PRINT
370 NEXT
380 PRINT:PRINT
390 INPUT " RUN or END )":j$
400 IF q$="RUN" OR q$="run" THEN RUN ELSE 410
410 IF q$="END" OR q$="end" THEN NEW ELSE 390

```



```

422 END
430 REM * move the horses across the screen *
440 REM * the race *
450 x(1)=1:x(2)=1:x(3)=1:x(4)=1
460 y(1)=8:y(2)=11:y(3)=14:y(4)=17
470 REM * h is the horse number *
480 FOR h=1 TO 4
490 IF x(h)=39 THEN PRINT FNpos$(39,y(h),CHR$(32)+CHR$(35)+o$)
500 IF x(h)=39 THEN PRINT FNpos$(39,y(h)+1,fallens$:60TO 530
510 PRINT FNpos$(x(h),y(h),t$)
520 PRINT FNpos$(x(h),y(h)+1,o$)
530 NEXT
540 FOR h=1 TO 4
550 IF x(h)=39 THEN 600
560 PRINT FNpos$(x(h),y(h),o$)
570 PRINT FNpos$(x(h),y(h)+1,o$)
580 IF x(h)>6 THEN PRINT FNpos$(6,y(h),CHR$(39)):PRINT FNpos$(6,y(h)+1,CHR$(39))
590 IF x(h)>39 THEN PRINT FNpos$(40,y(h),CHR$(35)):PRINT FNpos$(40,y(h)+1,CHR$(35))
600 NEXT
610 FOR h=1 TO 4
620 IF x(h)=39 THEN PRINT FNpos$(39,y(h)," C"+o$):60TO 670
630 IF x(h)=39 THEN PRINT FNpos$(39,y(h)+1,fallens$:60TO 670
640 move$=RND*5:x(h)=x(h)+move$
650 PRINT FNpos$(x(h),y(h),t2$)
660 PRINT FNpos$(x(h),y(h)+1,b2$)
670 IF x(h)=39 THEN PRINT FNpos$(34,h,"Number");h;"fell"
680 NEXT
690 PRINT home$;
700 IF x(1)=39 AND x(2)=39 THEN 60TO 710 ELSE 720
710 IF x(3)=39 AND x(4)=39 THEN winner$=PRINT FNpos$(55,12,"** WAIT! RACE ABANDONED **"):FOR pause=1 TO 3000:PRINT cls$:RETURN
720 FOR h=1 TO 4
730 IF x(h)>82 THEN winner=h:RETURN
740 NEXT
750 FOR h=1 TO 4
760 IF x(h)=39 THEN 790
770 PRINT FNpos$(x(h),y(h),o$)
780 PRINT FNpos$(x(h),y(h)+1,o$)
790 NEXT
800 PRINT home$;
810 GOTO 480
820 REM * f LOOP draws the race field *
830 REM
840 PRINT cls$;
850 FOR f=1 TO 24
860 PRINT FNpos$(40,f,CHR$(35))
870 PRINT FNpos$(6,f,CHR$(39))
880 PRINT FNpos$(88,f,CHR$(33))
890 PRINT FNpos$(38,25,"FENCE")
900 PRINT FNpos$(0,25,"START")
910 PRINT FNpos$(93,25,"FINISH")
920 PRINT FNpos$(0,8,"")
930 PRINT FNpos$(0,11,"2")
940 PRINT FNpos$(0,14,"3")
950 PRINT FNpos$(0,17,"4")
960 NEXT
970 FOR y=8 TO 17 STEP 3
980 PRINT FNpos$(1,y,t$)
990 PRINT FNpos$(1,y+1,b$)
1000 NEXT
1010 PRINT FNpos$(12,12,"They're under")
1020 PRINT FNpos$(12,13,"Starters are");
1030 PRINT home$
1040 FOR pause=1 TO 2000:NEXT
1050 PRINT FNpos$(12,12,STRING$(13,"")):REM * 1 space between inverted commas *
1060 PRINT FNpos$(12,13,STRING$(16,"")):REM * 1 space between inverted commas *
1070 PRINT CHR$(7)
1080 PRINT FNpos$(12,12," And-")
1090 PRINT FNpos$(12,13,"They're Off")
1100 PRINT home$
1110 FOR pause=1 TO 500:NEXT
1120 PRINT FNpos$(12,12,STRING$(8,"")):REM * 1 space between inverted commas *
1130 PRINT FNpos$(12,13,STRING$(12,"")):REM * 1 space between inverted commas *
1140 RETURN
1150 REM * the rules *

```



```

1160 PRINT cls$;
1170 PRINT FNPOS$(17,5,"THE CHIPSTOW
RACES" - by Courtesy of C.A. Berry")
1180 PRINT FNPOS$(17,6,"-----"
-----"
)

```

```

1190 PRINT:PRINT
1200 PRINT "There are 5 races.":PRINT
1210 PRINT "In each race there are fo
ur horses, all of equal form.":PRINT
1220 PRINT "The TOTE starting price,
therefore, has been set at 3 to 1 for
each horse.":PRINT
1230 PRINT "Up to 5 punters may parti
cipate and can back ANY horse.":PRINT
1240 PRINT "TOTE tickets are £1 each
and you can have up to 10 tickets for
each race.":PRINT
1250 PRINT "Each punter starts with o
nly £25 spending money.":PRINT

```

CPC Disk drive owners - don't miss the December issue of Computing With The Amstrad - we present a utility which could save you literally hundreds of dollars! Reserve a copy now!

```

1260 PRINT "If you win any money of co
urse, you can reinvest it.":PRINT
1270 PRINT:PRINT
1280 PRINT "-----"
-----"

```

```

1290 PRINT "- PRESS ANY KEY TO CONTIN
UE -"

```

```

1300 PRINT "-----"
-----"

```

```

1310 REM * generate the random seed *
1320 n=1
1330 WHILE INKEY$=""
1340 n=n+1:GOTO 1330
1350 WEND
1360 RANDOMIZE n
1370 RETURN
1380 REM * enter the players *
1390 PRINT cls$;
1400 INPUT "Enter how many punters - (
Maximum of 5 )";punters$
1410 IF punters$>5 OR punters$<1 THEN
PRINT CHR$(7); * INPUT ERROR *:GOTO
1400

```

```

1420 PRINT:PRINT
1430 FOR p=1 TO punters$
1440 PRINT "Enter name of punter numbe
r";p;
1450 INPUT punters$(p)
1460 PRINT
1470 NEXT
1480 RETURN
1490 REM * the race-card and placing

```

```

the bets *
1500 PRINT cls$;
1510 PRINT "Race Number";race$
1520 PRINT "-----"
1530 PRINT
1540 PRINT "Horses for this race are:
":PRINT

```

```

1550 FOR horsename$=1 TO 4
1560 READ horsename$
1570 horsename$(horsename$)=horsename
$
1580 PRINT horsename$; "- ";horsename$
1590 NEXT
1600 FOR p=1 TO punters$
1610 PRINT
1620 PRINT punters$(p); ", which horse
do you want? ( 1/2/3/4 )";
1630 INPUT horses$(p)
1640 horse$(p)=VAL(horses$(p))
1650 IF horse$(p)=1 OR horse$(p)=2
THEN 1680

```

```

1660 IF horse$(p)=3 OR horse$(p)=4
THEN 1680
1670 PRINT CHR$(7); * INPUT ERROR *:
GOTO 1620
1680 INPUT "- and how much do you bet
( 1 - 10 )... £";stake$(p)
1690 IF stake$(p)>10 THEN PRINT " * So
rry Tote maximum is £10 . Bet again!
*":GOTO 1680

```

```

1700 IF stake$(p)+totalstake$(p)-total
winings$(p)>25 THEN PRINT " * Sorry,
you only have * £";25-totalstake$(p)+
totalwinings$(p):GOTO 1680
1710 NEXT

```

