A Database Publication

Computing with the AMSTRAD

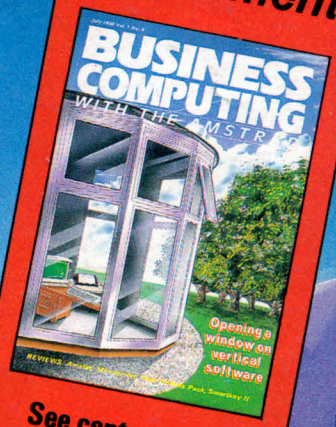
Premier Australian Edition 1986

The independent magazine for Amstrad computer users

Registered by Australia Post - Publication No. VBP7811

$3.60

Free Supplement



See centre pages

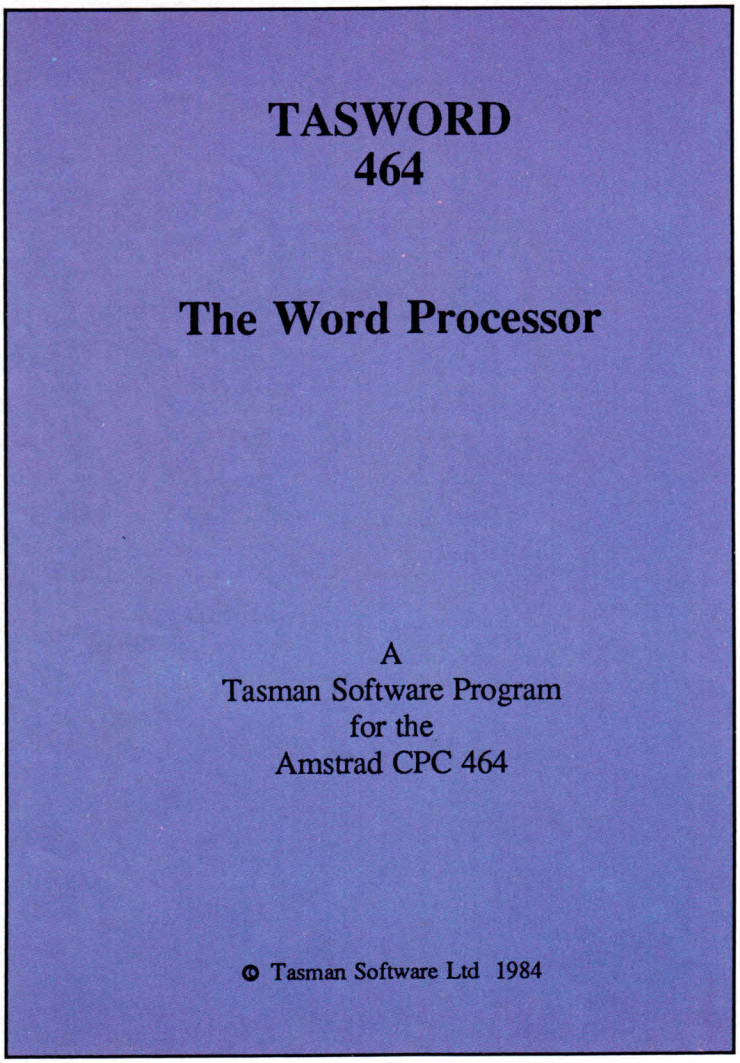
Digging into danger...

Hunt for diamonds in our great game

Map your way to Terrormolinos

Create extra colours for your CPC

DK'Tronics Speech Synthesiser review



PREMIERE AUSTRALIAN EDITION

CONTENTS

PREMIERE AUSTRALIAN EDITION

More on finding your way round loops

By PETE BIBBY

WE'LL take another look at loops this month. In a way it'll be a bit of revision, but also more than that.

I hope that by the end you'll have a greater understanding of how loops work, know which ones to use in different situations and how not to use them.

A word of warning though. We'll be using the dreaded GOTO again. This is just so that the way the program goes from line to line – the flow of control – can be demonstrated.

In practice, as we'll see, their use can be easily avoided. Do this. Avoid them like the plague.

After that cautionary word on to our first program, complete with GOTO.

```
10 REM Program I
20 REM Start of loop
30 PRINT "This is the"
40 PRINT "body of the loop"
50 GOTO 20
60 It never reaches here
```

Program I

Spectacular stuff this. You know that loops are meant to go round and round doing the same bit of code over and over again. Well that's what this one does, but it never stops.

The code that's repeated is known as the body of the loop. In this case it's made up by lines 30 and 40 which print the message that's filling your Amstrad's screen.

The reason that the loop never ends is the GOTO of line 50. As soon as this line is reached the program is sent back to line 20. It then works its way through the body of the loop, displaying the messages, and then comes to the GOTO again. This unceremoniously sends it back to line 20 and the whole cycle repeats.

There are no conditions attached to the GOTO. As soon as that line is reached it's obeyed without question, the program jumping to the line specified. Hence it's known as an unconditional jump. Figure I shows the flow of control for this endless loop.

Line 60 seems a bit odd as there's no Basic keyword there. Normally this would cause an error message, but in this case nothing happens because the program never reaches line 60. The closest it ever gets is line 50, which promptly sends it off elsewhere. !ncidentally if you're fed up of seeing a screen full of text just press Escape for relief.

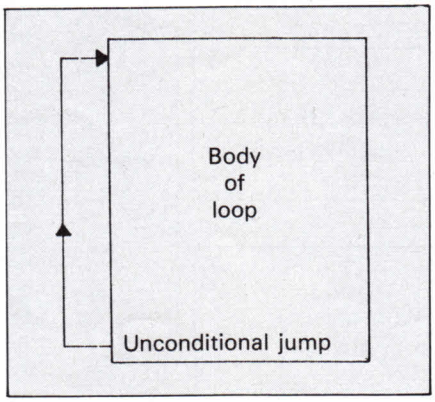


Figure I: An endless loop

Program II should stop of its own accord. Always provided that it's typed in correctly.

Let's go through it line by line and then look at the program as a whole.

The first line is just a REM labelling the program. The next line sets the numeric variable *loop* to 0. Now this isn't strictly necessary as the Amstrad automatically takes the value of a variable to be zero if it hasn't previously been given another value.

However this isn't always the case with other micros and languages so it's good practice to initialise the variable as I've done in line 20. It can also help in debugging recalcitrant programs if you have all your variables and their initial values explicitly stated somewhere in the listing.

```
10 REM Program II
20 loop=0
30 REM Start of loop
40 loop=loop+1
50 PRINT "loop is ";loop
60 IF loop<10 THEN GOTO 30
70 REM End of loop
80 PRINT
90 PRINT "Final value of loop is ";lo
op
```

Program II

Line 30 REMs the start of the loop with lines 40, 50 and 60 forming its body. Line 40 just adds 1 to the value of *loop*. At the start *loop* has the value 0, so the first time round the loop line 40 leaves *loop* with the value 1.

Should you doubt this you'll find that line 50 confirms it by printing its value. The next line is the one that controls it all.

The IF checks to see if *loop* is less than 10. Should this be the case – as it obviously is when *loop* is 1 – the part of the line after the THEN is

obeyed. This happens to be a GOTO, sending the program back to line 30, which is the start of the loop.

Now the program carries on as before, adding one to *loop* and displaying it. Again its value is tested at line 60 and again the program goes into the loop again.

Only when *loop* eventually reaches 10 does line 60's condition become false. Now the GOTO is ignored and the program goes on to line 70, which is the end of the loop. Here it "drops out of the loop", to use the jargon. The next line just gives the final value of *loop*.

The result of all this is that the numbers from 1 to 10 are printed out. This is known as a conditional loop as it has a condition for ending the loop.

Well really it's a condition for carrying on with the loop. If this condition isn't met the program drops out of the loop. Figure II shows the flow of control involved here.

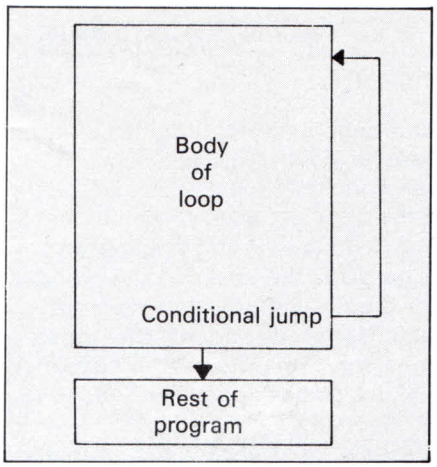


Figure II: A FOR . . NEXT loop

You might find Program II fairly familiar. After all it's very like a FOR . . . NEXT loop, as Program III shows.

This does the same job as Program II, but there is a slight difference. Here

```
10 REM Program III
20 REM Start of loop
30 FOR loop=1 TO 10
40 PRINT "loop is ";loop
50 NEXT loop
60 REM End of loop
70 PRINT
80 PRINT "Final value of loop is ";lo
op
```

Program III

the final value of *loop* is 11, as opposed to final value of 10 in Program II.

This is just a peculiarity of the way that FOR . . . NEXT loops work. It's always worth bearing in mind that the loop control variable will be one more than you might think reasonable.

Looking at the two programs, which seems easier to use and understand? I think you'll agree that Program III is to be preferred. However, as a test of your abilities, can you alter Program II so that it performs exactly like Program III?

You'll have to change the value of the condition in line 60 and put an IF on the PRINT of line 50. When you've done that can you make the program have a feature mimicking the FOR . . . NEXT loop's STEP?

Let's move on and look at another kind of loop, the kind shown by Program IV.

```
10 REM Program IV
20 endCondition=5
30 startValue=1
40 count=startValue
50 REM Start of loop
60 IF count>=5 THEN GOTO 100
70 PRINT "count is now ";count
80 count=count +1
90 GOTO 50
100 REM End of loop
110 PRINT
120 PRINT "The program has dropped ou
t of the loop"
130 PRINT "count is now ";count
```

Program IV

After the initial REM the next lines are just used for initialising variables whose use we'll see when we explore the main loop. This begins at line 50.

Notice that the first line of the loop is a condition. If the variable *count* is equal to or greater than 5 the program will go to line 100 which, as the REM informs us, is the end of the loop.

So there's a test right at the beginning of the loop which, if it is evaluated as true, has the program leaping over the body of the loop and carrying on from the following line.

The first thing the program meets is the loop exit condition. Should *count* be less than 5 the test fails and the GOTO is ignored. The program moves to the next line, displays the value of *count*, adds one to it and then comes to line 90.

This is an unconditional GOTO. When and if the program reaches this line it has no choice but to loop back to the start of the loop.

Here it is immediately subjected to another test for the exit condition and the program will continue looping or drop out of the loop depending on the outcome. Figure III shows the flow of control involved.

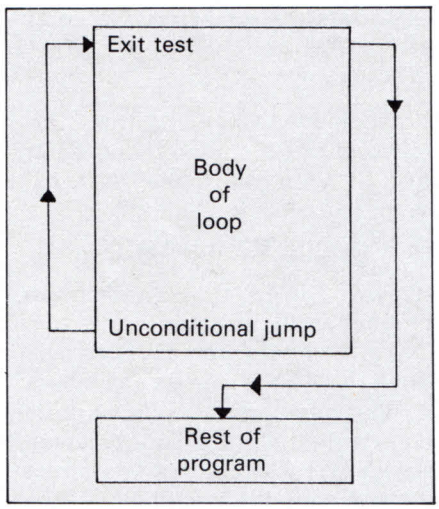


Figure III: A WHILE . . . WEND loop

As it is, Program IV displays the numbers 1, 2, 3 and 4, then tells you that the loop has finished as the final value is 5. Try changing the values of *endCondition* and *startValue* and see the effects on the loop.

One thing to notice is that with this type of loop the test for the exit condition is performed right at the beginning of the body of the loop.

If the condition is found to be true then that's that, the loop is effectively ignored. Try changing Program IV so that *startValue* is 5 or more and you'll see this for yourself. The loop is avoided.

Full marks if this reminds you of a WHILE . . . WEND loop because it is one. Program V is just a version of Program IV using WHILE . . . WEND. I'll leave it to you to decide which is the better method.

Notice that the condition seems to be reversed. Program V continues looping while the condition *count*<5 is true. However this is just the same as saying that the loop stops when

→

```
10 REM Program V
20 endCondition=5
30 startValue=1
40 count=startValue
50 REM Start of loop
60 WHILE count<5
70 PRINT "count is now ";count
80 count=count +1
90 WEND
100 REM End of loop
110 PRINT
120 PRINT "count is now ";count
```

Program V

the condition *count*>= is true. The two conditions complement each other. If you don't see that try changing line 60 to:

```
60 WHILE NOT (count)=5)
```

and you'll find that there's no difference in the results.

In a way you can look on the WHILE . . . WEND loop as an endless loop that has an exit condition at its start – if that's not a contradiction. Another type of loop which has its exit condition after the main body of the loop sometimes comes in useful. Program VI is one.

```
10 REM Program VI
20 endCondition=5
30 startValue=1
40 count=startValue
50 REM Start of loop
60 PRINT "count is now ";count
70 count=count +1
80 IF count)=5 GOTO 100
90 GOTO 50
100 REM End of loop
110 PRINT
120 PRINT "The program has dropped ou
t of the loop"
130 PRINT "count is now ";count
```

Program VI

The first few lines are the same as Program V, it's the loop that's different. It starts at line 50 with the labelling REM and the next line displays the current value of *count*. After that *count* is increased by one. Then comes the loop exit condition.

This is found in line 80. If the condition is true – which occurs when *count* is 5 or more – the GOTO sends the program to line 100, which is the end of the loop. Line 90 isn't obeyed, it's skipped over.

On the other hand if the condition isn't true, such as when *count* is 1, 2, 3 or 4, the GOTO in line 80 is ignored. The program goes on to the next line, line 90, where it promptly meets another GOTO.

This one is unconditonal, and once the program's reached this line it is forced to go back to line 50, which is the beginning of the loop.

Figure IV shows the flow of control involved in this kind of loop. It only ends when the exit condition is true, forcing the program to skip the unconditional jump that is the backbone of the loop.

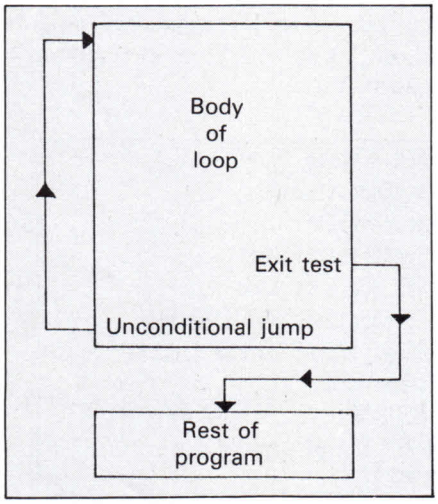


Figure IV: A REPEAT . . . UNTIL loop

There is a specific term for these kinds of loops where the exit test is at the end of the main body of the loop. They are known as REPEAT . . . UNTIL loops.

Some Basics, such as BBC Basic, have REPEAT . . . UNTIL loops, though they don't have the WHILE . . . WEND variety. Amstrad Basic doesn't have this, but Program VI shows you how to go about constructing one if you wish.

REPEAT . . . UNTIL loops have one interesting property. Because the exit condition lies at the end of the main body of the loop the loop is always performed at least once. Try giving *startValue* a value of 5 and you'll see what I mean.

Have you noticed anything about our loops? We've had loops that exited at the beginning and loops that could be left at the end. But so far we haven't had one that could be left in the middle. Why not?

The answer is that they are more trouble than they're worth. If you get your program to leave a loop in the middle then obviously half that loop hasn't been obeyed. This may not be important but at times, especially in longer, more complicated programs, it can lead to silly results.

```
10 REM Program VII
20 FOR loop=1 TO 10
30 PRINT "Give me a number 1-10"
40 INPUT number
50 PRINT "number is "number"
60 IF number=5 THEN GOTO 120
70 double=number*2
80 PRINT "double is ";double
90 NEXT
100 REM Normal end of FOR...NEXT loop
110 PRINT
120 PRINT "The loop has ended"
130 PRINT "number is finally ";number
140 PRINT "double is finally ";double
```

Program VII

Our final program, Program VII, is a case in point. This just loops 10 times. Each time you give it a number, and the program displays the number and also doubles it and displays that.

Line 60 is the snake in the grass. Here the programmer has obviously decided that if one of the input numbers is 5 the program is to leap out of the loop and stop. Try it and see what happens.

Silly isn't it? I always though that twice 5 was 10. Notice that it's an intermittent mistake. So long as you keep away from 5 the program works perfectly.

As it is, it's easy to see what the mistake is, but in real life these intermittents have seen the end of many a good programmer.

So don't jump out of loops in the middle – you may get away with it for a while but one day your sins will catch up with you.

● *And that's where we'll end this time. Play around with the various loops we've used, try out joint conditions and see if you can devise your own loop structures (WHILE . . . UNTIL?). It should keep you busy until next time, when we'll be looking at more routine Basic.*

Exercise your imagination in Sweevo's World

With help from ALEATOIRE

ONE of the objections to many computer games is that you are just playing against a random number generator which is mainly testing your reflexes rather than your intelligence, and in the real world all the important decisions are made by intelligent, logical analysis not by random guessing.

In real life however, the most important decisions can be made for a mixture of logical and random reasons and this combination is often the mark of a good computer game.

Another reason for condemning games is that they are unsociable – the player should be outside in the fresh air circuit training or running a marathon. The same arguments have been made against reading, and obviously some balance should be made – believe it or not you need a deal of both mental and physical stamina to sit and concentrate at a desk for hours at a time.

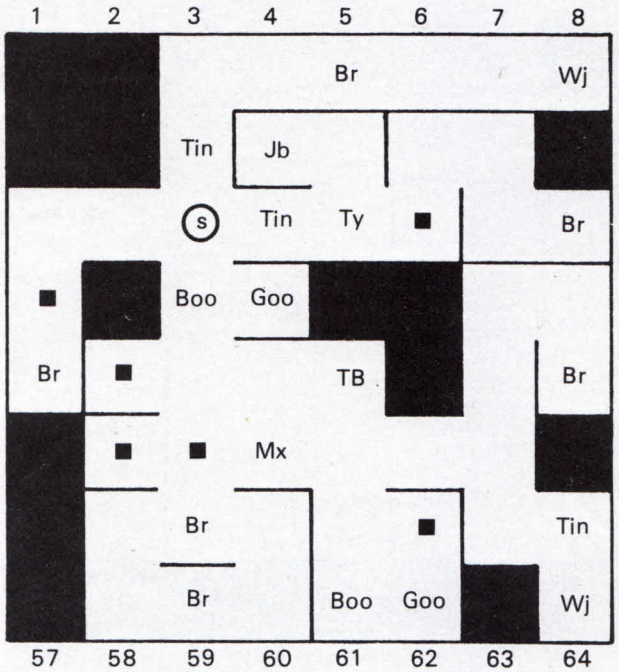
The main criticism is that until recently the majority of computer games were rubbish. However nowadays there are some ingenious and stimulating games on the market. Perhaps the best defence is to pick out and analyse what I consider one of the best games of 1985, Sweevo's World by Gargoyle. Like all good puzzles you are not told what you have to do, but it is fairly obvious that you must dispose of five different character types.

These are Brownies (16), Geese (8), Minxes (4), Tyrants (4) and Wijus (8).

Exactly how you do this is for you to find out. Some of the methods appear silly, but the game is cleverly designed on four levels of 50 rooms each. You can choose which level you start at, but until you have mapped the game you have no chance of solving it. Note, for example, that the only way to reach the Wiju in room 18, level 2 is via an underground tunnel.

Having made a map you then have to work out the most efficient and/or safest way of getting around and through the 200 rooms. Some of the time this requires observation and good timing but quite often – and I suspect this is because the designers of the game are middle aged – a bit of

Level 4

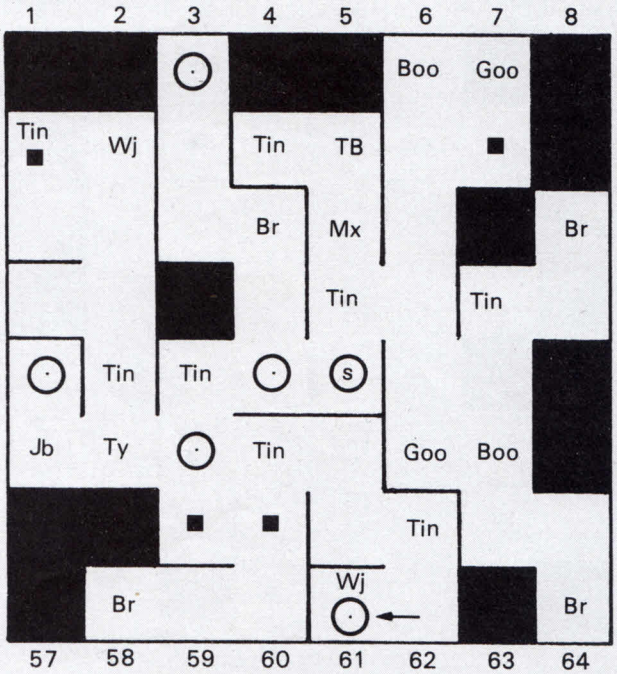


To reach the six Brownies:

Room

5 Drop tin on hidden lift.
24 Use R.H. lift and walk over three tons.
33 Use lift on Level 3.
59 & 40 Timing involved.
51 Use tins to move (not slides).

Level 3



To reach the four Brownies:

Room

20 Via lift on level 2.
24 Drop tin, two tons and walk over.
58 Timing.
64 Need two tins.

analysis will reveal a safe and simple method. For example, in room 51, level 4 it appears that you must walk over a hazardous sliding block, but a totally safe method is to drop and walk over a couple of tins.

The game is full of these deft twists where good reflexes and timing might get you through, but a bit of analysis reveals a safer alternative for the more mature player.

It took me about three weeks to solve the game. The main problem was that I did not find the last Brownie (hidden in room 15, level 2) until Greg Follis, one of the authors, told me about it.

If you have now been told to "report for duty at the Dust Bowl" you may also like to know that although 26 points for tidiness is enough to win it is possible to score 34 points by eliminating the Boos as well.

You will notice that the characters move around randomly but always start from the same position – all you have to do is say Boo to a Goose in the same corner that it starts. If you then leave and re-enter the room the egg and the associated Boo disappear.

Solutions to last month's puzzles

Listing III

There are 11 ways of writing down the digits 1 to 9 in order and interspersed with + or – so that the result equals 100. They are:

123+45–67+8–9
123–45–67+89
123+4–5+67–89
123–4–5–6–7+8–9

12+3+4+5–6–7+89
12+3–4+5+67+8+9
12–3–4+5–6+7+89

1+23–4+56+7+8+9
1+23–4+5+6+78–9
1+2+34–5+67–8+9
1+2+3–4+5+6+78+9

Listing IV

The following is amazing but true:

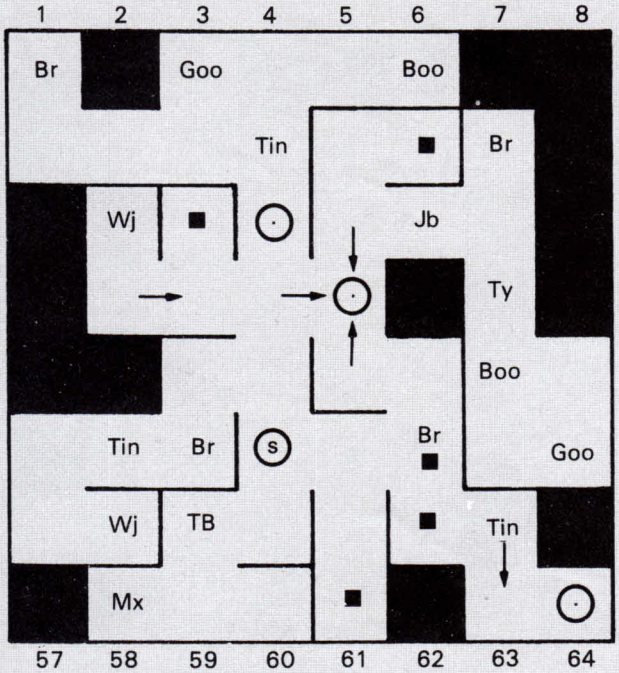
148/296+35/70
14586/7293
17469/5823
31824/7956
14865/2973
17658/2943
36918/5274
74568/9321
75249/8361

Boo Drop this behind Goo
Br Brownie. Just get close.
Goo Goose. Killed by ton weight.
Jb Jack boot.
Mx Minx. Killed by TB.
TB Teddy bear.
Tin Tin used to climb up or over.

Ty Tyrant. Killed by Jb.
Wj Wiju. Killed by finger.
← One way door.
■ Hole to floor below.
⊙ Lift to floor above.
— Wall.
(S) Start for that level.

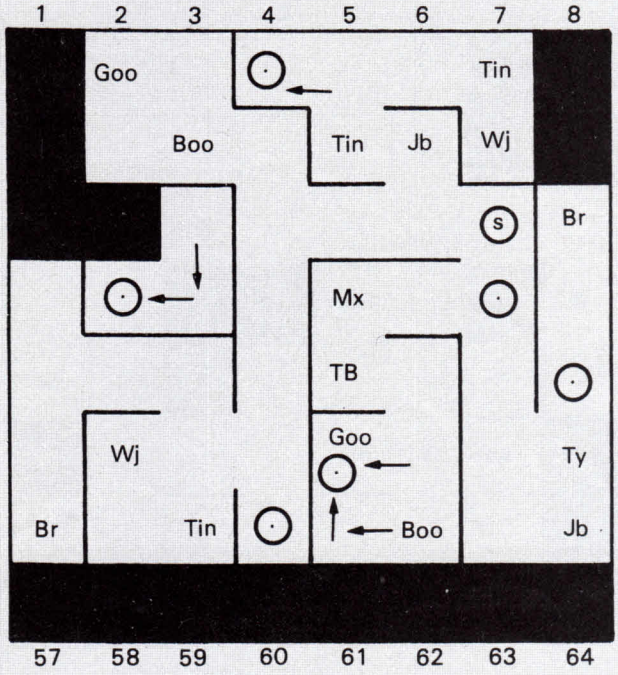
Key to Map of Sweevo's World.

Level 2



To reach the four Brownies:
Room
1 Timing.
15 Hidden.
46 Drop ton then tin (in skull).
43 Line up (thread).

Level 1



To reach the two Brownies:
Room
24 Need tin.
57 Drop tin, then ton weight and walk over.

Random reflections

JUST how random are the Amstrad's random numbers? Suppose the RND function had 100 goes at picking between 1, 2, 3 and 4. If they were equally likely you'd expect 25 occurrences of each. However run this month's program and you might change your ideas.

10 Labels the program.

20 Calls the subroutine that sets the initial conditions for the program to run.

30-50 Form a FOR . . . NEXT loop that cycles *trials* times.

40 Sends the program off to the subroutine that produces and counts the random numbers. This routine will be called *trials* times.

60 Draws the bar chart that summarises the results once the loop has finished.

99 Stops the program crashing into the subroutine definitions. Leave this out and see what happens.

100-199 Form the initialisation routine. All four running totals – used to count the number of times 1, 2, 3 and 4 are thrown up by the random number generator – are set to zero. This isn't strictly necessary but it is good practice.

110 *Scale* is used to decide how long each bar in the bar chart will be. Initially six graphics coordinate points are used for every one occurrence of a number.

160 *Trials,* used to decide the number of random numbers that the FOR . . . NEXT loop will generate, is set to 100.

200-299 Generate a random number and keep a tally of each of the four possibilities.

210 The expression produces a whole number in the range 1 to 4 and stores it in the temporary numeric variable *number.*

220 Has *number* running the gauntlet of three nested IF . . . THEN . . . ELSE statements. Depending on the value of *number* the relevant total is increased by one.

300-399 Is performed after the program drops out of the main FOR . . . NEXT loop. By this time the four *total* variables will contain the tallies for the number of occurrences of 1, 2, 3 and 4 in 100 trials. Three further subroutines are called to display these results.

400-499 Draw the two axes. Notice that the vertical axis is 300 (350-50) graphics coordinates in length.

500-599 Create the bars, one for 1, 2, 3 and 4 respectively. The length of each bar is directly proportional to the number of times that that particular number occurs.

510 Moves the graphics cursor to the point on the horizontal axis where the 1 bar will start.

520 A vertical line is drawn, its length representing the number of 1s generated in *trials* loops. The Y coordinate is made up of an offset of 50 – to start the line on the axis – added to the number of 1s multiplied by the scaling factor.

It's just possible that there will be 100 1s, and if this is the case the line would be 600 graphics coordinates long and go off the screen.

As it is, the vertical axis will only cover 50 (300/6) occurrences but this is nearly always enough. I leave it to the statistics freaks to figure out the chances of the scale being exceeded.

530-580 Move along the horizontal axis producing the lines for the other three numbers.

600-699 Number the axes.

605 Joins the text and graphics cursors. In effect you can use MOVE to decide where to print.

610-640 Form a FOR . . . NEXT loop which moves along below the horizontal axis, labelling each bar.

620 In the expression the 90 is the distance apart of the bars, to which an offset is added to start the count at the origin – 100,50. This offset is 84, not 100 as might be expected. Try 100 and you'll see that the figures are slightly out of line.

630 Prints the number at the graphics cursor. The semicolon is needed to suppress the usually automatic carriage return and line feed. Leave it out and you'll see them being treated as graphics control codes with weird results.

650-660 Label the highest value of the vertical axis. As the value of *scale* is changed the number that appears is automatically changed to the correct maximum value.

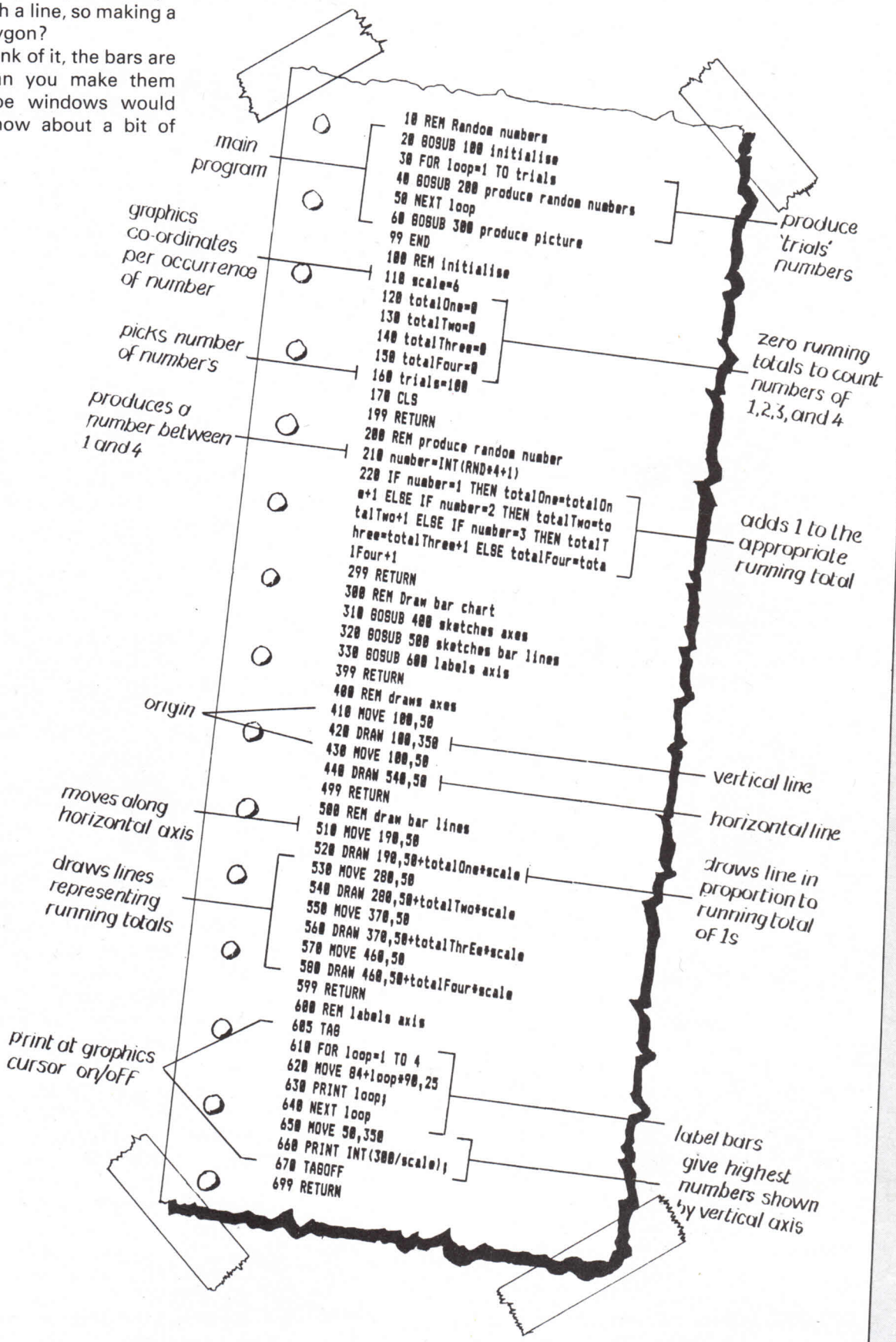
670 Undoes the TAGging.

