

© 1983 PROCOM SOFTWARE

All rights reserved by Procom Software

no part of this publication or program may be reproduced or transmitted in any form or by any means without permission in writing from the publishers, with the exception of the Sprite Subroutine created via the Sprite Subroutine compiler on Tape 1 side B.

COPYRIGHT TO THE SUBROUTINE CREATED VIA THIS PROGRAM

The purchaser of 'Dynamic Graphics' is free to use the subroutines that they create from this program in their own programs for commercial sale and distribution under the condition that on no account may a disassembled listing of this subroutine be published in any form. Nor any other form or listing be it Decimal, Hexadecimal, Octal or other data.

PS 004 4 Cover Design by Ray Owen

DYNAMIC GRAPHICS

by David Jones and John Campbell.

INTRODUCTION

You have before you a powerful graphics aid that will save you a great deal of time and effort and give you the pleasures of creating your own games or business programs. You will find DYNAMIC GRAPHICS easy to use and unlike other utility you do not have to read through the instruction manual before commencing. In fact, we would advise that the best way to get fully acquainted with this program is to treat it like a game. Look at the 'on screen' instructions. Try all the keys and then after a while when you get the hang of it - Turn to the manual and get down to some serious programming. Good luck.

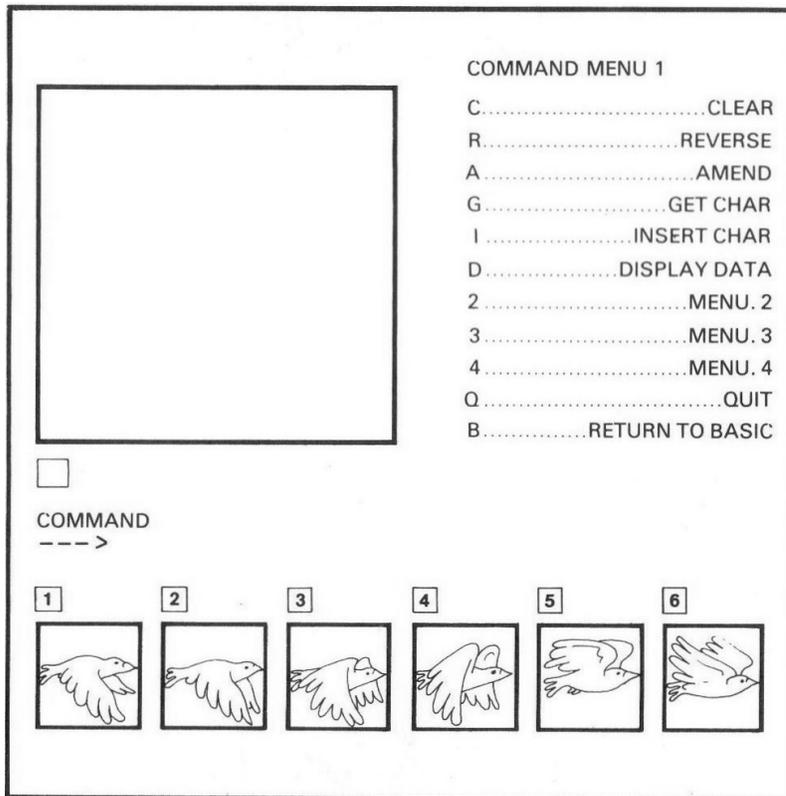


309 High Road, Loughton, Essex.
Tel: 01-5081216

TAPE 1 SIDE 1

INSERT TAPE, PRESS LOAD" " AND ENTER.

As soon as you have loaded Tape 1 side A the screen will look like this:-



For a good demonstration of DYNAMIC GRAPHICS capabilities press the following keys in sequence:

3 - A - 7 - 1 - 123456 - 0 - N -

You should now have a bird flying on the screen.

- Using your cursor keys you can make the bird move in any direction.

Tape 1 deals with the character design and is based around the concept of a 'widow' which displays the data of a character as large squares.

eg.