```

1720 RETURN

```

```

1730 REM * the names of the horses *
1740 DATA Gerontius,Balthazar,Icarus,
Caesar

```

```

1750 DATA Jazz,Dodo,Creamier,Rialto

```

```

1760 DATA Sputnik,Apollo,Pioneer,Mora
tio

```

```

1770 DATA Blazer,Blouson,Cous-Cous,Si
lvo

```

```

1780 DATA Gollightly,Caspar,Leonid,Syd
ney

```

```

1790 REM * the results *

```

```

1800 PRINT cls$;

```

```

1810 PRINT:PRINT " * Winning horse in
race";race$;"was -";winner;horsename$
(winner)

```

```

1820 PRINT

```

```

1830 FOR p=1 TO punters$

```

```

1840 IF horse$(p)=winner THEN PRINT p
unters$(p); " wins £";stake$(p)*3+stake
$(p); " ";

```

```

1850 NEXT

```

```

1860 PRINT

```

```

1870 FOR p=1 TO punters$

```

```

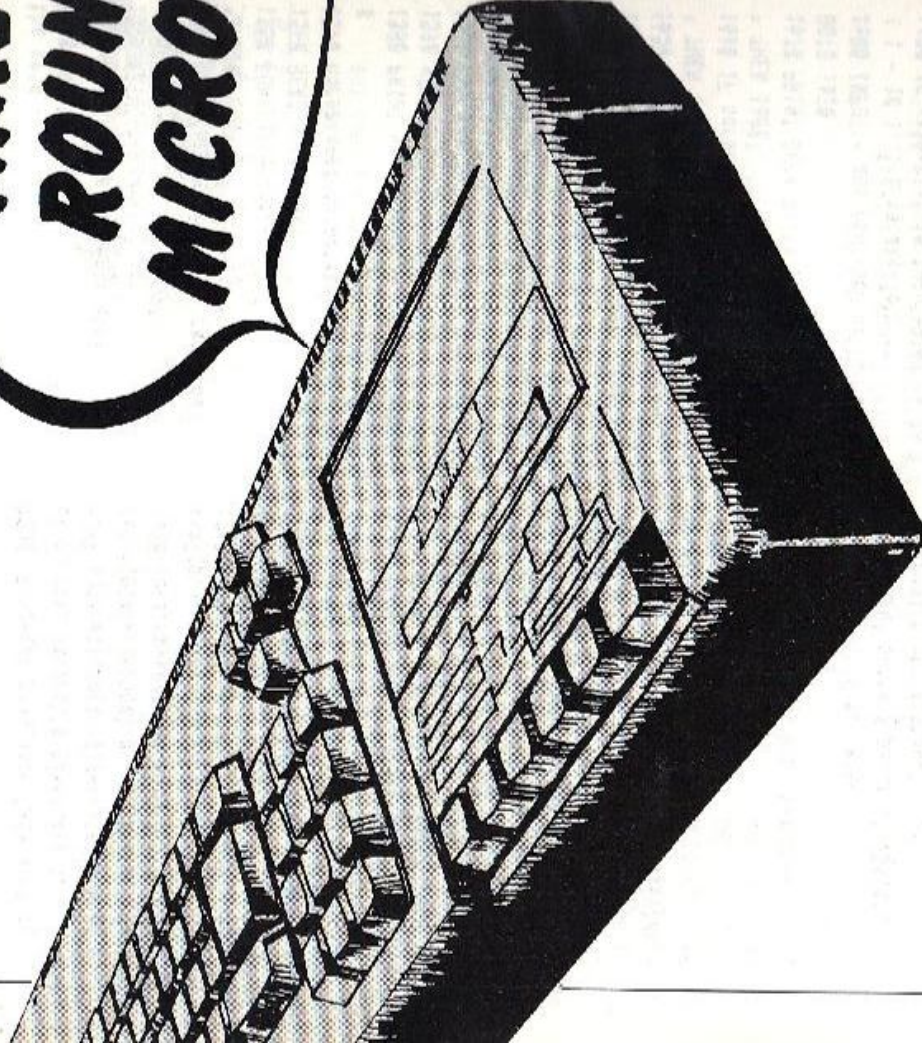
1880 IF horse$(p)=winner THEN winning
s$(p,race$)=stake$(p)*3+stake$(p)
1890 totalwinings$(p)=winings$(p,rac
e$)+totalwinings$(p)
1900 totalstake$(p)=stake$(p)+totalst
ake$(p)
1910 NEXT
1920 PRINT
1930 PRINT " * Total winnings to date:
-"
1940 PRINT
1950 FOR p=1 TO punters$
1960 PRINT punter$(p); " £";totalwinn
ings$(p)
1970 NEXT
1980 PRINT
1990 PRINT " * Total stakes to date: -"
2000 PRINT
2010 FOR p=1 TO punters$
2020 PRINT punter$(p); " £";totalstak
e$(p)
2030 NEXT
2040 PRINT
2050 PRINT " * What you have now: -"
2060 PRINT
2070 FOR p=1 TO punters$
2080 PRINT punter$(p); " £";25+totalw
innings$(p)-totalstake$(p)
2090 NEXT
2100 PRINT FNPOS$(30,28,"** PRESS [
C ] TO CONTINUE **");
2110 REM * generate the random seed *
2120 n=1
2130 answer$=INKEY$
2140 WHILE answer$=""
2150 n=n+1:GOTO 2130
2160 answer$=INKEY$
2170 WEND
2180 match$="Cc"
2190 answer=INSTR(match$,answer$)
2200 IF answer=0 THEN PRINT CHR$(7);G
OTO 2100
2210 RANDOMIZE n
2220 race$=race$+1
2230 IF race$<6 THEN 200
2240 RETURN

```

Because this program is in Mallard Basic and that language usually only exists on PCW machines, we have not included Chipstow Stakes on this months cassette. The program is however on our first Public Domain disk (see P. 72) and will also be on our second quarterly disk which will be available in January. Chipstow stakes will NOT run on CPC machines without considerable alteration.

TAKE A TOUR ROUND YOUR MICRO'S SOUND

First in an
informative series
by NIGEL PETERS



MOST micros nowadays have the ability to make sounds. The simplest way to get your CPC464 to make a noise is to enter:

PRINT CHR\$(7)

and press Enter. The trouble is that this solitary beep isn't all that exciting.

To enable you to create rather more interesting sounds the Amstrad has three very sophisticated commands - SOUND, ENT and ENV. However, the more sophisticated the command the more difficult it is to use, at least at first.

So over the next few months I will be covering the commands in detail, building up a step-by-step guide to making noises on the Amstrad CPC464. It will be a practical course with lots of examples, so make sure that your micro is switched on and ready for use while you read the articles.

And don't take everything I say on trust - try it out. The SOUND command is a big subject and only by trying things out for yourself will you grasp it in its entirety.

Having said all that, let's make a sound. Enter:

SOUND 1,200,100,7

and feel yourself thrill to the exciting

tones of your Amstrad.

Well, maybe I am overdoing it a bit. It's not the most exciting sound in the world. Still, once you understand how that SOUND command works you'll have gone a long way to conquering Amstrad sound.

The simplest form of the SOUND command is followed by four parameters. Don't be put off by parameters, they are just the numbers that follow the SOUND command and affect the way it works.

The parameters in the last sound we made are 1, 200, 100 and 7. The first number (1) selects the channel that the note will be played on, the second (200) tells the micro what pitch it will be. The third parameter (100) determines how long that note will last while the last figure (7) decides on the volume it will be played at.

By entering:

SOUND 1,200,100,7

we produce a note on channel 1 which will last for one second, be played at full volume and be fairly high pitched.

Don't worry if you don't fully understand all these terms, we'll be dealing with them later in this article.

The basic structure of the SOUND

command is:

SOUND channel,pitch,duration,volume
that is, the Basic keyword SOUND followed by four parameters which specify the particular sound wanted.

Have a go at a few more sounds entering:

SOUND 2,600,20,4

or:

SOUND 4,20,50,3

or be adventurous and make up your own.

As you'll see (and hear) from the above, the values following the SOUND command can vary and they vary so do the notes produced. However you can't just pick any parameter, you can only choose between certain limits.

Take the channel parameter, the first number that comes after the SOUND command. This is used to decide which of the CPC464's sound channels is to be used. The Amstrad has three such channels, and each can only play one note at a time.

However by having all three channels working at the same time you can have the micro producing three different notes simultaneously. The channels are known as channels A, B and C and the parameters that

select them are 1, 2 and 4 respectively.

Program I shows these parameters in action. First of all it plays a note on channel A (parameter 1), then waits for you to press a key. The next note is played on channel B as the parameter following the SOUND command is the figure 2.

```
10 REM PROGRAM I
20 REM Channel A
30 SOUND 1,478,100,7
40 WHILE INKEY$="" :WEND
50 REM Channel B
60 SOUND 2,379,100,7
70 WHILE INKEY$="" :WEND
80 REM Channel C
90 SOUND 4,301,100,7
```

Program I

When this note is over and you press another key, the last note, with channel parameter 4, plays on channel C.

Of course these notes were played one after another. I hope the more sceptical of you will be thinking: "How do I know there are three channels? There could be just one channel with the notes playing on it, one after another".

If you do have such doubts, have a go at Program II which plays the same notes at the same time, on different channels. Not exactly Mozart is it? Still, it should have convinced you about the three channels.

Now let's move on from the channel parameter — we'll be back to it in a later article — and look at the pitch parameter that follows it. As you might guess, this controls the pitch of the note produced by the SOUND command.

The pitch of a note is how high or low it is. The higher the note — like a soprano or a molested cat — the higher the pitch is. The lower the note

```
10 REM PROGRAM II
20 REM Channel A
30 SOUND 1,478,100,7
50 REM Channel B
60 SOUND 2,379,100,7
80 REM Channel C
90 SOUND 4,301,100,7
```

Program II

— like a bass or a foghorn — the lower the pitch.

Run Program III and you'll hear a note that changes pitch, going lower and lower. Press ESC a couple of times when you've had enough.

What's happened is that each time round the FOR ... NEXT loop the SOUND command of line 30 makes a note. However each time round the loop the value of the pitch parameter, *pitch*, varies. At first it's 32, then it's 33, then 34 and so on. As the pitch parameter varies so does the note produced.