Amstrad Analysis

by Trevor Roberts

Things to do: Why not try different values of *scale* and *trials*, such as 1 and 600 or even 1 and 1000? Also why not join the tops of the bars with a line, so making a frequency polygon?

Come to think of it, the bars are a bit thin. Can you make them thicker? Maybe windows would help. Finally how about a bit of colour?



Load up the stack and get things moving

Part XVI of MIKE BIBBY's guide to machine code

OVER the past two months we've been learning how to use the stack, an area of memory set aside by the Z80 as workspace.

We saw that we could PUSH the register pairs on to the stack for storage, and then POP them back again into the registers. The last pair pushed was the first pair popped so if we were to:

```
PUSH DE
PUSH BC
 POP DE
 POP BC
```

the contents of BC and DE would be swapped.

We also saw that the Z80 uses the stack to remember the address of the instruction to return to after doing a subroutine.

The RET instruction at the end of the subroutine POPs this address off the stack automatically and loads it into the Program Counter (PC), so this is the next instruction done.

This means that if we do any pushing in a subroutine so covering up this vital return address, we have to do an equal number of POPs to get rid of them and restore it to the top of the stack.

The Z80 itself keeps tabs on the area of stack memory with the Stack Pointer (SP). This is a 16 bit register containing the address of the last byte pushed on the stack. That is, the SP points to the end of the stack.

Program I lets us see where the SP is currently pointing by using the instruction ADD HL,SP (opcode &39). This adds what's in HL and the value of the stack pointer together. Since the first line loads HL with zero, the result – which is stored in HL – will simply be the current value of the stack pointer.

The program then stores HL in the first two bytes of Hexer's workspace – at &7FF8 – lo byte, hi byte fashion. So by looking at &7FF8 we can find the value of the stack pointer during Program I.

I ran the program using our Hexer/Raw combination described in the March and July 1985 issues of *Computing with the Amstrad.* If you're following the series I suggest you get this duo typed in.

On examining the workspace I saw that the stack pointer when I'd called the routine was &BFF8, stored as F8 BF. When you run the program your value might be slightly different, although it will be in the same region.

This is because Basic and the operating system also use the stack, so where your pointer is at present could be a combination of past history, the machine you're running it on and the way you call the code.

However whatever the value of the stack pointer the same thing applies – it's pointing at the last byte put on to the stack.

To see how the stack works try Program II. This loads HL with zero then pushes it on to the stack. This is an arbitrary value as I just want to put something on the stack and see how the pointer moves.

We then do our ADD HL,SP trick to get at the stack pointer. After I've got the stack pointer safely into workspace via HL, I then POP HL to restore the stack to normal ready for the RET.

When I ran this, the SP after pushing HL was &BFF6. That is, the stack pointer moved down in memory by two bytes when we pushed HL on to the stack.

This is an important discovery – as

```
RAW Assembler V.3

Pass... 2           ORG &8000

8000:               .workspace=&7FF8

8000:21 00 00       LD HL,00
8003:39             ADD HL,SP
8004:22 F8 7F       LD (workspace),HL
8007:C9             RET
8008:               END
```

Program I

```
RAW Assembler V.3

Pass... 2           ORG &8000

8000:               .workspace=&7FF8

8000:21 00 00       LD HL,00
8003:E5             PUSH HL
8004:39             ADD HL,SP
8005:22 F8 7F       LD (workspace),HL
8008:E1             POP HL
8009:C9             RET
800A:               END
```

Program II

you PUSH bytes on to the stack it grows downwards in memory. Figure I shows what's happening.

Notice how the bytes of HL end up in proper lo byte, hi byte order when they're pushed. This happens when we PUSH a register pair, or automatically when we CALL a subroutine. We'll prove this with Program III.

All Program III does is to call a subroutine, aptly named *routine*, and then RETurn. The clever bit is *routine*, which looks at the address stored on the stack when the subroutine is called.

According to our theory this should be the address of the instruction after the call: &8003 which is, coincidentally, our terminal RET. To find out what's on the stack *routine* finds the address in the SP by using the:

```
LD HL,00
ADD HL,SP
```

trick. SP now points to the last byte stored on the stack. We get this into the E register with:

```
LD E, (HL)
```

However the CALL pushed two bytes on to the stack, so let's get the other one in the D register. Remembering that the stack grows downwards in memory, the other byte that was pushed must be in the byte above HL. We get HL to point to this with:

```
INC HL
```

and then we can load it into D with:

```
LD D, (HL)
```

We next transfer both bytes into our workspace at &7FF8 with:

```
LD (workspace),DE
```

If you remember how this type of load works this will put the lower byte of the stack (in E) into &7FF8, and the higher byte (in D) into &7FF9. Looking at &7FF8 with something like Hexer you'll see the relevant memory locations are:

```
03 80
```

That is, the stack really does contain &8003, the address of the next instruction in lo byte, hi byte form, ready for use when it meets the RET of *routine*. The same happens when we PUSH a register pair.

To prove that the stack builds downwards as we think it does try Program IV. Here we PUSH the contents of HL – whatever they may be – on to the stack. This of course covers up the return address already on the stack, put there when we CALLed the routine via Hexer or whatever.

If we simply came to a RET we'd pull the contents of HL off the stack as

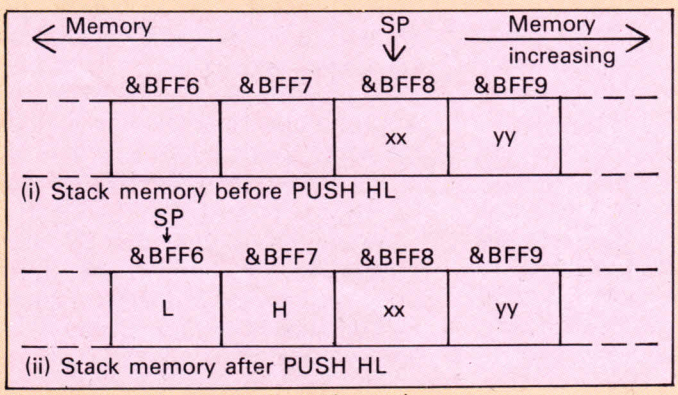


Figure I: How Program I affects the stack

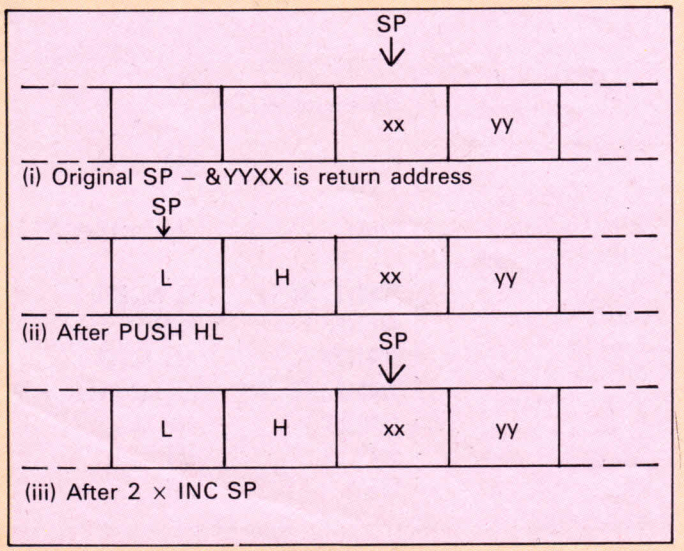


Figure II: How Program IV affects the stack

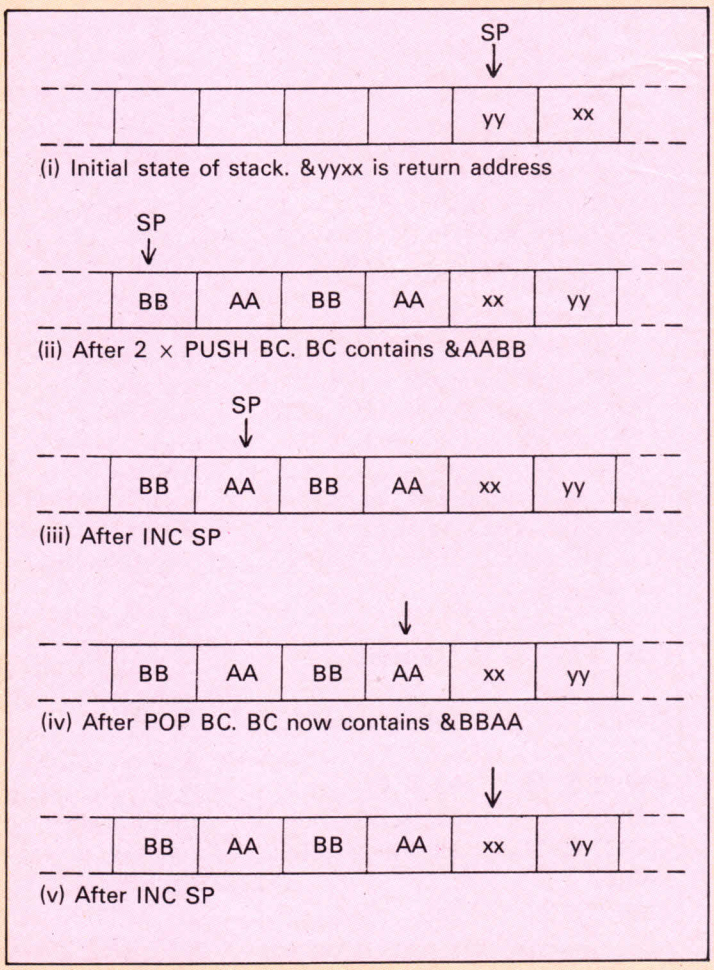


Figure III: How Program V affects the stack

```
RAW Assembler V.3

Pass... 2          ORG &8000

8000:              .workspace=&7FF8

8000:CD 04 80      CALL routine
8003:C9            RET
8004:              .routine
8004:21 00 00      LD HL,00
8007:39            ADD HL,SP
8008:5E            LD E,(HL)
8009:23            INC HL
800A:56            LD D,(HL)
800B:ED 53 F8 7F   LD (workspace),DE
800F:C9            RET
8010:              END
```

Program III

the return address and get our next instruction from there – probably crashing the system while we're at it.

However there's an instruction – INC SP (opcode &33) – which increases the stack pointer. That is, it makes it point to the next higher byte in memory.

In Program IV after we've pushed HL we INC SP twice, thus moving our stack pointer back over the two bytes we've pushed on so that it now points to the lo byte of the return address.

Effectively we've used INC SP to nullify the effect of the PUSH so when the RET comes along and gets the return address from where the SP is pointing it gets the correct address. Figure II shows what's going on.

If you take things carefully stack manipulation can be very powerful. However it pays to sketch out what you're doing with pencil and paper to see if it really works as you think it does. Making a mistake with the stack can cause spectacular crashes.

One standard trick with the stack

```
RAW Assembler V.3

Pass... 2          ORG &8000

8000:E5            PUSH HL
8001:33            INC SP
8002:33            INC SP
8003:C9            RET
8004:              END
```

Program IV

```
RAW Assembler V.3

Pass... 2          ORG &8000

8000:              .workspace=&7FF8

8000:01 BB AA      LD BC,&AABB
8003:C5            PUSH BC
8004:C5            PUSH BC
8005:33            INC SP
8006:C1            POP BC
8007:33            INC SP
8008:ED 43 F8 7F   LD (workspace),BC
800C:C9            RET
800D:              END
```

Program V

pointer is to use it to swap the contents of a pair of registers. Program V does this.

Initially we load BC with &AABB – an easily recognised marker pattern – then we push it on to the stack twice. If you look at Figure III you'll see that at this stage the stack reads BB AA BB AA.

Remember B goes on the stack before C. What we want to do is somehow POP the AA BB part of the pattern into BC. To do this we next INC SP so that it now points to the first – which, confusingly, was the last to go on to the stack.

After that POP BC will give you &BBAA in BC and moves the stack pointer to the last AA. Of course we don't want this to confuse our final RET, so we once more INC SP to point to our return address.

```
RAW Assembler V.3

Pass... 2          ORG &8000


8000:11 FF FF      LD DE,&FFFF
8003:21 00 00      LD HL,0
8006:39            ADD HL,SP
8007:31 00 80      LD SP,&8000
800A:D5            PUSH DE
800B:D5            PUSH DE
800C:D5            PUSH DE
800D:D5            PUSH DE
800E:F9            LD SP,HL
800F:C9            RET
8010:              END
```

Program VI

The result of all this is that the contents of BC have been changed from &AABB to &BBAA. That is, we've swapped the B and C registers.

Now the Amstrad CPC stack uses &BFFF downwards. However the Z80 lets us move our stack to wherever we want by directly loading our stack pointer with a two byte value using LD SP,nn (opcode &31).

Program VI changes the stack to &8000, then pushes &FFFF onto it four times. Remembering that the stack grows downwards the first push puts &FF into both &7FFF and &7FFE, leaving SP pointing to &7FFE.

The next push puts &FF into &7FFD and &7FFC and so on. The result of all four is to put &FF in the bytes from &7FFF down to &7FF8. Here's how it works in detail:

LD DE,&FFFF	Fill DE with &FFFF.
LD HL,0	Zero HL ready for.
ADD HL,SP	Now our stack pointer is saved in HL. We'll need it to get our return address from when we've finished fiddling with the stack.
LD SP,&8000	Our new stack. From now on PUSHes will occupy from &7FFF downwards.
PUSH DE x 4	Fills up &7FFF to &7FF8 with &FF.
LD SP,HL	Restores our stack pointer so that
RET	fetches the correct return address from the original stack.

● *Right, that's all for now. Next month we'll meet some new registers and see how to print out strings of characters.*

Studying trees with Dr Logo's assistance

AS an owner of a CPC464 who recently bought the DD1 I was delighted to discover that the Dr Logo language provided on the CP/M systems disc lent itself very well to the modelling of plant growth.

Though not as spectacular as the system shown on the BBC's Micro Live, models of considerable scientific value may be developed which manifest themselves in diagrams of trees possessing different branching angles and relative internode lengths found in nature.

Various effects of the environment on the growth and final habit may also be simulated by randomly or systematically killing or limiting the growth of some branches or altering the probability at which branching will occur from a given node.

If you're only interested in elegant and interesting graphics Amsoft's "Guide to Logo" by Boris Allan contains a chapter which describes how to draw trees.

This brings me to my main point – appendix D of the guide describes how machine code extensions may be added to Logo by lowering the top of the CP/M system allowing an RSX to be inserted into the space provided.

This may then be called from Logo by redefining a Logo primitive with pprop and .PRM, so that whenever that primitive is used the RSX is called.

The most obvious RSX was a screen dump and I loaded one into the space. But since it involved calls to the firmware, and CP/M has altered the location of the firmware, it didn't work.

I am not much of a machine code programmer, but since you are doing a series on CP/M could you explain where the firmware has gone and how to get around the problem described above?

Better still, since many people would probably like hard copies of their Logo drawings could you publish a screen dump that will work when called from Logo? – **Richard M. Woodfin, Bath.**

● To install an RSX under CP/M you must first create space for it between the firmware variables and the BDOS – normally, these are contiguous.

You do this by running the program MOVCPM to alter the system size – the amount of memory CP/M is allowed to use. The BDOS loads at the top of this system memory, so by reducing it you will free space, which CP/M will not use, above the BDOS. Type:

```
A>MOVCPM nnn *
```

where nnn is the system size in pages (1 page = 256 bytes). The standard system supplied is 179 pages in size, so if your RSX is 500 bytes long, (just under two pages) then you would reserve space for it by executing:

```
A>MOVCPM 177 *
```

Once you have done this use SYSGEN to copy the system image MOVCPM has created in memory onto a disc by typing:

```
A>SYSGEN *
```

The asterisk tells SYSGEN to read the system from memory instead of a source disc.

You then have to write a program to load the RSX routines into the space you have created in memory. The base of the firmware variables is at 0AD33h. Your RSX space is created immediately below this and extends down to the BIOS jump table at the top of the BDOS (which you lowered with MOVCPM).

CP/M does not alter the existing firmware routines in the CPC464. However, we have to call the jumpblocks in a different way. For example, where under AMSDOS we would have used the sequence:

```
ORG   &1000
.txt-output
EQU   &BB5A
LD    A,"A"
CALL  txt-output
RET
```

to print the letter A on the screen, under CP/M we must do it using an extra routine – ENTER_FIRMWARE as follows:

```
ORG   &1000
.txt-output
EQU   &BB5A
.enter-fware
EQU   &BE9B
LD    A,"A"
CALL  enter-fware
DEFW  txt-output
RET
```

ENTER_FIRMWARE must be used for all firmware calls under CP/M in order to avoid corruption of system variables which are held in the alternate register set.

On a more down-to-earth note, when you are assembling RSX routines make sure that you have ORGed them for the addresses at which they will load, that your loader program is putting them in the right places, and that you have reserved enough space for them with MOVCPM.

If you haven't, the routines will overwrite firmware variables or jump tables and the computer will start to behave oddly.

Helpful First Steps

AS a complete beginner I have benefited enormously from First Steps by Mike and Pete Bibby and from Al McLachlan's REMarks, and as an equally new adventurer, from the contributions of one of your newer writers, Gandalf.

I must admit that a lot of the articles over the past year are still beyond me at the moment, but I am sure that they are greatly appreciated by non-beginners.

The Software Surveys have always been interesting although, no doubt, I am not the only one who would have liked to have seen reference numbers in your now discontinued "Reviewed so far" section.

While I know that it is akin to using a sledgehammer to crack a walnut I now have them on the database from Mini Office.

Many thanks for an extremely well-presented and informative first 12 issues. – **Aubrey Sinden, Rye.**

● The "Reviewed so far" panel was OK while it was small, but it has had to go to make space for more reviews.

Screen surprises

ALLOW me to promote the nice routine at &BC44 SCR FILL BOX Fill a character area of the screen as it can embellish many screen layouts especially when using a green monitor in Mode 2.

Surprising effects can be obtained when applied as a background with a respectable choice of 256 different patterns. And in combination with the transparent mode – chr$(22)"1" – you can in some cases overwrite this background paper with both PEN 0 and PEN 1 in Mode 2, as if a third colour had been intro-

duced, without diminishing the readability too much.

It's just a matter of trial and error until the best one for your application shows up.

```
ORG &8000
LD A,170     ;pattern num
             ;or coded ink
LD DE,20243 ;D=right col
             ;E=bottom row
LD HL,3      ;H=left col
             ;L=top row
JP &BC44     ;fill area
END
```

The area in this example is the window 1,80,4,20 and the A register is initialised with the coded ink 170. This can be understood for Mode 2 as the bit pattern: 10101010.

When the alternation is regular it results in a smooth plane but when, for instance, A is loaded with 42 the jumpblock produces a vertical stripey pattern. Note also that the execution speed is remarkable.

Here's another interesting little diversion. Locomotive Basic has only string concatenation built in and you achieve it using the + operator. Other existing operators like *, – and / are not catered for.

The next Basic program shows in just a few lines how these three can also be of interest:
– takes the characters of the first stringset except when also belonging to the other set.
* takes only the common characters of the two stringsets.
/ takes the unison of both sets but deleted with the common characters of the two sets.
+ takes the unison of both stringsets.

If its meaning is still obscure then take a closer look at the following examples:

```
10 MODE 2:q$=CHR$(34):WHILE
 TIME
20 INPUT"First  character s
tring : ",a$
30 INPUT"Second character s
tring : ",b$
40 INPUT"Operator type (*,+
,-,/) : ",o$
50 ON INSTR("*+-/",o$)+1 GO
SUB 70,90,140,90,160
60 PRINT:PRINT q$a$q$" "o$"
 "q$b$q$" = "q$Lc$q$:PRINT:
WEND
70 Lc$="Illegal operator":R
ETURN
80 REM * and -
90 La$=a$:Lb$=b$:Lc$="":FOR
 x=1 TO LEN(La$):y=INSTR(Lb
$,MID$(La$,x,1))
100 IF y THEN Lb$=LEFT$(Lb$
,y-1)+RIGHT$(Lb$,LEN(Lb$)-y
)
110 IF (y AND o$="*") OR (y
=0 AND o$="-") THEN Lc$=Lc$
+MID$(La$,x,1)
120 NEXT:RETURN
130 REM +
140 Lc$=a$+b$:RETURN
150 REM /
160 o$="*":GOSUB 90:c$=Lb$:
o$="-":GOSUB 90:Lc$=Lc$+c$:
o$="/":RETURN
```

In practice:

```
database - amstrad = base
amstrad - database = mr
nixon * waddilove  = io
basic / pascal     = bipal
add + on           = addon
az za - z z        = aa
```

Preceding and trailing blanks remain undetected in my version. Normal and capital letters are distinguished.

The longer the sets are the slower is the outcome but this can be overcome when translated in machine code. – **Patrick De Geest, Belgium.**

Listing barrier

IN the July 1985 issue of Computing with the Amstrad you directed Mr Cannasso to enter a line 1 REM *** and then in direct mode to type in POKE &176,255 to stop people from listing his programs.

This is fine if you type LIST in directly, but if you type LIST 10–, or LIST 10 then the program or line is listed. The following will stop this:

```
1 REM ***
POKE &171,1 protect
POKE &171,0 unprotect
```

This prints line 1 and then stops.

To use an input, where no writing appears on the screen, use the following:

```
10 PRINT "Enter a number";
20 PRINT CHR$(21);:REM turn
s off text screen
30 INPUT n
40 IF n=7 THEN PRINT CHR$(6
);:END
50 GOTO 50
```

When RUN, anything you type will not appear but will still be held by N. If you type in 7 the text screen will be reactivated but if you type in any other number it won't. – **David Arnold, King's Lynn, Norfolk.**

Lunar correction

I REFER to the Easter Sunday program (Gauss' formulae) published in the April issue of Computing with the Amstrad. In the accompanying article Aleatoire wonders when Gauss' rule breaks down.

We have no real need to panic. The rule gives the correct date for the next 2,214 years. It starts to go wrong in the year AD4200!

Gauss' rule ignores the fact that a lunar correction was made in the year 1800 at the start of a 2,500 year lunar correction cycle.

Ignoring the change from the 300 year interval to the 400 year interval in the lunar correction gives April 13 instead of the correct April 20 for Easter in 4200.

In all, 24 errors occur in the 43rd century. Similar errors re-appear every 300 years thereafter.

I offer the following program as a correct alternative for those with Amstrads still going in the 43rd century:

```
10 DEFINT A-Z:INPUT "Enter
the year";x
20 a=x MOD 19:b=INT(x/100):
c=x MOD 100
30 d=INT(b/4):e=b MOD 4:g=I
NT((8*b+13)/25)
40 h=(19*a+b-d-g+15) MOD 30
50 u=INT((11*h+a)/319):i=IN
T(c/4)
60 k=c MOD 4:l=(2*(e+i)-k-h
+u+32) MOD 7
70 n=INT((h-u+l+90)/25):p=(
h-u+l+n+19) MOD 32
80 PRINT "Easter Sunday";x;
" = ";
90 IF n=3 THEN PRINT "March
";p:END
100 PRINT "April";p:END
```

Awkward invaders

I AM having trouble with your Galactic Invaders game that I typed in from the May issue of

Computing with the AMSTRAD Postbag

WE welcome letters from readers – about your experiences using the Amstrad, about tips you would like to pass on to other users . . . and about what you would like to see in future issues.

The address to write to is:

POSTBAG EDITOR
COMPUTING WITH THE AMSTRAD
BOX 5000
GLEN WAVERLEY
VICTORIA 3150

Computing with the Amstrad.

When I ran it a line check routine appeared on the screen and later announced 'Line 3380 is incorrect'.

I checked this line but found it to be identical to the magazine listing. After all it couldn't be simpler:

```
3380 DATA 00,00,00,00,
00,00,00,00,00
```

I don't know what the information conveys, nor do I really care, but I do know the characters are all zeros. Can you help me to find out what I've done wrong or is your program faulty?

Also in Ice Front I keep getting 'Improper argument' at line 2220. The program is OK when I first run it but when I Escape and run again the error appears. – **Paul Hindmarsh, Knutsford, Cheshire.**

● Your first problem is a splendid example of how the Amstrad's error reporting routine can fool you into believing that a typing error you have made is in one line when in fact it's in another.

Here are the lines in your version before and after the alleged error line:

```
3370 DATA 00,00,00,00,
00,00,00.00,00
3390 DATA 0f,02,28,00,
80,00,28,00,e1
```

Line 3390 is perfectly OK, but in case you haven't spotted it, you've actually typed in a full stop instead of a comma between the seventh and eighth items of data in line 3370.

Robin Nixon's error check routine, in simple terms, adds together the hex numbers in the first eight data items of each line, then compares the total with the number shown as the ninth number of the line. If these are not equal, the error message results.

As far as your micro is concerned line 3370 contains only eight zeros. The 00.00 counts as only one and the total, of course, is zero.

The number this is compared with for the total – what would have been the ninth number on line 3370 – is in fact the first number in line 3380 – 00. They are equal, line 3370 seems OK.

Next line 3380 is checked, now starting at the second number. It adds up the eight remaining zeros – 0 naturally, but now compares this with the first number of line 3390 – 0f.

They are unequal, hence the error message, but as far as the micro is concerned it is line 3380 that's at fault, not line 3370.

This is well worth bearing in mind for future reference.

There is another little problem with this program. A few people have been experiencing difficulties in so far as when they Escape and try to run it they keep getting DATA errors.

Unfortunately pressing Escape deletes some of the data, and as yet we haven't found out why.

We must regard this as a bug of course, and the only way to avoid it when developing the program is to save it before you run it, then reload the saved version to make any modifications.

You should always do this with a machine code program anyway, as it is always capable of doing strange things.

We'll inform you immediately the fault is discovered, whether we find it ourselves or one of our whizz kids out there finds it first.

Your problem with Ice Front is the result of a bug (sorry, feature) of the Amstrad's operating system. When it encounters the SYMBOL AFTER command the second time, after HIMEM has been altered, it can't handle it and the program crashes.

One solution is to add the following lines that use the error generated to bypass the offending line.

```
2215 ON ERROR GOTO 2225
2225 RESUME 2230
```

Opcodes recognised

THANKS for your article taking the mystery out of the so called hidden opcodes of the Z80.

I have adapted Roland Waddilove's disassembler to recognise all these codes by adding just two lines and changing one line.

*To add the SLL commands: Change "***" in line 960 to "SLL". To add the XH, XL, YH and YL commands add the two lines:*

```
295 IF class=0 AND index>0
THEN r$(4)="H":r$(5)="L"
375 IF class=0 THEN IF inde
x>1 THEN IF NOT byte=&74 OR
 byte=&66 OR byte=&6E THEN
IF index=1 THEN r$(4) ="XH"
:r$(5)="XL":ELSE IF index=2
 THEN r$(4)="YH":r$(5)="YL"
```

I use Roland's Disassembler all the time and I have only found one bug in it – which is much better than some commercial programs.

The bug is that it will not correctly disassemble the codes CB,DD (SET 3,L) or CB,FD (SET 7,L). This can be corrected by changing two lines:

```
360 IF byte=&DD THEN IF CLA
SS<>1 THEN index=1:GOTO 320
 ELSE 380
370 IF byte=&FD THEN IF CLA
SS<>1 THEN index=2:GOTO 320
 ELSE 380
```

– **D. Instone Brewer, Roath, Cardiff.**

● Thanks for the contributions, particularly the one pointing out the bug in Roland's program – we like our readers to keep him on his toes.

But just to have the last word, Roland says that your modifications don't quite achieve what you anticipated. Add the following line and all will be well.

```
350 IF byte=&ED THEN IF CLA
SS<>1 THEN class=2:GOTO 320
 ELSE 380
```

Similarly your attempts to add the XH, XL, YH and YL commands don't quite result in perfection as occasionally on screen the X is missing from the disassembled instruction XH. Can anyone come to the rescue?

Contacts wanted

I AM a 15-year-old boy in Singapore who wishes to get in contact with some other Amstrad users.

I have a CPC464 without disc drive. I hope that the users of the CPC664 and 6128 would make friends with me. – **Peter Cheong, Apt Blk 252, Ave 4, Ang Mo Kio, 03-211, Singapore 2056.**

Use for MIN

HAVING found a use for the MIN command I thought I should share it with your readers. In the following program it is used in line 90 to keep the display within the screen boundary. – **W.S.T. Hamley, Gainsborough,**

```
10 MODE 2:INK 0,0:INK 1,26
20 DIM x(13),y(13)
30 FOR go=1 TO 10
40 xo=RND(1)*550+40
50 yo=RND(1)*310+40
60 n=INT(RND(1)*8+5)
70 rx=INT(RND(1)*120+40)
80 ry=RX*0.8
90 q=MIN(xo,yo,400-yo,640-x
o)
100 IF rx>q THEN 70
110 GOSUB 150
120 NEXT go
130 k$=INKEY$:IF k$="" THEN
 130
140 END
150 count=0
160 FOR node=PI/2 TO PI*2.5
 STEP PI*2/n
170 count=count+1
180 x(count)=xo+rx*COS(node
)
190 y(count)=yo+ry*SIN(node
)
200 NEXT node
210 FOR k=1 TO n-1
220 FOR l=k+1 TO n
230 MOVE x(k),y(k)
240 DRAW x(l),y(l),1
250 NEXT l,k
260 RETURN
```


$50
$25
$45
$70
SPACE TO LET
$15
$15
$100
$25
$25
$150
(03) 560 4324

Adventuring with Gandalf

Infocom's best are available for Amstrads at last

ANYONE who has the slightest interest in adventures can't fail to be aware that some of the Infocom adventures have been released on the Amstrad machines.

What you may not know though is that all but one of them – Hitchiker's Guide to the Galaxy for the CPC 6128 – have only been available for the PCW8256.

This situation should have changed by the time you read this and all the Infocom games should be available for all the Amstrad machines.

