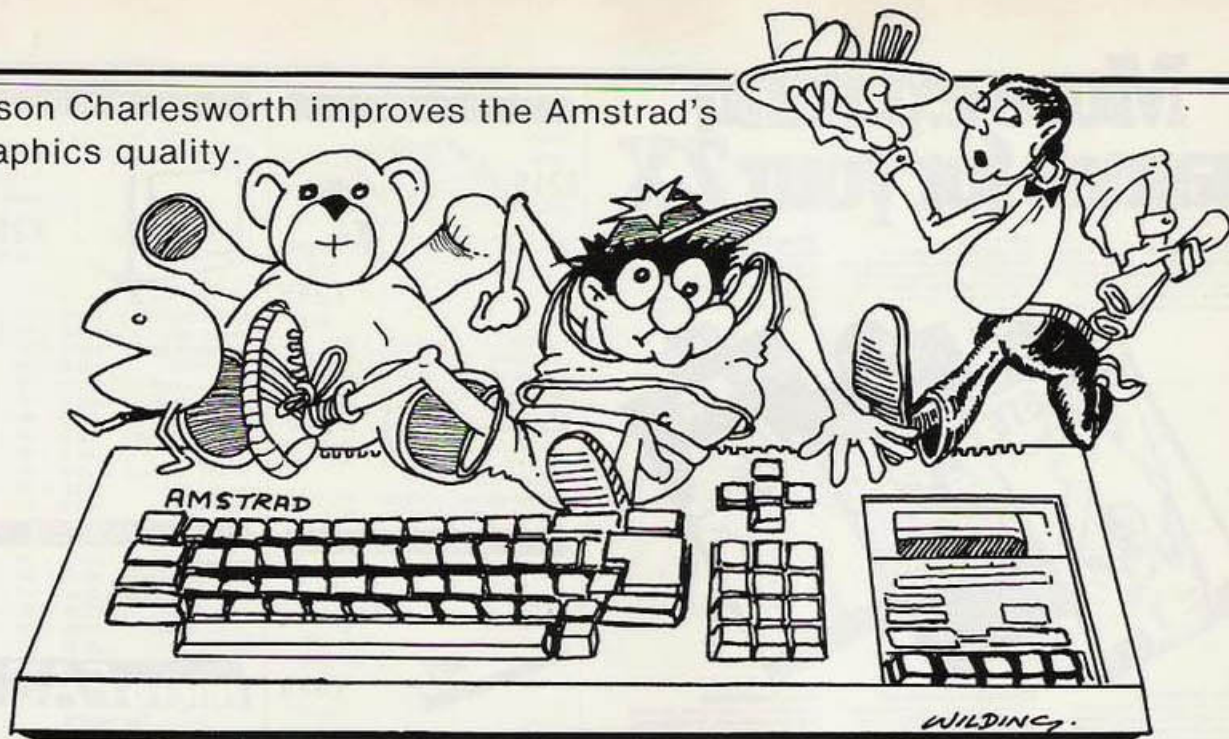


Jason Charlesworth improves the Amstrad's graphics quality.



# AMSTRAD SPRITES

THE AMSTRAD is a very well designed computer which is extremely strong in all areas except that of graphics. It has the usual Plot and Draw commands but does not contain any sprite routines, Fill commands or Scroll command unlike most recent computers. This article should rectify this by

adding five new Basic commands to Amstrad Basic. These are actually new commands, not just a sequence of Pokes and Calls.

Listing 1, when entered and run will provide these commands but before the commands can be used they must be activated by Call 37000. The commands provided are:

**SPRITEON,X,Y,Z** — This works only in mode 0. X is the X axis co-ord (0-143) and Y is the Y axis — top = 0, bottom = 183 — and Z is the Sprite number — 1 to 15. If any of these are out of range, the routine will return, having done nothing.