```
10 REM PROGRAM III
20 FOR pitch=32 TO 60
30 SOUND 1,pitch,100,7
40 NEXT pitch
```

Program III

You might have noticed that as the pitch parameter gets larger in value, the note gets lower in pitch (the sound gets deeper). This means that a pitch parameter of 100 produces a much lower note than a pitch parameter of 30.

Incidentally if you did use ESC to get out of the previous program you might have left a few notes still in the micro. These may cause the following programs to sound odd (or, rather, odder) so get rid of them by entering:

SOUND 129,100,0,0

I know that the channel parameter looks wrong, but all will be explained in a later article. Just use it for garbage collection for the time being.

To return to the pitch parameter, Program IV uses a step of -4 to decrease the pitch parameter each time round the FOR ... NEXT loop. The result is a note that increases in pitch.

The pitch parameter can take values that range from 0 to 4095 though it does get a bit ragged at the extremes of its range. The numbers have to be integers (whole numbers). If you use a number with a decimal part for the pitch parameter the CPC just ignores the decimal.

Don't just believe me, try it for yourself. Is there a difference between:

SOUND 1,200.5,100,7

and

SOUND 1,200,100,7

that you can detect?

For those with musical pretensions

```
10 REM PROGRAM IV
20 FOR pitch= 478 TO 253 STEP -4
30 SOUND 1,pitch,100,7
40 NEXT pitch
```

Program IV

Appendix VII of the User Instructions gives the full range of pitch parameters with their corresponding musical notes. They call the pitch parameter the tone period, but they're the same thing.

Table I shows 24 of these notes and their pitch parameter values.

Program V plays a selection of these notes in order, reading the values of *pitch* from the DATA statements of line 60.

```
10 REM PROGRAM V
20 FOR note=1 TO 12
30 READ pitch
40 SOUND 1,pitch,100,7
50 NEXT note
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

Program V

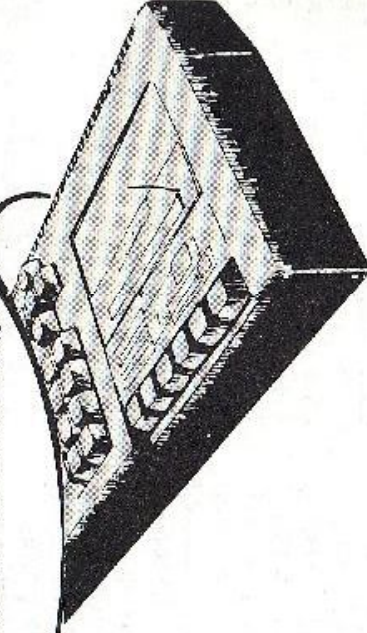
You might notice that this series of notes has a more "complete" feel about it than the previous ones we've had. We'll come back to this in a later article.

For the moment let's press on to the next parameter which determines

Pitch		Pitch	
NOTE	parameter	NOTE	parameter
C	239	F#	84
C#	225	G	80
D	213	G#	75
D#	201	A	71
E	190	A#	67
F	179	B	63
F#	169	C	60
G	159	C#	56
G#	150	D	53
A	142	D#	50
A#	134	E	47
B	127	F	45
C	119	F#	42
C#	113	G	40
D	106	G#	38
D#	100	A	36
E	95	A#	34
F	89	B	32

Table I

TAKE A TOUR ROUND YOUR MICRO'S SOUND



how long the note is going to last.

This third parameter, controlling duration, can have values from -32768 to 32767 but for the present we'll just use the values 1 to 32767. This number represents the number of 1/100ths of a second the note is to last.

So if the duration parameter is 100 the note should last for one second. If the duration parameter is 1000 it should last for 10 seconds. Enter:

```
SOUND 1,200,300,7
```

and you'll find the note lasts for three seconds.

Program VI, a variant of the previous one, uses the familiar FOR ... NEXT loop to vary the duration parameter via the variable *duration*.

```
10 REM PROGRAM VI
20 FOR duration=10 TO 120 STEP 10
30 READ pitch
40 SOUND 1,pitch,duration,7
50 NEXT duration
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

Program VI

As the notes change in pitch each time round the loop they also increase in length from 1/10th of a second to 1.2 seconds. This gives a sort of slowing down effect.

Program VII decreases the dur-

ation parameter each time round the loop, so giving the opposite effect.

The final parameter we're going to deal with is the volume parameter. This, as is obvious from the name, decides how loud the note is going to be and can have whole number

```
10 REM PROGRAM VII
20 FOR duration=120 TO 10 STEP -10
30 READ pitch
40 SOUND 1,pitch,duration,7
50 NEXT duration
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

Program VII

values from 0 to 7. If the parameter is 1, the note is at its quietest. The parameter 2 is a little louder and so on to maximum volume when the volume parameter is equal to 7. Program VIII shows them in action.

```
10 REM PROGRAM VIII
20 FOR volume=1 TO 7
30 SOUND 1,100,100,volume
40 SOUND 1,100,100,0
50 NEXT volume
60 DATA 119,113,106,100,95,89
70 DATA 84,80,75,71,67,63
```

Program VIII

The main part of the program is easy enough to follow. As the FOR ...

	Channel	Pitch	Duration	Volume
range	1 - A 2 - B 4 - C	0 to 4096	1 to 32767	0 to 7
default	none	none	20	4

Table II: Parameter ranges for SOUND command

NEXT loop cycles the volume parameter increases and the note gets louder. But what of line 40 which has a SOUND command with a volume parameter of 0? What's that doing there?

The answer is that it's there to provide an interval of no sound between the other notes. When the volume parameter has a value of 0 means that the sound has no loudness at all. You can't hear it.

However, the note still plays (albeit silently) for the full length of the duration parameter. This means that line 40 produces one second silences between the notes.

Although using a 0 volume parameter to produce a note that you can't hear seems a little strange at first sight, it is useful. You can use it to produce the rests that are often found in tunes. Also it makes things a lot clearer at times. Leave out line 40 in the last program and you'll see what I mean.

And that's the end of our tour of the four SOUND parameters, except for one thing. Can you explain why the notes produced by:

```
SOUND 1,200,20,4
```

and

```
SOUND 1,200
```

sound exactly the same?

The reason is that if we don't supply the SOUND command with a duration or volume parameter it provides them for itself. The default duration parameter is 20 and the default volume is 4, hence the two SOUND commands above produce the same note.

And that's it for this month. Table II sums up what we've covered so far while Program IX makes use of it to produce a familiar tune.

Why not write a few simple tunes yourself?

```
10 REM PROGRAM IX
20 SOUND 1,239,100,7
30 SOUND 1,213,100,7
40 SOUND 1,268,100,7
50 SOUND 1,536,100,7
60 SOUND 1,358,100,7
```

Program IX

● *Next month: More explanation of the sound capabilities of the Amstrad CPC464.*

It's amazing what you find in an envelope

LAST month we had a look at how to get the Amstrad to produce simple sounds using a very basic version of the SOUND command. If you understood what I said then you should be able to see why:

SOUND 1,200,100,5

produces a note of volume 5, on channel A with pitch 200, lasting for one second.

Make sure you follow this because we'll be using this particular note as the basis of our exploration of the Amstrad's volume envelope.

And what, you may ask, is a volume envelope? When we enter:

SOUND 1,200,100,5

we're rewarded with a fairly loud note. Notice that the loudness doesn't vary, it stays the same through the length of the note.

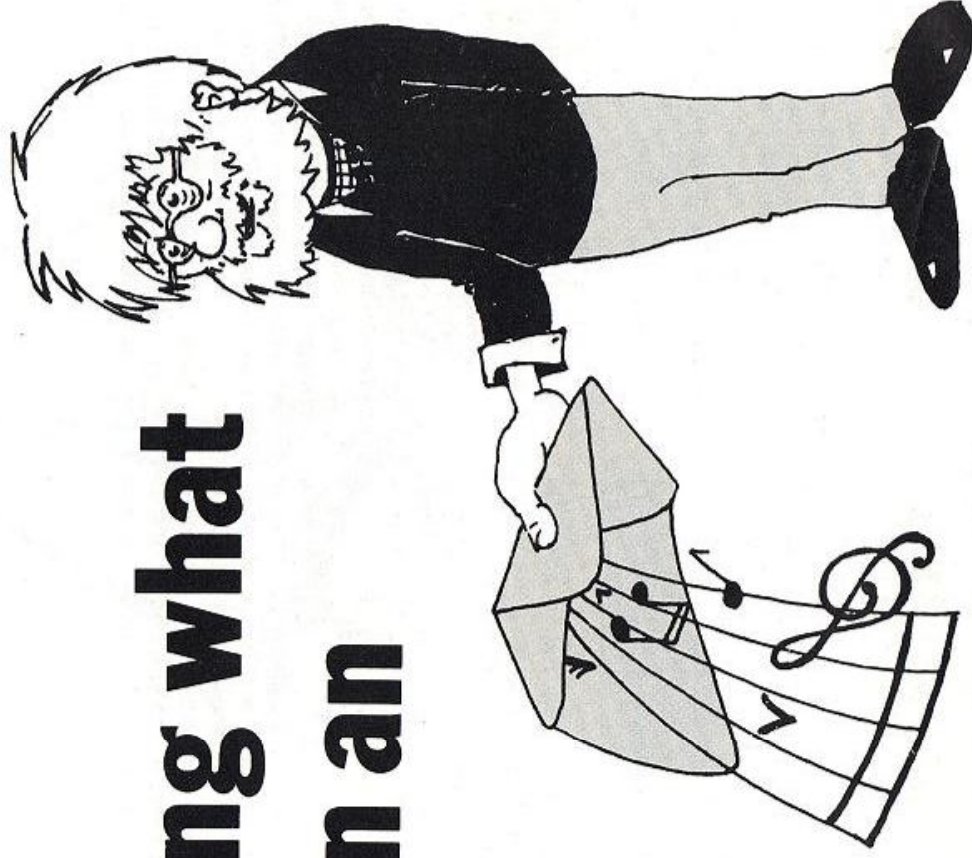
The trouble is that in real life notes don't always stay at the same loudness level. They fade away or grow louder as time goes on. We can't do this with a simple SOUND command we've used so far.