You can use a CPC6128 version of Hitchhiker's Guide to the Galaxy on your Amstrad CPC464 providing you are using the DK'Tronics ROM pack and their screen handling patch.

The patch is available direct from DK'Tronics and will allow the lucky owners of the RAM pack to run any 6128/464/664 version of the Infocom games. We may possibly see the rest of the range being released, as Activision have signed a licensing agreement with Infocom.

I'd like to thank Boogie Boots for his or her maps of Red Moon, and Jane Dearl who has sent in a complete solution to Pyjamarama. Anyone wanting a copy should enclose an sae.

Thanks also to Barry Newell for his map of Adventure Quest, Hilary Anderson for her map of Lords of Midnight and Adrian Steele for all of his maps. If you want an easier Level 9 game Barry try Lords of Time or Red Moon.

Frossie Economon has written in from Athens asking for help in deciding what his first adventure should be. I think that you would get most enjoyment out of one of the classics like Level 9's Dungeon Adventure. I admit that's being thrown in at the deep end, but the atmosphere is such that you would be hooked from the word go.

If you want to start with an easy one then why not try Forest at World's End – you'll find plenty of help with it from the May column – or Message from Andromeda, both from Interceptor Micros.

Chris Gibson and Moira Brooks have problems with saving a part-played game on the disc version of Lords of Midnight.

Chris says that saving the game overwrites anything on the disc used and Moira says she is unable to save the game to disc or tape.

Amsoft still haven't sent me a disc version so I am unable to offer any advice though I have heard that you must use the actual game disc to save your position.

Does anyone with the disc version know why these problems occur and can they offer any solution to them?

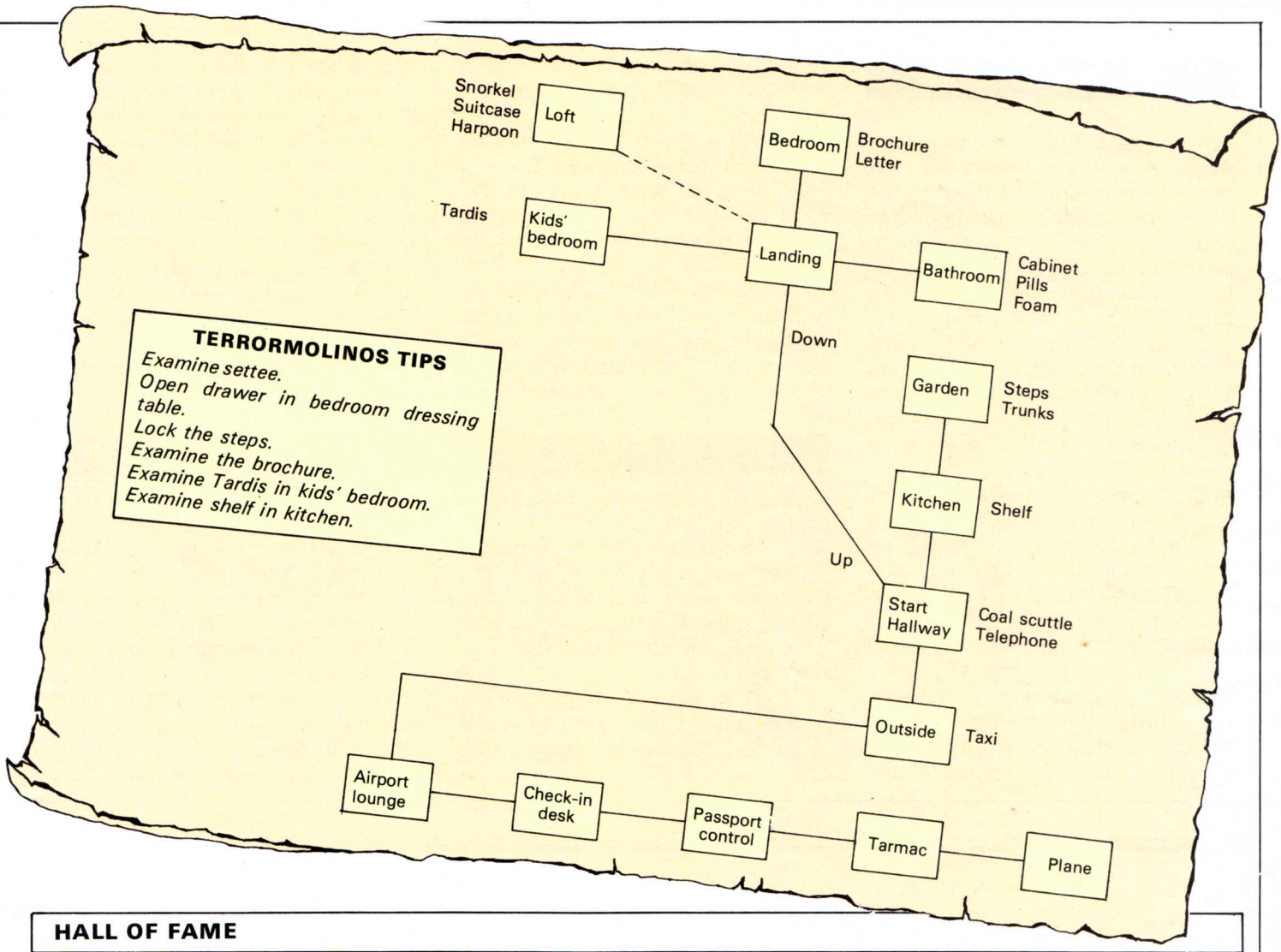
FEEDBACK

DAVID Marchant, Lee Neilson, D.W. Powell, David Kelly and Gavin and Andrew Smith have all written in with help for previous problems with Sorcery+.

The golden chalice will release the sorcerer in the wine cellar. To get the fourth heart needed to kill the necromancer take the jewelled crown through into part 2. To get into practice mode press the fire button and C.

S. Dempsey has answered previous problems raised with Gems of Stradus and Subsunk. To get past the ghost use the bottle. In Subsunk to open the safe wear the stethescope and then TURN LEFT, TURN LEFT and TURN RIGHT.

To get the aerial into the torpedo tube grease it first with butter. To join the map you need some glue. Mix the toast scrapings with yogurt to get some.



TERRORMOLINOS TIPS

Examine settee.
Open drawer in bedroom dressing table.
Lock the steps.
Examine the brochure.
Examine Tardis in kids' bedroom.
Examine shelf in kitchen.

HALL OF FAME

The response to last month's introduction of the Hall of Fame has been phenomenal, so much so in fact that I will have to ration them.

Dennis Goodwin has sent a solution and a map for **Mordon's Quest.** Thanks Dennis and yes, I would like to see a full solution to *Subsunk* and *Never Ending Story.*

Richard Hyams has sent in a comprehensive hint sheet for *Return to Eden.* Both of these will be serialised over the coming months. Thanks also to Mark Rodgers for his list of tips to his travels so far in **Heroes of Karn.**

★ ★ ★

Mordon's Quest *– Dennis Goodwin*

Climb the drainpipe to find Mordon who is back inside the house. The torch will get you through the mist into the jungle and the blanket will get you across the swamp if you drop it.

Use the thorns, berries and bamboo to make a blowpipe to kill the pygmy. The answer to Tarzan's question is frog – the map of the jungle looks like a frog.

To pass the waterfall sacrifice the frog on the altar – you will need the dagger.

To get the diamond you must smash the iron pyrites, but pick them up afterwards.

Finally use the geiger counter from the future to find the battery in the Roman barn.

★ ★ ★

Heroes of Karn *– Mark Rodgers*

The tinderbox ignites all gases and the barrowright dies when he sees holy objects.

Try kissing the frog. The bear likes honey and the songbird kills serpents.

Say "orion" among the stars. The flute smashes crystal and jade, so don't play it near the bottles.

Try acid on the black knight and bribe the guard. You need a cage for the songbird. Kill the dragon with the sword, the hydra with the spear and the witch with the water.

Wet the ashes with water and use a crowbar to open the clam. Finally have the wand over the chasm.

★ ★ ★

Return to Eden *– Richard Hyams*

Dig into the molehill. Wear the radsuit and sleep in the cave with leaves; dig in the loose earth with a spade.

Remove the radsuit before entering the forest and take the sweet pea and drop it. Eat the bean to carry more objects.

Plant the egg with the spade and give the fish fungus to the leviathan. Eat the pill to avoid radiation sickness.

The parrot's nest needs to be found in the maze of maize.

When you have everything go down the river to the waterfall but make sure you have the soggy log. On the tiny island squeeze the log and the bulb will become a parachute.

HINTS DEPARTMENT

In *Hitchiker's Guide to the Galaxy* Roger Wilson has solved the first three problems involved in getting the Babel fish, but is now having trouble with the upper half of the room cleaning robot. Try putting the object found in your front porch on top of the satchel.

David McCarthy wants to know how to get past the plant or pygmy in *Mordon's Quest.* I don't know how to get past the plant, but to get past the pygmy use a blowpipe – see Hall of Fame.

S. Dempsey is having problems with *Gems of Stradus.* The sword should be sharp enough to kill an alien. To open the doors you need a key of the same colour as the door you want to open.

D. Pye is deep into *Dungeon Adventure* but needs help to get any further. Throw the body to get past the jellies. The pedestals are a transport system. Wear a collar and experiment with the colours of the rainbow. The marble tower is a look-out point – use the transport system.

Ian Wave and Helen and Michael Peats need some help with *Emerald Isle.* The spider needs sticky threads, so throw some glue on them. The mine will collapse unless you prop up the roof.

W. Aspinall is in difficulty in *Return to Eden.* Every time he tries to get his credit card back from the big robot it gets stolen. The dumpy droid is a cleaning robot and leaves anything it finds at the lost property office.

Jane Dearl needs my help in *Message from Andromeda.* Point the rod at the plate. Bill Walsh is also stuck in this game and should rotate the sphere.

Finally I would like to remind you all to send in an sae if you want help. I will try to answer problems in this column if you don't send in an sae. However if you don't see an answer it is because it has been given in a recent issue in response to another reader's letter.

SOS

Several problems have arisen this month where I can't help.

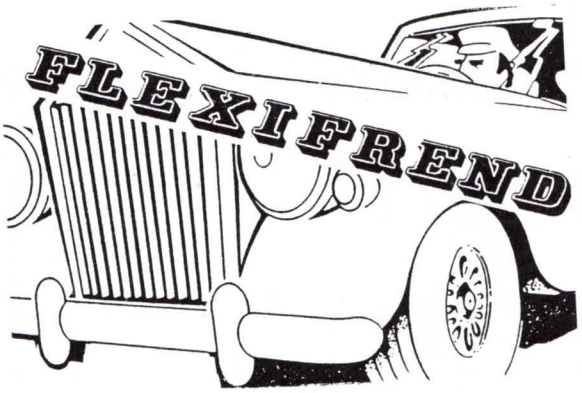
Jane Dearl is stuck in **Fantastic Voyage,** an adventure that I haven't heard of before. She says that she knows the eighth piece is in the colon but doesn't know how to get it.

E.T.V. Pidwell wants to know how to get the sword from the stone in **Castle Blackstar.** Can anyone help?

H. Wood can't get the droid out of the lift on the first floor in **Mission 1: Project Volcano.**

S. Smith has got the gold key but can't find anywhere else to go in **The Never-ending Story.** I could also do with help in this.

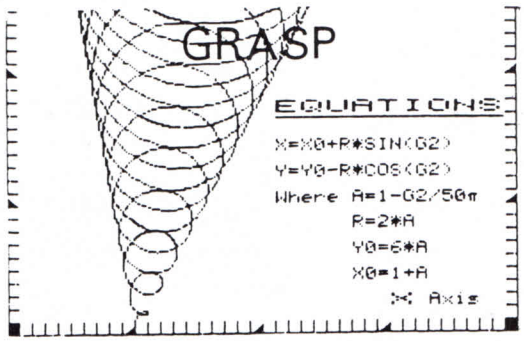
D. Pye would like help in moving the immovable stone in **Dungeon Adventure.**



FLEXIFREND

Amongst the many useful and creative applications for home computers, budget planning is one of the most popular. Flexifrend is a package which combines all the facilities needed for planning home finances in a program which is easy to use. Flexifrend features an integral calculator which can be called to the screen at the touch of a key at virtually anytime during the running of the program.

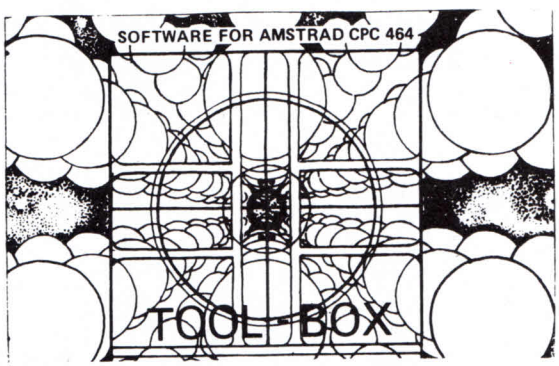
$16.95



GRASP

Grasp for the AMSTRAD CPC464 is a utility which has a myriad of uses for educational, business, and general home use. With it's rapid presentation of data in graphical form it provides the idean means of picking out trends, demonstrating principles, and enhancing the presentation of your ideas. Using inexpensive screen photographs, everyone can now utilise the techniques which have been used to great effect by advertising agencies, public speakers and professional writers.

$16.95



TOOLBOX

AMSPICT and AMSPRITE form two outstanding software tools for creating screen pictures and animated sprites with the option for easily recreating and using these in your own BASIC programs. Using these two, you can easily add graphics to your adventure games, or write arcade standard action games with smooth pixel motion of your characters.
Also supplied are: FCOPY and AMSMON. FCOPY can be used to make back-up copies of any cassette which conforms to the Amstrad file format. This includes programs saved using any of the usual SAVE commands, both Basic and code. AMSMON is a very useful utility which allows you to type machine-code programs into memory, examine the contents of either the random access (RAM) memory, or the read-only (ROM) memory. Experienced users will know that addresses 0 to 3FFF hex and C000 to FFFF hex are used for both ROM and RAM. Normally, reading from these addresses will access RAM (e.g. when using PEEK), so special action must be taken to read from ROM. This is catered for by AMSMON.

$16.95

STRATEGY SOFTWARE
BOX 5000
GLEN WAVERLEY
VICTORIA 3150

(03) 560 4324

TAPE ONLY

July 1986 Vol. 1 No. 4

BUSINESS COMPUTING

WITH THE AMSTRAD

Opening a window on vertical software

REVIEWS: Amstat, Microscript, Sage Comms Pack, Smartkey II

Networks market expanding

DISTRIBUTOR Northern Computers is appointing UK dealers with exclusive areas to supply the Amstrad network and hard disc systems. The company says the appointment of dealers, in addition to overseas distributors supplying the network system, is a response to the rapidly growing demand by Amstrad computer users for multi-user applications.

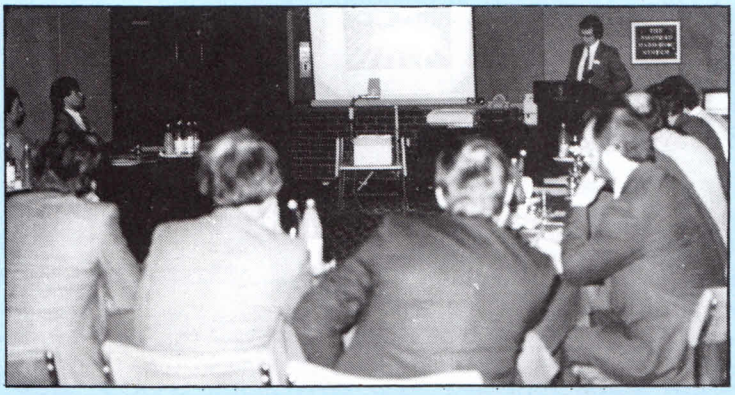
Northern Computers says it is now replying to enquiries from potential customers at the rate of 2,000 a month. About 80 per cent of these originate from the industrial and business sector, the remainder coming from education, health authorities and home users.

The company's UK dealer manager, Mark Sporne, says he intends to appoint about 60 dealer/installers. Nearly all network purchasers so far have asked for installation, user training and maintenance contracts, he says.

The dealers will provide these services in addition to selling Amstrad computers, network components and hard disc systems.

Northern Computer's product is a register insertion ring network system which can accommodate up to 120 stations over a maximum distance of 3km. It uses 20mb file and print server units called Amstores which also provide print spooling.

All Amstrad models operate on the network, in addition to Apple, Apricot, BBC and the IBM PC.



Northern Computers UK dealer manager, Mark Sporne addresses an Amstrad network and hard disc dealer seminar.

Nominated

AMSTRAD'S chairman, Alan Sugar, is in line for another personality award. Already named Personality of the Year by RITA (Recognition of Information Technology Achievement), he is now nominated for a similar title by the Marketing Society.

The other nominees are Richard Branson, boss of Virgin Records and Virgin Atlantic Airlines, Eddie Shah, publisher of Today, Michael Grade, Controller of BBC 1 and Lord Rayner, chairman of Marks and Spencer.

Sugar's Amstrad PCW8256 is also nominated for an award under the consumer durables category.

Pagemaker for Amstrad

AFTER three months' delay the headline-making publishing package AMX Pagemaker is now available for the Amstrad CPC464, 664 and 6128 range of micros.

Created by Advanced Memory Systems of Warrington, this latest version of the top-selling graphics program was to have been launched at the Amstrad Computer Show in Manchester in March.

"What at first looked to be a minor hiccup turned out unfortunately to be a major hold-up", explained Nick Pearson, managing director of AMS. "But people who know about the intricacies of programming will be well aware of the unforeseen pitfalls which occur from time to time".

AMX Pagemaker for the Amstrad combines a typesetter, graphics and word processor on two discs for £49.95. It lends itself to letters, posters, menus, in fact anything that needs to be created on A4 size paper. "We see it as a most useful business tool", said Nick Pearson.

Seeking to strengthen its position in the Amstrad market, the company has also made AMX Utilities – support software for the AMX Mouse, price £14.95 on cassette and £19.95 on 3in disc – available for the CPC range.

New modules widen scope

MINI Office II, successor to the 1985 Business Program of the Year Mini Office, is reaching an even wider audience than the first version.

The addition of two new modules – label printing and communications – to the original combination of word processor, database, spreadsheet and graphics has given the software even wider appeal.

As well as business users and computer training course operators, educationalists at all levels from college to primary schools are buying the program, as are hospitals, the Ministry of Defence and other government departments.

Mini Office II, which is available from all leading multiples like Boots and W.H. Smith, is being used in Health Service administration, to train youngsters in business technology in Itec centres, and in schools for children of British Army personnel.

"Mini Office II is training administrators of the future", says a spokesman for the publishers, Database Software. "Even primary schools are using it because its excellent graphics make it an attractive teaching tool for the youngest age groups, while at the same time it is sophisticated enough for any businessmen or office administrator".

Mini Office II costs £14.95 on cassette and £19.95 on 3in disc.

SINCLAIR STOCK IS SOLD OFF

AMSTRAD has already effectively recouped more than half the £5 million paid for the Sinclair name in one deal. This involved the unloading of 50,000 Sinclair QL and Spectrum computers to a Buckinghamshire firm.

Marlow-based PST paid £2.6 million for the machines which the company now intends to market world wide. Although PST only entered the computer field nine months ago, it has already figured prominently in a number of large shipments abroad.

The news that Amstrad has achieved such a spectacular deal so soon after handing over £5 million for a company once worth £136 million scarcely caused an eyebrow to be raised in the City.

"When it comes to Alan Sugar you can't be surprised by anything", said a leading broker. "There is little doubt that like with everything else, he knew he was entering a profit-making situation when he bought Sinclair. He just can't put a foot wrong – for the moment at least . . ."

Amstrad swamping the French market

AMSTRAD has the competition reeling in France with the news that it has captured 60 per cent of the market.

Yet it seems they even have more shocks in store. For experts are predicting that the recent arrival of the PCW8256 could see this increase to almost 90 per cent.

Writing in The Sunday Times, Anne-Elisabeth Moutet claims that leading rivals such as Thomson, Commodore and Apple France "barely know what has hit them".

She insists that the secret of Amstrad's success in France stems from its marketing techniques in general and its £470,000 advertising budget in particular.

Under the direction of Marion Vannier, a former hi-fi saleswoman, turnover has rocketed from 35 million francs in 1983-84 to 291 million francs in 1984-1985. Now the forecast is that it will reach 700 million francs for 1985-86.

Amstrad sold just 10,000 computers in France during 1984-85. However total sales last year amounted to a staggering 200,000 machines.

Asked for the reasons for her company's breakthrough, Madame Vannier cited four – a complete, ready to plug in package; low comparative cost, reliable service and good marketing.

"Yet it is the marketing more than anything else which has seen them come out on top", one French analyst told *Computing with the Amstrad.* "They have taken their computers and sold them like hi-fi with all the razamataz that that involves.

"Other companies have literally been caught with their pants down by this brash young company".

But will the situation remain like this for long? "The fact of the matter is that the other companies may have superior products but they have been left at the post", said the analyst.

"Amstrad are going to take a lot of catching here in France".

A Show spectacular

A SPECIALLY-built 150-seat Amstrad Theatre will have its own premiere at the Amstrad Computer Show at the Novotel, London.

The "show within a show" will offer continuous presentations aimed at business users during the three day event, all of them enhanced by special lighting effects and sophisticated sound equipment.

Amstrad's own specialists including technical manager Roland Perry, will be using the theatre to host You and Your Amstrad – a daily question and answer session with a panel of experts.

Also featured every day in the theatre is What is a Micro? – business systems computing on the Amstrad explained, with 20 free training courses being given away during the course of the show.

Other highlights in the Amstrad Theatre include presentations by the UK's fastest-growing electronic mail service, MicroLink, business software from Sagesoft, AMX Mouse and AMX Pagemaker from AMS, Cash Trader and other business products from Quest, professional software from Caxton, light pens and graphics from Electric Studio, and a demonstrations of the revolutionary six-module database software Mini Office II.

Amstrad Theatre is open 10am to 6pm Friday and Saturday, June 13 and 14, and 10am to 4pm Sunday, June 15.

IBM clone

MORE whispers about Amstrad's rumoured IBM clone from America. The machine forecast in last month's *Business Computing with the Amstrad* was a hot topic at the Comdex Show in Atlanta, Georgia.

A high ranking Digital Research executive confirmed that his company's GEM would provide the new machine's built-in graphics interface. "People are going to see an IBM compatible computer which comes bundled with GEM selling at the same price as an electric typewriter", the DR man promised.

The new computer is expected to cost around £499 and some pundits are forecasting that one version will also include a printer at £599.

Constructive criticism can pay

By JO STORK

IT is highly rewarding to observe that these pages are beginning to have an influence. I am therefore happy to incorporate two items of "feedback" this month.

Regular readers will recall a statistical package called Amstat 1 which I reviewed some months ago. I rated it an excellent product for performing parametric statistical tests but handicapped by its lack of a worthwhile manual. Amstat 2, which I received this month, shows how constructive criticism can develop a product.

This latest offering is a separate product from Amstat 1, permitting totally different statistical tests – the non-parametric for those who know what these are. The producers have wisely produced a 40 page manual which is more than adequate for those who understand basic statistics.

Newcomers to the subject needing a far fuller explanation of the uses to which these tests may be put will find cross references to what is arguably the current standard statistical text book by Sidney Siegel.

This product is not specifically geared to the business market, since it will probably have far more sales to students, but I now give it an unreserved recommendation for those organisations requiring:

- Quality control calculation.
- More complete management statistics than are available within most finance packages.
- Statistics utilities but are unwilling to generate their own in either a spreadsheet or Basic.

The ability to transfer data and results in Ascii to word processors will prove particularly useful to those producing reports, as you can see from Figure I. After a little experimentation I found that I could also use this Ascii file as input to my standard graph plotting program. Nevertheless can I propose that when the proposed CP/M version is released it incorporates histograms and line charts?

While I have not seen the latest version of Amstat 1, I understand it too has been revised to incorporate recommendations made by its many regular users. Consequently, at £49.95 for Amstat 1 and Amstat 2 combined, this represents good value for money, even if you do need to buy a copy of Sidney Siegel's text book.

The second example of feedback I

Tom,
Please examine the figures below, which represent the AVERAGE DAILY production of our staff during the first 13 weeks of this year, when producing different items in our range.

DAILY PRODUCTION AVERAGES
13-APR-86

Variables are listed in columns of five

	Alan	Bert	Chas	Dave	Eddy
Prod. A	40	32	35	28	37
Prod. B	32	37	37	35	33
Prod. C	29	41	42	37	39
Prod. D	34	39	41	48	31
Prod. E	37	26	40	29	40
Prod. F	32	46	34	40	25
Prod. G	43	50	33	39	46
Prod. H	31	26	33	36	41
Prod. I	27	28	37	31	29
Prod. J	46	29	26	29	27

	Fred	Greg	Hugh	Ivan	Jack
Prod. A	37	31	27	44	51
Prod. B	33	38	29	46	46
Prod. C	37	29	30	47	42
Prod. D	39	24	34	32	43
Prod. E	36	27	34	35	36
Prod. F	37	36	37	31	37
Prod. G	39	39	38	36	32
Prod. H	50	43	39	29	28
Prod. I	23	42	47	24	24
Prod. J	47	37	44	27	26

	Leon	Matt	Norm	Owen	Pete
Prod. A	22	41	46	27	37
Prod. B	32	42	42	24	34
Prod. C	34	36	25	28	36
Prod. D	36	37	28	29	38
Prod. E	51	33	33	32	39
Prod. F	30	32	37	51	32
Prod. G	38	36	26	45	37
Prod. H	39	41	29	43	32
Prod. I	38	37	40	42	33
Prod. J	43	24	37	39	35

Running a Chi-Squared against these results, it is obvious that a review is necessary in order to permit the staff to manufacture the products to which they are best suited, since the matrix shows significant variability.

DAILY PRODUCTION AVERAGES

13-APR-86

Chi-square K sample test

Chi-square = 179.561

DF = 126

Associated probability = 0.0012

I therefore intend to propose a re-scheduling of work at the next Management Review.

Best Regards,

Bill.

STATISTICAL OUTPUT SENT TO WORD PROCESSOR FILE BY AMSTAT 2

Figure I: Incorporating Amstat files into text

received this month came when I met a regular reader of *Computing with the Amstrad* who shook me rigid by saying: "Even if you rate a product very highly, you still point out its deficiencies". I had no time to reply before he added: "Will you never be satisfied?"

The honest answer is – no not ever. Before software houses refuse to send any further material for review I add that this is as true of systems I produce for my clients as it is for products I write about. Time pressures, client requests, marketing compromises and a host of other criteria invariably result in my having to hand over systems before every facility, feature and benefit I wish to offer is incorporated.

Similarly, when choosing software you must accept that you will be extremely lucky to find perfection on the shop shelves. A few minor moans must be accepted.

As long as they stay minor, you have all the essential features you require and you do not obtain more software than you will need during the life of the system then you have found the best software for you.

Understanding that the same problems as mine are faced by the software houses that I write about actually makes me highly tolerant of idiosyncracies which may grate with personal prejudices.

They rarely form any part of my general rating of a product. However I must mention them, since if you are going to live with a system for several years you will not thank me for ignoring something which may ultimately prove highly irritating, even when the basic facilities are first rate.

It is for this reason that I mentioned the question of graphics in the case of Amstat.

The next issue is that with any program, there are two distinct phases during its life. The first is the time needed to learn the commands which enable the options to be accessed. This should be a brief period, but even 30 hours may not be unreasonable for a major system. As a rule, the program which has a very short learning cycle is likely to be outgrown equally quickly.

MICROSCRIPT TEXT PROCESSING SAMPLE

This piece of text was produced using MICROSCRIPT. It works for the most part much as any other Word Processor, in that there are the normal range of editing, cutting, pasting, merging and printing options. Curiously the only function unavailable, (or which I failed to discover), is a 'Delete Line'.

Where it may prove most valuable is for users wishing to produce reports containing columns of figures. Most word processors only permit lines of text to be manipulated. Where MICROSCRIPT is unusual is in permitting columns to be moved. Since the majority of columns also contain figures automatic calculation of the values they are to contain is available. While certainly not as powerful as a Spreadsheet, it is a useful tool when transporting files from one program to another, eg. a Spreadsheet or Database is undesireable.

A further possible use is for those organisations wishing to produce their invoices using a Word Processor rather than a full-blown invoicing package. This is perfectly feasible in service oriented companies, which require these invoices to detail the descriptions of the services provided, while not having to process many such invoices each month.

All the features are explained in a 100 page manual accompanying the system.

FARM IMPLEMENT
- UNIT SALES -
- PRODUCT -

	A	B	C	D	E	F	G	TOTAL
JANUARY	22	31	43	16	9	17	4	142
FEBRUARY	29	33	42	22	7	26	7	166
MARCH	42	41	58	26	11	32	12	226
APRIL	38	33	73	38	20	40	18	260
MAY	51	38	60	35	27	31	22	264
JUNE	73	50	84	58	46	62	36	422
JULY	102	84	111	67	65	83	71	583
AUGUST	113	103	116	82	71	76	79	639
SEPTEMBER	28	25	84	60	42	65	50	454
OCTOBER	30	41	34	31	24	44	33	282
NOVEMBER	32	32	42	31	20	28	28	287
DECEMBER	29	36	45	24	13	25	19	191
TOTALS	688	631	831	493	357	532	375	3907

Table produced using MICROSCRIPT CALCULATOR options

Figure II: Sample Microscript output from its built-in calculator

In the case of Microscript, a word processor from Amsoft, I feel that this phase will last five to seven hours, which is only acceptable for a system of this size.

The manual is extremely comprehensive and has a massive index. This helps reduce the time required, but set against this is the sequence adopted, plus the absence of many potentially useful index entries. An on-line help facility would have been most welcome.

The second phase, when the program is in regular operation, should last years. There is therefore a major risk that anybody who has acquired rapid familiarity with the features may have to live a long time with the annoying quirks, complexities or omissions.

There is another factor to consider in any software purchase. The product you first heard about as the clear market leader a year ago has probably been bettered by now. This does not mean it is now a bad product, merely that time does not stand still in this business.

This point is particularly true of Microscript. The key problem that Amsoft faces is that the market for word processors has changed drastically in the last 18 months.

The early home micros had limited memory and therefore the programs had few facilities and/or were slow, particularly when doing string searches or cutting and pasting.

To overcome this irritation, ROM-based word processors appeared. Not only did this free up a considerable amount of memory for the text itself, but also made the program run very much faster.

Soon after this development Mr Sugar announced dedicated word processor machines – the PCW series – with every conceivable bell and whistle at a price which no separate computer/printer/software combination can touch.

The current situation results in traditional word processor systems such as Microscript needing either an exceptional price/performance ratio or some unique feature which other systems have not implemented to have any hope of commercial success.

Competence is no longer a guarantee of sales, no matter how well the product is marketed. This is why I doubt Microscript will sell in large numbers. In a buyers' market there is no excuse for buying a merely good product.

Word processors fall into two main types, those directed at the person with only an occasional need and those for whom word processing is a

major portion of their work. The first type must be kept as simple as possible, which unfortunately means that few features will be available. Only the simplest editing, cutting, pasting and print formatting should be offered, otherwise the user will forget how to use the program between runs.

The second type needs a considerably longer period of familiarisation, but should provide every facility which even the most fastidious editor or office manager can demand.