WOULD REPRESENT [!] By changing the data in the window, pixels that make up a character will be changed.

There are four main sections in tape 1, these are - Command menu's 1, 2, 3 & 4.

Command menu 1 does the more mundane functions and Command menu's 2, 3 & 4 do some of the more dramatic functions.

COMMAND MENU 1.

FUNCTION KEY	FUNCTION NAME.	DESCRIPTION
C	CLEAR	Clears the contents of the window.
R	REVERSE	Reverses the contents of the window such that a negative image is formed.
A	AMEND	Whilst this option is active, pressing the cursor keys will move the flashing cursor about the window. To set a block under the cursor press 'S' and to remove it press 'R'. Pressing 'enter' will end the amend.
G	GET CHARACTER	Takes the character from the graphic character set and places it in the window.
I	INSERT CHARACTER	Takes the contents of the window and inserts it into the graphic character set.
D	DISPLAY DATA	Will display within the window the numbers which must be poked in your own programs to create the character as a User Defined Graphic. There are two other methods for including the new characters and these are detailed later in this manual.
2	MENU 2	Switches to Command menu 2.
3	MENU 3	Switches to Command Menu 3
4	MENU 4	Switches to Command Menu 4
Q	QUIT	Warning this is the same as pulling the plug out. Press Q and you will have to reload the program.
B	RETURN TO BASIC	Enables basic commands to be entered - RUN will return to menu 1. Do not type NEW, COPY, LLIST, LPRINT or RANDOMIZE USR, 0 if you wish to return to the machine code.

GET AND INSERT

With both of these commands a flashing cursor will appear in the character set. The cursor may be moved using the cursor keys. When 'getting' a character press 'enter' and the character under the cursor will be transferred to the large window. When inserting a character the contents of the large window will be copied to the character space under the cursor upon pressing 'enter'.

COMMAND MENU 2

<u>FUNCTION KEY</u>	<u>FUNCTION NAME.</u>	<u>DESCRIPTION</u>
5	SHIFT LEFT }	The shift functions move the entire contents of the windows in the direction specified. No data is lost on the edge as it appears on the side of the window.
6	SHIFT DOWN }	
7	SHIFT UP }	
8	SHIFT RIGHT }	
H	HORIZONTAL FLIP	Imagine an axis half way up the window. Thus:-  and the character in the window will be flipped over Thus:- 
V	VERTICAL FLIP	Imagine the axis is now vertical. Thus:-  and the character in the window will be flipped around. Thus:- 
R	ROTATE	The contents of the window are rotated clockwise by 90 degrees.
1	MENU 1	Return to command menu 1.
3	MENU 3	Switches to Command Menu 3
4	MENU 4	Switches to Command Menu 4

COMMAND MENU 3

<u>FUNCTION KEY</u>	<u>FUNCTION NAME.</u>	<u>DESCRIPTION</u>
S	SAVE CHARACTER SET	Saves the character set with a file name of "Chars" (see tape use later).
V	VERIFY SAVE	Verifies a character set against a tape copy.
L	LOAD CHARACTER SET	Loads a previously saved character set
A	ANIMATE SPRITES	Animation.
Z	ZAP CHARACTERS	Copies the normal A.S.C.II into the character set memory area.
1	MENU 1	Switches to command menu 1.
2	MENU 2	Switches to Command Menu 2
4	MENU 4	Switches to Command Menu 4

COMMAND MENU 4

<u>FUNCTION KEY</u>	<u>FUNCTION NAME.</u>	<u>DESCRIPTION</u>
C	COPY SPRITE	Copies one of the six sprite boxes to another sprite box.
S	SWOP SPRITE	Will swop two sprites.
D	DELETE SPRITE	Will blank out a sprite box
P	PIXEL SHIFT SPRITE	Will move a sprite up, down, left or right by cursor keys. Press 'enter' to end.
H	HORIZONTAL FLIP	Will do to a sprite what horizontal flip does to a character in menu 2.