The note starts at loudness 5 and stays at that level until it stops a second later. However, by using a previously defined volume envelope we can make a note vary in loudness as it plays.

So, let's define a volume envelope with:

ENV 1,5,2,20

Don't worry just yet about what that



Part II of our series on putting sound on the Amstrad by NIGEL PETERS

means, just accept that this command defines a volume envelope that we can refer to as envelope number 1.

Now let's hear what effect this has on the note by entering:

SOUND 1,200,100,5,1

You should be able to hear the note getting louder and louder. It still plays for one second but the volume is changing.

The command is the same one we used previously except that, 1 is stuck on its end to tell the micro that the volume envelope previously defined

as number 1 is to be used.

You can define up to 15 of these envelopes, numbered from 1 to 15. Once defined, you call them by attaching the appropriate number to the end of our simple sound command. The Amstrad then produces the note, varying its loudness in line with the volume envelope specified.

Before we go into the mechanics of how the volume envelope is defined, I must first admit that last month I told you what might be considered a small lie.

You'll remember I wrote that the

	Channel	Pitch	Duration	Volume		Volume Envelope
				without envelope	with envelope	
Range	1=A 2=B 4=C	0 to 4095	1 to 32767	0 to 7	0 to 15	0 to 15
Default				4	12	0

Table 1: Parameter ranges for SOUND command

volume parameter can vary from 0 to 7, with a default value of 4. Well this is true, but only if you aren't using a volume envelope.

If you *are* using a volume envelope then the actual volume parameter can vary from 0 (silence) to 15 (maximum loudness). Actually there is no difference in the absolute loudness of maximum volume, for a volume parameter of 7 without an envelope is just as loud as one of 15 with an envelope.

It's just that when you specify a volume envelope the range is divided into 16 parts as opposed to the usual eight parts.

This new range of parameters is shown in Table I.

So we can define up to 15 volume envelopes and, by tagging on the appropriate number to the end of the SOUND command, we can use them to vary the loudness of a note as it plays. The question is, how do we define a volume envelope?

The answer is that we use the ENV command. Now this command can be a fairly formidable-looking beast, having up to 16 parameters following it.

When you get to know it, however, it's not all that bad. The secret is not to let it see that you're afraid of it.

Let's listen to the effects of envelope 1 again. Unless you've switched off, or redefined it, it should still be lurking in your micro, but just to make sure type in:

ENV 1,5,2,20

Notice that the ENV command doesn't make a noise itself. You can type them in until you're blue in the face but the Amstrad will stay mute. All a volume envelope does is (when asked) affect a SOUND command. The SOUND command is the one that makes the noise. If you don't believe me, enter:

SOUND 1,200,100,5,1

and you'll hear a note being affected by a sound envelope.

If you listen carefully, you should hear five distinct changes of loudness during the second that the note is playing. The note gets louder in steps of two and each step lasts for 20 hundredths of a second.

It's important to realise that the

rise or fall in volume caused by an envelope isn't smooth. It takes place in steps. To be formal, the change is discrete, not continuous.

Let's have a look at envelope 1 in more detail. You'll see that it is in the form:

ENV N,P,Q,R

that is, the envelope command followed by four parameters.

All N does is label the envelope you're defining with a number between 1 and 15. There is an envelope 0 but this is the default

volume envelope, fixed at two seconds at the same volume level as the normal volume parameter.

The P parameter just tells the micro how many steps the envelope is going to take. There can be up to 127 of these steps. In our volume envelope 1 there are five of these steps.

The Q parameter tells the Amstrad the increase or decrease in volume that is to occur with each step, the initial volume is taken from the volume parameter of the SOUND command. In the case of envelope 1

Parameter	Number N	Number of steps in section P	Volume change per step Q	Time length of each step R
Range	0 to 15	0 to 127	-128 to 127	0 to 255

Table II: Parameter ranges for ENV command

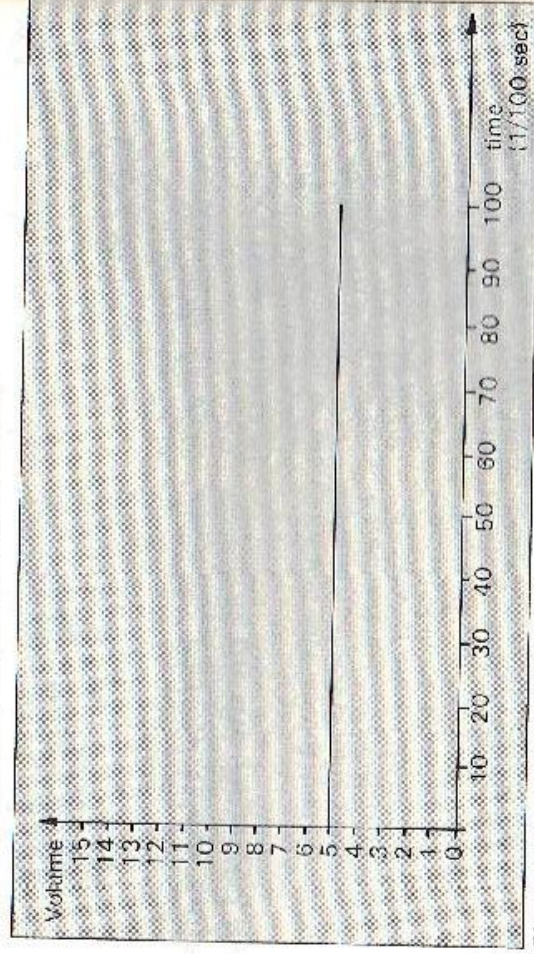


Figure 1: SOUND 1,200,100,5

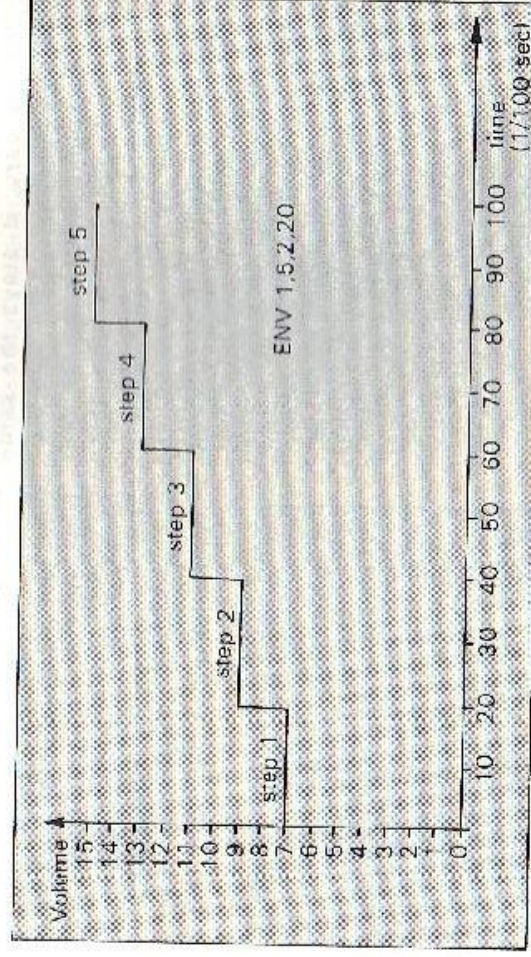
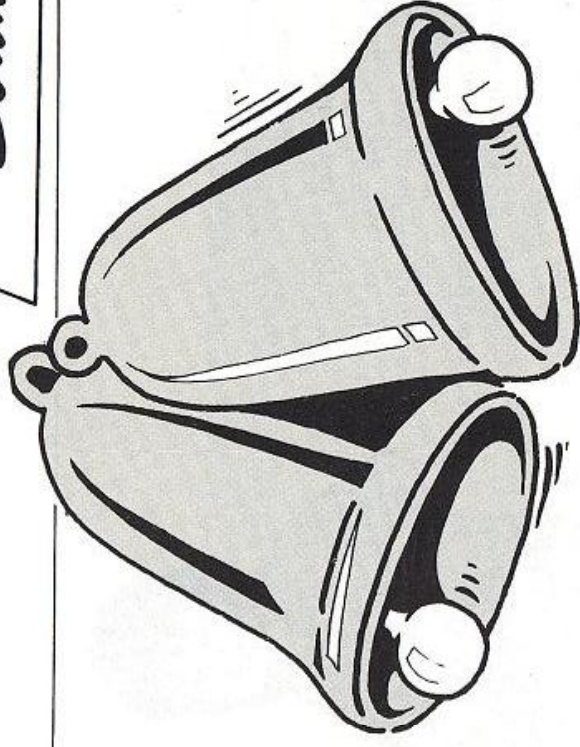


Figure 2: SOUND 1,200,100,5,1



the volume increases by 2 volume units for each step of the envelope. The Q parameter can vary from -128 to 127 in value.

Finally the R parameter decides how long each step of the envelope takes. It is measured in hundredths of a second and can range in value from 0 to 255.

Table II sums up these parameters and their ranges.

Figure I shows the effects of:

SOUND 1,200,100,5

in graphic form.

As you can see, the volume level stays the same throughout the second that the note plays.

Now try our old favourites:

ENV 1,5,2,20

(if you've switched off or redefined the envelope) and:

SOUND 1,200,100,5,1

Figure II shows what happens. The envelope causes the volume of the note to increase during the second that it plays.

There are five steps, each lasting 20 hundredths of a second. Each step causes the volume to increase by two; the result of all the steps being that the volume goes from 7 to 15 in the second the note is playing.