Microscript is a good product but I suspect it does not offer enough to satisfy those with a heavy word processor requirement, who should be seriously considering the 8256/8512 route. Most of the essential features needed are there but the relatively small memory available for text, together with some limitations of the print style variations available, makes me think that the semi-professional user would soon be looking for a replacement.

Conversely there are a full range of text manipulation options including Text Merge to learn and remember, which certainly places Microscript beyond the occasional user.

However before dismissing this as an unworthy compromise, Microscript has three important shots left in its locker. Firstly, while aimed at the 464/664 market this is a CP/M 2.2 based word processor. Secondly Microscript contains a built in calculator – see Figure II. This could well be the extra feature it needs to make it stand out from the crowd of its competitors.

Finally up to 10 lines of text may define a series of instructions used to program it to carry out specific actions. The instructions are called up with a couple of key strokes as often as necessary anywhere in the text.

This last feature is normally only available on much larger word processor systems, which makes me wonder if Microscript is not another migration to the Amstrad. If it is then it has certainly been successful, but I fear it has come too late as this market must be coming close to saturation.

Nevertheless don't dismiss Microscript out of hand. This thoroughly competent product may well be ideal in some organisation. Take a look.

Using the Mini Office II database

I HAVE been asked by readers to discuss databases in more detail. However it is hard to talk about them in general, so over the next couple of months I'll introduce them with reference to Mini Office II, which offers a word processor, spreadsheet, database and other facilities for under £20.

The lessons we learn will be readily applicable to the more complex and expensive packages you might want to consider once you've learned on Mini Office. Then again, you might find its combination of power and ease of use ideal for your needs.

Anyway, this introduction to databases is primarily intended to make you broaden your horizons. With a little imagination you may avoid needing to make do with an inadequate financial, stock or club membership package.

The good ones will obviously offer far more facilities than can be obtained using a database alone, but they may not be entirely suitable and also the application may be sufficiently simple not to justify a specialist system.

To illustrate this we'll use Mini Office II's database to design an application to post and analyse the entries into a cricket club's cashbook. Next month we will discuss the data manipulation features which could prove invaluable. I'll assume you've got the manual by you for reference – what follows is meant to be a tutorial.

Since Mini Office II requires the whole of the file to be in memory the first stage is to analyse the best mix between the number of characters per field, the number of fields per record and the number of records per file. We will assume an average of 70 postings per month, and that the organisation is not VAT registered. More postings or VAT registration would mean a specialist book-keeping program is required.

To make life easier, as a record is designed the software will inform you of its size and the maximum number of records that may be handled by a single file. If the record length must be increased the number of them that may be held is correspondingly reduced.

With our record length established the next phase is to split it up into fields. To keep this example simple we will assume that only the 12 fields shown in Figure III are required in each record.

Since we need only 12 fields, and Mini Office II can handle a maximum of 20, this leaves us eight spare lines which may be used for gaps or for guidance. In the first instance we will assume the desired field lengths are possible.

There are four other points to note at this stage – firstly, the club treasurer assumes he will need to sort/select records which are marked with an asterisk, and secondly, Mini Office II will provide totals of the two amount fields.

Thirdly it automatically allocates three characters to the Date and Numeric fields and, finally, Mini Office II can handle 196 records of this length before a new file must be created – in other words approximately 11 weeks' transactions.

With the record designed we can

→

now start placing it into the computer:

Step 1: Place Mini Office II disc in drive.

Step 2: Type RUN "Office"

Step 3: Select Database

Step 4: Place data disc in drive

Step 5: Select edit structure

At this point we can begin to define the record structure to the database. The information that may be entered for each field is explained in the manual, except in the case of the formula.

If a type of formula is selected use the cursor to select the first field number in the calculation and press Enter.

Use the cursor to select the arithmetic expression required, such as + or * and press Enter. Use the cursor to select the second field number in the calculation and again press Enter.

The entries needed to produce our record are shown in Figure III. If an error is made at this stage there is no need for concern, since you may make as many changes as are required. If a few more records were required you could trim some characters off some of the fields. If the wrong type had been defined these could be altered and if a title was incorrect this could be modified.

Particular attention should be paid to the title chosen because this is the prompt given to the operator during data entry, as well as being the column headings during listing.

Only when absolutely sure that the structure is correct should you press Escape and return to the main database menu. At this point, even though there is no data yet held on the database you are strongly urged to do a save in order to secure the structure itself.

Furthermore, as a point of good systems design, whenever fields such as Transaction Type or Reference are used as prompts a list of permitted entries should be prepared. These should form part of the eventual operations notes, otherwise in time sorting and particularly selecting will become increasingly difficult.

What started as a transaction type EQPTPURC, meaning EQUipment PURChase, is likely to pass through EQUPTPUR via EP to EQPURC.

Sort	Select	Field	Size
*		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	93 characters

Figure III: Sample record structure

Come the day when a listing and total of all the equipment purchases are needed searching would at best be laborious given the four way split above.

It is only the largest, most powerful databases that allow the creation of a table of permitted entries and force a re-input if an erroneous value is made. The preparation of operator notes is even more important in such cases, otherwise the operators would never be able to get beyond the field for which they did not know the acceptable values.

To return to our record structure data entry may now take place, therefore select Edit Data from the main menu. Were this a live system the first entry to be made would be carried forward value for the amount received and the second would be the carried forward for the amount paid, thus ensuring that the totals could be used at any time to determine the cash-in-hand of the club.

Note that in my example these two fields are computed automatically as a result of multiplying the quantity by the unit price.

Entry of new records is made by pressing N when the record screen appears. Figure IV shows our record scheme. The fields which were defined as MT = N – that it, they aren't allowed to be empty (MT) – must have values placed in them before Mini Office II will permit you to exit from this record.

Similarly an attempt to put letters into a numeric field will be rejected. An example of a purchase entry for three new cricket bats is shown.

Now that we have the database designed and records being entered into it we will stop for now. Next month we will look at how the other features of Mini Office II's database may be used in order to make the treasurer's life easier.

```
Date                11/05/86
Transaction Type    EQPTPURC

Name                S. SURRIDGE
Purpose             CRICKET BATS
          REVENUE
Quantity                     0
Unit Price                .00
Amount Received           .00
       EXPENDITURE
Quantity                     3
Unit Price          -    37.95
Amount Paid         -   113.85

Reference           0102
Notes               12.5% Discount Received


+ive Price RECEIVE
-ive Price PAID
```

Figure IV: Records structure

A comms link for the PCW series

By GABRIEL JACOBS

THE Sage Communications Pack for the PCW series is a complete kit, consisting of a modem, a serial cable, and a communications program called Chit Chat which is designed to replace Mail232, supplied free with the PCW.

All you need with the Sage Pack is a serial interface and you're up and running – no wiring diagrams, no frustrating trial and error installation procedures.

The modem is housed in a black plastic and metal case measuring 7½in by 6½in. It is 2in deep, and thus well out of the fashionable slimline category, but nevertheless looks and feels like a quality product. It also has British Telecom's blessing, so you needn't worry about being fined £1,000 for hooking up an unapproved device.

At the front of the case are a connect button, two LEDs indicating carrier detect and power/data transmission, and two further buttons for selecting V21 (300/300 bps) or V23 (1200/75 bps), the most common modem standards.

The mains power and telephone cables come out of the back where there is also a self-test button, a standard 600 series socket to take a telephone handset and a DIN socket for the serial cable. The other end of the serial cable has a female D socket which fits the RS232 port of the PCW's interface.

As modems go this one has relatively crude facilities. Everything has to be done manually, including dialling from a handset, logging-on to the remote carrier signal, and toggling between V21 and V23.

You have to push the correct baud rate buttons before a call, and to disconnect the modem after it, or waste time and money watching a blank screen when the host computer answers.

Although Chit Chat software comes configured to run with the Sage modem it can be bought separately and will work with other manual modems, as well as smarter systems, if correctly installed.

Included on the Chit Chat disc are several installation files customised for popular makes of modem, including Thorn EMI, Dacom, Master Systems and Hayes. There is also a file which should work with nearly all straight manual modems, and one for manual modems with speed translators.

If the appropriate file fails to install your particular modem correctly you can resort to an all-purpose installation file, also provided on the disc. But be warned – if you're not familiar with communciations procedures and protocols you'll probably need professional support.

Sage will be able to help, since as an authorised user you can call its hotline for free advice over a period of 90 days from the date of purchase – one reason why it pays to buy from a reputable company.

Chit Chat has been carefully configured for the PCW – not an unimportant consideration, as you'll know if you've used certain pieces of patched software originally designed for other machines. It works on the PCW precisely as intended, makes use of many of the machine's facilities and takes into account base-line limitations such as a single disc drive.

Chit Chat loads automatically into the RAM disc, is driven by the function keys and knows that it is talking to the PCW's dedicated printer.

The general approach is different from that of Amstrad's Mail232, and if you have no special requirements such as DEC terminal emulation, you'll find the Sage software preferable.

The core of the program is the phone directory (see Figure I), which is much more than a simple database of phone numbers. Each entry is the tip of an iceberg – the summary information displayed on the screen hides a mass of previously entered data stored on disc.

By calling the directory editing screen various parameters and swit-

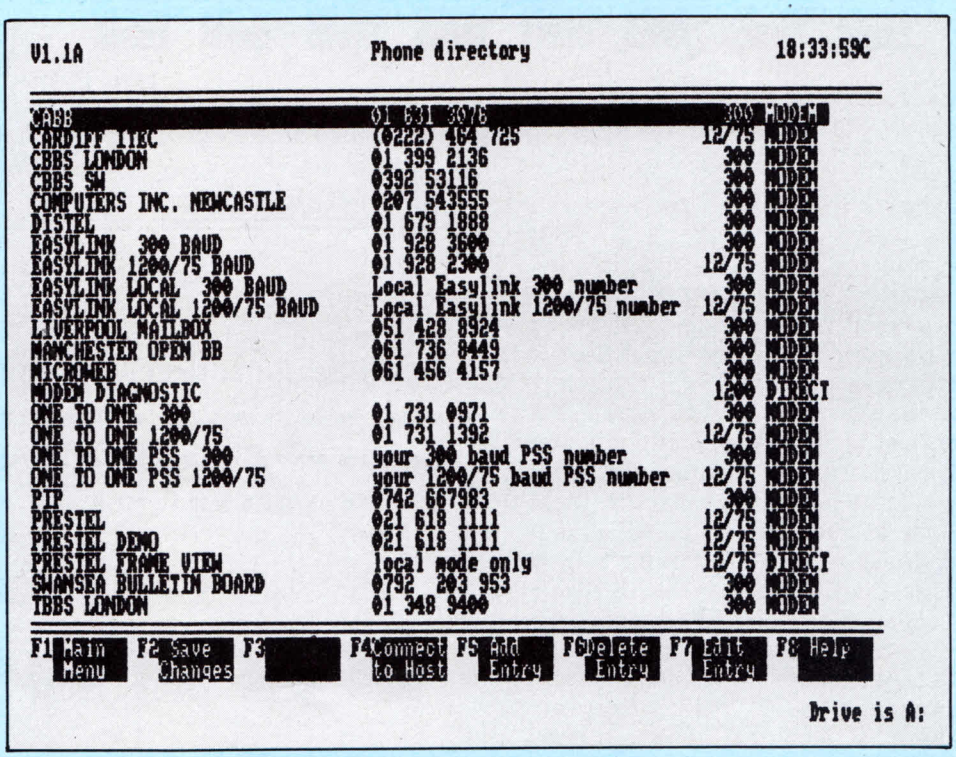


Figure I: A page from the phone directory

ches – from parity to password strings – can be set for new or existing entries (Figure II). Once an entry has been set up all its data is implemented when the connection is made.

In addition, the program can use the PCW's internal clock to carry out tasks at predetermined times, provided you have entered the current time at boot-up. A task file is just a sequence of commands with a start time, and can be used, say, to send telexes at off-peak rates, or to empty your electronic mailbox and write the messages to disc while you are away.

Interactive data capture is even easier. In terminal mode at the touch of a function key everything echoed to the screen can be saved to a disc file for later viewing.

In Viewdata mode there are special facilities for capturing either Prestel frames or those of other viewdata systems, recalling them from disc and printing them, without the need to strip them of the control codes they contain.

Everything has been designed for convenience and ease of use. The clock continuously displays current time and elapsed time during a call. There is a context-sensitive Help facility, and a disc manager which allows you to view the disc directory, and delete or rename files without leaving the program.

Most important of all, there is a built-in text editor – Figure III.

It does not have the wide range of features found on full-blooded word processors, but it uses many of the PCW's dedicated word processing keys, and has enough flexibility to make it useful for preparing and, if necessary, printing chunks of text off-line, thus saving connect time.

You can exit to LocoScript, create a page-image Ascii file and send that instead, but there are times when a page or so can be more conveniently entered using an internal editor.

The supremely lucid documentation for all these features is entirely specific to the PCW, unlike that of many other configured programs where a couple of typed pages inserted into the manual are the only indication that you've bought the right configuration.

Chit Chat comes in three different versions – E-Mail for mail networks such as MicroLink and Telecom Gold, Viewdata for public systems like Prestel (both cost £70), and Combo which will handle the two (£100). Combo is the version provided with the complete kit, which costs £200.

These prices, do include VAT, but do not represent outstanding bargains – however, you can be sure of quality and support. Furthermore, buying Chit Chat gives you the option of taking out a free registration on MicroLink.

So at first the world can be your oyster for practically nothing, and just when you begin to wonder how you ever managed without electronic mail and remote databases, British Telecom can start to cash in.

Product: Sage Communications Pack
Price: £199.99 (with either E-Mail or Viewdata software), £239.99 (with both)
Supplier: Sagesoft, NEI House, Regent Centre, Gosforth, Newcastle upon Tyne NE3 3DS. Tel: 091-284 7077

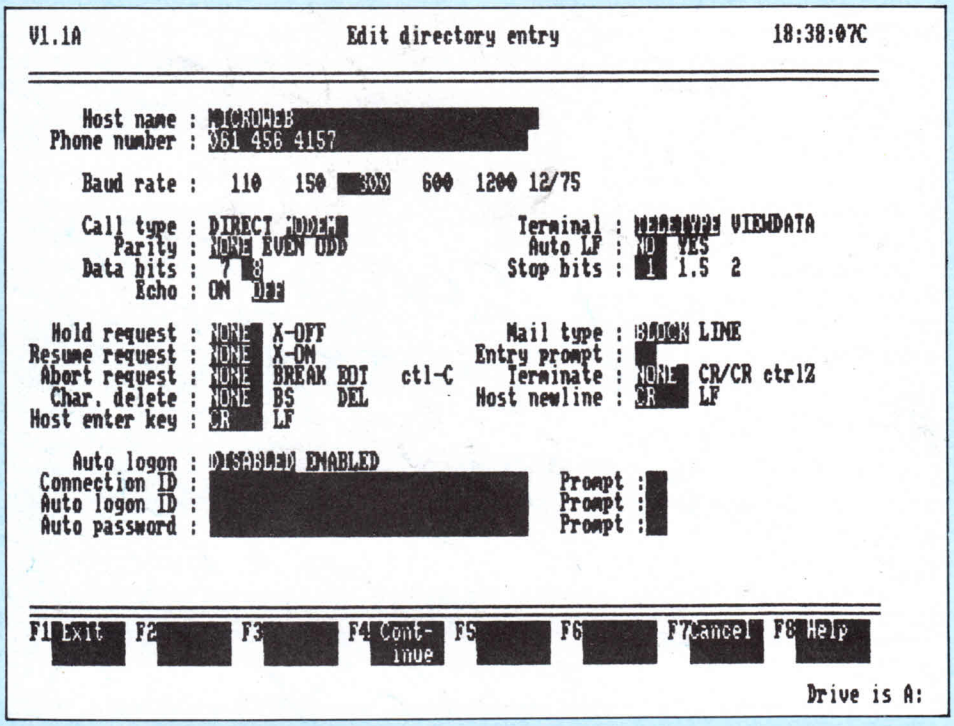


Figure II: Setting up parameters for a directory entry

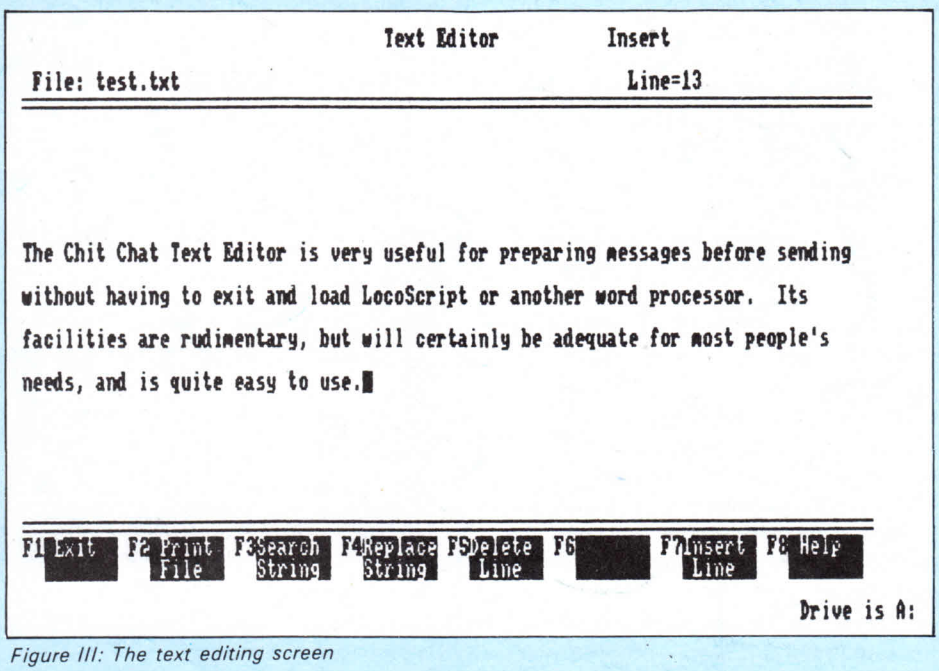


Figure III: The text editing screen

Mike Aylett . . . "It reduces the chances of human error"

Opening a window on vertical software

AFTER spending 15 years as a computer programmer and analyst for Corning Glass, Joseph Williams struck out on his own in search of fresh fields as a consultant – only to find himself writing software for a company selling glass.

The long arm of coincidence has meant that his first commercial package since setting up his own company, C.J. Systems Software Support, is for Amstrad PCW8256 owners who sell double glazing.

Coincidence number one was that Williams' wife was working for a London window replacement company called SAS Conversions. Coincidence number two was that the boss of SAS Mike Aylett was an Amstrad owner looking for a vertical program for his business that didn't cost an arm and a leg.

The outcome is that Williams has written a package designed for Aylett's company, but which can be easily tailored to meet the requirements of similar small firms in the double glazing field.

As Mike Aylett put it: "We are not another Everest, and my partner and I

By TONY LEAH

enjoy our social life too much to want to be. But I'm a reformed Spectrum owner with a CPC 6128 at home and another Amstrad in the office, and I believed the latter was capable of doing much more than basic word processing and clerical work".

Williams' solution is the Quaff – Quotation And Full Fabrication – system which costs less than £700 including a PCW8256, compared to similar vertical packages for the industry which start at £1,400 and go up to £4,000.

The package is tailored for the small aluminium replacement window business to save time and reduce waste by eliminating complex calculations.

Management information about costings and labour time is generated, and a professional quotation is produced for the customer – all replacing the old pen and paper methods previously used.

There is even a cutting sheet printout for the fabrication shop and glass department, and a list of the details and sizes on which the job is based. Prices are presented as record cards and are easily updated for both cost and profit margin. They can also be printed out as a directory showing

→

the information on file for glass window types and extrusions.

Very user friendly, Quaff takes the operator through each stage keeping the required keyboard skills to a minimum. Many of the commands are by single key strokes, though numbers for sizes and prices obviously need typing in full. At the completion of each stage the user is given the option to check for errors.

To create a window the user chooses from one of the styles available and inputs the outer frame measurements. The computer then draws the outline of the window to scale on the screen and asks for details of the individual panels, if any.

When details of opening vents and hardware are fed in the computer adjusts the drawing to scale. Glass specification and types of handle are also input. The program calculates for supply and fit, or supply only, with details of subframe charges and labour. If the order is for several windows, details of the individual units can be merged into one estimate and factory order.

Quaff is supplied on a double sided, single density 3in disc with an instruction manual. The system reports display the company name and addresss of the purchaser and the passwords and security codes are defined by him before installation.

The basic system is intended to operate for a single profile of aluminium extrusion, with off-sets specified by the user. Details of labour charges and materials costs entered by the user need not be disclosed on print outs.

Williams strongly urges his clients to take advantage of his training and installation option which involves one day on-site and costs £50. Support contracts are available for the software and also for the computer equipment purchased from C.J. Systems, as well as telephone help and advice.

Williams reckons his system should pay for itself in a matter of months if for no other reason than its accurate costing and pricing of every order.

Mike Aylett is enthusiastic about the benefits of Quaff. "It reduces the chances for human error", he said. "If you made a mistake early on in a quotation using pen, paper and calculator it would magnify itself as you went on. Now we are able to get all our figures right first time.

"Windows make up about one third of our business and they are all made to measure. No two are exactly the same, so we've got that terrific room for error. But using Quaff means we can feed in all the basic details about measurements and materials and get back a detailed worksheet right down to the lengths of aluminium and wood to be cut and the sizes of the glass panes.

"Another nice touch is that we no longer have to submit an estimate with a crude hand-drawn sketch of the job, or use a form with a standard box diagram bearing no relationship to the actual shape of the window the customer wants.

"With Quaff the computer draws the window to scale in the actual shape it will be, making it much more attractive to the customer who sees exactly what the finished job will look like, complete with fastenings and other details".

C. J. SUPER-SYSTEMS LTD.

38, HODDER DRIVE, PERIVALE, MIDDX TEL: 998 9257

CUSTOMER REF :- 4C2504 DATE 25/04/86

CUSTOMER NAME:- MR. JOHN SMITH

COMMENT/LOCATION UPPER FRONT BAY

CUSTOMER QUOTATION

To Supply Fabricated Window Unit(s) to the Following Specifications.

4 WINDOW(S)

1200 MM

[350]

A /\ AREA A = VENT, TOP HINGED

B AREA B = FIXED PANEL

890 MM

FRAME TYPE : Solid - Equal Leg.
DOUBLE-GLAZED 4 MM Clear X 4 MM Clear
LOCKING HANDLES Fitted.
OPENING VENTS 1
SUPPLY AND FIT, WITH NEW SUB-FRAME.

PARTS AND LABOUR	£ 868.80
+ VAT @ 15 %	£ 130.32
TOTAL COST	£ 999.12

This Quotation is Valid for 30 Days from the above Date.
If you have any queries , Please Telephone the above Number.

A customer quotation provided by the system

SMART BY NAME AND NATURE...

GABRIEL JACOBS reviews Smartkey II

SMARTKEY II is a neat little key-configuration package which will run under CP/M on the CPC6128 and the two PCWs. It sits invisibly in memory, taking up about 7k of the transient program area, and enables you to alter the codes assigned to any key on the keyboard.

The assignments available are unlimited – control codes, commands with pauses for input if required, single characters, phrases and even whole paragraphs. The program can therefore be used very effectively both at the level of the operating system, and with all kinds of applications software from word processors to business planners.

With one keystroke you can generate chunks of frequently used text or foreign characters which require a backspace sequence, save or load a file, find a record in a database, display a window in a spreadsheet . . . the possibilities are endless.

You can also extend the effective range of the keyboard by "Supershifting" keys without altering their other values. So after defining a key as the Supershift character itself, you can assign Supershift definitions to any other keys. Thereafter any keystroke following that of the Supershift character will generate the extra definition.

The changes you make can be merely temporary, for a particular job in hand, or they can form sets of definitions which can be saved, then tagged to a particular piece of software and included in a boot-up file.

All in all, Smartkey is in effect a souped-up version of CP/M Setkeys with a dash of the Submit utility. But the package is not only more powerful than Setkeys, it is also far easier to handle.

Redefining a key is simply a matter of pressing the Setup key – the default is left squared bracket, but this can be changed – followed by the key to be redefined and the new definition. The original values of both the Setup and Supershift keys are not lost – they can be generated within an applications package or at the CP/M prompt by pressing the key twice.

Once a key is defined it becomes effective immediately, so you can experiment with your keyboard until you are satisfied with it, then save all the definitions to disc in one operation.

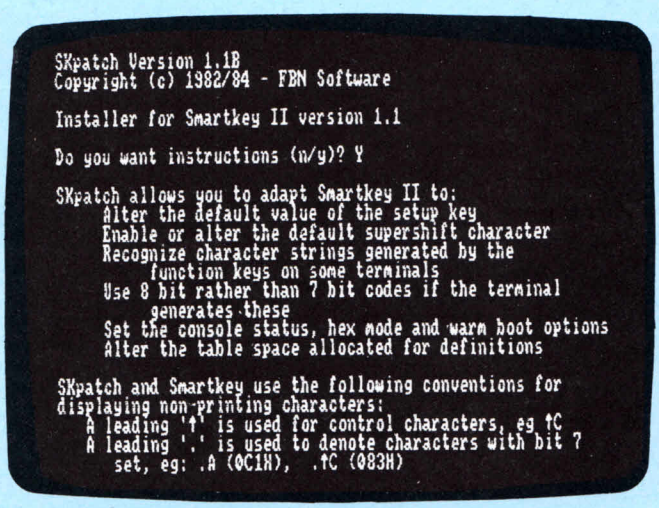
As you would expect from Caxton, the instruction manual is exceptionally well organised and clearly written. It does, however, contain one error which I had better point out for those who intend to rush out and buy the program.

You will save yourself a phone call to Caxton (or a letter to Postbag) by changing Step 11 in Section 3-10 from "Press Return" to "Press Setup". At least on Amstrad machines, pressing Return at that point (while programming a Pause in a command line) will throw the whole sequence out of joint.

It's also worth mentioning that although the Supershift key is supposed to be ready-implemented as a left curly bracket instead of the default backslash (which is not available in a single key on the Amstrad PCW), it has to be redefined using a subsidiary program called SKPATCH before it will work.

But these are minor irritations. Caxton's own description of Smartkey as "a stroke of genius" is perhaps more a clever play on words than an unbiased assessment, and at £49.95 the program can hardly be called exceptionally good value for money, but on the whole it does a useful job very well indeed. What is more, you can get it free if you buy Caxton's enhanced spreadsheet, Scratchpad Plus, at £69.99 while the offer lasts. It then becomes a genuine bargain.

One final, but important point, unfortunately Smartkey will not work at all with LocoScript, which bypasses CP/M. Pity.



The SKpatch installation option menu

Product: Smartkey II
Price: £49.99 (inc VAT)
Supplier: Caxton, 10-14 Bedford Street, Covent Garden, London WC2E 9HE. Tel: 01-379 6502

PRINTER WITHOUT SWITCHES

JOHN HUGHES explains how easy it really is to use the PCW printer

Although the PCW is thoroughly documented owners have been finding Mastering the PCW 8256/8512 by JOHN HUGHES a far clearer, user-friendly introduction to their micro.

We're grateful to the publisher Sigma for permission to reproduce its section on using the printer to show you just how simple it makes things.

And for those of you who know a good thing when you see it there's a special offer to readers on Page 66.

THE printer included with the PCW8256 differs from most others in that it is almost entirely lacking in external controls or switches. This is because it is controlled by the program stored in the computer at any given time.

One result of this is that all the printer's facilities can be manipulated directly from LocoScript and all the LocoScript commands are compatible with the printer.

If you have never had to install a printer for a particular computer you may not appreciate what a great advantage this is: Underlining, italics, special symbols – in short, everything which can be produced on the screen – is all printed accurately and without fuss.

Printer control is entered through the Ptr key, located to the left of the Exit key on the bottom row of the keyboard. This key puts LocoScript into printer control mode and lists the printer control commands on the status bar. Alternatively the same effect can be obtained by using the paper-feed knob, or pulling the bail bar forward.

Provided that none of the pull-down printer control menus are being displayed you can return at any time from printer control mode to the editing text or the disc manager screens by pressing the Exit key. If you want to abort the use of a menu use Can in the usual way.

It must be admitted that the operation of the various function keys in printer control mode is not as clear as it might be nor are the various key names which appear on the status bar terribly helpful.

Because of this it is wise to develop a fixed procedure which can be used for setting up the printer at the beginning of each printing session. Careful adherence to this routine will ensure that your documents always look as you intended them to.

Begin by checking the one control which is actually located on the printer itself, namely the strength of impression lever.

This is located inside the printer case on the right (see Figure I) and to ensure maximum clarity it should be set to low pressure – pulled back – for single sheets and to high pressure – pushed forward – for multiple copies. Select the lowest pressure consistent with good quality.

The next task is to set the printer for the size of paper which will be used and the quality of the printout required. This is done by pressing *f1* options which displays a pull-down menu allowing the user to choose between draft and high quality mode, and between continuous stationery and single sheets – see Figure II.

You can select the options you require by the usual combination of moving the cursor bar and pressing +. If you change the form of stationery LocoScript will suggest a suitable length in lines which you can amend if necessary.

Line pitch is usually set at six lines to the inch, though eight lines to the inch can be selected – this is described later. Single sheet length is

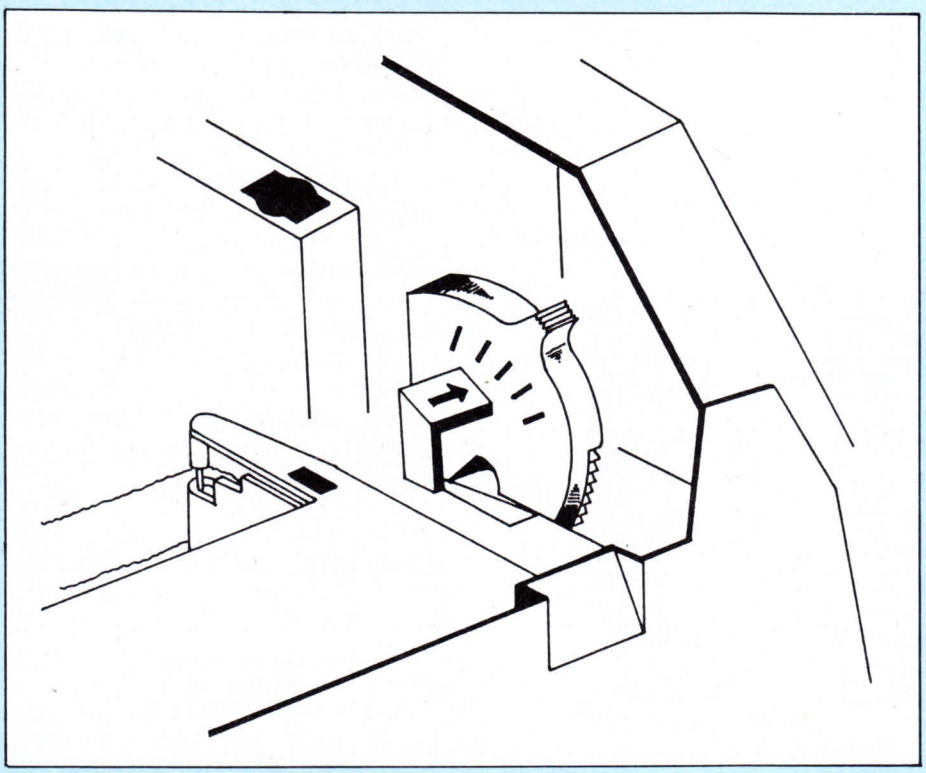


Figure I: Strength of impression lever

pre-set for A4 paper and continuous stationery length is pre-set for standard 11in by ½in perforated paper.

