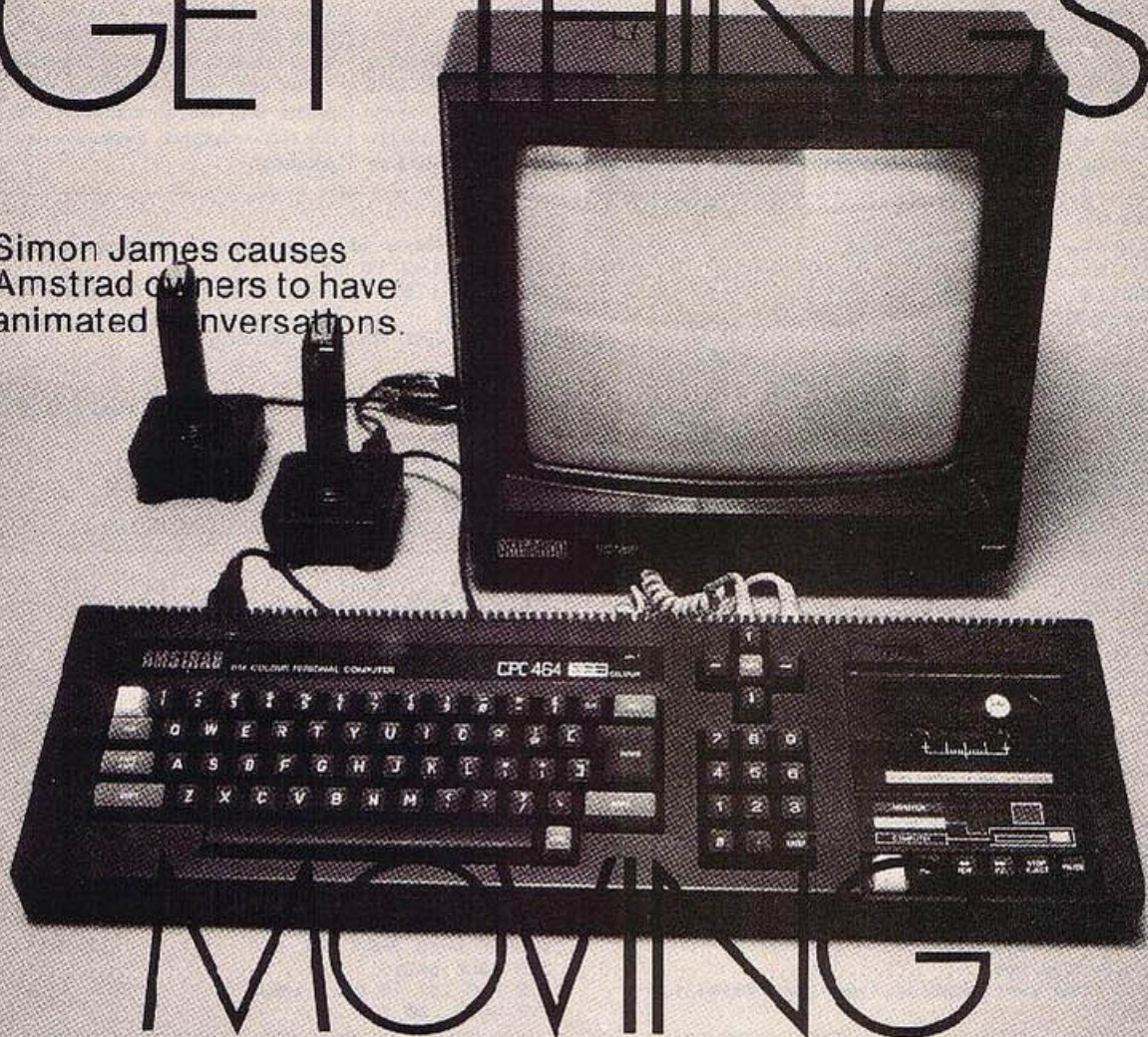


GET THINGS

Simon James causes
Amstrad owners to have
animated conversations.



MOVING

AN INTERESTING and very useful feature of the new Amstrad computer, is its Ink command. This enables you to redefine the colours available in any mode to those which suit you. The User Guide explains it by reference to pens and ink. For instance, in mode 0 there are 16 colours available, from a palette of 27 colours. This is like having 16 pens and 27 ink pots available and the ink in any pen can be changed. It is something like VDU 19 command on the BBC micro. As an example type in:

```
'Reset' (Press 'Ctrl' and 'Shift' and then  
'Esc', holding all three down) INK 1,9  
'Enter'
```

You will see that all the writing on the screen has changed colour — or shade if you have a green monitor. This is because when you switch on or Reset the computer, pen 1 is being used, and is set to colour 24, bright yellow. The above Ink command told the computer to change the ink in pen 1 to number nine, green. Try other numbers, for instance,

```
INK 1,15 (changes the ink in pen 1 to
```

orange) until you get the hang of it.

This command can be used to produce animation — movement — on the screen. It is done by redefining all the available colours to the background colour, and making one colour at a time visible. Type in listing 1 and run it. You will see at the start, the screen being filled up with multi-coloured vertical lines. Then most of them disappear, and a few start to move across the screen.

In actual fact, nothing is really moving. What happens is that lines are drawn using the 16 different pens. These pens are defined as various colours after a Reset or switch on. You can see what these colours are by reference to pages F3.2 and F3.4 in the manual. The program then changes the ink in all of these pens to black — the background colour — so the lines appear invisible. Then each pen in turn is filled with bright white, and the effect of movement is achieved.