Now have a change and define another volume envelope with:

ENV 2,5,1,20

You can hear this in action by entering:

SOUND 1,200,100,5,2

Volume envelope 2 is called by the parameter at the end of the SOUND command. Now the note increases in loudness again, but since the Q parameter is 1, it doesn't get as loud as before.

Incidentally, volume envelope number 1 is still lurking around the Amstrad's memory. You can hear this for yourself by entering:

SOUND 1,200,100,5,1

and getting the same result as before.

The volume envelope doesn't always have to increase the volume. It can decrease it as well. If you don't

believe me define a new envelope with:

ENV 3,5,-1,20

and then call it with:

SOUND 1,200,100,5,3

Now the volume is decreased by a factor of 1 for each step forward in time.

While we're decreasing the volume, try:

ENV 3,5,-2,20

and:

SOUND 1,200,100,14,4

Bell-like, isn't it? Notice that the volume parameter of the SOUND command - the fourth figure - is 14. This means that the volume envelope starts just below maximum volume and decreases in steps of 2.

Try defining a few envelopes of your own and seeing the effects you can create. It's great fun.

While you're playing you may come across a few weird effects.

You may have noticed how, in the examples above, the duration of the SOUND command has always been one second. Similarly, there have always been five steps, each of which lasted 20 hundredths of a second.

In other words, the five steps of the volume envelope added up to one second, the duration of the SOUND command. You might wonder what happens if the SOUND duration is longer or shorter than the time taken by all the steps of the volume envelope. Let's see.

Define a new version of envelope 1 with:

ENV 1,5,2,10

There are now five steps, each of 10 hundredths of a second so the

envelope span is 0.5 seconds. The time taken by an envelope is given by multiplying P*R, the number of steps by the time each step lasts.

Now see what happens when you enter:

SOUND 1,200,100,5,1

which tells the Amstrad to play for one second.

As you can hear the note quickly achieves maximum volume and stays at that level until the SOUND command finishes. In other words, the envelope only effects the note for the duration of the envelope span (P*R). The note carries on for its full duration, playing at the volume reached when the envelope finishes.

To see what happens in the opposite case, redefine envelope 2 with:

ENV 2,5,1,20

which gives five steps lasting a total of one second.

Now let's have a sound lasting half a second and hear what happens. Enter:

SOUND 1,200,50,5,2

and you'll hear the poor old envelope being cut off in its prime. If you want it to have its full effect use:

SOUND 1,200,100,5,2

Apart from discrepancies between the duration of volume envelopes and the SOUND commands that call them, there are other problems that can arise. What if there are too many steps in the envelope?

Try it and see. While:

ENV 1,10,2,20

expects 10 steps, each lasting a fifth

of a second and needs a full two seconds to do its work.

Unfortunately:

SOUND 1,200,100,5,1

only lasts one second, so the envelope gets cut off in its prime, only having five steps.

The opposite case, where there are two few steps is illustrated by:

ENV 1,2,2,2,20

and:

SOUND 1,200,100,5,1

As you can hear, you get two steps - taking up 40 hundredths of a second, - and the note remains at the final volume for the remaining 60 hundredths.

Now have a look at Table 1 again. You'll notice when an envelope is used, the volume parameters can range between 0 and 15 in value. So far, all the examples have been picked so that the parameters stay in this



range.

What happens if a volume envelope tries to go out of range?

Define an envelope with:

ENV 5,5,3,20

This increases the volume in steps of three each time. Now consider the effect of this envelope on a SOUND

command such as:

SOUND 1,200,100,5,5

The initial volume is 5. The first step of the envelope will make this 8, the second 11, and the third 14. But what happens when the fourth step tries to give it a value of 17 which is out of range? Try it and see.

As you can hear, the Amstrad is a crafty beast and, seeing that it is being asked to do something naughty, it just takes away 15. As you can hear the volume "wraps around", going suddenly from very loud to quiet again.

It can happen the other way as well. The envelope defined by:

ENV 6,5,-3,20

decreases the volume by 3 for each step of the envelope. When it works its wicked way on:

SOUND 1,200,100,5,6

the volume goes from 5 to 2 and then,

Don't miss it!

If you own a CPC computer with disk drive you'll know how expensive disks have become. If you make back-ups it's costing you twice as much. No more! If you own a CPC 464 you already have the equipment needed to save lots of dollars. If you own a CPC 664 or CPC 6128 you'll need a cassette recorder.

In next month's Computing With The Amstrad we publish the listing for Archiver - a disk to tape utility that allows you to place up to two full disks on one 60 minute cassette.

If you never buy another issue, don't miss December's CWTA - you'll save hundreds of dollars!

Place an order with your Newsagent now or send in the subscription form, on page 76 and get in before our price increase.

CWTA - December issue -don't miss it!

rather than go out of range, wraps around to 14 and carries on decreasing again. Try it and see.

By now you should be fairly comfortable with the volume envelope. You might also be aware of some of its limitations. One of them is that so far we can only use a volume envelope to make a note increase or decrease in volume.

In real life, however, notes sometimes do both, growing gradually louder, then fading away. In its present form of:

```
ENV N,P,Q,R
```

our volume envelope can't handle this. And once again I must confess to telling you rather less than the truth.

You see, the actual definition of a volume envelope isn't:

```
ENV N,P,Q,R
```

it is really:

```
ENV N,P1,Q1,R1,P2,Q2,R2,
P3,Q3,R3,P4,Q4,R4,
P5,Q5,R5
```

Instead of our friendly little ENV with four parameters, we've got a huge

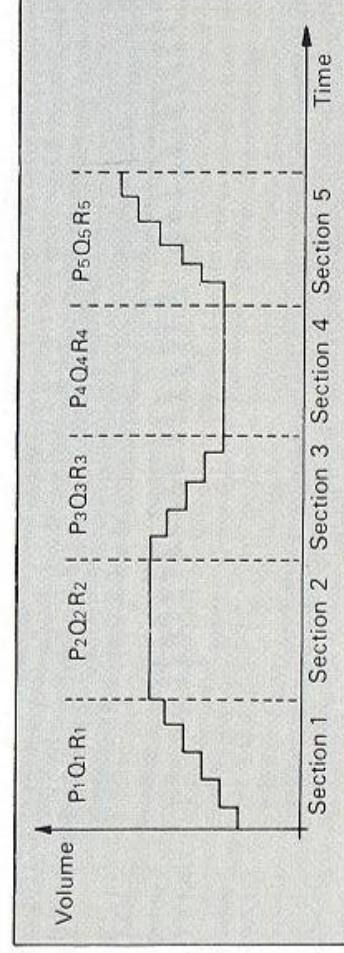


Table III: Parameters for all five sections of volume envelope

beast with 16 numbers after it. However, don't give up in disgust or fear, it's quite simple really.

Up until now we've only been using a fraction of the volume envelope's parameters. Each envelope can have up to five sections, each section affecting the same SOUND command in a different way.

So far, we've only used the first section as a kindness to those of a nervous disposition.

Each of the five sections of the volume envelope works in exactly the same way as the first one that we've been concentrating on. All you have

to do when figuring out an envelope is to treat each of its five sections as we did before.

The only difference is that instead of P, Q and R, the first section has parameters P1, Q1, R1, the second P2, Q2, R2 and so on. Figure III shows how the parameters relate to the sections.

Although you can have five sections in a volume envelope, as should be obvious from the above you don't have to have all five sections in use.

Let's define an envelope that will have the volume increasing, then decreasing, then increasing again.

We do this with:

```
ENV 1,5,2,20,5,-2,20,5,2,20
```

Here the first section of the envelope has the volume increasing by two for five steps. The second section has the volume decreasing by two for five seconds. I leave it to you to figure out what the third section does.

```
SOUND 1,200,300,5,1
```

will let you hear this three-sectioned envelope. Notice I've increased the duration of the SOUND command to three seconds. This ensures that all the sections get a chance to play.

And really that is all there is to the volume envelope. Although it looks difficult it's not that hard so long as you take it section by section. All it needs is a little practice to provide familiarity, and that's where Program I comes in.

Run it and you'll find it will allow you to create your own volume envelopes and hear the results. After an hour or so you'll find that you'll understand them completely.

And you may even be looking forward to next month's article on pitch envelopes.

```
10 REM PROGRAM I
20 REM VOLUME ENVELOPE
30 DIM P(5),Q(5),R(5)
40 WHILE -1
50 MODE 1
60 INPUT "How many sections in volume
envelope?", sections
70 IF sections<1 OR sections >5 THEN
CLS:GOTO 60
80 CLS
90 FOR loop=1 TO sections
100 LOCATE 3,5:PRINT "Section" loop
110 LOCATE 3,8:PRINT "Number of steps
?"
120 LOCATE 30,8:INPUT P(loop)
130 IF P(loop)<0 OR P(loop)>127 THEN
LOCATE 30,8:PRINT SPACES(8):GOTO 120
140 LOCATE 3,13:PRINT "Size of each s
tep?"
150 LOCATE 30,13:INPUT Q(loop)
160 IF Q(loop)<-128 OR Q(loop)>127 TH
EN LOCATE 30,13:PRINT SPACES(8):GOTO
150
170 LOCATE 3,18:PRINT "Duration of st
ep?"
180 LOCATE 30,18:INPUT R(loop)
190 IF R(loop)<0 OR R(loop)>255 THEN
LOCATE 30,18:PRINT SPACES(8):GOTO 180
200 LOCATE 14,23:PRINT "PRESS SPACE"
210 WHILE INKEY(47)=-1:WEND:CLS
220 WHILE INKEY(47)=-1:WEND
230 NEXT loop
240 ENV 1,P(1),Q(1),R(1),P(2),Q(2),R(
2),P(3),Q(3),R(3),P(4),Q(4),R(4),P(5)
,Q(5),R(5)
250 duration=P(1)+R(1)+P(2)+R(2)+P(3)
+R(3)+P(4)+R(4)+P(5)+R(5)
260 SOUND 1,200,duration,5,1
270 CLS
280 duration$=RIGHT$(STR$(duration),L
EN(STR$(duration))-1)
290 PRINT "SOUND 1,200,";duration$; ",
5,1
300 FOR loop=1 TO sections
310 loop$=RIGHT$(STR$(loop),1)
320 PRINT "P(";loop$;)" ";P(loop)
330 PRINT "Q(";loop$;)" ";Q(loop)
340 PRINT "R(";loop$;)" ";R(loop)
350 NEXT
360 LOCATE 14,23:PRINT "PRESS SPACE"
370 WHILE INKEY(47)=-1:WEND:CLS
380 WEND
```

Program I

PDI = Public Domain Information

Welcome to CWTA's inaugural Public Domain Information column. Over the coming issues Shane Kelly will disseminate some of his vast wealth of knowledge of the subject of CP/M and MS-DOS public domain software.