When the paper has been loaded using the paper-loading control it may be necessary to command the printer to carry out certain additional tasks. This is done with *f3* Actions. This menu – see Figure III – offers two external printer actions and one internal action.

The first external action forces the printer to advance the paper by either one line or one page. Locate the cursor bar over the appropriate choice and press +. Several lines can be advanced by pressing + repeatedly, but the key does not auto-repeat.

The second external action is to move the printhead away from the left-hand edge of the paper by a certain number of characters. Put the cursor bar over the bottom line of the menu, enter the new value and press +.

The internal action resets the top-of-form position. This may need to be done when continuous stationery is used and the paper has been manually advanced in order to inform the printer that the current paper position is at the top of a sheet.

In every case press Enter to confirm your choice and leave the menu. The printer control facilities already described are all that you will normally use – indeed if you standardise on A4 paper you will rarely need to change any of the settings from their initial values, except perhaps to alter print quality.

When you are happy with the settings use Exit to leave printer control. Subsequent printing of your document when Exiting from an editing session, or with P from the disc manager screen, will be done according to the settings you have already given, though you can revise them at any time by re-entering printer control mode with the Ptr key.

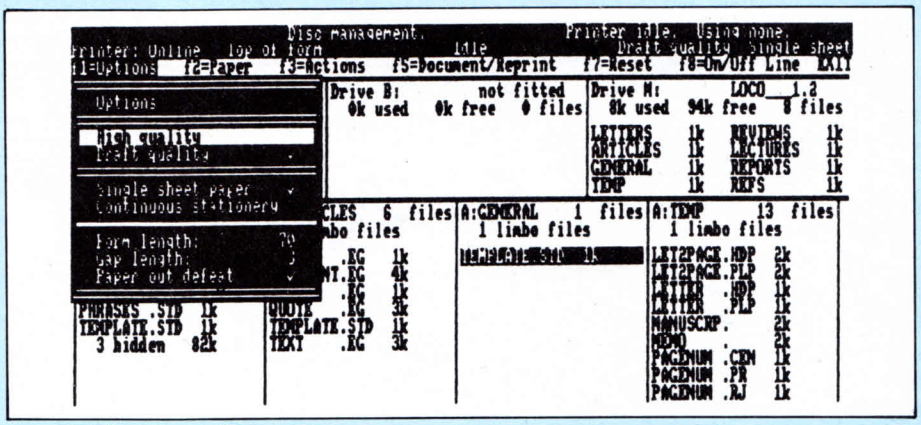


Figure II: The print options menu

Thus if anything goes wrong while printing is taking place, such as the paper tearing or jamming, the Ptr key will stop the printer almost immediately. Although the printer will stop when you press Ptr it will start again as soon as you press Exit. To prevent an immediate restart you can put it off-line, in other words on stand-by.

This is done with *f8* on-line/off-line. When the printer is on-line it responds to signals from the PCW8256 and when it is off-line it does not. Pressing *f8* toggles the printer between the two states – in other words it changes it to the other state regardless of what state is current. If you go off-line during printing you must both restore the on-line state (f8) and signal – using Exit – that you are ready to proceed before the printer will restart.

Occasionally you will need to abandon printing in the middle of a document – perhaps because you realise that there is some error in it which you need to correct. This cannot be done using the Ptr key alone. Instead press *f7* (Reset), which is also used to reset the printer and return the print head to its rest position.

An advanced feature of the PCW8256 is that the printer can be working on a copy of one document at the same time as you are editing another document on the screen. The document/reprint menu (Figure IV) reached by pressing *f5*, gives you details of the document which is being printed or informs you that the printer is not in use.

A second purpose of this menu is to allow you to reprint either the previous or the current page of the document being printed or the whole text of it in case the paper has been misaligned or something else has gone wrong with the printing.

The only other occasion on which you may need to return to printer control mode is when the printer stops and the status line displays the

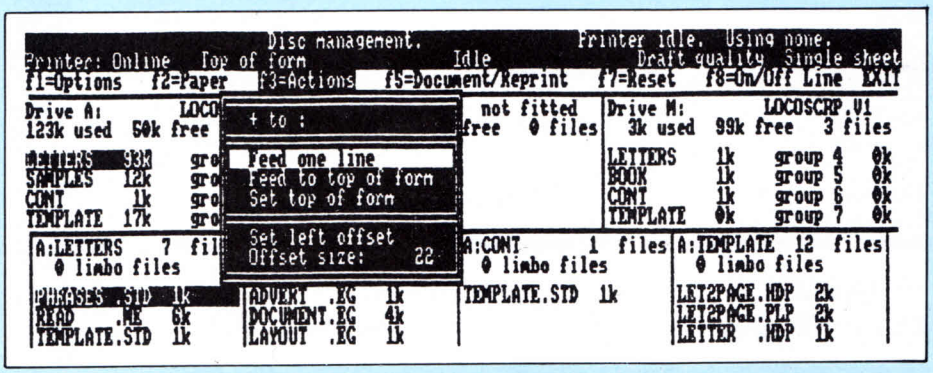


Figure III: The print actions menu

message "Waiting for paper", although paper is in fact loaded.

This may happen if you are using continuous stationery but have not set f1 accordingly, or if you have wound the paper in by hand without using the automatic paper feed. Pressing *f2* (Paper) and Enter will enable the printer to continue.

When using single sheets the printer stops after every page to enable you to insert more paper. Inserting the new sheet automatically returns you to printer control mode, so when the new sheet is ready press Exit to start printing the next page.

Finally an image of the screen at any time – a screen dump – can be obtained on the printer by holding down Extra and then pressing Ptr This is always in high-quality print, and the print image is reversed from the screen images, for example light green areas on the screen appear as

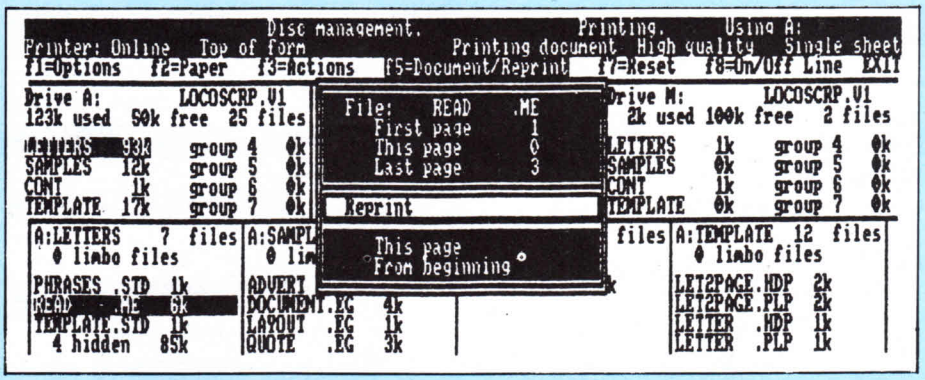


Figure IV: The document/reprint menu

black ink and vice versa. All the screen images in this article were obtained in this way.

Many printer manufacturers suggest that over-use of so-called bit-image printing of this sort can eventually result in damage to the print head. We are not aware of any such problems with the PCW8256 printer, but caution is probably wise. Certainly prolonged printing of any sort can generate considerable heat at the print head, so be sure to keep your fingers well clear of it.

Many LocoScript commands can be given in several different ways. In general it will help you to work faster if you use the methods which require the fewest key-strokes. At first however you will find it easiest to use the function key menus freely, and only move on to the Set and Clear menus, and to give direct commands, when you feel ready.

To clean or not . . .

I HAVE occasionally seen adverts for disc drive cleaners, although never for the 3in size used by Amstrad, and I am wondering whether I ought to clean the drive on my PCW8256.

Is this necessary and if so where can I get a suitable cleaner? – **Peter Owen, Liverpool.**

● A head cleaning kit usually consists of a disc made of an absorbent lint-free material and a bottle of cleaning fluid, such as fluorocarbon and isopropyl alcohol preparation.

You put the disc soaked in the fluid into the drive and activate the read/write head for a few seconds – by giving some Directory commands, say.

You have to take particular care if there are children around as many of these fluids can be nasty if they get into the eyes.

With all the makes of drive cleaners we have used the instructions have insisted on the absolute necessity of frequent cleaning to avoid sudden data loss (daily for heavily used drives, and under no circumstances less than once a week).

For years we took such advice deadly seriously, and spent a small fortune on keeping the innards of the drives spotless.

In countless hours of disc access using overlay files on an IBM PC and a Sirius we only ever had two discs so badly corrupted that we could not recover the files they contained, and on both occasions the culprits were faulty tracks.

Proof enough of the effectiveness of regular cleaning?

Well we later acquired an ACT Apricot, which uses 3½in discs something like the 3in variety for the PCW, and did not immediately come across a suitable cleaning kit.

So we did not bother for over two years – and never had any trouble whatsoever.

We bought a PCW8256 when it first appeared, and it has been in daily use ever since. We don't think it's an exaggeration to say that the drive has been given a real hammering.

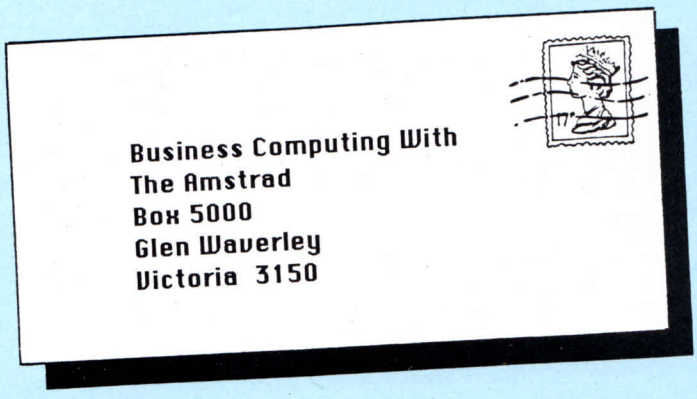
On top of that, we work with people who smoke pipes and cigarettes, and the desk is a dust trap.

Never once has either the drive or a disc let us down.

So in our experience cleaning kits are a waste of time and money, falling into the category of the short-handled geniune squirrel hair brush customised for removing dirt from between the keys on the keyboard.

PCW drives are protected from dust by a flap when the disc is removed, there is no fan to suck smoke into them, and there is nothing in the documentation to suggest that a cleaning fluid need be applied to them (and it does recommend using a vacuum cleaner on the printer!).

Fortunately PCW owners would appear to have little option but to leave well alone as we know of no manufacturer who produces cleaning kits for 3in drives.



Search facility

GABRIEL Jacobs' comparison of LocoScript and WordStar in your April issue seems a very fair review.

I have been using a dedicated office word processor machine running on a simplified WordStar type of software for some years and I have been using a PCW8256 at home for the past few weeks.

LocoScript is much easier for a beginner to use but it does have some shortcomings. Putting in a series of hard spaces seems particularly cumbersome.

The reviewer's criticisms of LocoScript's Find facility are also justified, but I should point out that the system can be made to find whole words only by typing a space at the start and end of the find text prompt.

So although a search for "bat" will also stop at "battle", "abattoir" and "acrobat", it will not do so if one asks it to find " bat ".

I do wish that one could make a single search for a word whether in upper or lower case.

In terms of value for money I think LocoScript v1.2 is marvellous. Maybe later versions will improve on these relatively small faults. – **M.H.Evans, Linton, Cambridge.**

● Your method of finding whole words is unfortunately, too unreliable. "Space + bat + Space" will find "bat" of course, but it won't find "bat.", "bat;", "bat," or other combinations you can think of.

What you want out of a global Search and Replace is the sure knowledge that everything has been found and changed.

The method you suggest requires either a visual check or several consecutive searches containing every possible combination.

We also hope that future versions of LocoScript have an improved Search facility. Life would be so much easier.

Spreadsheet pitfalls

I WRITE in answer to the spreadsheet feature in the May Business Computing with the Amstrad *and have a few comments to make from my experiences with the 6128.*

All user manuals are a trap for the unwary and uninitiated. Practical examples are worth 100 explanations. Often it is what they don't say that's the most important, and that is true here.

Supercalc 2 solved my financial modelling problems, but not quite in the way I expected. So I thought others would benefit from the pitfalls I fell into.

The manual is pretty comprehensive and very informative, but many answers had to be found by trial and error with little cross referencing.

The most valuable part was the Answer Card summary at the back. After 15 years of reading software manuals I have been brainwashed into expecting inadequacies, as the writer of manuals is unable to put himself in the place of the user or predict his requirements.

My first problem was I had no usable £ sign, as this was disabled. The people who sold me the package at the Amstrad Show in January said to use the hash symbol which I found later to be correct.

I set out to draft a complete spreadsheet with index, management expenses for four years, monthly capital/interest/product for three types of capital (shares being nine types, mortgages and investments), depreciation, corporation tax, ratios for liquidity, reserves and growth followed by profit and loss and finally the balance sheet.

After I had reached the nine types of shares using 43 columns the input delay

became 10 seconds with the weight of formulae that I had entered.

By the time I had set about the balance sheet I had loud warnings of 'Memory full' so my 31k had disappeared.

So far I had made no data entries, and had used about one third of the spreadsheet.

My first call to Amsoft was misconstrued as disc being full and on the second call the next day it was erroneously explained that blanking out filled the memory.

I also used the delete option which deleted the lines, but unsuitably moved them up on space. Not what I wanted at all.

The whole plan needed a rethink, which resulted in the model being divided into three spreadsheets with 5k, 15k and 17k each saved from their original locations.

Saving was another experiment only answered in the Answer Card, where you learn to save as a block from top left hand corner to bottom right. Maybe the % sign does work but I ended up by creating brackets separated by minus and plus with long formulae on the three occasions they were used for ratios.

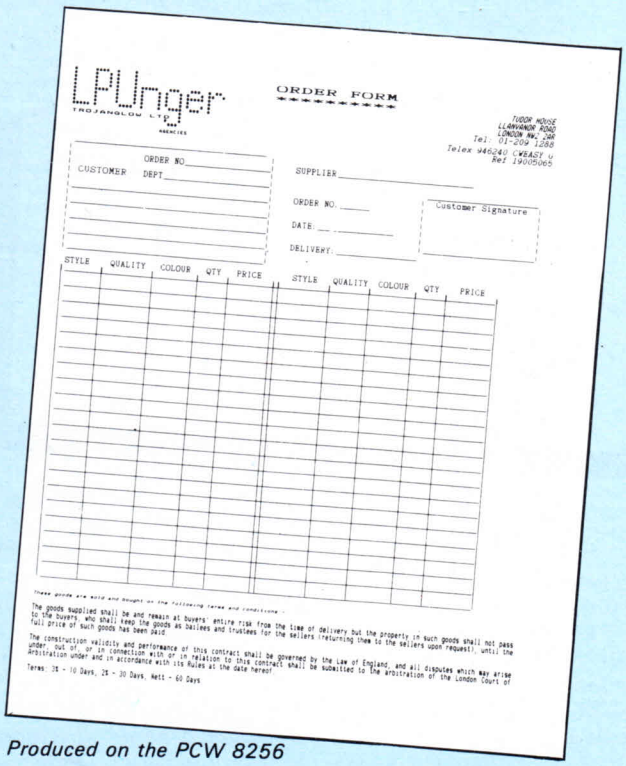
It was only at the end of my creation that I discovered that the program calculates sequentially from top to bottom.

So any result that had to be transported to a higher position needs an ! to force recalculation from the top. It is as well to enter a footnote to avoid disastrous results. – **Michael Varney, Battle, East Sussex.**

PCW 8256 in action

AS one of the first UK purchasers of the Amstrad PCW8256 I take pleasure in writing to you with regard to the use to which we have put this versatile system.

We are an import/commission agency and our plan was to use the 8256 as a basic accounting computer to go on line by July of this year giving us adequate time to establish our requirements and purchase the most suitable software and additional hardware if required.


LPUnger
TROJANGLOW LTD.

ORDER FORM

TUDOR HOUSE
LLANVANOR ROAD
LONDON NW2 2AQ
Tel: 01-209 1288
Telex 946240 CWEAST G
Ref. 19005065

ORDER NO.
CUSTOMER DEPT.

SUPPLIER

ORDER NO.
DATE:
DELIVERY:

Customer Signature

STYLE	QUALITY	COLOUR	QTY	PRICE	STYLE	QUALITY	COLOUR	QTY	PRICE

These goods are sold and bought on the following terms and conditions:-

The goods supplied shall be and remain at buyers' entire risk from the time of delivery but the property in such goods shall not pass to the buyers, who shall keep the goods as bailees and trustees for the sellers returning them to the sellers upon request, until the full price of such goods has been paid.

The construction validity and performance of this contract shall be governed by the Law of England, and all disputes which may arise under, out of, or in connection with or in relation to this contract shall be submitted to the arbitration of the London Court of Arbitration under and in accordance with its Rules at the date hereof.

Terms: 7% - 10 Days, 2% - 30 Days, Nett - 60 Days

Produced on the PCW 8256

Since the PCW8256 was supplied with LocoScript we decided to first fully investigate the system as a word processor.

As a result of the power and flexibility of the 8256/LocoScript combination we feel that the system has already fully paid for itself as a word processor and are considering purchasing separately the new PCW8512 for our accounting needs.

Where previously we had produced hand-written documentation we are now able to produce far more professional results with regard to credit notes, debit notes, statements to suppliers and customers.

Additionally, since all our price lists are stored on disc, we are in the position of knowing that as long as we have paper in the office we can run off copies of any or all of our price lists as required.

This begs the obvious question – what about out photocopier? Since we work with a restricted customer base and we only have a tiny, slow, coated paper copier our output from the 8256, albeit slower, looks much more professional.

Finally, our most important use to date – order forms.

With our restricted customer base and the variations in detailed information such as size scales which are dependent on a particular supplier's products we have produced order forms which exactly meet our requirements and which can easily be varied if those requirements alter.

An example of one of these order forms is enclosed. It is in NLQ but our normal copies are produced in draft. The origination of this order form took me around four hours work one evening earlier this year and I consider that time most profitably spent.

While I am typing this letter to you our printer is merrily producing further order forms for tomorrow's customers.

On the debit side, the system is slow in most of its operations and we would dearly love a mailmerge program to tie in with LocoScript. However at £399 these are shortcomings which we are happy to live with.

We look forward to the arrival of the PCW8512 when we have selected the appropriate accounting software. – **N.M.D. Unger, managing director, Trojanglow Ltd, London NW2.**

Forum for Joyce

THANK you for the new Amstrad Business Computing *in your magazine. One hopes that it may become a forum for queries and answers about the inestimable Joyce (I must be very green, but how did she come by this name?) and presumably, in the fullness of time, her more formidable sister.*

I think what we users would like from you would be some tips of how to tailor Locoscript, as one can Wordstar, to one's own needs by altering default settings.

For instance, among other things I would prefer to have a default gap setting of 1 rather than 3 for A4 stationery.

What pray, is a Unit? As far as I can see it is a simple text place marker, but am I missing something? I can find no reference to it in the manual at all.

And the tricky bit with headers and footers. Quite apart from the mix up in the manual, it took me some time to realise that Joyce reads a document backwards.

So if you are only doing a one pager, she will treat that page as the last rather than the first if you are not printing a header.

To get the right first page footer it seems you have to put in an 'End page here' and at least a CR on page 2. Then, of course, you have to remember to reset the printer when you have printed your single page. **– R. Villiers Lister, Basingstoke.**

● As far as we know Joyce was – and perhaps still is – a respected member of the team that produced the PCW8256 specifications. The machine was called after her as a kind of thank you.

As for customising LocoScript, we're afraid it's bad news. Because the software is not supplied with an Install program, there is simply no way for the ordinary user to

→

re-configure the default values.

All we can do is put pressure on Amstrad to produce a list of patch areas in the LocoScript command file, or hope that someone has the patience – and it will need a lot – to discover which values need changing, and how to change them using something like the CP/M Debug utility.

We'll keep you informed of any developments.

A Unit is, as you say, nothing more than a place marker, and the facility is briefly described on page 11 of the addendum which comes with the LocoScript manual.

The Unit key will send you to the next marker, and Alt + Unit to the previous marker. Unlike many other word processing packages, in LocoScript Units are saved with a file. They are ignored by the printer.

You're right about the difficulty with footers on single page documents, and your solution is the only one we know.

But you don't have to remember to reset the printer with f7 – the program will let you know you've forgotten next time you come to print.

Unfair table

GABRIEL Jacobs' article in the April Business Computing comparing LocoScript with Wordstar seemed fair and reasonable – until I came to Table I where features are compared.

I don't know what version of Wordstar Mr Jacobs uses but it must be truly antique.

The version I use (version 4 for the IBM) is the best, as you can see both underlining and bold print displayed on the screen, but even the old CP/M version did have the facility to store ruler lines, reform paragraphs and most of the other features the author denies it. The approach is just different.

Personally I miss the file merge facility within Wordstar, which as far as I know can't be recreated with LocoScript, and anyone who can afford an IBM would probably buy Business Wordstar with the mailmerge and indexing facilities.

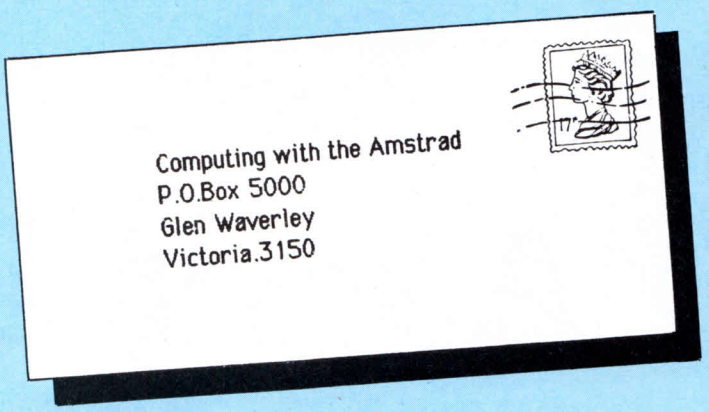
I agree that the Amstrad dedicated printer means that fancy printing is infinitely easier than with Wordstar – but please make fair comparisons about the other features. After all, in terms of value for money, it's a one horse race, so even an Amstrad user magazine can afford to be magnanimous. – **Juliet Cairns, London SE3.**

● I don't mean to be partisan or ungenerous in my assessments, but unfortunately the version of WordStar implemented on the PCW is the old CP/M Version 3, which will not handle any of the non-scoring features in the table.

I wish Version 4 was available for the PCW. I use it myself on an IBM clone, and it's fantastic.

I hear that a new version of Pocket WordStar is soon to be released, with many added features. My scoring would then have to change accordingly.

Gabriel Jacobs



Praise for Microscript

I FOUND your article on word processors very interesting. I found myself comparing my word processor, Microscript running on my CPC 464, with WordStar and New Word. I think it better than Pocket WordStar but has perhaps less features than New Word. I am surprised that with Locoscript you cannot delete a whole line in one go and that Pocket WordStar uses only one ruler. I don't see how anyone could live with such limitations.

But on to the main point. Why no review of Microscript, or don't you consider it to be in the same class? I paid only £19.95 (at Comet) for a very powerful and flexible program.

It has 10 standard rulers, plus as many others as you wish to create within the document for total flexibility of layout.

You can even use a ruler as a five function calculator plus horizontal and vertical totalling – great for doing invoices. You can even leave them in, as they aren't printed.

User defined and standard codes can be embedded in the document for printing and you can see exactly what you will get.

The printing facilities seem endless to me as I was used to Easi-Amsword. Start and stop printing at any page, underscore and double strike even on the DMP1!

In summary, with Microscript you can cut and paste, reformat, use as many rulers as you wish, use temporary memory stores for often-used phrases, change to upper or lower case without retyping, lift and put down a line, move around the documents sensibly using the four arrow keys, merge files wherever you want to and in any format.

I've even used the search and replace function to debug a Basic program (saved using SAVE "NAME",A). The list is endless. It's probably easier to say what Microscript can't do.

The only thing I can think of is a spell check and global, rather than manual, reformatting.

The documentation misses out the important aspect of customisation, for instance how to get rid of the awful black letters on a blinding white background.

Amsoft supplied by return of post full details of how to customise the System Information File.

It's also a little slow because of the use of overlay files but a DK'Tronics ram disc should cure that. All in all a very good product and worthy of a mention in your magazine. – **Alex Aird, Birmingham.**

● We have been inundated with letters about stored ruler lines, and in last month's issue explained what we meant by them. So let's clear up any final misunderstandings and then call it a day.

Most word processors, like WordStar and MicroScript, will let you insert any number of non-printing ruler lines into a file.

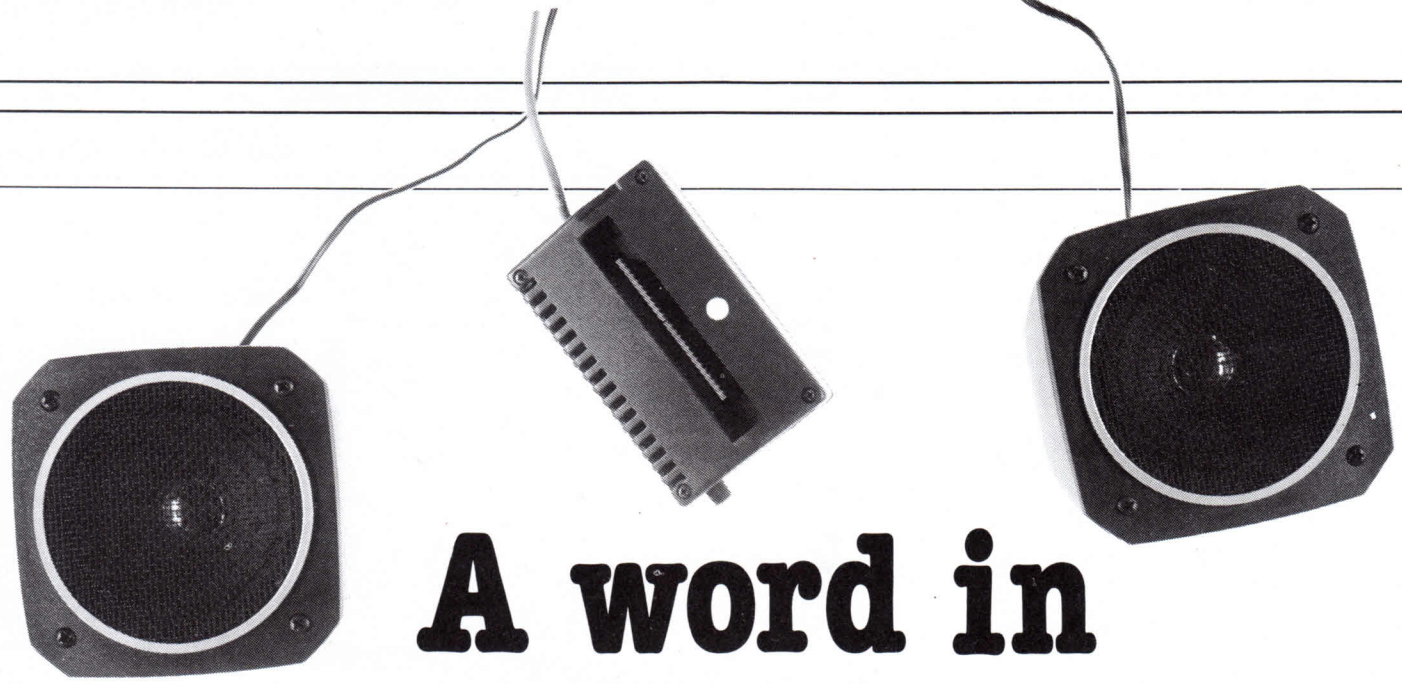
But advanced programs, like LocoScript and New Word, will allow a global reformat which takes these embedded ruler lines into account.

That's the difference, and it's an important one if you have many layouts in a single document.

On this score, incidentally, LocoScript goes much further than New Word. It will allow you not only to insert a new ruler line, but also a complete new layout containing information on margins, pitch, line spacing, line height, justification, italic script and so on.

Each layout is taken into account as the cursor passes over it, whether the codes are displayed on screen or hidden.

MicroScript is not in the New Word class when it comes to complex page layouts, including columns, nor in the LocoScript class for mixed typestyles, but it certainly represents excellent value for money, no doubt about that. Jo Stork reviews it in this issue.



A word in your ear..

ROBIN NIXON reviews the DK'Tronics Speech Synthesiser

WITH DK'Tronics newly released ROM version of its Speech Synthesiser you can add speech and stereo sound to your Amstrad accessible from the moment you turn it on.

The package consists of the standard style DK'Tronics add-on box, two four inch speakers and a comprehensive manual. There are versions for both the CPC 464 and 6128.

The synthesiser uses allophones, the smallest parts that speech can be divided into, to create speech. The ones used in this synthesiser are shown in Table I.

Fitting it takes 30 seconds and once connected you can test whether it's working by typing:

```
|SPEAK
```

If the system is working it says: "DK'Tronics Speech Synthesiser." It then displays a list of new | (bar) commands it provides. These commands, shown in Table II control the speech.

On the ROM version as well as the bar commands provided by the cassette version you get two new ones, |LEFT and |SAY. |LEFT returns the amount of free space left in the speech buffer.

To use |SAY you might, for example, type:

```
A$="Hello everyone"
|SAY,@A$
```

A$ will then be converted into a sequence of allophones by the ROM software which is then spoken by the speech synthesiser.

The speech chip itself is located at &FBFE. You can send data straight to it, but you must have previously worked out the allophones required and turn them into data.

You also have to check when the speech chip has finished saying each allophone by reading location &FBFE until its value is less than 128. Program I shows the idea.

```
10 REM Program I
20 FOR x=1 TO 8:REM Amount of data
30 WHILE INP(&FBFE)>127
40 WEND:REM Wait until ready
50 READ a
60 OUT &FBFE,A
70 NEXT x
80 END
90 DATA 27,7,45,53,3,18,47,0:REM Hell
o there
```

Program I

The allophones used by this program are:

HH1	EH	LL	OW	PA4	DH1	XR	
27	7	45	53	3	18	47	0
H	e	ll	o	(sp)	th	ere	.

PA4 is a 100 millisecond pause which is used for separating clauses and sentences. The 0 at the end tells the synthesiser to stop talking – if only some humans had this facility!

The method illustrated in Program I would be useful for machine code, but for Basic programmers the command:

```
|FEED,27,7,45,53,3,18,47,0
```

would achieve the same effect.

In fact a lot of time and effort can

|SPON	Switches on interrupts to read the speech buffer.
|SPOF	Switches off interrupts and stops reading the speech buffer.
|FEED,n	Feeds bytes directly to the speech buffer.
|FLUS	Clears the speech and text buffers.
|SPED,n	Controls the speed at which text is spoken.
|OUTM,1	Sets the mode where text printed in the following format: `PRINT "'Some text'"` will be spoken but not output to the screen.
|OUTM,2	With this option all print outputs, including LIST, will be spoken but not printed on the screen.
|OUTM,3	When this is set all printed output is both spoken and printed on screen.