(continued on page 147)

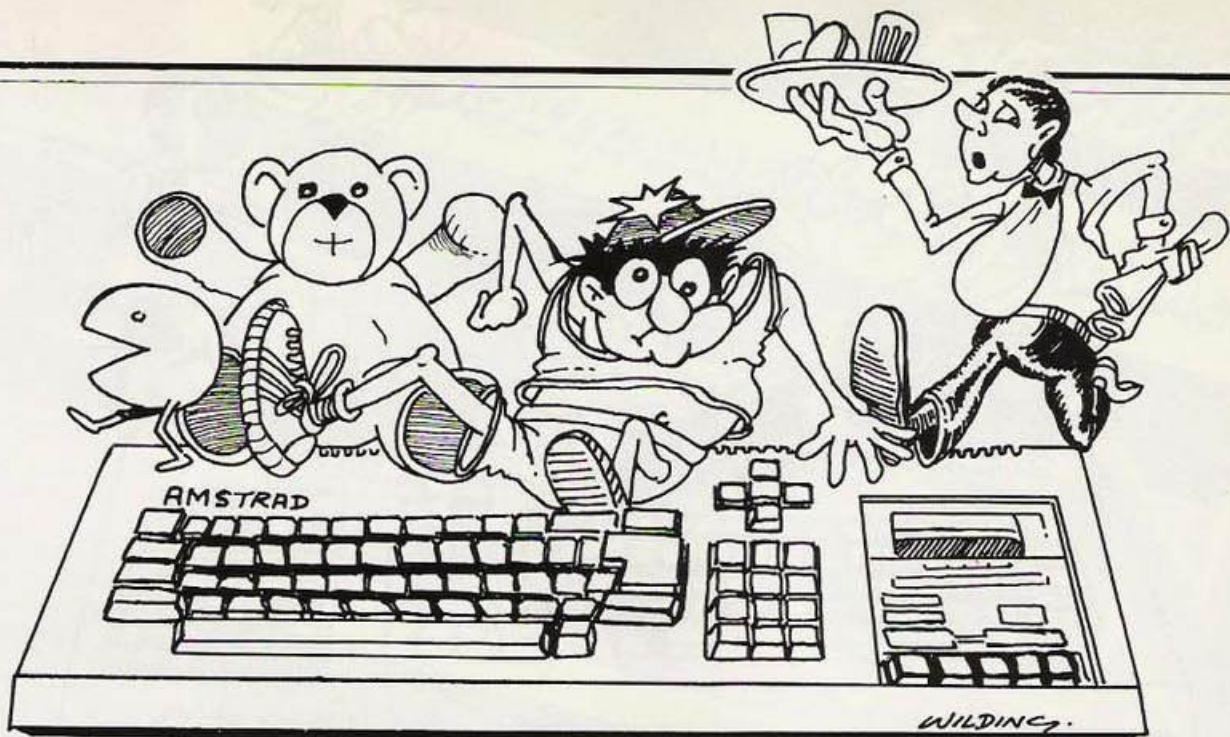


Listing 1.

```

10 MEMORY 36999
20 CLS:PRINT "Poking in progress, do not disturb !"
30 x=37000:RESTORE
40 READ a$:IF a$="9999" THEN GOTO 80
50 FOR a=1 TO LEN (a$) STEP 2:POKE x,VAL("&"+MID$(a$,a
,2))
60 x=x+1:NEXT a
70 GOTO 40
80 CLS:PRINT "Finished"
90 DATA 01929021CB90CDD1BCC9A390C3D090C35091C35B91C345
92C351925350524954454FC5350524954
100 DATA 454F46C646494CCC5343524F4C4D55343524F4C4CC40
00000009290000000000D7E00DD4602DD4E
110 DATA 04FE00C8FE10003D6778FEB9D079FE91D07C87B787876
F117C9219E53EC7906F5997675779E60120
120 DATA 1ACD1DBC545DE13E10F5D05010B00EDB0D1EBDC26BCEBF
13D20EFC9CD1DBC545DE1D511B00019D13E
130 DATA 10F5D51AE6AA4F7EE655B1122313010700EDB01AE6554
F7EE6AAB11223D1EBDC26BCEBF13D20DAC9
140 DATA 3E0FDD4600DD4E02C3E09DD7E00FE10D04FDD6603DD6
E027CFE0238047DFE900DD5605DD5E047A
150 DATA FE0338047BFE0B0079F5
160 DATA CD11BCFE0020020E04FE0120020E02FE0220020E01060
0CDE1BB32B92F1CDDEBB5C491CDE991
170 DATA CD139228F5E1CD2D922008C491CDE99110F33A2B92C
DDEB0C7E5D5E5D5C5CDF0BBC1D1E1FE0020
180 DATA 13E5D5C5CDEABBC1D1E1EBA7ED42E87AFEFF20E0D1E1C
9E5D5E09EB7AFE02200578FE0301BE5D5
190 DATA C5CDF0B0C1D1E1FE00200BE5D5C5CDEABBC1D1E110BDD
1E1C92B287CFE02200E5D5C5CDF0BBC1D1
200 DATA E1FE00C93E03FE04C9010023237CFE0120057DFE0B30E
DF5D5C5CDF0BBC1D1E1FE00C90601CD998B
210 DATA CD2CBCCD40BCC906001BF20000000000000000000000
0000000000000000000000000000000000
220 DATA 00000000000000000000000000
230 DATA 9999