As some of you would already know, Public Domain software is software which is freely available from various sources such as user group libraries and bulletin boards. Generally, PD software is not copyrighted and is available for copying without restriction. Some PD software however is also referred to as Shareware or other similar names. This indicates that the author, whilst making the software available is seeking a donation towards recouping his costs in producing the software. Shareware is generally supported by the author with updates and documentation being made available to registered users only.

Software which is not CP/M or MS-DOS and which exists in user group libraries should be marked as Public Domain software and should indicate that it is available for copying without charge for the software. If no such statement is made, the user should be extremely careful about copying as the software may be copyright.

Where PD software is not available from bulletin boards etc., distributors generally make a charge for media costs, preparing documentation and the time involved in configuring the software for a particular machine which is in some way non-standard (as is the Amstrad because of its disk format and the CPC464 in particular because of the lack of memory under CP/M). PD distributors will of course charge for postage and mailing costs as well as the above.

Over to Shane:

Welcome to the inaugural Public Domain column of Computing With The Amstrad. This column is here for only one reason - to bring you software that is freely available BUT in a format that your Amstrad can understand. Where possible, source listings will be provided with alterations to allow CP/M 2.2 or 3.0 operation. Where this is not possible, all programs will be tested to check that they run on the Amstrad range of computers. This will prove superior to the current situation where PD programs are down-loaded from bulletin boards or other obtained by other means and are found to run critically or are squashed up into a window 10 characters wide and lock up completely.

This month's disk contains a number of utilities of the type which will prove invaluable to all disk users. You may find that you won't use these utilities that often but will find yourself in trouble if you haven't got them.

This month's star program is NUSWEEP - the definitive file manipulation program. You will find the relevant documentation for NSWP and FINDBAD on the opposite page.

The balance of our first PD disk (PDI) is made up with the following programs:

CHECKS.COM/DOC - A cheque book maintenance program [3.0 only]
ALLOC.COM - Disk utility to show allocation vector [2.2 only]
COMPARE.COM - compares two files and reports on # of bytes [3.0 and 2.2]
DISPLAY.COM/DOC - allows formatted display of files [3.0 and 2.2]
CRC.COM/DOC - Compute cyclic redundancy check of file/disk [3.0 and 2.2]
DIR-DUMP.COM - Dumps directory sectors on Dump format [2.2 only]
DIRCHC.COM/DOC - Checks correctness of directory sectors [3.0 and 2.2]
EXAMPL.CHK/NAM/DOC - Example file for CHECKS.COM [see Checks]
FILTER.COM/DOC - Removes all control characters except: tab, cr, and lf from ASCII files [3.0 and 2.2]
FINDBD54.COM/DOC - checks entire disk and locks out bad sectors
MENU.ASM/COM/DOC - Displays disk contents in menu form and invokes programs by keypress [2.2 only]
NSWP207.COM - see article [3.0 and 2.2]
PROTOCOL.DOC - Discussion of XMODEM protocols for comms programs [doc file]
SUPERDIR.COM/DOC - Rather like 3.0 DIR command [3.0 and 2.2]
VLIST.COM/DOC - Varies the speed of a file typed on the screen [3.0 and 2.2]
DU-77.COM/DOC - a disk disector of high caliber [2.2 only]
FIND.COM - Search for an ASCII string in a file [3.0 and 2.2]
LASM.COM/DOC - A linking version of ASM [3.0 and 2.2]
LISTT.COM/DOC - Formatted output to printer [3.0 and 2.2]
PGLST.COM/DOC - Changes an ASCII file to columnar format [3.0 and 2.2]
PRINTPRN.ASM - Prints a CROWE assembler file in expanded form (included so non-programmers can see an ASM file) [unknown]
SORT.COM - Sorts an ASCII into ascending order [2.2 only]
WASH.COM - Older version of NSWP - include for interest to see how a file can progress and improve over time [3.0 and 2.2]
ZIP.COM - A buffered file transfer utility that's really quick! [3.0 and 2.2]

That's it for this month - you'll find ordering information elsewhere in this issue. Next month we'll look at programs to help you communicate.

FINDBAD.COM

THIS utility lets you test a disc for damage or bad sectors non-destructively. It will test all sectors on the disc and group any bad ones it finds into a dummy file to prevent their use.

It should then be safe to use the disc as normal — however if you don't know why a sector has failed then treat the whole disc with suspicion and check it regularly if it is still in use.

FINDBAD should really only be used on blank discs immediately after formatting, otherwise if a bad sector is found any information in it will be lost. It will only run under CP/M 2.2 — not CP/M 3!

NSWEEP.COM

THIS is the latest version (2.07) of the popular utility NewSweep. It is of particular use if you have two disc drives, but even on a single-drive system it allows you to selectively delete files, squash and unsquash text files, alter file attributes, rename files and much more.

Type ? once you are into NSWEEP to display a menu of functions. We'll explain a little about each in turn.

A — Retag files, T — Tag file for mass; W — Wildcard tag of files

Many of NSWEEP's functions operate on groups of files which you have selected for a common operation. T will tag the file the cursor is at to mark it for participation in any future multifile operation. This will be shown on the file list by an *.

After any mass operation tagged files are put into limbo and marked with a #. If you want to retag the same files for another operation type a, if not the first new t you do will untag all the old ones.

W allows you to specify groups of files for tagging using wildcard names — for example, typing s*.com in response to NSWEEP's Tag what? prompt will tag all .COM files with names starting with S.

B — Back one file; cr, sp — Forward one file; F — Find file

Pressing Enter or the spacebar will step you forward through the file list and B will move you

CP/M users are fortunate in having a wealth of public domain software. As a service to our readers this month's software disc will contain — in addition to all the main listings from this issue — two invaluable CP/M public domain utilities — NSWEEP.COM the 2.07 version of the popular NewSweep utility, and FINDBAD.COM with which you can test for bad sectors. Read on for full details of these vital CP/M routines.

backwards which is useful for getting to files nearer the end of the alphabet. A faster way of reaching a particular file is to use f — NSWEEP then prompts with Find what? to which you can type any or all of the letters of the filename.

D — Delete file; E — Erase T/U files

D allows you to delete the file which the cursor is at in the list — NSWEEP prompts for confirmation. E allows you to delete some or all of either the files you have tagged for mass operations or the ones which you have not tagged.

NSWEEP prompts for tagged or untagged files and whether you want to confirm individual files. If you type a in response to the Delete (Y/N/A)? prompt the operation will abort.

L — Log new disc/user

Type 1 or L and you get New drive/user/mask? To select drive B:, for example, just type b. To log to user 2 on drive A: type a2. To only deal with .COM files in the currently logged drive/user area type *.com — and so on.

M — Mass file copy

This command allows you to copy any number of files which you have previously tagged to another disc and/or user area. Typing a v after the drive/user specification will force NSWEEP to verify each file after copying it.

V — View a file — and P — Print a file on the LST: device

These commands will work on squeezed files as well as normal ones — you don't even need to unsqueeze a file before you look at it or print it out.

Q — Squeeze/Unsqueeze files

NSWEEP uses Huffman coding to reduce the size of a text file by between a third and a half — this is useful on systems like the Amstrad where we have very small discs.

When you type q you are prompted with Squeeze, Unsqueeze or Reverse (S, U, R)? Select as appropriate and you are

prompted with Copy to drive/user? Type a for drive A:, current user; b2 for drive B:, user 2 and so on. All files you have previously tagged will then be squeezed or unsqueezed as requested.

R — Rename file(s)

Type r and you are prompted with New name, or *? Typing a new name will rename the file that you are currently at in the list. Typing * takes you to a further prompt Old name?

You can specify the name(s) of the file(S) you wish to rename. The real power of the command lies in its ability to accept wildcard filenames, for example, *.xxx means all files with extensions of .XXX.

After this you are prompted for New name? which again will accept wildcards. So, for example, if we answered *.yyy to this we would take all files with extensions of .XXX, keep their names the same but change their extents to .YYY. It gets better — this command also allows us to change a file's user number.

The following dialogue Old name? O:*.xxx New name? 2:*.yyy would not only change the filenames as before but would also move the affected files from user 0 to user 2.