Table II: The commands provided by the Speech Synthesiser ROM

Pauses

0	PA1	(10 mS)	use before voiced
1	PA2	(30 mS)	stops & affricates use before voiced
2	PA3	(50 mS)	stops and affricates before voiceless stops & voiced fricatives also between words
3	PA4	(100 mS)	Between clauses and sentences
4	PA5	(200 mS)	Between clauses and sentences

Short vowels – these can be repeated

7	EH	E	b**e**nd
12	IH	I	**i**tting
15	AX	U	s**u**cceed
23	AO	AU	**au**ght
24	AA	O	c**o**t
26	AE	A	f**a**t
30	UH	OO	c**oo**k

Long vowels

5	OY	OY	t**oy**
6	AY	Y	sk**y**
19	IY	E	s**ee**
20	EY	EA	gr**ea**t
22	UW1	O	t**o**
31	UW2	OO	f**oo**d
32	AW	OU	**ou**t
53	OW	OW	sn**ow**
62	EL	L	**a**ngle

R-coloured vowels

47	XR	AI	h**air**
51	ER	ER	comput**er**
52	ER2	IR	b**ir**d (monosyllables)
58	OR	OR	st**or**e
59	AR	AR	f**ar**m
60	YR	R	cl**ear**

Affricates

10	JH	J	**j**ury
50	CH	CH	**ch**urch

Resonants

14	RR1	R	**r**ead
39	RR2	R	bra**i**n
49	YY1	U	compu**t**er
25	YY2	Y	**y**es
45	LL	L	**l**uck
46	WW	W	**w**ool

Voiced fricatives

18	DH1	TH	**th**ey
54	DH2	TH	ba**th**e
35	VV	V	e**v**en
43	ZZ	Z	**z**oo
38	ZH	GE	bei**g**e

Voiceless fricatives

29	TH	TH	**th**in
40	FF	F	**f**ire
55	SS	S	**s**at

(29, 40, 55, double for initial position)

27	HH1	H	**h**e
57	HH2	H	**h**oe
37	SH	SH	**sh**irt
48	WH	WH	**wh**ip

Voiced stops

28	BB1	B	r**ib**
63	BB2	B	**b**ig
21	DD1	D	cou**ld**
33	DD2	D	**d**o
36	GG1	GU	**g**uest
61	GG2	G	**g**o
34	GG3	IG	wi**g**

Voiceless stops

17	TT1	T	**t**ime
13	TT2	T	i**t**s
42	KK1	C	**c**omputer
41	KK2	K	**k**ey
9	PP	P	**p**ub

Nasal

16	MM	M	**m**ilk
11	NN1	N	**n**o
56	NN2	N	ba**n**s
44	NG	NG	ea**rn**

Table I: The allophones used by DK'Tronics Speech Synthesiser

be saved by using the ROM's in-built text to speech converter. With this you could replace Program I and the above example with:

```
PRINT "'Hello there'"
```

But, as DK'Tronics points out, there are many more exceptions than rules in the English language, so a number of words will be mispronounced. For example, it is difficult to program exceptions such as the following:

Word	Pronunciation
Rough	*(Ruff)*
Bough	*(Bow)*
Bow	*(Bo)*
Cough	*(Coff)*
Furlough	*(Ferlow)*

However using these pronunciations like this:

```
PRINT "'COFF'"
PRINT "'RUFF'"
```

you could get around any difficulties.

I mentioned that the system provides you with stereo sound. In fact the Amstrad already has this. What the speech synthesiser does is provide you with a good quality stereo amplifier and two speakers to obtain full stereo.

In my opinion for a price of less than £40 the DK'Tronics Speech Synthesiser is an excellent add-on which has many uses ranging from playing games to education and providing blind people with access to computing.

Psst... wanna colour that isn't strictly legit?

IAN SHARPE has some . . .

XCOL is a program that allows you to display more than the legal number of colours on the screen in any mode on the CPC series.

Impossible? Not quite. Perhaps you've seen the mode switching trick used in Sorcery that sets the upper part of the display to Mode 0 and the lower part to Mode 1? Well this program does a similar thing with palette switching so that the values of the inks may be varied on different parts of the screen. It doesn't change mode to do it.

Type in Program I which is the Basic loader for the machine code. Save it before trying to run it because it will erase itself from memory once debugged.

You will now find three extra commands added to Basic – ¦XCOL, ¦XDIS and ¦SETHT,n where *n* is a parameter. ¦XCOL enables the new facility, ¦XDIS disables it and ¦SETHT allows some control over the position of the switching.

Having typed ¦XCOL you will find that the normal INK and BORDER commands act in a slightly different way. Normally issuing an instruction such as BORDER 0,1 would set the border flashing between colours 0 and 1 – black and blue.

With XCOL enabled the same command would set the top part of the border to the first colour and the bottom part to the second. INK 1,22,26 will produce a similar result with anything written with PEN 1, and so on.

¦SETHT expects one parameter of value 0, 1 or 2 which will vary the level of the split to either $\frac{1}{3}$, $\frac{1}{2}$ or $\frac{2}{3}$ of the way down the screen.

¦XDIS will disable XCOL and restore all inks as they were before XCOL was used. Further use of XCOL will resume use of the ink values employed the last time XCOL was in operation. Therefore if you are writing a program that runs with XCOL disabled for part of the time you will need to define the inks twice – once in the usual fashion and again after ¦XCOL has been executed.

There are a few points to note before using XCOL in your own programs:

- The horizontal division does not fall exactly on the boundary between two text lines.
- Make the dividing line visible by typing BORDER 0,1:INK 0,1,0. If you are lucky the line will be steady and will not jitter when you press a key. If you aren't lucky the division will appear to waver by one graphics line vertically, accentuated by key presses. It is the luck of the draw when you switch on whether or not you get a steady line.

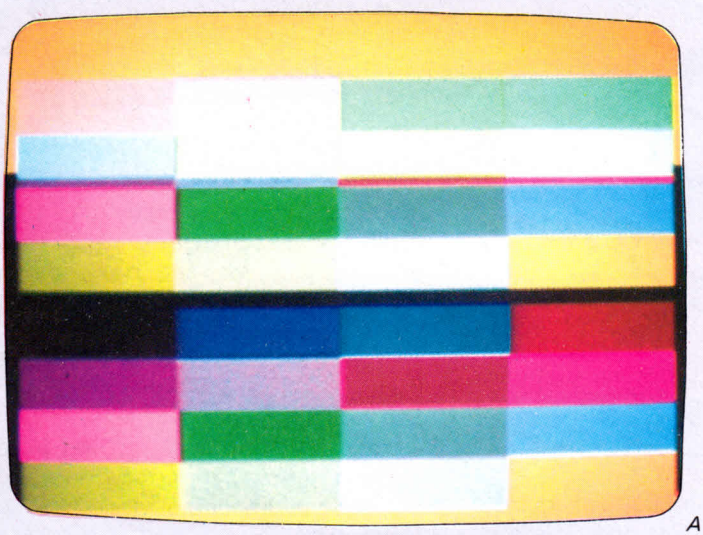
For these two reasons it is best to avoid producing screen displays where the dividing line is made plainly visible. In other words, keep your background and border colours the same in both halves of the screen and avoid putting graphics or text over the split.

- During disc or cassette operations your Amstrad disables interrupt routines. If this happens with XCOL enabled the result will be unsightly because normal ink handling is temporarily restored. When XCOL reactivates it will probably be out of sync with frame flyback. It is therefore advisable to first disable XCOL when using cassette or disc.
- Don't assume that XCOL is compatible with commercial software. The worst thing that can happen is a crash. I found this to be the case with the text editor in Maxam – it doesn't like the substitute screen clearing routine which is used whenever XCOL is enabled.

There's nothing to cause any great hardship there, just so long as you are aware of the limitations.

Program II gives the source code. It works using a fast ticker interrupt, which occurs every 1/300 of a second. A frame flyback happens every 1/50 second, so that during one frame cycle there are six fast ticker interrupts.

Fast tickers are synchronised with frame flyback so it is possible to set



A sample screen

up an interrupt to change the values of the inks while the cathode ray beam is only part-way down the tube. When the beam reaches the bottom of the screen – at frame flyback – the interrupt routine changes the values back, ready to refresh the top part of the display.

The values of the inks and border are stored in the operating system workspace in the form of two lists known as vectors. Two vectors are required because each ink can be set to two colours when flashing. XCOL takes its ink values from the same two lists but uses the second one for inks to be displayed on the lower part of the screen.

When enabling or disabling XCOL the contents of the vectors are swapped with those in a buffer so that you can work with two different sets of inks depending on whether or not XCOL is being used. The address of the vectors depends on which model of CPC you are using, but the program sorts that out for itself.

The machine operating system has its own arrangements for refreshing the screen display and some of the routines involved compete with XCOL to produce unsightly effects. You can't alter the guilty sections of program because they thumb their noses at you from the safety of ROM.

This mainly causes problems when changing mode, or would do if I hadn't substituted a screen clearing routine that favours XCOL. This is the bit that Maxam – and who knows what else – doesn't get on with, so be careful.

An example of how XCOL could be used would be with a graphic text adventure where the text could be printed in two or three colours in one part of the screen and the picture could employ three different foreground colours in the other part. This would help to avoid the monotony of four colours using Mode 1 as you could effectively display seven colours.

Another possible use would be in a screen or sprite designer where, by using Mode 0, it would be possible to display all 27 colours for comparison when making an initial pallette selection. Program III is an example of displaying all 27 colours at one time. I hope you find this utility as useful as I have.

Program I

```
10 REM XCOL
20 REM By Ian Sharpe
30 REM (c) Computing with the Amstrad
40 REM
50 MEMORY &9FFF
60 FOR ADDR=&A000 TO &A19B:READ BYTE$
70 BYTEVAL=VAL("&"+BYTE$)
80 POKE ADDR,BYTEVAL
90 CHECK=CHECK+BYTEVAL
100 NEXT
110 IF CHECK<>37357 THEN PRINT"TYPING
 ERROR IN DATA":END
120 CALL &A000
130 PRINT"|XCOL to enable"
140 PRINT"|XDIS to disable"
150 PRINT"|SETHT,n to set level where
 n=0,1 or 2"
160 PRINT"BORDER a,b or INK x,a,b set
s top(a) and bottom(b) values"
170 NEW
180 DATA 21,47,A1,CB,46,C0,CB,C6
190 DATA CD,00,B9,3A,02,C0,A7,CA
200 DATA 1B,A0,21,E5,B7,11,D4,B7
210 DATA C3,21,A0,21,EA,B1,11,D9
220 DATA B1,22,43,A1,ED,53,45,A1
230 DATA CD,0C,B9,2A,45,A1,11,57
240 DATA A1,01,22,00,ED,B0,21,EB
250 DATA BD,11,40,A1,01,03,00,ED
260 DATA B0,01,4A,A0,21,63,A0,C3
270 DATA D1,BC,55,A0,C3,67,A0,C3
280 DATA 93,A0,C3,C8,A0,58,43,4F
290 DATA CC,58,44,49,D3,53,45,54
300 DATA 48,D4,00,00,00,00,00,21
310 DATA 47,A1,CB,4E,C0,CB,CE,CD
320 DATA AF,A0,11,EB,BD,21,3D,A1
330 DATA 01,03,00,ED,B0,21,48,A1
340 DATA 0E,00,06,81,11,D7,A0,CD
350 DATA E0,BC,3E,01,CD,19,BD,32
360 DATA 55,A1,C9,21,47,A1,CB,4E
370 DATA C8,CB,8E,CD,AF,A0,21,40
380 DATA A1,11,EB,BD,01,03,00,ED
390 DATA B0,21,48,A1,C3,E6,BC,2A
400 DATA 45,A1,DD,21,57,A1,06,22
410 DATA 7E,F5,DD,7E,00,77,F1,DD
420 DATA 77,00,DD,23,23,10,F1,C9
430 DATA FE,01,C0,DD,7E,00,FE,03
440 DATA D0,C6,02,32,56,A1,C9,3A
450 DATA 55,A1,3D,32,55,A1,CA,ED
460 DATA A0,21,56,A1,BE,C0,ED,5B
470 DATA 45,A1,C3,25,BD,3E,06,32
480 DATA 55,A1,ED,5B,43,A1,C3,25
490 DATA BD,2A,43,A1,E5,11,79,A1
500 DATA CD,31,A1,2A,45,A1,E5,11
510 DATA 8A,A1,CD,31,A1,21,79,A1
520 DATA 22,43,A1,21,8A,A1,22,45
530 DATA A1,21,00,C0,01,00,40,36
540 DATA 00,23,0B,78,B1,C2,1F,A1
550 DATA E1,22,45,A1,E1,22,43,A1
560 DATA C9,7E,12,23,13,7E,06,10
570 DATA 12,13,10,FC,C9,C3,F9,A0
580 DATA 00,00,00,00,00,00,00,00
590 DATA 00,00,00,00,00,00,00,00
600 DATA 00,00,00,00,00,00,03,00
610 DATA 00,00,00,00,00,00,00,00
620 DATA 00,00,00,00,00,00,00,00
630 DATA 00,00,00,00,00,00,00,00
640 DATA 00,00,00,00,00,00,00,00
650 DATA 00,00,00,00,00,00,00,00
660 DATA 00,00,00,00,00,00,00,00
670 DATA 00,00,00,00,00,00,00,00
680 DATA 00,00,00,00,00,00,00,00
690 DATA 00,00,00,00,00,00,00,00
```

Program II

```
 ORG &A000

;*** initialise ***

 LD HL,flag   ;done before?
 BIT 0,(HL)
 RET NZ
 SET 0,(HL)
 CALL &B900   ;find OS version
 LD A,(&C002) ;and set vector
 AND A        ;address to suit
 JP Z,osv1    ;is v 1.0
 LD HL,&B7E5  ;is v 1.1
 LD DE,&B7D4
 JP xcol1

.osv1

 LD HL,&B1EA
 LD DE,&B1D9

.xcol1

 LD(vec1),HL
 LD(vec2),DE
 CALL &B90C     ;restore ROM state
 LD HL,(vec2)   ;copy initial ink
 LD DE,vecstore ;values to store
 LD BC,34
 LDIR
 LD HL,&BDEB    ;store jumpblock
```

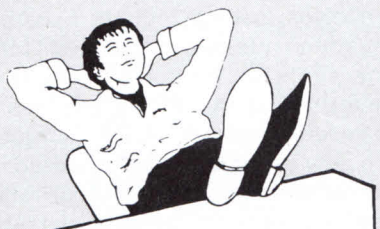
```
LD DE,jpstore ;entry for later
LD BC,3       ;use
LDIR
LD BC,rsxtable  ;enable RSX
LD HL,workspace
JP &BCD1

.rsxtable

WORD nametable
JP enable
JP disable
JP setht

.nametable

DEFB   "XCO","L"+&80
DEFB   "XDI","S"+&80
DEFB  "SETH","T"+&80
NOP

.workspace

DEFS 4

;*** enable interrupt ***

.enable

LD HL,flag ;already enabled?
BIT 1,(HL)
RET NZ
SET 1,(HL)
CALL swap       ;swap vector/store
LD DE,&BDEB     ;patch
LD HL,clrpatch  ;SCR MODE CLR
LD BC,3
LDIR
LD HL,tickblock ;enable interrupt
LD C,0
LD B,&81
LD DE,rtn
CALL &BCE0
LD A,1
CALL &BD19
LD(counter),A
RET

;*** disable interrupt ***

.disable

LD HL,flag ;not enabled?
BIT 1,(HL)
RET Z
RES 1,(HL)
CALL swap
LD HL,jpstore   ;restore
LD DE,&BDEB     ;SCR MODE CLR
LD BC,3
LDIR
LD HL,tickblock
JP &BCE6        ;disable

;** swap store/vectors ***

.swap

LD HL,(vec2)
LD IX,vecstore
LD B,34

.swaploop

LD A,(HL)
PUSH AF
LD A,(IX)
LD(HL),A
POP AF
LD(IX),A
INC IX
INC HL
DJNZ swaploop
RET

;***** set height *****

.setht

CP 1
RET NZ
LD A,(IX)
CP 3
RET NC
ADD 2
LD(count),A
RET

;*** switch colours routine ***

.rtn

LD A,(counter) ;decrement counter
DEC A
LD(counter),A
JP Z,rtn1      ;0=frame flyback
LD HL,count
CP(HL)         ;switch point?
RET NZ
LD DE,(vec2)
JP &BD25       ;set new colours

.rtn1          ;recharge counter

LD A,6
LD(counter),A
LD DE,(vec1)
JP &BD25       ;reset colours

;**** clear screen ****

.cls

LD HL,(vec1) ;set inks to bgrnd
PUSH HL
LD DE,zvec1
CALL setvec
LD HL,(vec2)
PUSH HL
LD DE,zvec2
CALL setvec
LD HL,zvec1
LD(vec1),HL
LD HL,zvec2
LD(vec2),HL
LD HL,&C000
LD BC,&4000

.clsloop     ;scr mem=0's

LD(HL),0
INC HL
DEC BC
LD A,B
OR C
JP NZ,clsloop
POP HL        ;restore cols
LD(vec2),HL
POP HL
LD(vec1),HL
RET
.setvec
 LD A,(HL)
 LD(DE),A
 INC HL
 INC DE
 LD A,(HL)
 LD B,16
```

From Page 45

```
.setvecloop

 LD(DE),A
 INC DE
 DJNZ setvecloop
 RET
;*** program w. s. ****

.clrpatch

 JP cls

.jpstore   DEFS 3
.vec1      DEFS 2
.vec2      DEFS 2
.flag      DEFS 1
.tickblock DEFS 13
.counter   DEFS 1
.count     DEFS 3
.vecstore  DEFS 34
.zvec1     DEFS 17
.zvec2     DEFS 17

end
```

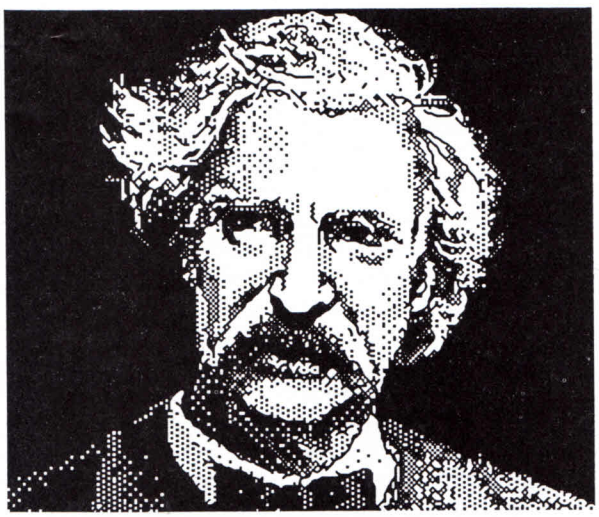
Program III

```
100 MODE 0
110 CALL &BC02
120 CALL &BB4E
130 DEFINT a-z
140 !XCOL
150 FOR x=0 TO 15
160   INK x,x,(x+16) MOD 27
170 NEXT
180 BORDER 9,18
190 FOR i=1 TO 14 STEP 13
200   FOR j=0 TO 3
210     FOR k=0 TO 2
220       FOR l=1 TO 16 STEP 5
230         LOCATE 1,i+j*3+k
240         PAPER j*4+(l-1)/5
250         PRINT STRING$(5,32);
260       NEXT l
270     NEXT k
280   NEXT j
290 NEXT i
300 WHILE INKEY(47)<0
310 WEND
```



Give your fingers a rest . . .

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



BUDDING GENIUS?

If you're a budding Einstein (i.e. Amstrad programmer) we'd like to see what you have to offer.
We are constantly in need of good quality software for our aggresive marketing program and would like to make contact with you. Genuine devalued Aussie dollars are offered! Give us a call or send us a sample.

Strategy Software

Box 5000
Glen Waverley
Victoria 3150

(03) 560 4324

Non-disc BDOS function calls

Part VII of COLIN FOSTER's exploration of CP/M 2.2

LET'S have a detailed look at the 13 BDOS function calls which CP/M provides us with to carry out simple input and output operations.

There is also a larger number of more powerful function calls available which deal with disc operations, but we'll deal with them in the future.

Last month we used function number 2, Console Input, as an example to demonstrate the way we make a function call. This simply involves loading register C with the function number, loading register D – or register pair DE – with any information the BDOS requires and executing a CALL 0005 instruction.

Any result is normally passed back to us in either register A or register pair HL when the function call returns control to our program. This system is the same for all BDOS function calls.

So now we'll examine these 13 non-disc BDOS function calls.

The only parameter System Reset requires is the function number 0 in register C, and it returns nothing. It is rarely used because it's effect is identical to executing a:

```
CALL  0000h
```

or

```
RST   00h
```

instruction, either of which is easier and quicker to do. All of these alternatives will abort any program which is running, reload the CCP and BDOS from disc and return us to CCP command level, which is the A> prompt.

Console Input requires function number 1 in register C, and returns to us in register A the Ascii value of the next character read from the logical CON: device, which is normally the Amstrad's keyboard. If the character is printable or is a CR, LF or backspace it is echoed to the screen.

If it is a tab it is expanded on the screen in columns of eight characters. The BDOS will wait until a character is typed before returning control to our program.

Console Output requires function number 2 in register C and the Ascii value of the character we wish to send to the logical CON: device – normally the Amstrad's screen – in register E.

Similarly to function 1 tabs are expanded to columns of eight characters. Also the BDOS checks to see whether a Ctrl+S has been typed on the keyboard.

If a Ctrl+S has been typed this tells CP/M to temporarily halt the execution of the program which is running, so if you are outputting a lot of information to the screen you can pause it to give yourself a chance to read it.

Once a Ctrl+S has been pressed you can do two things – pressing Ctrl+C will abort the program and return you to the CCP prompt, while pressing any other key will resume execution of the program without it knowing it has been paused.

Something which causes a deal of confusion in CP/M is the printer echo facility. Function 10, which is Buffered input and described below, will accept a Ctrl+P character from the keyboard, and uses this to set or reset the printer echo toggle flag in the BDOS.

If set, this flag causes all characters printed on the screen to also be sent to the LST: device, which is normally the Centronics printer port on the Amstrad. All the functions which either print or echo characters on the screen (1, 2, 9, 10) check the flag and echo to the printer if appropriate.

However only function 10 will read the Ctrl+P command from the keyboard to alter the state of the flag.

I have read several books which state that other functions will also do this under CP/M 2.2 – probably based on the Digital Research manuals which are themselves ambiguous on the point – but they are wrong. CP/M Plus is smarter in this respect – as in many others – but that is a separate issue.

Reader Input requires only function number 3 in register C, and returns the character read from the logical RDR: device – normally assigned to the RS232 interface on an Amstrad – in register A.

Characters read by this function are not echoed to the screen and the function does not return until a character has been received.

Punch Output requires function number 4 in register C and the Ascii value of the character to be sent in register E. The character is sent to the logical PUN: device, which is normally assigned to RS232 output on the Amstrad. Unlike Console Output no checks are made for pause or abort commands from the keyboard.

List Output requires function number 5 in register C and the Ascii value for the character to be printed in register E. This character is sent to the logical LST: device – normally the Amstrad's Centronics printer port. Again no checks are made for keyboard input.

Direct Console I/O is where things start getting a bit complicated. This function can be used to let us check the keyboard to see whether a character has been typed, input a character from the keyboard or print one on the screen. In all cases we must call it with function number 6 in register C.

If we wish to print a character on the screen we must also put the Ascii code for the character into register E, just as for the normal Console Output function. However this one does not check for pause or abort commands from the keyboard.

To read from the keyboard we must call the function with the value &FF in register E. It then checks to see whether a character has been typed on the keyboard, and if so it

Continued on Page 58

Sam's no poker face

Program: Samantha Fox Strip Poker
Price: £8.95 (cassette), £13.95 (disc)
Supplier: Martech, Martech House, Bay Terrace, Pevensey Bay, East Sussex BN24 6EE. Tel: 0273 692224

THESE days many software houses advertise their products using personalities such as Daley Thompson and Barry McGuigan to underwrite their games.

It's now time for the ladies to get in on the act, and who better to start us off than Samantha Fox. She has got together with Martech to produce Strip Poker.

There are a number of jokes that could be used during this review, but because you are above such things I will stick to the main points – and I don't mean Sam's.

Poker is one of those games which is relatively easy to learn but difficult to master. Side one contains Sam's Strip Poker in which you play against the computer and start off with 1,000 credits.

During the game you bet on your hand as it is being dealt, and if you win you gain credits off your opponent.

As the credits total of the computer drops down below certain levels, digitised pictures of Samantha appear in various stages of undress. The pictures are in Mode 2 and look a bit like what you would expect from a newspaper print.

Seven Card Stud on side two is very much the same, but you can play against three opponents which is more demanding.

Side one is a little predictable and it shouldn't take even a novice long to outwit Martech's pin-up girl. Side two needs far more skill and patience to master the other opponents, and is much the better of the two games.

The music that goes along with the credits is well done – an excellent rendition of The Entertainer.

If you wish to learn how to play poker and you enjoy looking at semi-naked pixels this game will suite you.

Ian Duerden

Sound:	*7*
Graphics:	*7*
Presentation:	*7*
Value for money:	*7*
Overall:	*7*

Wartime realism

Program: Dambusters
Price: £9.95
Supplier: US Gold, Parkway Industrial Estate, Heneage Street, Birmingham B7 4LY. Tel: 021-359 3020

US Gold's Dambusters has now appeared for the Amstrad, and considering the complexity of the original mission I was interested to see if this program could live up to its bold title.

First of all the aircraft must be configured for take-off. Separate screens allow access to the engine and other major controls, and the runway is portrayed by a stream of red lights.

The view out of the Lancaster can be seen from three different screens – cockpit with flying instruments and joystick control, and front and rear gunner positions.

Once airborne you must navigate to the German target areas. Several clearly designed maps make this infinitely easier than it must have been originally.

They show your aircraft and many different ground locations and eventually the three dams and their reservoirs.

You will have to fight off attacking German planes using the two gunner screens, as well as dodge searchlights, barrage balloons and gunfire from the ground.

The sound of tracer bullets and gunfire are effectively recreated.

Once near a dam you must make fine adjustments to your speed and height and make an accurate measurement of the target distance.

There seem to be no major omissions from this game and it provides some thought-provoking entertainment. It successfully captures the spirit of the problem-filled mission, though inevitably some aspects are better portrayed than others.

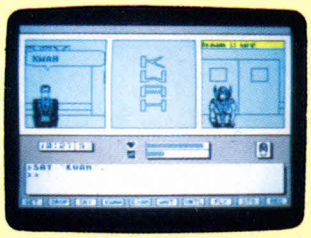
What this product might lack in high speed action is made up for by its realism and atmosphere.

Phil Murfitt

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

Look out, Batman!

Program: Redhawk
Price: £8.95
Supplier: Melbourne House, 60 High Street, Hampton Wick, Kingston-upon-Thames, Surrey KT1 4DB. Tel: 01-943 3911



ONE of the problems that software houses face is in trying to make the games they produce sufficiently different.

This applies even more to adventure type games. Redhawk from Melbourne House is certainly different in a number of ways.

Kevin Oliver who has lost his memory is about to leave hospital and only one word means anything to him – KWAH. When used, this word changes Kevin into the superhero Redhawk. Like all superheroes Redhawk has superhuman powers. As well as being incredibly strong Redhawk can fly. In the game he must use these powers to battle the arch villains and criminals who threaten the city.

Redhawk is a typical comic strip hero and is displayed as such on the screen. The top half is split into three panels, and the graphics are produced in the right hand panel and scrolled across to the left hand panels as the game progresses.

The bottom half of the screen contains the text area which enables you to control the action of Kevin/Redhawk by typing in English-type sentences. Responses or instructions appear on the right hand graphics panel.

At the bottom of the screen are 10 commonly used words which are programmed on to function keys for easier use.

Melbourne House, with adventures such as The Hobbit and Lord of the Rings to their credit, are no strangers to adventure games. This shows in the professional way that Redhawk has been designed and the vocabulary that can be used.

The game itself has a high level of intelligence and quite a sense of humour. In fact if you really want to be different you could forget about your popularity rating and become the worst of the supervillains.

Ian Duerden

Sound:	*N/A*
Graphics:	*9*
Presentation:	*7*
Value for money:	*9*
Overall:	*8*

Grunts and groans

Program: Rock'n Wrestle
Price: £9.95
Supplier: Melbourne House, 60 High Street, Hampton Wick, Kingston-upon-Thames, Surrey KT1 4DB. Tel: 01-943 3911

AFTER boxing and Kung Fu here is computer wrestling from Melbourne House. In Rock'n Wrestle you are Georgeous George, tenth ranked contender for the world championship belt.

You have to work through all the other contenders such as Angry Abdul, Vicious Viviane and ultimately Lord Toff to become world champion. Each contender has his own characteristics.

There are 23 different moves in all, which is quite a lot to learn and get used to.

It is probably best to watch the demo first and then go into two player mode so you can practise without your opponent being able to get the better of you.

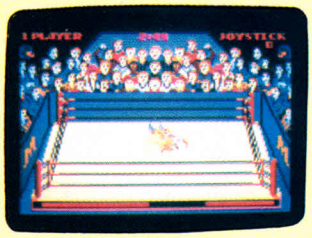
The main screen shows a ring in 3D with the crowd looking on in the background. The contestants start from opposite corners of the ring.

You control the moves either by joystick or keys, and the position you are in relative to your opponent determines the moves you can make.

The list of moves looks endless and all the famous ones are there, such as body slam, headlock and drop kick.

The graphics are colourful if a little square and block-like. More detail and less colour would have probably given a better overall impression.

If you have an interest in wrestling this game may well appeal to you.



Ian Duerden

Sound:	*3*
Graphics:	*5*
Presentation:	*5*
Value for money:	*6*
Overall:	*6*

State of the arts

Program: The Way of the Tiger
Price: £9.95 (cassette), £14.95 (disc)
Supplier: Gremlin Graphics, Alpha House, 10 Carver Street, Sheffield S1 4FS. Tel: 0742 753423

AS you may have guessed from the title, The Way of the Tiger is a martial arts simulation, and once you have seen this game I have a feeling that your Exploding Fist or Yie Ar Kung Fu will be left to gather dust.

It is supplied on two cassettes or a single disc. Side one of the first cassette contains the master program.

Having loaded this a menu is displayed, which gives you the option of practicing any of the three skills or playing the game for real. The first of the three disciplines is unarmed combat. You control the hero of the game, a young would-be ninja who is cast out into the desert to face unknown adversaries.

The first of these is a ninja who is easily disposed of, and the second is a knee-high goblin sporting a sword. The third enemy is a horned demon who delivers a mean head butt.

After defeating another few ninjas and goblins you begin to gain confidence, but at this point the ground begins to tremble and on lumbers the biggest troll you have ever seen.

Your guess is as good as mine as to what follows him because your humble reviewer got beaten senseless every time he met the beast.

Your second task is to guard a slippery horizontal log which stretches across the river. To protect yourself you are armed with a big stick, and wielding this mercilessly you batter your way through a new series of opponents.

Finally you are provided with a sword and sent out to hack up a few more bad guys. One particular movement of the joystick will cause your ninja to execute what is described as a sword ritual.

Talk about a flash of steel this chap would put any self respecting samurai to shame.

Even if you have to raid the piggy bank you must buy this game.

James Riddell

Sound:	*8*
Graphics:	*10*
Playability:	*10*
Value for money:	*10*
Overall:	*9*

Cavern adventure

Program: Starquake
Price: £8.95 (cassette), £12.95 (disc)
Supplier: Bubble Bus, 87 High Street, Tonbridge, Kent TN9 1RX. Tel: 0732 355962

AN unstable planet emerging from the inner reaches of a black hole threatens the existence of the whole universe. Unless its fragile core can be rebuilt a starquake will result.

As the other starship pilots have pressing engagements elsewhere it has fallen to Blob to perform this task.

Starquake is an arcade style adventure game in which you must guide the Pacman-esque Blob through the caverns which honeycomb the planet. The characters are well detailed but drawn in single colours only, thus revealing the game's Spectrum origins.