```



(continued from page 145)

**SPRITEOFF,X,Y** — This is the same as for Spriteon except that as it removes a Sprite instead of printing it, no sprite number is needed. **FILL,X,Y,Z** — X = X co-ord and Y

= Y co-ord using normal co-ordinates. Z = pen in which the filling is to be done. This works on any mode.

**SCROLLU** — This scrolls the screen up one line.

**SCROLLD** — This scrolls the screen down one line. All the commands need to be prefixed by the extended colon — character above the @ on the keyboard. This is the  
(continued on next page)

**Listing 2.**

```

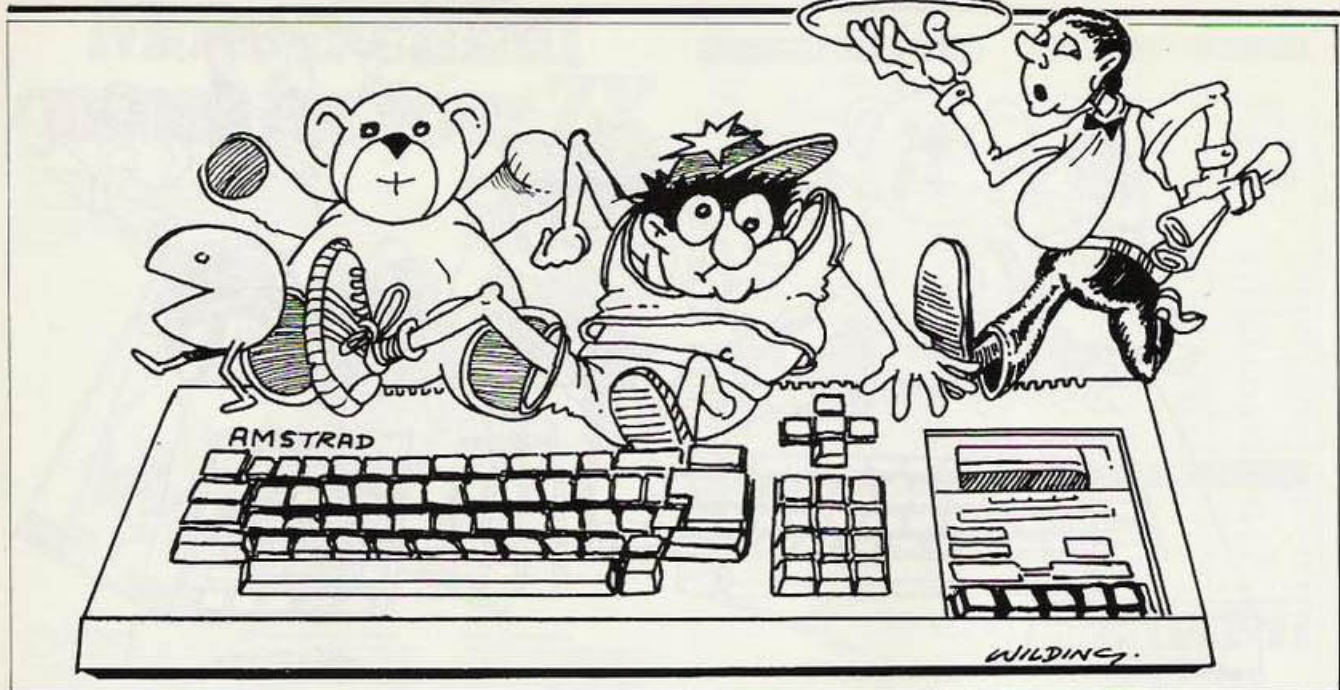
10 MODE 1:INK 1,13:INK 0,1:PAPER 0:PEN 1
20 MEMORY 36999
30 PRINT"Sprite designer by J.Charlesworth":LOCATE 1,1
0
40 INPUT "How many sprites (1 to 15) ",spr
50 IF spr<>INT(ABS(spr)) OR spr>15 OR spr<1 THEN 40
60 spr=spr-1:DIM sp%(spr,15,15)
70 KEY DEF 72,1,&F0,&F4,&FB
80 KEY DEF 75,1,&F3,&F7,&FB
90 KEY DEF 73,1,&F1,&F5,&F9
100 KEY DEF 74,1,&F2,&F6,&FA
110 KEY DEF 76,1,&E0,&E0,&E0
120 tsp=0:pn=1:xp=0:yp=0:GOSUB 550
130 GOSUB 490:GOSUB 140:GOTO 130