V	VERTICAL FLIP	As vertical flip - menu 2.
R	ROTATE	As rotate - menu 2.
1	MENU 1	Switches to command menu 1.
2	MENU 2	Switches to Command Menu 2
3	MENU 3	Switches to Command Menu 3

Tape Use

The tape files saved using command 3 are saved from basic. There are several reasons for this:

1. It is easier for the user to change if you so wish (adaptions to microdrives etc)
2. It is easier for the user to check the size of the code saved and its location.
3. There will be no problem running the program on future issues of the Spectrum even if the Rom changes since there are no Rom calls in the machine code.

Using the new characters

One method of using the characters that you have created is to display the data using the menu 1 option D and and poke this data into user defined graphics. Details of this can be found in chapter 14 of the Spectrum users manual. Another method is to load the code saved on menu 3 and change the Chars System Variable to point to it. To demonstrate this enter (BASIC) from menu 1 and add two lines to the basic program as follows:

```
100 POKE 23607, 122: LIST
110 POKE 23607, 60 and RUN 100.
```

What happened was this:-

Chars System Variable normally contains the values 0, 60 which together form a value of 15360. This value is found from $0 + 256 \times 60$. 15360 is 256 bytes less than the start of the character set in Rom. So by changing this value to 0, 122 (31232) the character set will be expected to start at 31488 (which it does). So to use your new character set simply LOAD" "CODE (start) and Poke 23606 and 23607 such that the value obtained is 256 less than (start).

There is a third method of using the new characters, but this is described in the next section as it relates to the 'sprites'

TAPE1 SIDE B

SPRITE USER SUBROUTINE COMPILER

Briefly you will be able to create your own 'sprite' sub-routines for use in your own basic programs. The sub-routine that is created can be loaded anywhere in normal user memory and is re-locatable. Within the sub-routine there are up to 10 Frames of 4 x 4 Character 'Sprites' these 'Sprites' can be placed at any character position on the screen and may appear with the options over 1 and over 0 as per normal printing when the sub-routine is loaded you must Randomise User at the same address that you loaded at eg. Load" "Code 40000 Rand Usr 40000 This is to tell the sub-routine where to find the data for the 'sprite'.

COMMAND MENU

FUNCTION KEY	FUNCTION NAME.	DESCRIPTION
L	LOAD CHARACTER SET	Loads a character set created by the DYNAMIC GRAPHICS character designer. (Tape 1 side A).

M	MOVE SPRITE TO SUB-ROUTINE	Copies one of the six sprites within the current character set to one of the ten frames as indicated under the message sub sprites (meaning sub-routine sprites).
S	SAVE USER SUB-ROUTINE	Will tak a copy of the user sub-routine including the sprites transferred to it from the character set.
Q	QUIT PROGRAM	Re-boots the machines, same as pulling the plug out.
B	RETURN TO BASIC	Will return to basic to allow changes to the control program, peeks and pokes etc, do not do NEW, LPRINT, LLIST, RANDOMISE USER 0, RUN, if you wish to return to the machine code without problems, typing run will continue the program.

For the demonstratin which follows you must prepare a tape with the animated bird graphics in a sprite routine, to do this enter 'M' and type 1 then 0 then 'M' and '2' then 1 etc to transfer each of six birds characters from 1 to 6 to sprites 0 to 5. Then save the sub-routine to tape. The sub-routines created by this program produce large graphic characters 4 by 4 normal characters which can be easily manipulated. Since it is easiest to learn from an example. Try the following program.

Have a tape of the bird sprites saved ready for loading in this program-

```

10 CLEAR 31079
20 LOAD" "Code 31080
30 LET START = 31080
40 RANDOMISE USR START
50 INPUT "y coordinate"; y:POKE START + 28,y
60 FOR x = 0 to 28: poke start + 27, x
70 FOR I = 0 to 5
80 POKE START + 30, I
90 RANDOMISE USER (START + 26)
100 PAUSE 1
110 NEXT I
120 PRINT AT y,x;" ";ATy + 1,x:" "; AT y + 2,x:" "; AT y + 3,x:" ";
130 NEXT x:CLS:GOTO 60

```

Try a Y-coordinate of 10 to make the bird fly in the middle of the screen.