However don't let this fact deter you from buying it. The animation is faultless, and the action is fast and furious.

Control of Blob is simple – you can move right and left, and fall down holes. The instructions provided are sparse – it is an adventure game after all.

Following some trial and error, and much joystick waggling, you will find you can lay small pieces of platform.

By dropping these and jumping on them before they dissolve you can raise yourself into the air, thereby escaping from seemingly exitless pits.

Your final mode of transport is a hover platform which enables you to defy gravity and fly in all directions.

You are armed with a zap gun which is essential as the planet is riddled with alien life forms. Contact with any of these creatures will drain your energy cells, and touching the metallic spinning tops results in instant destruction.

Rapid access can be gained to various regions of the planet via the local teleport system. Upon entering one of these terminals you are asked for the name of your destination.

One of the major problems you encounter is that you don't know what the parts of the planet's core look like, and you wouldn't know where to put them even if you did.

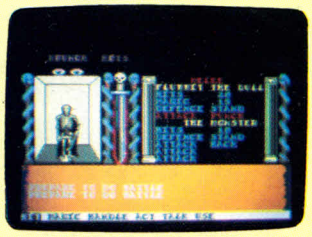
Starquake is extremely well written, packed with features too numerous to mention and, it's fun.

Jon Revis

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

Catacomb magic

Program: Swords and Sorcery
Price: £9.95 (cassette), £14.95 (disc)
Supplier: PSS, 452 Stoney Stanton Road, Coventry, CV6 5DG. Tel: 0203 667556



DO you have what it takes to enter the catacombs of Zob and survive? If so then you had better buy yourself a copy of Swords and Sorcery from PSS.

It is an adventure game in which the computer acts as your eyes. The screen displays a series of three windows.

The first contains an animated view of your immediate surroundings and the second is a plan view of the catacombs showing the positions of yourself and the monsters.

Finally a text window displays any dialogue spoken by yourself or a monster – this normally consists of insults related to dogs, bottoms, and dung.

Before beginning a game you are given the opportunity to load a previously stored character from tape. Failing that you can use the default character Flubbit the Dull, or create a new character of your own.

You do the latter by spending various portions of a 14 day training period with the masters of skills you may need. One such skill is immortality – spend a day with this chap and you are allowed 99 lives.

Once in the dungeon you will notice that everything you do has to be selected from a menu. The main one consists of options to hit, do magic, handle, act, talk, and use.

When engaging a monster in combat both your own strategies for attack and defence and those of the monster, are displayed.

You can just sit back and watch the computer display the results of each round of combat until one of you is dead.

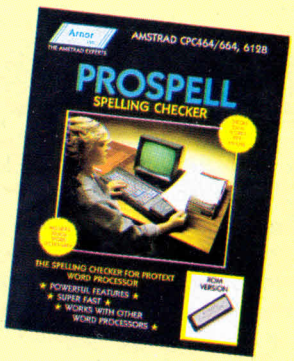
Should you ever complete or grow bored with the game, PSS will provide you with a second dungeon tape for £1.

James Riddell

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

Lost for right word?

Program: Prospell
Price: £24.95 (disc), £34.95 (Rom)
Supplier: Arnor, The Studio, Ledbury Place, Croydon CRO 1ET. Tel: 01-688 6223



NATURALLY a good speller like me (I?) doesn't need a spelling checker, and I had to deliberately insert some howlers into an old text file in order to test Arnor's Prospell.

Imagine my feelings when it discovered one more than I'd actually put in.

Prospell comes on disc or 16k Eprom and although designed to work with its sister program Protext it can stand alone and check any Ascii file generated by any word processor. Two methods are offered in the main menu. Firstly Spell check file throws up each possible error as it comes to it, requiring you to deal with it before it moves on to the next.

Secondly a two pass check runs through and lists all the possible errors before offering them individually for correction. I say possible errors because it will also fling out words not in its dictionary.

You can view the word in context, see what the dictionary thinks it should be, correct or leave it alone and, if you wish, add the word to the dictionary. When the check is finished the corrected text file is saved automatically.

Finally there are two functions which will be a boon to all crossword addicts. You can enter a word with question marks substituted for all the missing letters and the Find function will throw out a list of all words fitting the pattern.

In Anagram mode you can enter the letters and any words containing all the same letters will be displayed. You can also enter an asterisk to stand for any number of unknown letters in either function.

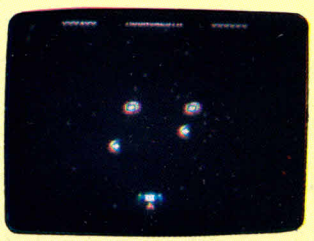
I'd certainly recommend Prospell, whether you think you need it or not.

Dorene Cox

Sound:	*N/A*
Graphics:	*N/A*
Ease of use:	*9*
Value for money:	*9*
Overall:	*9*

Arcade in your home

Program: Moon Cresta
Price: £7.95
Supplier: Incentive Software, 54 London Street, Reading RG1 4SQ. Tel: 0734 591678



INCENTIVE Software's Moon Cresta recreates the arcade game of the same name.

There is the more-tall-than-wide type display, 10p slot, two player option, keyboard/joystick, selection, pause key, hall of fame and finally a reassurance that you will have fun and thrills.

The stars cascade gently downwards, and your triple-tiered ship arrives gracefully. It divides into two – the lower part then descends from whence it came, and what remains drops to become your laser base, Space Invaders-style.

After that there is organised chaos arranged by the aliens. There are several types of these hyperactive menaces, and they know a lot more about tactical trajectories than I do.

One dirty trick involves disappearing off the side of the display and reappearing underneath your laser – quite a problem as you can only fire upwards or run sideways, often into another alien.

Either way obliteration is a fairly frequent feature of the game, after which you get 20 seconds to recover while the aliens take the screen over again, coming together in a smart formation.

The formalities dealt with, the second and third stages of your ship reappear and split again, providing two more double lasers for further battle.

If you can score 30,000 points there is a special competition to enter, but normal people can still enjoy what is a well crafted replica of a slightly dated arcade game.

Perhaps there will be more to come in this style – hopefully it will be a better game next time as this one seems too hard for some. It has visual appeal, but shows less imagination than more recent designs.

Phil Murfitt

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

DIAMOND DIGGER

By ARAMELLO CHAPMAN

IN 2019 the world's biggest mine is controlled by one computer, the Amstrad CPC 189234 mark II. It deals with everything from commanding the Mole-Droids to selling the mined diamonds on the world market.

You take the part of a poor computer operator who has devised a means of earning some extra money.

You have invented a program to mine 10 of the zones which make up the underground workings. To do this the main computer has been told to set aside three droids for extra duties.

But a bug in your program has left you only a limited time in each zone to collect the diamonds before the main computer logs your terminal off.

The droids are pretty weak and will be crushed if a rock falls or rolls on to them.

However, they are able to use the suction controlled rastor organisor and labelled line (Scroll for short) mover.

This invaluable piece of equipment allows the droid to move the line on which it is standing, either left or right.

The heavy rocks will crush a droid if one is silly enough to go under it. They move in the following three ways:

- If the space under a rock is vacant or contains a droid it will fall.
- If there is a rock under another rock, the space to the left is vacant and the space below that is vacant, then it will roll to the left.
- Similar to above, but it will check to the right and below. If both are vacant the rock will roll to the right.

Diamonds remain fixed until a droid collects them. After all the diamonds are collected from a zone the droids will move to another.

If all the droids are destroyed the main computer will give you a rating and log you out.

KEYS

Z	Left	**X**	Right
/	Down	**;**	Up
A	Scroll left	**S**	Scroll right

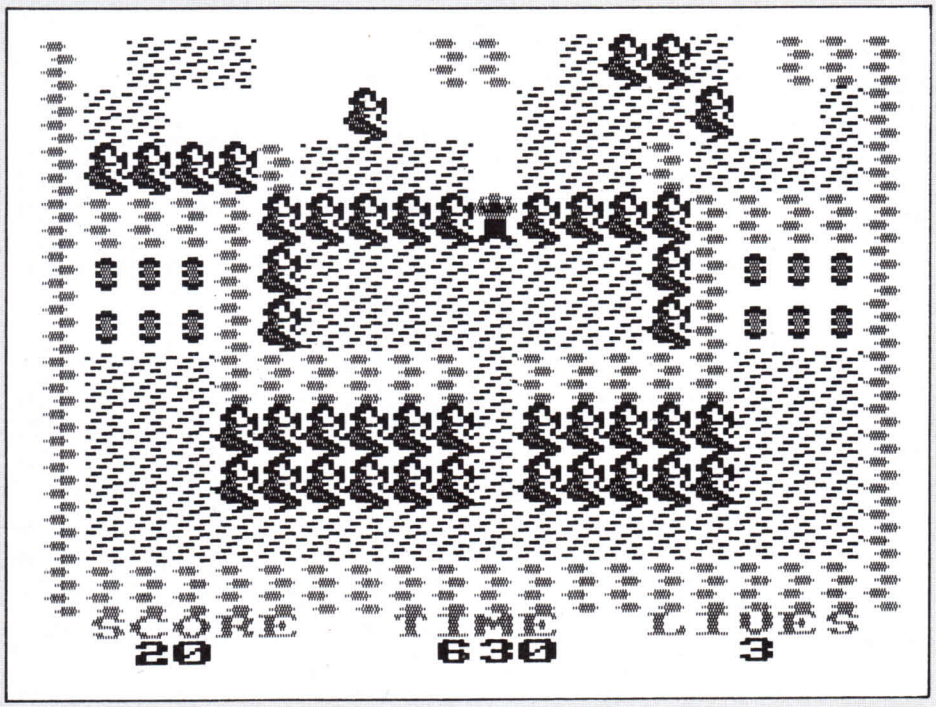
ROUTINES

100	Main loop.
180	Move droid.
550	Scroll left.
590	Scroll right.
650	Check rock below.
770	Rocks fall right.
890	Rocks fall left.
1050	Set up screen.
1260	Cleared screen.
1350	Lost a life.
1410	End of game.
1500	Variables.
1550	Instructions.
1790	Initial set up.
2460	High scores.
2880	Title tune.

VARIABLES

yopos	Location of droid in user screen map.
x,y	Location of droid on screen.
rockx, rocky	Location of rock on screen.
sc	Score.
screen	Mine zone which droid is in.
lives	Number of droids left.
ti	Time before computer logs out.
jewel	Number of jewels collected.
jeget	Number of jewels to collect.
gra	Graphics character droid uses.
hs	High score.
na$	Names in high score table.
left	Address of left scroll routine.
right	Address of right scroll routine.
rockb	Address of rock fall routine.
rockr	Address of rock fall to the right routine.
rockl	Address of rock fall to the left routine.
ch53	Address of routine to change all 5s to 3s in user screen.
block	Address of routine to transfer data into user screen work area.

```
10 REM Diamond Digger
20 REM By Aramello Chapman
30 REM (c) Computing with the Amstrad
40 REM
50 GOSUB 1790:REM Initial Set Up
60 GOTO 1550:REM Instructions
70 GOSUB 1500:REM Set Up Variables
80 IF screen=11 THEN 2460
90 GOSUB 1050:REM Set Up Screen
100 REM**********Main Loop**********
110 GOSUB 180
120 IF jewel=0 THEN 1260
130 POKE &A048,&A1
140 GOSUB 650
150 IF death=1 THEN 1350
160 ti=ti-1:LOCATE 9,25:PEN 6:PRINT t
i:IF ti=0 THEN lives=lives-1:GOTO 135
0
170 GOTO 100
180 REM**********Move Droid**********
190 POKE &A048,&A2
200 GOSUB 630
210 IF INKEY(69)>-1 THEN 550
220 IF INKEY(60)>-1 THEN 590
230 IF INKEY(71)>-1 THEN GOSUB 360:GO
TO 310
240 IF INKEY(63)>-1 THEN GOSUB 410:GO
TO 310
250 IF INKEY(28)>-1 THEN GOSUB 460:GO
TO 310
260 IF INKEY(30)>-1 THEN GOSUB 500:GO
TO 310
270 head=head+1:IF head=20 THEN grank
=5
280 IF head=40 THEN head=0:grank=4
290 CALL &A000,x1,y1,x1,y1,grank:CALL
 &A000,x1,y1+1,x1,y1+1,6
300 RETURN
310 GOSUB 640
320 CALL &A000,x1,y1,x,y,gra:CALL &A0
00,x1,y1+1,x,y+1,gra+1
330 SOUND 2,3500,5,15,1
340 POKE 39699+yopos,0
350 RETURN
360 addr=PEEK(39699+yopos-1)
370 gra=0
380 IF addr=4 THEN sc=sc+10:jewel=jew
el-1:yopos=yopos-1:SOUND 1,30,20,15,1
:GOSUB 540
390 IF addr<2 THEN yopos=yopos-1
400 RETURN
410 addr=PEEK(39699+yopos+1)
420 gra=2
430 IF addr=4 THEN LET sc=sc+10:jewel
=jewel-1:yopos=yopos+1:SOUND 1,30,20,
15,1:GOSUB 540
440 IF addr<2 THEN yopos=yopos+1
450 RETURN
460 addr=PEEK(39699+yopos-20)
470 IF addr=4 THEN sc=sc+10:jewel=jew
el-1:yopos=yopos-20:SOUND 1,30,20,15,
1:GOSUB 540
480 IF addr<2 AND yopos-20>0 THEN yop
os=yopos-20
490 RETURN
500 addr=PEEK(39699+yopos+20)
510 IF addr=4 THEN sc=sc+10:jewel=jew
el-1:yopos=yopos+20:SOUND 1,30,20,15,
1:GOSUB 540
520 IF addr<2 THEN yopos=yopos+20
530 RETURN
540 LOCATE 2,25:PEN 6:PRINT sc:RETURN
550 REM**********Scroll Left*********
560 FOR f=1 TO 4:CALL left,y1-1:CALL
left,y1:NEXT
570 addr=39700+(INT(yopos/20))*20:dum
p=PEEK(addr+1):FOR f=1 TO 18:POKE add
r+f,PEEK(addr+f+1):NEXT:POKE addr+18,
dump:yopos=yopos-1:x1=x1-1:IF x1=1 TH
EN yopos=yopos+18
580 RETURN
590 REM**********Scroll Right********
600 FOR f=1 TO 4:CALL right,y1-1:CALL
 right,y1:NEXT
610 addr=39700+(INT(yopos/20))*20:dum
p=PEEK(addr+18):FOR f=18 TO 2 STEP -1
:POKE addr+f,PEEK(addr+f-1):NEXT:POKE
 addr+1,dump:yopos=yopos+1:x1=x1+1:IF
 x1=20 THEN yopos=yopos-18
620 RETURN
630 y1=INT(yopos/20):x1=yopos-(20*y1)
:y1=y1*2+2:RETURN
640 y=INT(yopos/20):x=yopos-(20*y):y=
y*2+2:RETURN
650 REM*******Check Below Rock******
660 CALL rockb
670 IF PEEK(39698)=0 THEN GOTO 770
680 FOR f=1 TO PEEK(39698)
690 addr=PEEK(39599+f)
700 IF addr=yopos-1 THEN GOSUB 1030
710 rocky=INT(addr/20)
720 rockx=addr-(20*rocky)+1
730 rocky=rocky*2
740 CALL &A000,rockx,rocky,rockx,rock
y+2,2:CALL &A000,rockx,rocky+1,rockx,
rocky+3,3
750 SOUND 1,addr+100,2,15,1
760 NEXT
770 REM*******Rocks Fall Right******
780 CALL rockr
790 IF PEEK(39697)=0 THEN GOTO 890
800 FOR f=1 TO PEEK(39697)
810 addr=PEEK(39500+f)
820 IF addr=yopos-1 THEN GOSUB 1030
830 rocky=INT(addr/20)
840 rockx=addr-(20*rocky)+1
850 rocky=rocky*2
860 SOUND 1,addr+100,2,15,1
870 CALL &A000,rockx-1,rocky+2,rockx,
rocky+2,2:CALL &A000,rockx-1,rocky+3,
rockx,rocky+3,3
880 NEXT
890 REM******Rocks Fall Left******
900 CALL rockl
910 IF PEEK(39699)=0 THEN GOTO 1010
920 FOR f=1 TO PEEK(39699)
930 addr=PEEK(39399+f)
940 IF addr=yopos-1 THEN GOSUB 1030
950 rocky=INT(addr/20)
960 rockx=addr-(20*rocky)+1
970 rocky=rocky*2
980 SOUND 1,addr+100,2,15,1
990 CALL &A000,rockx+1,rocky+2,rockx,
rocky+2,2:CALL &A000,rockx+1,rocky+3,
rockx,rocky+3,3
1000 NEXT
1010 CALL ch53
1020 RETURN
1030 lives=lives-1:death=1:SOUND 1,15
0,10,15,1,1
1040 RETURN
1050 REM******Set Up Screen*******
1060 MODE 0:INK 1,7
1070 CALL block,screen
1080 POKE &A048,&A1
1090 no=0:FOR f=2 TO 22 STEP 2:FOR g=
1 TO 20
1100 addr=PEEK(39700+no)
1110 IF addr=0 THEN gra1=7:gra2=7
1120 IF addr=1 THEN gra1=0:gra2=0
1130 IF addr=2 THEN gra1=1:gra2=1
1140 IF addr=3 THEN gra1=2:gra2=3
1150 IF addr=4 THEN gra1=4:gra2=5
1160 CALL &A000,g,f,g,f,gra1:CALL &A0
00,g,f+1,g,f+1,gra2
1170 no=no+1
1180 NEXT g,f
1190 LOCATE 2,24:PEN 9:PRINT"SCORE  T
IME  LIVES"
1200 LOCATE 2,25:PEN 6:PRINT sc:LOCAT
E 9,25:PRINT ti:LOCATE 16,25:PRINT li
ves
1210 LET yopos=2:gra=0:grank=5:death=
0
1220 ti=850-(screen*50):jewel=jeget(s
creen)
1230 GOSUB 630
1240 POKE &A048,&A2:CALL &A000,x1,y1,
x1,y1,grank:CALL &A000,x1,y1+1,x1,y1+
1,6
1250 RETURN
1260 REM*****Cleared Screen*****
```




```
1270 IF screen=10 THEN a$="...Mega-Ro
cker...":ti=ti+1000 ELSE a$="..Zone C
ompleted.."
1280 FOR f=1 TO ti\10:LOCATE 2,25:PEN
 6:sc=sc+10:PRINT sc:SOUND 1,100,1,15
,1:NEXT
1290 GOSUB 1320
1300 screen=screen+1
1310 GOTO 80
1320 FOR f=1 TO LEN(a$):LOCATE 19,1:P
RINT MID$(a$,f,1):FOR d=1 TO 4:CALL l
eft,0:NEXT d,f:FOR f=1 TO 20*LEN(a$)-
4:CALL left,0:NEXT
1330 FOR f=1 TO 20:LOCATE 19,1:PRINT"
 ":FOR d=1 TO 4:CALL right,0:NEXT d,f
1340 RETURN
1350 REM*******Lost a Life******
1360 IF ti=0 THEN LOCATE 3,1:PEN 3:PR
INT"Device Time Out" ELSE LOCATE 3,1:
PEN 3:PRINT"Mining Droid Out"
1370 FOR f=100 TO 800 STEP 100:SOUND
1,300,10,15,1:NEXT
1380 FOR f=1 TO 3000:NEXT
1390 IF lives=0 THEN GOTO 1410
1400 GOTO 90
1410 REM*******End Of Game******
1420 GOSUB 1330
1430 RESTORE 1490:FOR f=1 TO screen:R
EAD a$:NEXT
1440 GOSUB 1320
1450 LOCATE 2,1:PEN 3:PRINT"Player Lo
gged Out"
1460 RESTORE 1480:FOR f=1 TO 50:READ
p,d:d=d*10:GOSUB 2950:SOUND 1,pn,d*0.
5,15,1,2:SOUND 2,pn+3,d*0.5,15,1,2:SO
UND 4,pn/0.5,d*0.5,15,1,2:NEXT:FOR f=
1 TO 2000:NEXT
1470 GOTO 2460
1480 DATA 8,4,8,2,8,2,8,4,10,2,12,2,1
3,4,5,2,6,2,8,4,1,4,13,4,13,2,13,2,15
,4,15,4,17,16,5,4,5,2,6,2,8,4,6,4,5,4
,8,4,13,6,12,2,10,4,8,4,10,4,12,4,13,
4,12,4,10,4,8,4,6,4,5,4,3,4,1,6,3,2,5
,4,3,4,1,6,3,2,5,4,8,4,6,6,5,2,3,4,8,
4,1,16
1490 DATA Device Defunct,Pile of Junk
,Load of Rubbish,Fairly Bad,Just Plai
n Bad,Could Be Better,Class 2 Rocker,
Class 1 Side Kick,Super-Rocker,Hyper-
Rocker
1500 REM*******Variables*********
1510 screen=1:lives=3
1520 sc=0
1530 head=0
1540 RETURN
1550 REM*******Instructions******
1560 MODE 1
1570 PEN 3:LOCATE 1,1:PRINT CHR$(150)
;STRING$(38,CHR$(154));CHR$(156):FOR
f=2 TO 5:LOCATE 1,f:PRINT CHR$(149);T
AB(40);CHR$(149):NEXT
1580 PEN 3:LOCATE 1,5:PRINT CHR$(147)
;STRING$(38,CHR$(154));CHR$(153)
1590 LOCATE 1,20:PRINT CHR$(135);STRI
NG$(38,CHR$(131));CHR$(139):FOR f=21
TO 23:LOCATE 1,f:PRINT CHR$(133);TAB(
40);CHR$(138):NEXT
1600 LOCATE 1,24:PRINT CHR$(141);STRI
NG$(38,CHR$(140));CHR$(142)
1610 LOCATE 14,3:PEN 2:PRINT"Diamond
Digger":PEN 1:LOCATE 14,4:PRINT STRIN
G$(14,CHR$(131))
1620 PEN 2:LOCATE 8,22:PRINT"PRESS ";
:PEN 1:PRINT"<SPACE>";:PEN 2:PRINT" t
o continue..."
1630 WINDOW#1,1,40,6,19:WINDOW SWAP 1
,0
1640 PEN 2:PRINT"Player Logged in....
.":PRINT" Move droids through mine zo
nes and     collect all the diamonds.
Avoid falling  rocks."
1650 PRINT" Collect all the diamonds
before device time runs out to collec
t a bonus.Beware computer logs out wh
en time runs out."
1660 PRINT" There are 10 mining zones
 to clear and the computer will give
a rating when allyour droids are dest
royed."
1670 PRINT" The rocks move down if no
thing is belowthem or across if there
 is a space to   either the left and
down or right and   down."
1680 WHILE INKEY$<>"":WEND
1690 GOSUB 2880
1700 LOCATE 1,1:PRINT STRING$(20,11)
1710 LOCATE 17,3:PEN 2:PRINT"The Keys
":LOCATE 17,4:PEN 1:PRINT STRING$(8,C
HR$(131))
1720 PRINT:PEN 2:PRINT"   Left:'Z'
         Right:'X'"
1730 PRINT:PRINT:PRINT"   Up:';'
         Down:'/'"
1740 PRINT:PRINT:PRINT"   Scroll Left
:'A'      Scroll Right:'S'"
1750 WHILE INKEY$<>"":WEND
1760 GOSUB 2880
1770 LOCATE 1,1:PRINT STRING$(20,11)
1780 GOTO 2460
1790 REM******Initial Set Up******
1800 MODE 1:CALL &BC02:INK 0,0:BORDER
 0:PAPER 0:CLS:LOCATE 15,10:INK 9,15:
PEN 2:PRINT"PLEASE WAIT!!"
1810 RESTORE 1880
1820 MEMORY &8C9F:check=0
1830 ENV 2,3,2,2,3,-2,2:ENT 1,30,10,1
:ENV 1,10,-1,2
1840 FOR f=0 TO 98
1850 READ n$:POKE &A000+f,VAL("&"+n$)
1860 check=check+VAL("&"+n$)
1870 NEXT
1880 DATA dd,7e,00,87,87,87,87,87,32,
47,a0,26,c0,dd,7e,04,3d,87,87,6f,11,5
0,00,dd,46,02,05,19,10,fd,e5,26,c0,dd
1890 DATA 7e,08,3d,87,87,6f,11,50,00,
dd,46,06,05,19,10,fd,11,fd,07,06,08,3
6,00,23,36,00,23,36,06,23,36,00
1900 DATA 19,10,f2,e1,11,00,a1,06,08,
```


```
1a,77,13,23,1a,77,13,23,1a,77,13,23,1
a,77,13,78,01,fd,07,09,47,10,e9,c9
1910 IF check<>7948 THEN LOCATE 10,10
:PRINT"ERROR IN DATA":PRINT CHR$(7):E
ND
1920 FOR f=0 TO 191:READ x$:POKE &A10
0+f,VAL("&"+x$):NEXT
1930 DATA 00,54,a8,00,a8,00,00,54,00,
fc,00,00,00,00,54,a8,54,a8,00,00,00,0
0,fc,00,fc,00,00,00,00,00,54,a8

1940 DATA 80,00,40,00,11,33,00,00,33,
33,22,80,11,33,00,00,00,00,40,00,00,9
1,33,00,80,33,33,22,00,11,33,40
1950 DATA 00,14,28,00,00,3c,3c,00,00,
2c,34,00,14,18,34,14,14,0c,30,3c,00,0
e,30,3c,00,2d,0c,34,14,2d,30,34
1960 DATA 14,0f,18,28,2d,0c,30,28,2d,
0e,18,34,14,0f,0c,20,14,3c,0e,08,00,3
c,2d,1c,00,00,14,1e,00,00,3c,28
1970 DATA 00,10,20,00,00,10,20,00,00,
60,90,00,00,64,98,00,10,1c,2c,20,10,1
9,26,20,30,91,62,30,30,19,26,30
1980 DATA 30,19,26,30,30,91,62,30,10,
19,26,20,10,1c,2c,20,00,64,98,00,00,6
0,90,00,00,10,20,00,00,10,20,00
1990 FOR f=0 TO 223:READ x$:POKE &A20
0+f,VAL("&"+x$):NEXT
2000 DATA 00,00,33,00,00,11,33,22,00,
40,33,22,cc,99,33,22,cc,99,19,22,00,0
4,33,22,00,00,33,00,00,00,33,22
2010 DATA 00,44,cc,88,00,14,3c,7c,00,
44,cc,dc,00,14,3c,7c,00,44,cc,88,04,0
c,0c,08,18,9a,9a,8e,04,0c,0c,08
2020 DATA 00,33,00,00,11,33,22,00,11,
33,80,00,11,33,66,cc,11,26,66,cc,11,3
3,08,00,00,33,00,00,11,33,00,00
2030 DATA 44,cc,88,00,bc,3c,28,00,ec,
cc,88,00,bc,3c,28,00,44,cc,88,00,04,0
c,0c,08,18,9a,9a,8e,04,0c,0c,08
2040 DATA 00,11,22,00,11,33,33,22,11,
33,62,22,33,66,99,33,33,66,99,33,33,0
c,0c,33,11,33,33,22,00,33,33,00
2050 DATA 00,11,22,00,11,33,33,22,11,
91,62,22,33,66,99,33,26,66,99,19,33,0
c,0c,33,11,33,33,22,00,33,33,00
2060 DATA 00,cc,cc,00,00,3c,3c,00,00,
cc,cc,00,00,3c,3c,00,00,cc,cc,00,04,0
c,0c,08,0c,08,04,0c,04,00,00,08
2070 REM**Rock Fall Routines**
2080 check=0:FOR f=0 TO 248:READ x$:P
OKE &9E34+f,VAL("&"+x$):check=check+V
AL("&"+x$):NEXT:IF check<>23947 THEN
MODE 1:PEN 3:LOCATE 5,10:PRINT"ERROR
IN ROCK FALL ROUTINES":END
2090 DATA 21,11,9b,36,00,23,23,23,7e,
fe,03,28,06,7e,fe,fe,20,f5,c9,e5,01,1
4,00,09,7e,fe,03,28,03,e1,18,e7,23,7e
,fe,00,28,03,e1,18,de,ed,42,7e,fe,00,
28,03,e1,18,d4,36,05,01,11,9b,0a,3c,0
2,e5,21,4c,9a,01,00,00,4f,09,d1,eb,01
,14,9b
2100 DATA ed,42,eb,73,e1,36,00,18,b5
2110 DATA 21,12,9b,36,00,23,23,7e,fe,
03,28,06,7e,fe,fe,20,f5,c9,e5,01,14,0
0,09,7e,fe,00,28,03,e1,18,e7,01,12,9b
,36,05,0a,3c,02,01,00,00,e5,21,af,9a,
01,00,00,4f,09,d1,eb,01,14,9b,ed,42,e
b,73,e1,36,00,18,c5
2120 DATA 21,13,9b,36,00,23,7e,fe,03,
28,06,7e,fe,fe,20,f5,c9,e5,01,14,00,0
9,7e,fe,03,28,03,e1,18,e7,2b,7e,fe,00
,28,03,e1,18,de,ed,42,7e,fe,00,28,03,
e1,18,d4,36,05
2130 DATA 01,13,9b,0a,3c,02,e5,21,e7,
99,01,00,00,4f,09,d1,eb,01,14,9b,ed,4
2,eb,73,e1,36,00,18,b5
2140 DATA 21,14,9b,3e,dc,f5,23,7e,fe,
05,20,03,3e,03,77,f1,3d,fe,00,20,f0,c
9
2150 REM***Scroll Routines***
2160 check=0:FOR f=0 TO 135:READ x$:P
OKE &9C40+f,VAL("&"+x$):check=check+V
AL("&"+x$):NEXT
2170 IF check<>18498 THEN MODE 1:PEN
3:LOCATE 5,10:PRINT"There is an error
 in Scroll Routines":END
2180 DATA dd,6e,00,26,01,cd,1a,bc,e5,
7e,f5,54,5d,eb,23,cb,f4,cb,fc,0e,47,7
e,12,23,cb,f4,cb,fc,eb,23,cb,f4,cb,fc
,eb,0d
2190 DATA 20,ef,f1,12,e1,7c,c6,08,67,
e6,38,20,d7,c9
2200 DATA dd,6e,00,26,01,cd,1a,bc,01,
47,00,09,cb,3a,cb,fc,e5,7e,f5,54,5d,2
b,cb,f4,cb,fc,0e,47,7e,12,2b,cb,f4,cb
,fc,eb,2b,cb,f4
2210 DATA cb,fc,eb,0d,20,ef,f1,12,e1,
7c,c6,08,67,e6,38,20,d8,c9
2220 DATA dd,7e,00,21,d8,8b,11,c8,00,
19,3d,fe,00,20,fa,11,14,9b,01,c8,00,e
d,a0,e2,c7,9c,18,f9,c9
2230 REM****Coded Screens****
2240 check=0:RESTORE 2260:no=35996:FO
R f=1 TO 10:FOR g=0 TO 49:no=no+4:REA
D n:check=check+n:POKE no,(n\64)+1:PO
KE no+1,((n AND 63)\16)+1:POKE no+2,(
(n AND 15)\4)+1:POKE no+3,(n AND 3)+1
:NEXT g,f
2250 IF check<>35612 THEN PEN 3:MODE
1:LOCATE 3,10:PRINT"There is an error
 in the Coded Screens":END
2260 DATA 64,150,40,150,1,64,0,190,0,
1,64,16,0,4,1,85,106,162,169,85,127,9
6,0,9,253,127,96,0,9,253,64,85,81,85,
1,64,170,162,170,1,64,170,162,170,1,6
4,0,0,0,1
2270 DATA 64,0,0,0,1,106,170,170,138,
169,64,0,0,10,161,72,8,21,15,193,72,8
,17,12,1,72,136,17,15,1,72,136,17,12,
1,74,168,21,12,5,67,240,0,0,21,64,0,0
,0,85
2280 DATA 64,0,0,0,1,74,160,0,10,161,
74,0,0,0,161,72,138,170,162,33,64,32,
40,8,1,64,27,195,228,1,64,27,195,228,
1,64,5,105,80,1,64,161,105,74,1,64,16
0,60,10,1
2290 DATA 74,170,170,170,129,74,170,1
70,170,129,79,255,63,254,129,64,0,0,1
0,193,64,0,0,43,1,66,34,0,172,1,64,0,
2,176,1,74,34,138,170,161,64,0,42,170
,161,64,0,3,252,1
2300 DATA 66,106,168,170,65,66,106,16
8,169,1,66,127,0,4,1,66,85,84,80,253,
64,0,0,26,169,64,84,81,80,129,64,4,66
,160,129,64,16,16,0,129,64,64,4,0,129
,65,0,4,3,241
2310 DATA 64,85,85,124,9,64,106,170,1
24,9,64,115,51,85,73,96,64,0,64,9,112
,69,84,64,9,80,71,244,5,73,72,5,84,6,
73,72,85,85,86,73,105,15,128,2,9,72,1
5,128,0,9
2320 DATA 64,0,40,0,45,72,10,136,0,41
,108,3,10,160,41,96,0,8,192,9,96,85,6
9,85,1,102,0,0,173,129,108,0,0,170,25
3,84,21,96,170,85,64,23,32,0,9,64,23,
96,0,13
2330 DATA 74,170,170,170,1,79,255,255
,255,193,69,85,85,85,65,71,232,0,43,6
5,71,232,8,41,1,70,162,160,36,1,70,10
,168,36,1,68,2,160,4,1,120,8,8,0,1,12
0,0,0,20,1
2340 DATA 68,5,85,80,17,100,5,170,80,
25,100,11,0,224,25,100,47,195,248,25,
```

IF YOU CAN WRITE SOFTWARE LIKE THIS - WE WANT YOU!!