140 REM KEYSKAN
150 IF INKEY(62)=0 THEN 280
160 IF INKEY(27)=0 THEN 320
170 IF INKEY(60)=0 THEN 360
180 IF INKEY(46)=0 THEN 400
190 IF INKEY(35)=0 THEN 440
200 a$=INKEY$:IF a$="" THEN RETURN
210 IF a$<>CHR$(&E0) THEN 520
220 IF sp%(tsp,xp,yp)=0 THEN sp%(tsp,xp,yp)=pn:GOTO 24
0
230 sp%(tsp,xp,yp)=0
240 GOSUB 520:RETURN
250 x=xp+(a$=CHR$(&F2))- (a$=CHR$(&F3)):y=yp+(a$=CHR$(&
F1))- (a$=CHR$(&F0))
260 IF x>15 OR y>15 OR x<0 OR y<0 THEN RETURN
270 GOSUB 520:xp=x:yp=y:RETURN
280 REM CLEAR
290 a$=INKEY$
300 CLS#1:INPUT #1,"Confirm (y or n) ";a$:IF a$<>"y" T
HEN GOSUB 770:RETURN
310 FOR a=0 TO 15:FOR b=0 TO 15:sp%(tsp,a,b)=0:NEXT b:
NEXT a:GOSUB 550:GOSUB 740:RETURN
320 REM PEN
330 a$=INKEY$
340 CLS#1:INPUT #1,"Which ink ";a$:IF a>15 OR a<0 OR a<>I
NT(ABS(a)) THEN 340
350 pn=a:GOSUB 770:LOCATE 8,8:PAPER pn:PRINT " ":PAPER
0:RETURN
360 REM SAVE
370 GOSUB 780
380 SAVE "spritecode",b,37500,4352
390 RETURN
400 REM SPRITE
410 a$=INKEY$
420 CLS#1:INPUT #1,"sprite ";a:a=a-1:IF a>spr OR a<0
R a<>INT(ABS(a)) THEN 400
430 tsp=a:GOSUB 550:GOSUB 740:RETURN
440 REM INK
450 a$=INKEY$