Changing the values poked before calling the subroutine will enable you to have any of the 10 available frames to any character position on the screen at high speed.

HOW THE SUBROUTINE IS CONTROLLED

The subroutine is re-locatable machine code, which means that you can load and run it any where in memory that will not overwrite the system variables or stack pointer. See the table at the end of this section for list of advised locations. For explanation purposes I will assume that you have loaded the machine code at location 'start' which could be any address in Ram. Before the subroutine can be used you must call it at 'START' using 'RANDOMISE USR START.'This is to let the routine find the sprite data. To display a sprite use RANDOMISE USR START + 26.

The following pokes will control the routine:-

POKE START + 27, x POSITION (0 to 28)

POKE START + 28, y POSITION (0 to 20)

POKE START +30, FRAME (0 to 9)

POKE 23681, OVER (0 to 1)

The x and y positions are the coordinates of the top left character of the sprite.

The frame is the sprite number of the 10 available. 'OVER' refers to printing over as with PRINT 'OVER 1' etc.

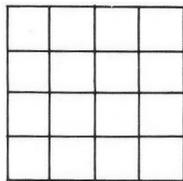
Addresses 23728 and 23729 are also used by this subroutine. It is possible to reduce the size of the subroutine by 128 for every sprite not used. E.g. if only the first 8 sprites are used, the routine can be 256 bytes shorter. This is one of the reasons that the tape routines are held in basic. To create a smaller subroutine when saving simply edit the line that contains-save "sprite" CODE a, 1420 and reduce 1420 accordingly.

Recommended highest locations for the sub routines

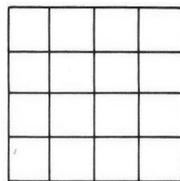
Machine	Location	
48K	63947	
16k	31080	1420 Bytes less than normal ramtop