S — Check remaining space

This allows us to check the space free on any disc before copying files to it — NSWEEP prompts for the drive to check.

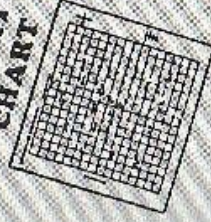
Y — Set file status

Type y or Y and you get Which flags (1-4, R,S,A)? One to four are the user bits in the filename — used for file protection by some programs such as MBasic, for example.

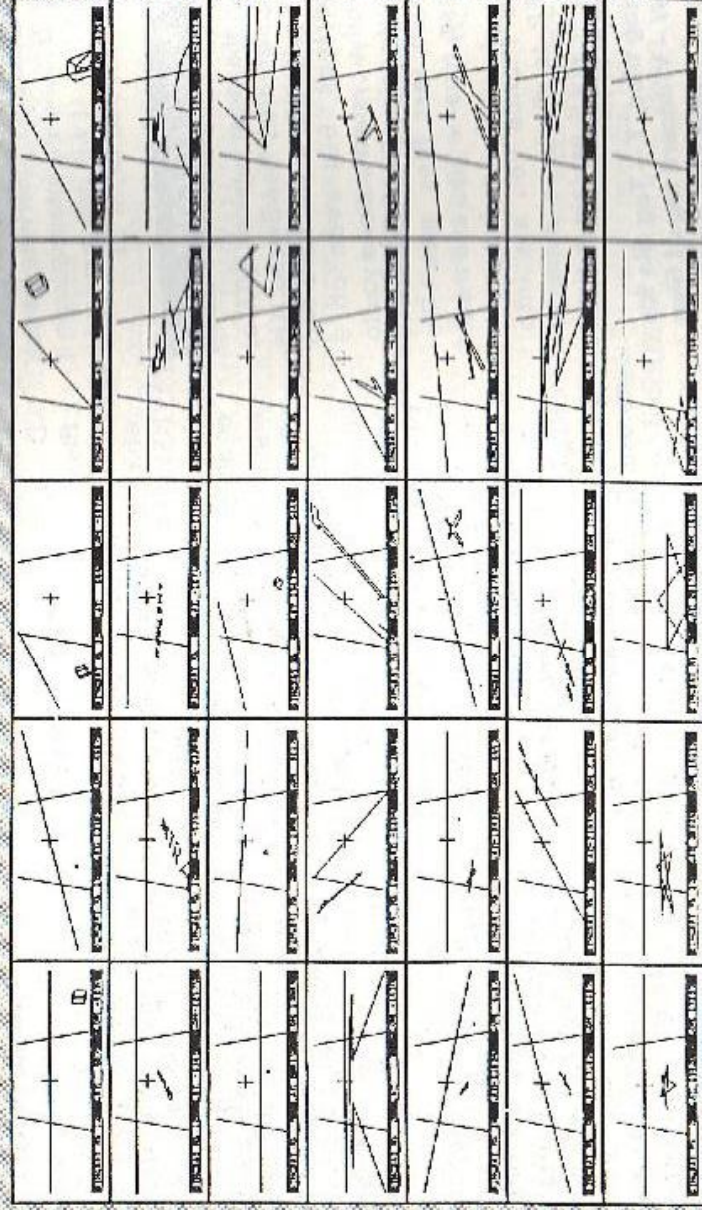
R is the \$R/O/\$R/W flag which when set writes/deletes-protects a file. It is reset with W. S is the \$SYS/\$DIR flag which when set hides a file from a directory listing. It is reset with D. A is the "Archive" flag which is used under CP/M 3 only. NSWEEP will work happily under both CP/M 2.2 and CP/M 3.

MYRDDIN**3D LANDMARKS
YOU CAN FLY AROUND****SUPERB REAL
TIME SIMULATION**

MYRDDIN FLIGHT SIMULATION

**MANUAL****AMSTRAD CPC 464****FULL SCREEN
DISPLAY****FLIGHT
CHART**

Here are some screens from a typical flight showing the view from the cockpit (top half of screen) produced as printouts of the actual simulator.



A real time simulation with 3D graphics uses a massive 64000 x 64000 longitude & latitude flying area, making each flight completely different. Developed under pilot instruction to give realistic flight effect. The view through the cockpit gives moving 3D graphics.

Comprehensive instrument panel with moving needle meters & digital displays. 15 aircraft types with varying control sensitivities & speeds of between 100 - 500 knots.

3 runways available for refuelling, take off & landing. Ground and landmark orientation correct with all flying attitudes (rolls etc.).

The 3D graphics are still accurate when you fly upside down.

3D landmarks you can fly around.

Comes complete with manual & fully detailed chart of landmarks & airfields.

Joystick or keyboard operation.

STRATEGY SOFTWARE
BOX 5000
GLEN WAVERLEY
VICTORIA 3150
(03) 560 4324

TAPE \$17.95**DISK \$29.95**

Dk'tronics 64k Silicon disk

App. \$145

Various distributors

THIS smart unit from Dk'tronics enables CPC6128 owners to use its second 64k of banked RAM as an extra disc drive. It's rather like the RAM disc on the PCW micro.

The Silicon Disc is a small neat box in the same style as the micro and it plugs into the 6128's expansion socket at the rear of the keyboard:

It's not very deep, but it is quite wide and completely covers the printer port. This doesn't mean that you can't use the printer, as there is a slot in the case for the printer cable and plug to pass through.

On power-up, or after a complete reset, the unit announces its presence with the start-up message Silicon Disc 0.1. Before you can use the extra RAM as a disc it needs to be initialised with iSDISC.

It will then tell you which drive it is — B or C — and the amount of storage space you have. You can't use the full 64k since the directory takes up 2k leaving you with 62k free.

The RAM disc operates exactly like a disc drive and from Basic it is selected with iB. You can now load, save, cat, open files, close files and so on just as you would with the built-in drive. There are no differences at all.

Well that's not quite true. Since this disc drive is really a section of the micro's RAM everything will be lost when you switch off though not when you reset, unlike a normal disc which remembers all the data stored on it.

So if you do use the RAM disc you'll have to remember to transfer all the files to a real disc before you switch off.

This can be a bit of a chore, but fortunately there are short cuts available. Two extra RSXs have been added — iLOADDISC which will transfer the contents of a disc to the RAM disc, and iSAVEDISC which saves the RAM disc contents to a real disc.

Make sure that the format of the disc you load from and save to are the same. For instance, if you use

Silicon Disc rams home the message

**ROLAND WADDILOVE checks
out a cheaper and sometimes
faster alternative to buying
a second drive for the CPC 6128**

iLOADDISC to copy a CP/M Plus formatted disc to the RAM disc you must copy it back to a CP/M Plus disc with iSAVEDISC.

CP/M 2.2 does not use the extra 64k of banked RAM in the CPC6128, so this is available as a RAM disc and operates just like a second drive. CP/M Plus also supplied with the CPC6128 uses all of the micro's memory including the second 64k of banked RAM.

This doesn't leave any for the Silicon Disc to implement as a RAM disc. However if you add one of Dk'tronics' RAM packs it can use this for its RAM disc instead.

To use the RAM disc under CP/M a file called SDISC.COM is required on the CP/M disc. The iSETCPM command will write this to the disc. CP/M now needs reconfiguring before it can be used using MOVCPM 176 * and SYSGEN * on the utilities disc.

Boot up CP/M and enter SDISC, and you've now got a second drive. Alternatively you could use SETUP to automatically initialise the RAM disc by entering SDISC into the initial command buffer.

All CP/M programs are capable of using the RAM disc by prefixing the filename with B: and the RAM disc can be selected with B: on its own. Files can be PIPped across on to the RAM disc at the start of a session and PIPped back at the end.

The Silicon Disc acts exactly like a real disc drive, so is there anything to

be gained in choosing this unit? The advantages are speed and price — it's cheaper than a disc drive and under certain circumstances it is faster to access.

For instance, when it comes to loading and saving Basic or binary files the RAM disc is up to 10 times faster.

Although after using cassettes for years on several micros I don't consider waiting five seconds for a program to load from an Amstrad disc an inconvenience, the RAM disc will take less than a second.

When saving files in Ascii form there is very little difference in speed. In fact the RAM disc can be slightly slower when reading files.

The RAM disc is very useful for storing machine code utilities when programming in Basic and utilities like PIP, STAT, DUMP, DDT and so on when using CP/M. Once you've used a dual drive system you'll wonder how you ever managed without one.

The Silicon Disc is also available for the CPC464, but you'll need a Dk'tronics RAM pack and a disc drive to use it. The CPC6128 can also make use of these RAM packs to give you a larger RAM disc if you find yourself short of space.

To sum up the Silicon Disc is a well made and designed add-on which will instantly provide you with a 62k RAM disc ready to use. It's cheap, fast and I couldn't fault it. I can thoroughly recommend it. ■

Subscriptions

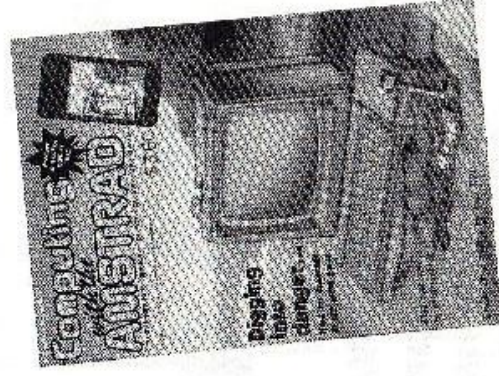
In just
three
months
you've had:

5 Games Listings
5 Hardware Reviews
32 Software Reviews
6 Utilities Listings
18 Feature Articles
and lots, lots more!

Special offers to subscribers alone have been worth up to \$74 if you have taken advantage of all of them!