```

```

460 CLS#1:INPUT #1,"Ink ";a:INPUT #1,"New colour ";b
470 IF a>15 OR a<0 OR a<>INT(ABS(a)) OR b>26 OR b<0 OR
b<>INT(ABS(b)) THEN 460
480 INK a,b:GOSUB 770:RETURN
490 REM PCURSER
500 PLOT xp*16+4,yp*16+2,1:DRAWR 8,12:PLOT xp*16+12,yp
*16+2:DRAWR -8,12
510 RETURN
520 REM PSPLOT
530 FOR a=4 TO 12 STEP 4:PLOT a+xp*16,yp*16+2,sp%(tsp,
xp,yp):DRAWR 0,12:NEXT a
540 PLOT 304+4*xp,272+2*yp:RETURN
550 REM SCREEN
560 MODE 0
570 FOR a=0 TO 256 STEP 16
580 PLOT a,0,1:DRAWR 0,256
590 PLOT 0,a:DRAWR 256,0
600 NEXT a
610 FOR a=0 TO 15:LOCATE 14,a+9
620 PRINT a:PAPER a:LOCATE 18,a+9:PRINT " ":PAPER 0
630 NEXT a
640 LOCATE 15,7:PRINT"inks"
650 PLOT 639,0:DRAW 639,316
660 DRAW 416,316:DRAW 416,0
670 DRAW 639,0
680 WINDOW#1,2,19,2,4
690 PLOT 0,399:DRAW 639,399:DRAW 639,320
700 DRAW 0,320:DRAW 0,399
710 GOSUB 770
720 LOCATE 1,7:PRINT"Sprite";tsp+1:PRINT"Pen "":PAP
ER pn:PRINT " ":PAPER 0
730 RETURN
740 x=xp:y=yp:FOR xp=0 TO 15:FOR yp=0 TO 15:IF sp%(tsp
,xp,yp) THEN GOSUB 520
750 NEXT yp:NEXT xp
760 xp=x:yp=y:RETURN
770 PRINT#1,"P Pen C ClearS Save","N Sprite I I
nk":RETURN
780 'compile
790 FOR a=0 TO spr:CLS:PRINT:PRINT:PRINT:PRINT"COMPI
NG IN":PRINT"PROGRESS"
800 FOR b=0 TO 15:FOR c=0 TO 15:PLOT 4*b,368+2*c,sp%(a
,b,c):PLOT 4*b+101,368+2*c,sp%(a,b,c):NEXT c:NEXT b
810 FOR c=0 TO 7:FOR b=0 TO 7:POKE 37500+272*a+b+8*c,P
EEK(49152+b+2048*c):NEXT b:NEXT c
820 FOR c=0 TO 7:FOR b=0 TO 7:POKE 37500+272*a+b+8*(c+
0),PEEK(49232+b+2048*c):NEXT b:NEXT c
830 FOR c=0 TO 7:FOR b=0 TO 7:POKE 37628+272*a+b+9*c,P
EEK(49164+b+2048*c):NEXT b:NEXT c
840 FOR c=0 TO 7:FOR b=0 TO 7:POKE 37628+272*a+b+9*(c+
0),PEEK(49244+b+2048*c):NEXT b:NEXT c
850 NEXT a
860 FOR a=(SPR+1)*272+37500 TO 41852:POKE A,0:NEXT A
870 RETURN

```



(continued from previous page)

method of telling the Amstrad that the new commands exist.

If you have typed in the run listing 1, save the resultant code with save "commands",b, 37000,500 and any time you need these commands, load them in with memory 36999:Load"":Call37000. The second listing, listing 2 is a sprite editor and this creates the code for the sprite design — without it the first two commands are almost useless.

Type in and run listing 2 and type how many sprites you want to define. You will then be presented with a grid, a list of the colours to one side and a list of the commands at the top. You can move your cursor round the grid — using joystick 1 or cursor keys — and setting or resetting the colour to Pen colour or background colour respectively using the fire or Copy key.

This means pressing the fire or copy key

will set a cell to the present pen colour and pressing it again will return the cell to background colour. The commands at the top are:

- P — Change the present pen — the colour of the pens are printed to the right of the screen.
- I — Change the colour of one of the pen's ink.
- N — Start working — or continue working — on another sprite.
- C — Clears a sprite and returns to a clear grid.
- S — Saves the sprites to tape.

Using all these commands, complex sprites can be designed and saved to tape. In future they may be loaded into memory using Load"" and used by the sprite commands as previously described. The final listing — listing 3 gives a demo of the commands. To run this, type in the program then load in the

new commands with Memory 36999:Load"":Call 37000 and then run it.

This demo illustrates how to use all the commands but the most interesting is the sprite demonstration. By using the Amstrad interrupt facility, the Amstrad keeps removing the sprite, moving it a bit then replacing it, animating it. If, however, several sprites are used and/or the amount they move is large the movement is jerky — the first sprite demonstration. However if a limited number of sprites are used and these move perhaps 1 pixel at a time, the movement is very smooth — sprite demonstration 2.