Plus 128 bytes per sprite removed as described above

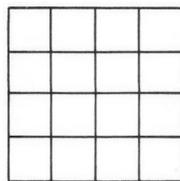
SPRITE PLANNER



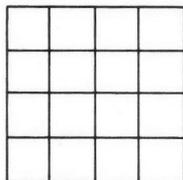
1



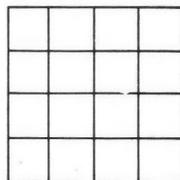
2



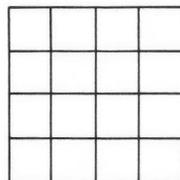
3



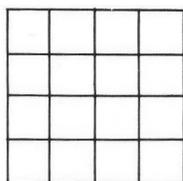
4



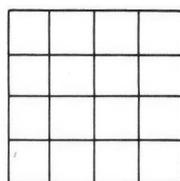
5



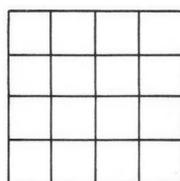
6



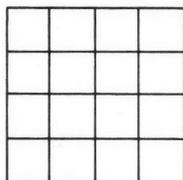
1



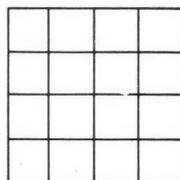
2



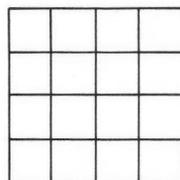
3



4



5

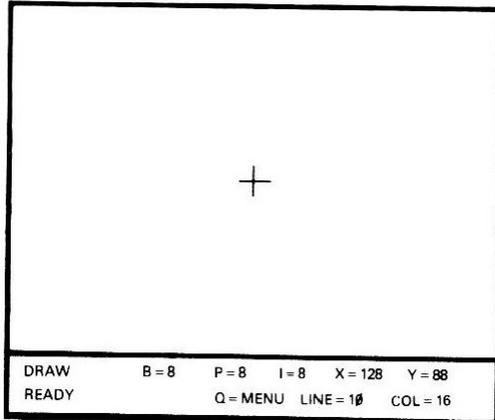


6

TAPE 2

Dynamic Graphics Drawmaster is the ultimate utility for composing HI-RES pictures on the ZX Spectrum. As well as essential hand draw cursor controls there are many advanced features such as FAST FILL and RESCALE. There are in all over 50 commands. A constantly updated STATUS DISPLAY in lines 22 and 23 presents details of cursor mode and cursor X, Y co-ordinates and ATTRIBUTES status. DRAWMASTER generated prompts are displayed here.

● INSERT TAPE, PRESS LOAD " " AND ENTER.



Once the program has loaded you will see a flashing cross-wire cursor in mid-screen, and the Status display in line 22 and 23, Lines 0 to 21 are yours for drawing.

You will, if you press BREAK or enter invalid data, find yourself in Spectrum Basic Command mode. There are two ways of returning to DRAWMASTER.

GOTO 30

This will restore the program as it was when you accidentally left it.

RUN

RUN resets all variables and clears from memory any drawing or stored window.

● MENU

Press and hold 'Q' to display the menu. Release the key to return to your drawing. Note that the commands are grouped by type of function.

KEY: [c] caps shift [s] symbol shift

● PEN COMMANDS

PEN commands define the mode of the cursor and show this in the status display.

D DRAW mode. This is the normal mode for drawing. The cursor leaves a trail of INK as it is moved.

E ERASE mode. In this mode the cursor erases or 'undraws' upon encounter any ink previously laid down.

O OVER mode. The cursor leaves a trail of INK or PAPER so as to contrast with what has already been drawn.

T TRANSPARENT mode. This, in effect lifts the pen off the paper. However, as the cursor is moved, colours will still be affected by the attribute status.

Experimentation is suggested.

● DRAW COMMANDS

The DRAW commands are all concerned with steering the pen or creating other INK effects.

1 - 8 The numeral keys 1 to 8 control the direction in which the cursor moves and allow you to draw one pixel at a time. Consult the menu for each of the 8 directions available.

[c] 1 - 8 Causes the cursor to 'hop' 8 pixels at a time. Used in TRANS mode it allows for rapid transfer of the cursor around the screen.

H You can draw dotted lines by entering a 'hop' interval other than 1. Upon loading or after RUN the value is reset to 1 to provide a continuous line.

z Z Press once to change the cursor speed. The status display carries a '*' when slow speed is selected and the computer 'beeps' when the cursor moves. The co-ordinate change, pixel by pixel, is also recorded in the status display. When fast is selected, the cursor co-ordinates are updated only when the cursor stops.

SPACE Holding this key down allows you to freeze the cursor for complicated manoeuvres.

L Draws a line in any of the 8 directions specified until it meets either a pixel or the screen edge. Has effect in DRAW and OVER modes only.

C Asks for radius - then draws a circle centred on the cursor.

X Upon loading or after RUN the pixel at screen centre is defined as 'point X'. The next four commands all relate to point 'X'. **X** redefines 'point X' at the present cursor position. This allows you to move 'point X' around the screen. This is very useful, as you can draw lines between 'point X' and the cursor - wherever they are.

A Draws an arc between the cursor and 'point X'.

J Draws a line from the cursor to 'point X'.

[s]**X** This command swaps the cursor with 'point x'. If you are unsure where 'point x' is - press twice.

F Fills with INK, hunting left, right, up and down from the cursor position, and when complete leaves the cursor at the original pixel but in TRANS mode. Since INK pixels delimit the process you cannot directly unfill but . . .

[s]**F** . . . to reverse PAPER/INK - then . . .

F . . . fill - then . . .

[s]**F** . . . again.

If the fill goes wild - hold down SPACE to abort.

