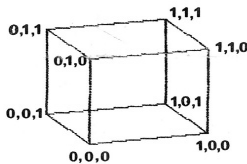
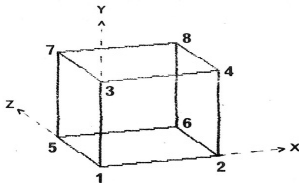


VIEWPOINT

Viewpoint is a program that allows you to model three-dimensional objects using the high-resolution graphics of either 16K or 48K versions of the Sinclair ZX Spectrum. To load the program use LOAD "viewpoint" or LOAD "".

WHILE USING VIEWPOINT, REMEMBER THAT YOU MUST NOT USE 'RUN' SINCE THIS WOULD DESTROY YOUR DATA, IF FOR ANY REASON YOU NEED TO RESTART THE PROGRAM, USE GO TO 1.



DATA ENTRY

When the program has loaded the main menu will be displayed. Data can be entered in two ways. Choosing option 1 will allow data to be entered from the keyboard while choosing option 7 will allow pre-recorded data to be entered from tape.

Let us assume that you want to enter the data for a cube from the keyboard. First draw your cube. This is probably best done on graph paper. Number the points where two or more lines intersect. Imagine that there is a set of three axes intersecting at the bottom left front corner (point 1). The x-axis is left/right, the y-axis is up/down and the z-axis is forward/back. The position of any point on your cube can be given a "map reference" relative to these three axes. The "map reference" takes the form of three numbers i.e. the x, y & z co-ordinates. For example the co-ordinates of the bottom left front corner (point 1) are 0,0,0. All the other corners are also given co-ordinates as shown. (This assumes that the sides of the cube are 1 unit long. It does not matter what units you choose or where you put the origin of the axes with respect to your object since the program will automatically alter the scale and origin so that the final image of your object fits neatly on the screen.)

Load Viewpoint and select option 1. You will be asked the number of points in your object. You are allowed a maximum of 43 points for the 16K Spectrum and about 770 for the 48K Spectrum. In the case of our cube there are 8 corners so enter 8. You will next have to enter the spatial co-ordinates of each point. Enter them as follows:

	<u>spatial co-ordinates</u>		
	x	y	z
point 1	0	0	0
point 2	1	0	0
point 3	0	1	0
point 4	1	1	0
point 5	0	0	1
point 6	1	0	1
point 7	0	1	1
point 8	1	1	1

You now need to tell the computer which points have to be connected by lines. The program provides for each point to connect to up to three other points. At first sight this might seem rather restrictive. However you will soon see that having connected point A to point B, it is not necessary when dealing with point B to connect it to point A since that would result in the

same line being drawn twice. In practice, three connections per point is more than adequate. For our cube enter the data as follows (# means press ENTER):

```
point 1 connects to      2 # 3 # 5 #
point 2 connects to      4 # 6 # #
point 3 connects to      4 # 7 # #
point 4 connects to      8 # #
point 5 connects to      6 # 7 # #
point 6 connects to      8 # #
point 7 connects to      8 # #
point 8 connects to      #
```

From this you will see that as a general rule, you should only enter data such as "point A connects to point B" if $B > A$. Hence point 8 is not connected to any other point because all of its connections have already been made. (If you are a bit confused by now, look at the drawing of the cube and trace out the connections in pencil. Don't trace over any line twice.)

The program will now return you to the menu. We would urge you to try out this exercise in data entry. If you want to check your results, use option 7 to load in the data for this cube from the data file called "cube" which is also on your cassette.

USE OF THE PROGRAM

The program is controlled from the main menu. All operations return to the menu unless the program is stopped (e.g. by BREAK). If the program does stop, remember to restart it with GO TO 1 rather than RUN.

The options available from the menu are:

- Option 1 - manual entry of data
- Option 2 - rotation in three dimensions about any point
- Option 3 - rotation about a line between two points
- Option 4 - rotation such that one point is brought to the front of the image and a second point is moved to the rear
- Option 5 - view of the position from which rotation will occur
- Option 6 - save object as data file to tape
- Option 7 - load new data file from tape

In options 2-4, the starting position is first drawn and you are then asked which point or pair of points you want to use. These are then identified and confirmed before you are prompted for any angular information required. The co-ordinates of the object are then recalculated and adjusted if necessary to keep the object on the screen. The new view is then drawn and you are given the opportunity to copy the drawing to the ZX printer. You can make this view the new starting position for further rotations if you want, otherwise the old starting position is used. Finally you are asked if you want to choose another option and if so are returned to the menu.

Stereo views can be printed out as follows:

Enter the data for your object by options 1 or 7. Use options 2 or 3 to obtain the view that you want. The use option 2 to rotate the object 90 degrees about the x-axis. Make this the new starting position. Then rotate 3 degrees about the x-axis and obtain hard copy. Do not make this the new starting position. Rotate 357 degrees about the x-axis and obtain hard copy again. You now have two views of the object on your printout. If you turn the paper so that its width is vertical you have a stereo pair which you can view with a stereo viewer, (though with

practice you may find that you can see the stereo image by squinting so that the two images coincide and then concentrating on the fused image so that it comes into focus).

If you would like some images to practise with, you will find three data files on your tape called "letter", "cube" and "alanine". The first is a solid letter "A", the second is the cube which you may have already entered and the last is a stick model of the amino acid L-alanine. ($\text{H}_3\text{N}^+ - \text{CH}(\text{CH}_3) - \text{COO}^-$).

ABOUT THE PROGRAM

The data describing the object is kept in two arrays c & g. Array c contains the data describing the starting position of the object and array g contains the data describing the object after rotation. Both arrays have 6 columns (x,y,z & 3 connections) and the number of rows is one more than the number of points. The element c(1,1) is reserved for the number of points and c(2-6,1) are empty. Array c is saved to tape when the data file is saved.

Lines 100-105	set up the arrays
Lines 106-120	input the spatial co-ordinates
Lines 121-137	input the connections
Lines 140-198	handle the menu
Line 200	saves the data file
Line 250	loads a new data file
Lines 300-320	gets the data for the rotation about a point
Lines 400-420	gets the data for rotation about a line
Lines 500-520	gets the data for option 4
Lines 600-650	takes the data from array c, adjusts it and transfers it to array g
Lines 700-811	draws the object whose data is in array g (if you want to do ball-and-stick molecular modelling, you might put some CIRCLE commands in line 711)
Lines 812-813	give hard copy
Lines 820-860	change the starting position if necessary
Lines 900-930	return to the menu or end the program
Lines 1000-1601	scale and draw the starting data
Lines 6000-6005	calculate the new co-ordinates after rotation at a point
Lines 8000-8005	calculate the new co-ordinates after rotation about a line
Lines 9000-9003	deal with the rotation required for option 4
Lines 9110-9200	check to see if the object has moved off the screen and bring it back if necessary

Please note that you are welcome to use these routines in your own programs, but if you publish any program that uses them please acknowledge ACS Software.

(N.B. Option 4 uses ATN in its calculations. For some angles, ATN can be rather imprecise and this may lead to cumulative errors which give rather unexpected results. If you don't get what you expected then rotate the image and try again from the new starting position.)

ACS Software has available cassettes of data for molecular modelling (e.g. the amino acids). If you would like copies of these for 5 pounds each, please write to us. If you produce very good 3-D images, let us know because we will pay 5 pounds for each very good image.

Copyright 1982 ACS Software



ACS SOFTWARE

7, Lidgett Crescent, Roundhay, Leeds LS8 1HN