Program 2 is the part of the program that actually changes the colours. Study carefully how it works and then type it in and save it. It

can then be merged — see Ch.8 page 27 — into any programs you write which use this technique.

Program 3 shows the spectacular effects that can be produced by this usage of the Ink command. 15 circles are drawn — using Sin and Cos, see page F3.13 — using different pens, and each one illuminated in turn.

This method can also be a lot faster than other ways of animating, because all that is required is a few lines of Basic, and several objects at once can move, whereas other methods would have to deal with each in turn — and would therefore be slower. Program 4 is a good example of this. Pens 2 to 15 are used for the movement, pen 0 being the background colour, and pen 1 the pen used for the writing, and the boxes. If pen 1 was used in the animation sequence, the boxes would flash on and off. Note that this effect could be quite useful, but was unnecessary in this program.

Another point to note is that in programs 1, 3 and 4, if you break, using (Esc) (Esc), and Run them again, the initialisation sequence
(continued on next page)

(continued from previous page)

where the multicoloured lines are drawn, will be invisible since all the pens have been filled with black ink, and are being written onto black paper. So do not assume something has gone wrong with the program, if nothing happens for a few minutes. This is most noticeable in program 3 with the circles, in which the circles take a while to draw.

In programs 3 and 4 I have included a delay loop which can be altered by changing the variable 'delay' defined at the beginning of the programs. If you remove the loop altogether,

you will get some idea of the speed that can be achieved by this method. And don't forget that no matter how many moving objects you had on the screen, the speed would be the same!

Another thing that was put into the programs was:-

CALL &BD*9

This is a call to the machine operating system which makes the computer pause until the next frame, — i.e. until a full sweep of the screen is finished by the mechanism inside the monitor. Each frame takes about 1/50 of a se-

cond, so this does not slow down the program noticeably, but prevents some strange effects taking place. As an example of what can happen without this instruction, remove line 205 from Program 1, by typing:

205(Enter)

and run the program. You will see that sometimes there are two vertical lines close to each other crossing the screen, sometimes just one.

You can now try writing your own programs — if you feel confident enough! — using this technique, because it has enormous potential.

Program 1.

```
10 REM *** PROGRAM ONE ***
15 REM Initialise
20 MODE 0
30 INK 0,0
40 PAPER 0
50 col=1
55 REM Draw multicoloured lines
60 FOR x=10 TO 639 STEP 10
70 MOVE x,5
80 DRAW x,395,col
90 col=(col+1) MOD 16
110 IF col=0 THEN col=1
```

```
120 NEXT
125 REM Make all the pens black except pen 1
130 FOR i=2 TO 15
140 INK i,0
150 NEXT
155 REM Get things moving
160 x=1
170 INK x,26
180 IF x=1 THEN INK 15,0 ELSE INK x-1,0
190 x=x+1
200 IF x=16 THEN x=1
205 CALL &BD19
210 GOTO 170
```

Program 2.

```
3000 REM ROUTINE TO SWAP COLOURS
3001 REM Use -or merging into your own programs
3002 FOR i=2 TO 15
3003 INK i,0
3004 NEXT
```

```
3005 x=1
3006 INK x,26
3007 IF x=1 THEN INK 15,0 ELSE INK x-1,0
3008 x=x+1
3009 IF x=16 THEN x=1
3010 CALL &BD19
3011 GOTO 3005
```

Program 3.

```
10 REM *** PROGRAM THREE ***
20 MODE 0
30 delay=20
40 INK 0,0
50 DEG
60 FOR i=1 TO 15
70 FOR j=1 TO 360 STEP 1
80 ORIGIN 320,200
90 PLOT (180-12*i)*COS(j), (180-12*i)*SIN(j),i
100 NEXT
```

```
110 NEXT
120 FOR i=2 TO 15
130 INK i,0
140 NEXT
145 REM FIRE AWAY!
150 x=1
160 INK x,26
170 IF x=1 THEN INK 15,0 ELSE INK x-1,0
180 x=x+1
190 IF x=16 THEN x=1
200 CALL &BD19
210 FOR i=1 TO delay:NEXT
220 GOTO 160
```

Program 4.

```
10 REM *** PROGRAM 4 ***
20 MODE 0
30 RESTORE
40 delay=100
50 PEN 1
60 LOCATE 8,2:PRINT "BOXES"
70 FOR i=1 TO 4
80 READ x,y
90 GOSUB 290
100 NEXT
110 DATA 100,300,100,50,500,50,500,300
120 RESTORE
130 FOR a=1 TO 4
140 READ x,y
150 GOSUB 430
160 NEXT
170 FOR i=3 TO 15
180 INK i,0
190 NEXT
200 x=2
210 INK x,26
220 IF x=2 THEN INK 15,0 ELSE INK x-1,0
230 x=x+1
240 IF x=16 THEN x=2
```

```
250 CALL &BD19
260 FOR q=1 TO delay:NEXT
270 GOTO 210
280 END
290 REM SUBROUTINE TO DRAW BOXES
300 MOVE x,y
310 DRAWR 0,-50,1
320 DRAWR 65,0,1
330 DRAWR 0,50,1
340 DEG
350 MOVE x,y
360 FOR z=1 TO 360 STEP 20
370 ORIGIN x,y
380 DRAW 5*COS(z),5*SIN(z),1
390 NEXT
400 ORIGIN 0,0
410 RETURN
420 REM SUBROUTINE TO DRAW LIDS
430 DEG
440 col=2
450 FOR z=1 TO 90 STEP 6.9
460 ORIGIN x,y
470 DRAW 60*COS(z),60*SIN(z),col
480 col=col+1
490 NEXT
500 RETURN
```