● ATTRIBUTE COMMANDS

P Set the PAPER colour. 1 to 7 as normal: 8 = no change: 9 = contrast: 0 = Black.

I Sets the INK colour. 1 to 7 as normal: 8 = no change: 9 = contrast: 0 = Black.

The status display provides a constant guide to the setting of the attributes. Unless changing colours use values of 8.

B Sets the BRIGHTNESS to 0 = dull: 1 = BRIGHT: 8 = no change.

V Sets FLASH to 0 = steady: 1 = FLASH: 8 = no change.

● COMMANDS AFFECTING THE WHOLE SCREEN

U Changes all INK to black and all PAPER to dull white until key is released.

[s]**M** Inverts PAPER/INK at the cursor character only.

[s]**K** This command allows you to SAVE your picture as a SCREENS. You may change your mind by pressing ENTER.

[s]**J** LOADS a SCREENS from tape. You may change your min by pressing SPACE/ENTER.

[s]**S** This command SAVES the program. We have no objection to our users taking legitimate back-up copies. Saving the program via

[s]**S** will also save any window stored in RAM.

● WINDOW COMMANDS

A window is defined as a rectangle of whole lines and columns and is initially the whole screen. Windows of any size may be defined, stored and later restored to any portion of the screen.

W Asks you to define your window.

[s]**B** Thos draws in OVER mode a pixel border around the defined window.

[c]**F** Paints the window in current PAPER/INK/BRIGHT.

[s]**F** Paints as above but reserves PAPER/INK.

S Stores the window in RAM. The status display asks you to confirm with 'Y'. A stored window may be restored later.

[s]**6** This command allows you to redefine the window so that the top left corner is at the cursor position. This is sometimes easier than using command **W**.

[c]**R** Restores the window previously stored in RAM. You may restore it to any line and column but the dimensions must be the same as those saved.

[c]**X** Clears all INK from the window. If the whole screen is the window, then it will clear the whole screen.

W Reminds you where your window is by displaying the border and putting the line/column co-ordinates in the status display.

● RESCALE

[s]**W** This is the most powerful function which copies any rectangular area of the screen to any other parts of the screen at pixel resolution, and further allows the X and Y dimensions to be independently increased and decreased. You supply X and Y factors e.g. 2 = double the dimension, .75 = reduce the dimension accordingly.

You must supply the height and width of the source in pixels. The originating point is the cursor and the destination is 'point X'. Both origin and destination are taken to be at the bottom left corners of the areas concerned. It is best not to overlap areas.

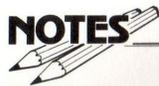
Note: Most of the DRAWMASTER menu was done by copying the Spectrum text using a factor of .75 for X and 1 for Y, thus giving legible text at 6 pixel width.

● TEXT

M Allows you to enter text directly. CAPS and SYMBOL keys work as does DELETE for backspace. TEXT entry is terminated with ENTER. Check PEN status as this determines whether text appears normal or inverted. If you overshoot line 22 the screen will scroll - BEWARE.

We hope you like this graphics package and find it value for money.

Should you encounter any problems do not hesitate to contact us and please watch out for other titles in the 'Dynamic' series. These will include DYNAMIC PROGRAMMING, DYNAMIC SOUNDS, DYNAMIC FILING, EDITOR ETC.



Procom Software will consider for publication any programs submitted which have been developed using any of the programs in the new 'Dynamic' utility series.

DYNAMIC GRAPHICS

- This package contains two cassettes and one manual. All you will ever need to create professional screens and smooth flowing sprites.
- Easy to operate and fun to use. A program that you will use time and time again!
- Programmed by David Jones and John Campbell, both well known in the field of computer programming.

©1983 PROCOM SOFTWARE

All rights reserved. No parts of this program may be duplicated, copied, transmitted or reproduced in any form without the prior written permission of Procom Software.



309 High Road, Loughton, Essex. Tel: 01-508 1216

Designed & Printed by RAINBOW STUDIO LTD, Tel: Harlow (0279) 442995.