Finally, it should be noted that the Fill and the Spriteon commands assume that the background colour is pen 0 and unpredictable events may occur if this is not so but this should be no major problem as on switching on, the computer immediately makes pen 0 the paper colour.

Listing 3.

```

10 MEMORY 36999:CALL 37000
20 x=37500:PRINT"please wait,poking in progress"
30 READ a$:IF a$="9999" THEN 210
40 FOR a=1 TO LEN(a$) STEP 2:POKE x,VAL("&"+MID$(a$,a,
2)):x=x+1:NEXT
50 GOTO 30
60 DATA 0044CCCCCCCC8800449933333333336688993333333333
6699323031323031669933308172303366
70 DATA 49933281723133664433333333333800099333333366
0000449933336688000000443333880000
80 DATA 004499333366880000993366993366004433669C6C9933
889933883C3C4433669966001428009966
90 DATA CC880000000044CC00000000000050A0000000000050A000
33CC0044333333333333884433303033
100 DATA 30303388443332703380313388443337033803333880
099333333333660000443333338800
110 DATA 000CC333333333000000000099336600000000CC333
333CC000000443333CC3338800009933CC
120 DATA 3CCC336600443366143C28993388443388003C004433B
844CC000000000CC8B
130 DATA 00000050A000000000000F0F0F00000000050F0F0A0
00000055AA50A00000000005AA5000000
140 DATA 0000050A00000000000050A000000000050A0000
0000F0F0F0F0F0F0F0F0CCCCCCCC26A0
150 DATA E4CCCCC8B94CDBE4CCCCC26CCDB50CCCC8B94CC
CA00F0F0F0F0F0F0F00000300000003000
160 DATA 10002000010002000000000F00000000000050F0
A00000000000F0F0F0F00000000000FF0F
170 DATA 0F00000000005F0A00000000000F0000000000
000000F0000000000000F000000000
180 DATA 0050F0F0F0F0F0A00000E4CCCCC8B9580050CCCCC

```

```

CDC26CCA050CCCCCCC8B94CCA000E4CCCC
190 DATA DC26CCDB00050F0F0F0F0A00000102000000010200
0002010000000201000
200 DATA 9999
210 FOR a=41580 TO 41851:POKE a,0:NEXT a
220 MODE 0:a=2:b=40:c=1:p=2:q=100:r=2:x=2:y=140:z=3
230 PRINT"Jerky sprites"
240 EVERY 3,3 GOSUB 260
250 FOR t=0 TO 2000:NEXT t:GOTO 310
260 DI:SPRITEOFF,x,y:x=x+z:ISPRITEON,x,y,1:ISPRITEOFF
,p,q:p=p+r:ISPRITEON,p,q,2
270 :SPRITEOFF,a,b:a=a+c:ISPRITEON,a,b,1:IF a=142 OR a
=1 THEN c=-c
280 IF p=142 OR p=0 THEN r=-r
290 IF x=143 OR x=2 THEN z=-z
300 EI:RETURN
310 Z=REMAIN(3):DI:CLS:r=1:z=1:y=180:q=60:PRINT"Smooth
sprites":EI
320 FOR a=0 TO 600:CALL &BD19:GOSUB 390:NEXT a
330 CLS:PRINT"Fill and Scrolling":FOR t=0 TO 5000:NEXT
t:CLS
340 FOR a=0 TO 640 STEP 40:PLOT a,0:DRAW 0,400,1:NEXT
350 PLOT 0,0:DRAW 636,0,1:DRAW 636,398,1:DRAW 0,398,1:
DRAW 0,0,1
360 FOR a=0 TO 15:IFILL,a*40+20,200,a:NEXT
370 FOR a=1 TO 24:FOR b=0 TO a:SCROLL:NEXT b:FOR b=0
TO a:ISROLL:NEXT b:NEXT a
380 STOP
390 DI:SPRITEOFF,x,y:x=x+z:ISPRITEON,x,y,1:IF x=0 OR
x=142 THEN z=-z
400 :SPRITEOFF,p,q:p=p+r:ISPRITEON,p,q,2:IF p=0 OR p=1
42 THEN r=-r
410 EI:RETURN

```