An increase in size to 80 pages beginning this month forces us to increase the cover price to \$3.95 but subscriptions will still be available until the end of this year at only \$36 plus optional cassettes or disks.

Give the ideal Christmas present - one that will be given 12 times in a year!



Subscription Order Form

Name

Address

State

Postcode

Telephone

Magazine Only ☐ \$36 Magazine + Tape ☐ \$72

Bankcard ☐ Mastercard ☐

Number: Expiry Date:

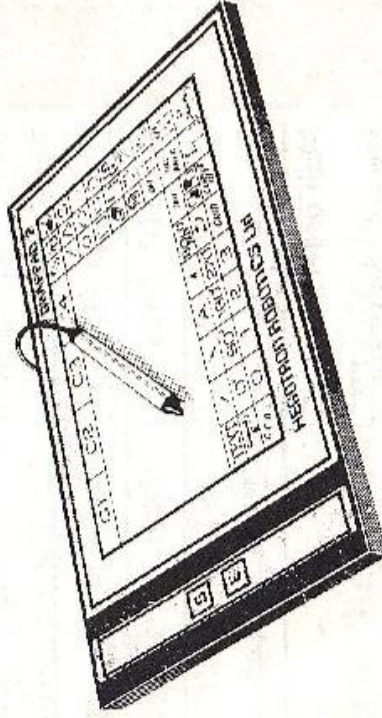
Signed:

HUGE PRICE LIFT ON DISK DRIVES

Hold on to your hats! The dollar debacle has now hit Amstrad computers, peripherals and software. Most items have risen only 5-8% but the DDI-1 disk drive for Amstrads has risen by almost 50%! Sources within Mitsubishi-AWA say that the old price (\$449) was a special price due to the large quantities that were held in stock from prior to the dollar drop. The dollar has now dropped approximately 40% over the last 18 months and the new price for DDI-1s reflects this at a whopping \$649. Other drives to have risen in price are the FD-1 (second CPC drive) which is now \$565.50 and the FD-2 (second drive for PCW 8256) now priced at \$617.50. The 6128 has risen in price to \$1099 and most software has gone up by a dollar or two. three inch disks have risen again and now retail at \$10.50 each. The good news is that these are probably the last increases in store for some time. The dollar seems fairly settled at an exchange rate of around \$2.50 to the U.K. pound and (hopefully) we won't be getting any more tax increases on our Amstrads!

GRAFPAD 2 ARRIVES

Mitsubishi-AWA have just received their first shipment of this well known digitizer pad from the U.K. Grafpad 2 should prove to be extremely popular for those into graphics - Grafpad 2 comes complete with software on a very expensive 3" disk and has the capability to do screen dumps of your glorious art work.



The Mouse that snarled

Mitsubishi-AWA have temporarily halted imports of that CPC rodent from AMX due to a price squabble. Seems AMX want their share of the price increase cake too!

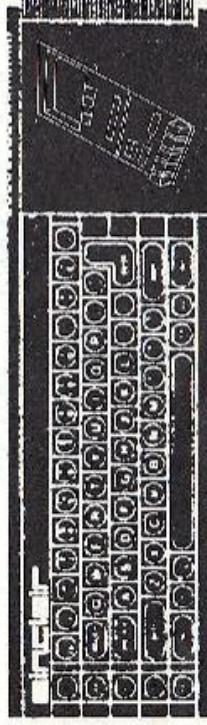
New software for PCW owners...

Hot off the wharf are: Locoscript Mail Merge, Spell Checker and the Typing One typing tutor. The first two will retail around \$125 each and Typing One for about \$89.

And for CPC owners

Hear there are lots of new games available for CPC machines (provided you can afford a disk drive!). About twenty new titles will be in the shops around mid-November and should help solve those Christmas present problems.

Is this it?



An artists impression of the new Amstrad/Spectrum we forecast last month. The QL-based system has 128k of RAM coupled with a Straddy cassette deck. The new machine however will only run Spectrum QL software. The unit will be available without monitor but may NOT be available in Australia. If it did become available we'd tip a price of around \$350 with it selling as a low-end games machine.

SPECIAL!

In a limited pre-Christmas offer, the Giltronic Accounting package includes:-

PCW-8512, Accounting 1 software, 2 * CF2-D discs,
2 * CF2-DD discs, an assortment of pre-printed
computer paper, Locoscript Tutor audio tapes and,
for your entertainment, Cyrus II chess.

Total value in excess of \$2650

SAVE OVER \$400

Our price (inc. insured freight).....\$2250

Accounting One for PCW-8256 owners -
\$1099

Package consists:-

Accounting 1 software, 1 * Giltronic 5.25" 1
megabyte disc drive, Interface and full fitting
instructions. (Minor alterations are required to
monitor case to allow cable access)

PCW-8256 RAM upgrades available mid-November
price to be announced - call for further details.

Why use expensive 3" discs?????

Fit a Giltronics 5.25" Disc Drive to your
Amstrad!

Drives suitable for:

CPC-464/6128

PCW-8256/8512

from \$319
from \$349

1 Meg 5.25" "B" Drive for PCW-8256
(including interface etc.....\$535)

Installation can be arranged (in Melb.) for \$36

**Full range of Software and
Hardware for Amstrad
computers.**

**CALL
(03) 580 9839
NOW!**

Ordering with the Amstrad - November 1986

Giltronic

AUSTRALIA

AMSTRAD

MAIL ORDER

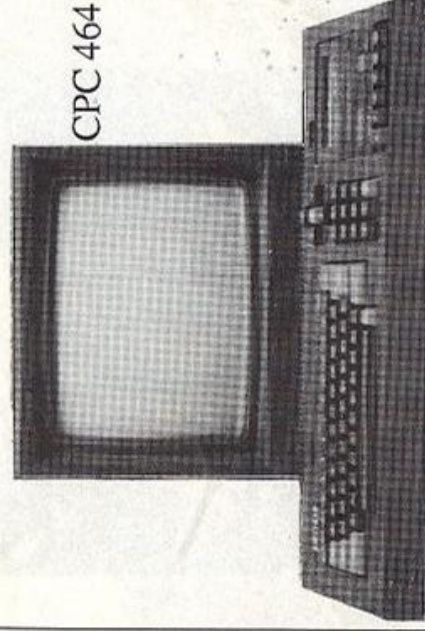
SOFTWARE & HARDWARE

(03) 580 9839

AMSTRAD PC1512

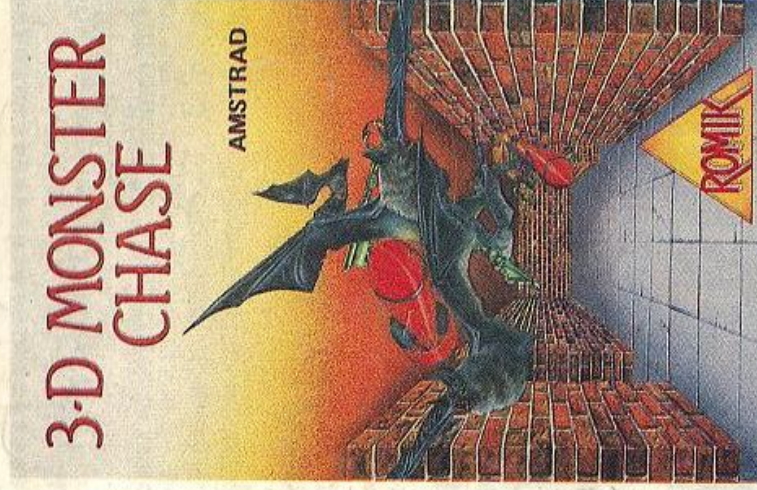
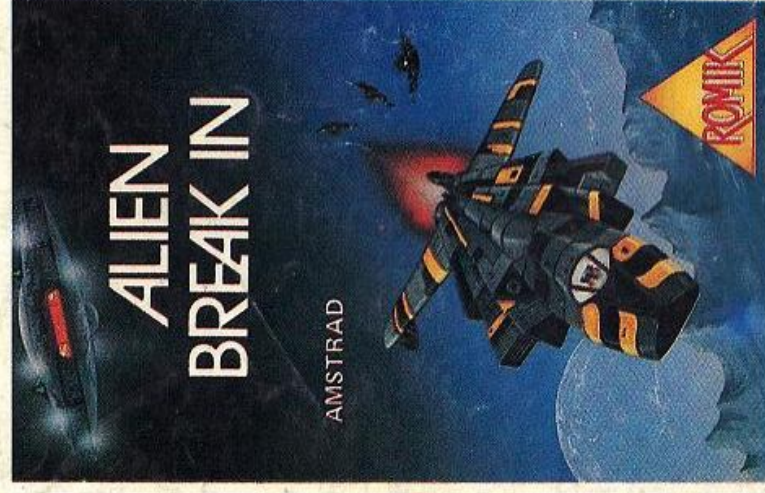


**IBM
COMPATIBLE**
(from \$1495)



CPC 464

FOUR OF A KIND



\$9.95 Each on cassette
 \$29.95 for all four on cassette
 \$39.95 for all four on one disk

Strategy Software, Box 5000, GlenWaverley, Victoria 3150
 (03) 560 4324
 Bankcard & Mastercard welcome