(03) 560 4324

```
116,43,170,232,29,116,42,170,168,29,1
16,26,170,164,29,84,5,85,80,21,64,0,0
,0,1,64,0,0,0,1
2350 DATA 64,0,128,32,1,64,0,44,32,33
,106,165,81,86,137,106,6,162,150,137,
106,7,250,102,9,64,4,0,4,9,106,128,0,
4,9,127,213,85,68,41,106,128,10,128,9
,64,0,0,21,1
2360 FOR f=1 TO 20:POKE 39899+f,2:NEX
T:POKE 39920,254
2370 DIM hs(8),na$(8),jeget(10)
2380 RESTORE 2390:FOR f=1 TO 8:READ n
a$(f):hs(f)=4500-f*500:NEXT
2390 DATA CPC 189234,CPC 189234,CPC 1
89234,CPC 189234,CPC 189234,CPC 18923
4,CPC 189234,CPC 189234
2400 RESTORE 2410:FOR f=1 TO 10:READ
jeget(f):NEXT
2410 DATA 14,11,10,20,9,16,12,22,16,8
2420 a$=STRING$(255,"*"):x$=STRING$(2
55,"*")
2430 left=40000:right=40050:rockb=405
82:rockr=40500:rockl=40647:ch53=40727
:block=40107:a$=STRING$(255,"*")
2440 SYMBOL 255,198,165,198,165,6,40,
40,16
2450 RETURN
2460 REM******High Score*****
2470 MODE 1:INK 1,24
2480 LOCATE 4,5:PEN 3:PRINT CHR$(150)
;STRING$(32,CHR$(154));CHR$(156)
2490 LOCATE 16,3:PEN 2:PRINT"HIGH SCO
RE":PEN 3
2500 FOR f=6 TO 15:LOCATE 4,f:PRINT C
HR$(149):LOCATE 37,f:PRINT CHR$(149):
NEXT
2510 LOCATE 4,16:PRINT CHR$(147);STRI
NG$(32,CHR$(154));CHR$(153)
2520 FOR f=1 TO 8
2530 IF sc>hs(f) THEN GOSUB 2640:f=10
2540 NEXT
2550 FOR f=1 TO 8:PEN 1:LOCATE 8,f+6:
PRINT na$(f):LOCATE 18,f+6:PEN 3:PRIN
T".....";hs(f):NEXT
2560 LOCATE 1,17:PRINT STRING$(220,"
")
2570 IF INKEY$<>"" THEN GOTO 2570
2580 PEN 2:LOCATE 1,20:PRINT STRING$(
40,CHR$(154)):LOCATE 1,22:PRINT STRIN
G$(40,CHR$(154)):PEN 1
2590 LOCATE 1,23:PRINT STRING$(40," "
)
2600 LOCATE 11,21:PEN 3:PRINT"PRESS";
:PEN 1:PRINT"< SPACE >";:PEN 3:PRINT"
TO PLAY.":PEN 1
2610 WHILE INKEY$<>"":WEND
2620 GOSUB 2880
2630 GOTO 70
2640 a$="ABCDEFGHIJKLMNOPQRSTUVWXYZ
#%&()!{}?*+"+CHR$(255)
2650 c=19:LOCATE 1,20:PEN 1:PRINT a$
2660 LOCATE 3,17:PEN 1:PRINT"USE CURS
OR KEY'S LEFT,RIGHT AND COPY":LOCATE
4,18:PRINT"TO SELECT LETTERS.(MAXIMUM
 OF 10.)":PEN 2:LOCATE 1,19:PRINT STR
ING$(40,CHR$(154)):LOCATE 1,22:PRINT
STRING$(40,CHR$(154))
2670 LOCATE 12,23:PEN 3:PRINT"PRESS";
:PEN 1:PRINT"< 'X' >";:PEN 3:PRINT"TO
 EXIT.":PEN 1
2680 x$=""
2690 FOR z=1 TO 10
2700 LOCATE c,21:PEN 2:PRINT" "
2710 IF INKEY(1)=0 AND c<40 THEN c=c+
1
2720 IF INKEY(8)=0 AND c>1 THEN c=c-1
2730 IF INKEY(9)=0 AND c=40 THEN LOCA
TE 7,f+6:PRINT"          ":z=100:GOT
O 2780
2740 IF INKEY(63)=0 THEN z=11:GOTO 27
80
2750 IF INKEY(9)<>0 THEN LOCATE c,21:
PRINT"*":FOR a=1 TO 50:NEXT:GOTO 2700
2760 x$=x$+MID$(a$,c,1):LOCATE 7+z,f+
6:PEN 1:PRINT MID$(a$,c,1)
2770 FOR a=1 TO 200:NEXT
2780 NEXT
2790 IF z=101 THEN GOTO 2680
2800 hs(8)=sc:na$(8)=x$
2810 f=0
2820 FOR z=1 TO 7
2830 IF hs(z)<hs(z+1) THEN t=hs(z+1):
hs(z+1)=hs(z):hs(z)=t:a$=na$(z+1):na$
(z+1)=na$(z):na$(z)=a$:f=1
2840 NEXT
2850 IF f=1 THEN GOTO 2810
2860 fr=FRE("")
2870 RETURN
2880 REM********TUNE********
2890 RESTORE 2980
2900 READ p,d:d=d*10:IF p=999 THEN RE
STORE 2980:GOTO 2900
2910 GOSUB 2950
2920 SOUND 1,pn,d*0.8,15,1,2:SOUND 2,
pn+3,d*0.8,15,1,2:SOUND 4,pn/0.5,d*0.
8,15,1,2
2930 IF INKEY(47)<>0 THEN 2900
2940 RETURN
2950 fr=440*(2^(0+((p-10)/12)))
2960 pn=ROUND(125000/fr)
2970 RETURN
2980 DATA 181,1.5,181,1,3,2,6,2,10,2,
15,4,17,2,15,2,12,2,15,2,10,4,6,2,3,2
,6,2,10,2,15,4,17,2,15,2,12,2,15,2,10
,6,10,2,12,2,15,2,6,4,10,2,6,2,5,2,3,
2,5,4,6,2,10,2,12,2,15,2,10,2,10,2,3,
2,6,2,5,3,3,1,3,8,181,1.5,161,1.5,161
,1,3,2,6,2,10,2,15,4
2990 DATA 17,2,15,2,12,2,15,2,10,4,6,
2,3,2,6,2,10,2,15,4,17,2,15,2,12,2,15
,2,10,6,10,2,12,2,15,2,6,4,10,2,6,2,5
,2,3,2,5,4,6,2,10,2,12,2,15,2,10,2,10
,2,3,2,6,2,5,3,3,1,3,6,181,8,181,8,18
1,8
3000 DATA 181,1.5,6,4,3,4,6,4,15,6,12
,2,8,4,12,4,6,8,10,6,12,2,13,4,12,4,1
0,4,8,4,6,6,5,2,3,4,6,4,12,4,12,4,15,
4,12,4,8,4,10,4,12,4,13,2,13,2,12,4,1
0,4,8,2,6,2,6,8
3010 DATA 181,1.5,6,4,8,4,12,6,6,2,8,
4,12,4,6,6,6,2,8,4,8,4,17,6,15,2,15,2
,13,6,181,1.5,6,4,18,4,15,4,12,4,10,4
,8,4,10,4,6,6,6,2,8,4,17,4,15,6,13,2,
13,2,12,6,181,1.5
3020 DATA 6,4,3,4,6,4,12,6,6,2,8,4,12
,4,6,6,6,2,8,4,17,4,15,6,13,2,13,2,12
,6,181,1.5,6,4,3,4,6,4,12,6,6,2,8,4,1
2,4,6,6,6,2,6,4,6,4,17,6,15,2,15,2,13
,6,181,1.5,6,4,18,4,15,4,12,4,10,4,8,
4,12,4,6,6,6,2,8,4,17,4,15,6,13,2,13,
2,12,6,181,2,181,20,181,20,181,20
3030 DATA 999,999
```

Give your fingers a rest . . .

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

AMSTRAD BOOSTS DISCS OUTPUT

AMSTRAD has ramped up production of its 3in discs to in access of one million a month from a previous high of 750,000 units.

The news has been released by the company in order to dispel rumours of severe shortages which have been openly circulating within the industry in the UK.

As evidence of the current situation, Amstrad made available a letter from its Japanese supplier detailing current production levels.

Writing to Amstrad chairman Alan Sugar, two directors of Matsushita Electrical Industrial agreed that the correspondence should be made public in the light of "bad rumours" in Europe.

"We know that writing this letter to you is somewhat irrelevant because the information within it is quite common knowledge between our two companies", wrote Masami Nakata and Masayoshi Horie in the joint communication.

"However, we hope that it serves as some evidence for you to show to important customers and journalists".

The letter then went on to confirm that the manufacturers were enjoying "good business" supplying 3in floppy disc mechanisms to Amstrad.

It pointed out that at no time did Matsushita maintain a big stock, always making the units as a result of firm orders placed three months in advance.

On the critical subject of the discs themselves, the Japanese confirmed they had increased production from 750,000 to one million a month to meet the demand.

"And we can continue to increase production at your request", the letter concluded.

However, reports still persist of shortages within the UK.

"The only reason for this is that overseas markets are getting more than their fair share – or perhaps total production needs to be much higher than even a million a month", commented one leading dealer.

Scenery on software

A FORMER Scottish university lecturer who took voluntary redundancy to write software for the Amstrad range has launched his third major package.

The latest offering from Brian James, who used to teach environmental physics at Aberdeen, is Landscape Utility. It follows his previously successful Country Cottages and Landscape Creator programs.

"This program is an intelligent utility, a logical extension of Landscape Creator", said James.

Using a unique application of machine intelligence, it spontaneously designs landscape views. Every day you get a different show.

Inspiration for this work came from the magnificent scenery in Scotland.

Landscape Utility costs £19 on tape, £23 on disc, and is marketed directly by the author from 21 Lamond Place, Aberdeen AB2 3UT.

Record line-up for the big show

THE fourth Amstrad Computer Show has attracted a record number of new companies among its 70 exhibitors.

About 20 firms who have only recently boarded the Amstrad bandwagon will be exhibiting for the first time when the show opens at the Novotel, Hammersmith, on June 13.

In each case they will be offering new products, swelling the total number of launches to an all time high of more than 120.

"The demand for exhibition space has been phenomenal", says Derek Meakin, head of Database Exhibitions. "There is no doubt this has been helped by the number of additional companies that are springing up as the Amstrad boom continues unabated.

"As a result we have had to more than double the available floor space – from 14,000 sq ft to 32,000 sq ft".

One exhibitor, Software City, plan to unveil three new products – Job Costing for the construction industry at £99.95, Bar Stock Control for hotels, public houses and restaurants at £99.95, and Solicitor's Accounting program, at £299. All are for the PCW8256 and the PCW8512.

Software City said they would also offer 13 per cent discount because they had booked Stand 13!

Another newcomer, Data-Gem from Gemini, is a database for the PCW range which holds up to 32,000 records.

Instant Access, from Minerva, is for the CPC series. It costs £29.95.

Electric Studio will launch its Light Pen and Graphics package at £79.95. It plugs directly into the expansion port of the PCW and simulates a "real world" pen by allowing the user to accurately freehand draw directly onto the screen.

Micro Maths, from LCL, is for the PCW8256 and PCW8512, and is a £24 self-tuition course of 24 easy-to-use programs, covering 59 topics. It takes beginners from the age of eight to O-level and GCSE standards.

There is even an examination paper at the end when the program is completed.

Prospell, a spelling checker, and Protext, a word processor, mail merge, spelling checker and disc utility, is Arnor's latest for the PCW8256 and PCW8512. Prospell costs £29.95 and Protext £79.95.

A new improved version of Money Manager, which includes full VAT facilities, has been announced by Connect Systems at £24.95.

Babani Publishing's 192-page, illustrated book, Practical Reference Guide to Word Processing on the Amstrad PCW8256 and PCW8512, is priced at £5.95.

A.K. Marketing has produced a new range of joysticks, while Garwood Wholesale has new dust-covers, beige with red piping, at £11.45, for the PCW8256.

The Cavalier Stock Invoice program is new from Load and Run.

Vanguard Leisure has two musical products, the Maestro, at £39.95, and The Music Master, at £12.95 (cassette) and £16.95 (disc) for the CPC range. These use the keyboard letters to reproduce piano notes.

Mirrorsoft's Biggles game, now out for CPC machines, is backed-up with pilot scarves.

Mobile Amstrad mounts drive for youth jobs

Members of the Marshall Youth Centre at Sheffield work on an Amstrad CPC6128

PHYSICALLY and mentally handicapped children, youth club members and the young unemployed are being introduced into the computer world by a mobile Amstrad CPC6128.

Liz Prior, computer projects supervisor at the Earl Marshall Youth Centre at Sheffield, South Yorkshire, runs a daily mini-bus with nine other helpers, visiting special clubs for the handicapped and "drop-in" centres for the jobless.

The project has developed within the centre with the help of youth club leader Gwyneth Harris.

Recently Gwyneth and Liz were delegates to the national conference in London where the application of computers in youth work was debated.

Youth leaders were told by Ed Berman, founder of the educational charity Inter-Action: "Youth clubs could transform themselves into occupational springboards if they used computers in an imaginative way".

In the Sheffield area Liz is aiming for just that: "If we can give the youth of today the opportunity to have a go with computers they could go on, guided by us, to evening classes and colleges providing computer training".

Arcade shoot-out type games are discouraged. "We carefully select software and encourage the use of strategy games, adventures and quiz-based programs which require logical skills or participation with others", she said.

While the computer is taken about the area youth clubs priorities are given to unfunded organisations, special clubs for the disabled and youth clubs which have integrated handicapped people. Said Liz, "When working with any disabled young people we try to use a touch pad and joysticks to reduce the need for the keyboard and this seems successful.

"We also encourage sessions just for girls, as we feel they can gain more confidence working together without intimidation from the boys.

"As our users progress from games they are encouraged to develop their skills. We have had many who have gone on to college and evening classes".

The Department of Education and Science has announced a £75,000 grant over the next three years to Inter-Action for their youth club community computer projects, subject to them being able to raise sufficient funds to match.

Day Arnold hit the bullseye

WHEN the boss invited Arnold out to his first pub the pint-sized office lad scored a big hit with the locals, and in no time at all he helped raise more than £700 for charity.

Colin Watkinson, a director of Graycol Enterprises, a Manchester typesetting company, said: "He was superb. But there was one complaint – he never bought a pint all night".

Not surprising, for Arnold is the firm's micro, a one-year-old Amstrad 464.

Colin and his partner, Philip Gray, pressed Arnold into use when the local darts team held a 24-hour-long sponsored darts marathon. "I suggested Arnold would be an expert scorer because of his experience of working in accounts and invoicing", said Colin.

"Many distinguished guests popped in to see him, including the Lord Mayor of Manchester", he added.

Arnold ran without a blink for 23 hours and 35 minutes, recording the final target score.

"We were proud of him, especially considering he had just finished a full week's work", said Colin.

Arnold should be retired now because his big brothers, the 664 and the 8256, have joined the office staff. However, Colin did not have the heart to pension him off, so he is being kept on in a part-time capacity.

Colin concluded: "He has proved one of the best, and I would not hesitate in writing the perfect reference for him".

Taped tutor tells all

AMSTRAD users can now use a taped tutor to help understand their machines. Reel-Time's new series covers the range of business software for the PCW8256 and 8512.

The user just sits at his own computer, slots a tutor cassette into a tape recorder and follows the instructions.

First titles released are Word Processing with LocoScript, Using and Understanding CP/M and Basic, and Using and Understanding SuperCalc2.

Another range being developed includes NewWord and Multiplan. Cassettes cost £9.95.

Program to create catalogues

MAGIC Filer from Sagesoft is a new program to help Amstrad owners create, update and read computerised catalogues or lists organised by classifications.

The program provides a logical system of creating and using a catalogue with simple menu choices and freedom to classify information without a formal structure.

There are two operating modes – read only for information users and read/write for information providers.

Each page of information can be labelled with up to 32 keywords making selective searching quick and easy, and there is a simple built-in text editor. Price: £69.99.

Focus on fun

TWO battle games have been released by Ariolasoft on disc for Amstrad users.

Skyfox entails piloting an advanced jet plane and fighting off attackers. The other, Panzadrome, is more down to earth – building and running a tank while avoiding anti-tank guns.

Skyfox costs £15.95, Panzadrome £12.95.

★ ★ ★

A SEQUEL to last year's chart-topping adventure Red Moon has been released by Level 9 Computing to run on Amstrad machines.

Price of Magik has more spells and independent creatures to control than Red Moon and features full sentence commands, a 1,000 word vocabulary, 200 illustrated locations and multi-tasking. Cassette price is £9.95.

Level 9 has also issued Colossal Adventure, Adventure Quest and Dungeon Adventure as a triology on disc for £19.95.

★ ★ ★

ROBIN Hood turns detective to right the wrongs in Sherwood Forest in the new adventure game for Amstrad users.

CRL's light-hearted graphic adventure, Robin of Sherlock, is full of dangers lurking in Delta 4's strange forest – including an exploding Friar Tuck.

The price of £7.95 includes a bonus program, the Delta 4 Guide to Greater London, a humorous look at various places in the capital.

★ ★ ★

AMSOFT has released the Cyrus II chess game for the PCW8256 and the PCW8512.

The program, with 12 different levels of play from beginner to expert, now covers the full Amstrad range.

The package provides a print-out of the game, the current position of the pieces and a listing of the moves.

Price: £13.95.

Rainbird adventures on way

FOLLOWING major deals with two top independent software houses a new series of Amstrad adventure games from Rainbird Software is in the pipeline.

The Pawn is the first of seven adventures to be written for Rainbird by Magnetic Scrolls over the next two years, and it will be available for users of the CPC6128 and the PCW8256 within the next four months.

This will be followed by The Guild of Thieves in the autumn.

Level 9 Computing has also signed an agreement to provide three Rainbird products for the Amstrad machines over the coming year.

Publisher Tony Rainbird said: "Part of the Rainbird expansion will involve reversing the current trend of imported American software.

"We have already made a considerable impression on the American market with British games and utilities", he added.

Anita Sinclair, managing director of Magnetic Scrolls, said: "This agreement with Rainbird enables us to continue our research into natural language and other artificial intelligence related projects".

Rainbird will initially market Level 9's rewritten Colossal trilogy – Colossal Adventure, Dungeon Adventure and Adventure Quest – as a disc.

A second multi-load adventure, provisionally called Knight Orc, is due out in September, with a third probably following two months later.

NOT even the diligence of "Policewoman" Rachael Davies could stop the intrepid Redhawk from releasing the new adventure for the Amstrad CPC464, 664 and 6128.

"PW" Davies, editorial manager for publishers Melbourne House, tried to make a fun arrest outside their offices at Hampton Wick, but failed miserably.

Redhawk took five months to write, and the programmers Simon Price – topless above – and Mike Lewis – alias Redhawk's alter ego, Kevin – were determined to let everyone know the program was complete.

Rachael said: "There is no stopping Redhawk once he gets into action, even when you pose as a woman cop".

The Amstrad CPC cassette version costs £8.95.

Forecast for four

MOSAIC has already announced release dates of four new computer games for Amstrad players.

The first, Snow Queen, the Hans Christian Andersen fairy tale, will be out soon, followed in summer by The Story of the Amulet.

The late summer will bring The Growing Pains of Adrian Mole, which allows the sharing of the agonies of life of the popular youngster.

The Archers, based on Britain's longest-running serial, comes out in the autumn.

Dr Who back

DR WHO is back early for the Amstrad screens. The popular TV series restarts in autumn, but Micro Power has already brought out Dr Who and the Mines of Terror.

The game uses the Doctor's ingenuity to solve space-boggling problems.

To help him the Time Lords have given the Doctor a programmable cat, Splinx, a feline version of TV's K9. Price: £11.95.

Realism the name of the game

ONE for Amstrad soccer fans is Virgin Games' new release, FA Cup Football, which can be played by up to eight people at once.

You select your favourite teams, with the option of a fantasy side, in the hope of winning through to the final.

There is a giant-killer factor built in which can send any team crashing out of the cup.

Sports writer Tony Williams, editor of the League and Non-League Directories, did the calculations of the performance figures of every team that competes in the FA Cup and the Football Association has approved the game.

Virgin boss Nick Alexander says: "It's like a football manager simulation program, but Tony Williams' figures mean you are basing results on how the teams perform in reality".

The game costs £7.95.

From Page 46

returns it in register A, just like the normal Console Input.

However if there is not a character ready it also returns immediately, but this time has the value 00 – an Ascii null character – in register A. Unlike the normal Console Input this does not wait until a character has been typed before returning.

This function is extremely useful if we wish to check for input occasionally – for example, an abort command – while doing something else. If we used the normal Console Input for this our main program would stop until someone pressed a key.

Get IOBYTE requires only the function number in register C, and returns the current value of the IOBYTE bit field in register A. This is a particularly useless function as the IOBYTE is always found at address 3 of the SPA in any CP/M system we might as well just get it direct with an instruction such as:

which is all the BDOS does anyway, without going through the rigmarole of a function call. The function was originally provided for compatability with early versions of CP/M which did not support the IOBYTE concept, but nowadays it is redundant.

Set IOBYTE is the companion function to the previous one, and is equally useless. Again it can be replaced simply by a single instruction such as:

```
ld    (0003h),e
```

If you really do want to use it you need function number 8 in register C and the new pattern for the IOBYTE in register E. It returns nothing.

Print String is extremely useful and requires function number 9 in register C and the memory address of a string of characters in register pair DE.

It takes a string of Ascii characters starting at address DE and terminated with a $ character, and prints the string on the logical CON: device, which is normally the Amstrad's screen.

As it works internally by calling Console Output for each character to be printed it obeys the same rules – typing Ctrl+S will pause the output, Ctrl+C will then abort and any other key will restart the output. If the Ctrl+P toggle has previously been set the string is also sent to the LST: device.

Read Console Buffer requires function number 10 in register C and the memory address in register pair DE of an input buffer. The layout of this buffer is shown in Figure I.

This function lets us input a string of characters from the keyboard, complete with line editing facilities like those available on CCP command lines. On an Amstrad these are as shown in Table I.

We must have previously initialised the first byte of the buffer with a value for its maximum size, which is the maximum number of characters we are going to allow the person running the program to type at the keyboard before we cut him or her off. This can be any number between 1 and 255.

The function will return either when the buffer has filled to the size we have specified or before that if Enter is pressed to show end of input. When the function returns CP/M has worked out the length of the string which was typed in and has put this number into the second byte of the buffer.

Ctrl+C	Aborts and warm boots if at start of line.
Ctrl+P	Printer echo toggle – typing this echoes everything printed on the screen from then on to the LST: device. Typing Ctrl+P again turns this off.
Ctrl+R	Retypes the line.
Ctrl+X	Deletes all characters already typed and repositions cursor to start of line to let you start again.
Del	Deletes last character typed.
Enter	Terminates input and returns the string to our program.

Table I: Read Console Buffer line editing facilities

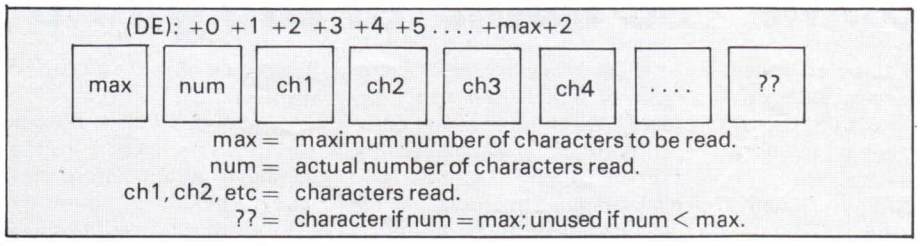


Figure I: Console buffer structure

Get Console Status requires function number 11 in register C, and returns a value in register A. If the value is 00 no character has been typed since the last character was read. If the value is &FF a new character has been typed and is waiting to be read.

This function is another pretty useless one. If you remember Direct Console I/O will scan the keyboard and return not only the status but also the character if one is waiting. This is quicker and easier than calling this function to check the status, and then having to check the value returned and call another function to actually read the character if one is ready.

Return Version Number requires function number 12 in register C, and returns values in registers H and L which specify the version of CP/M present on the machine.

This allows programs to work out exactly what facilities and functions are available to them on different machines which may be running different versions of CP/M.

Older versions such as 1.4 and 2.0 do not have as many BDOS functions as we do with version 2.2, and some common functions work slightly differently.

Similarly newer or more powerful versions such as 3.0 (CP/M Plus) and MP/M (the multi-user system version of CP/M) can do many things which we can't.

If register H returns with a value of 01, then we are in an MP/M system. Otherwise register H will contain 00 to signify normal CP/M. In this case register L will contain 00 for versions before 2.0, and values of &20 or higher for newer versions. So our CP/M version 2.2 will return a value of 00 in register H together with &22 in the L register.

● *Next month we'll look at an assembler/editor combination.*

MYRDDIN

Presents:-

FLIGHT SIMULATION

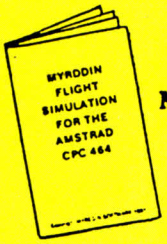
LOADING.....

3D LANDMARKS YOU CAN FLY AROUND

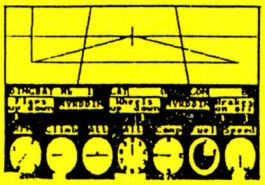
SUPERB REAL TIME SIMULATION

MYRDDIN FLIGHT SIMULATION

AMSTRAD CPC 464

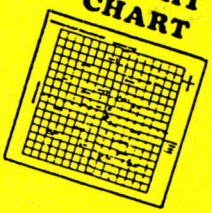


MANUAL

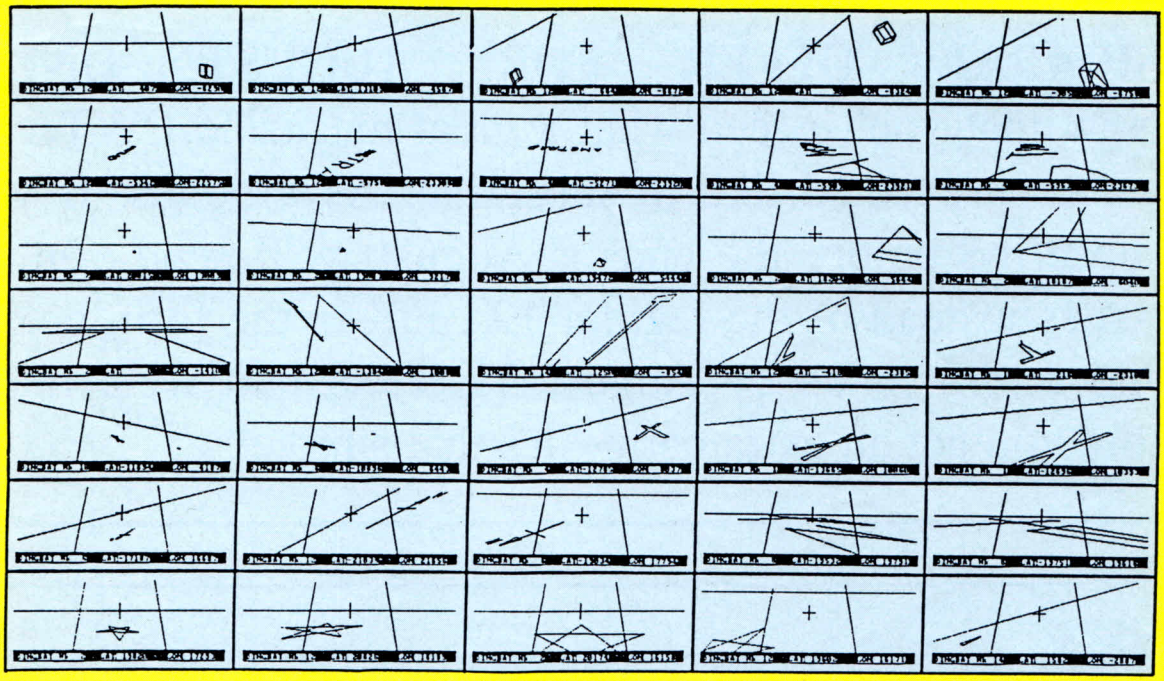


FULL SCREEN DISPLAY

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 $14.95 **DISK $26.95**

THE BAKERS DOZEN!

SPECIAL OFFER FOR AUGUST ONLY.
Now that you've seen what a great magazine CWTA is, we're sure you'd like to subscribe. As a bonus, If you subscribe before the end of August we'll send you 13 issues instead of the regular 12 . Fill in the form below and return to us by August 31, 1986 and we'll make sure you get an extra copy - it's that simple!

NAME:_____

ADDRESS: _____

TELEPHONE: _____

SIGNED: _____

BANKCARD OR MASTERCARD WELCOME - YOUR NUMBER AND EXPIRY DATE PLEASE:

NUMBER: _____ EXPIRY DATE:_____

MAGAZINE ONLY : $36 MAGAZINE PLUS MONTHLY TAPE : $72

We regret that, at this stage, we are unable to offer tapes of individual issues and that tapes are only available with magazine subscription.

PLEASE NOTE: TAPES ARE ONLY SUITABLE FOR CPC MACHINES.

STRATEGY SOFTWARE, BOX 5000, GLEN WAVERLEY, VICTORIA 3150
(03) 560 4324