

# ATARI®

## Supergraphics Three-Dimensional Display System

P. Lutus, Kerby, Oregon 1980

### Contents

Page 1: Introduction

Pages 2 - 12: Part I: Basic Supergraphics

(Source of Images - Cartesian Coordinates - Allowable Image Number Range - Command Numbers - Color Entries - Coordinate Numbers - BASIC DATA table format - Enabling Supergraphics - Image Table Identification - Example Images - Display Commands - Scale - Rotation - Translation - Position - Default Graphics Color - Erase - Text Color - Quit - General Discussion)

Pages 13 - 16: Part II: Advanced Supergraphics

(Command Loops - Multiple Commands - Multiple Images - Images with moving parts - Value Query - Text Clear - Screen Position Control)

Pages 17 - 19: Part III: Use of the GRAFBASIC utility

(Menu - Draw - Entry Format - Show - Show Commands - File)

Pages 20 - 22: Part IV: System Technical Description

(Display method - Screen Sequence - Procedure - Matrices - Plot format)

Page 23: Appendix A: Short-form list of Supergraphics Commands

Page 25: Appendix B: System Memory Map

Page 26: Appendix C: Getting SUPERG up and running

Page 28: Appendix D: DEMO Program BASIC Listing

Page 34: Appendix E: GRFBAS Program BASIC Listing

Page 39: Appendix F: Shape Editing

Page 40: Appendix G: Creating Images

Page 43: Appendix H: System Addresses

## Supergraphics Three-Dimensional Display System

P. Lutus Kerby, Oregon 1980

### Introduction

Supergraphics is a software system that projects three-dimensional color images on the Atari screen. The system draws images so fast that animated displays are possible. The program is completely compatible with the Atari BASIC language, the Disk Operating System, and other assembly-language programs. Required are an Atari 800 with 40K of Random Access Memory and a minimum of 1 disk drive.

The quality of the graphic images is much better than that normally available from Atari. All high-resolution graphics colors are available, both for image drawing and text displays.

Access from BASIC is very simple. After initialization, simple printed commands such as are used by the Disk Operating System command the drawing, rotation, scale, etc. of the images.

Demonstration and image development programs are provided to aid the software developer. Because Supergraphics is a software development tool, these instructions assume minimum familiarity with the BASIC programming language and the Atari.

In these instructions, segments of particular interest are preceded by ">>>" in the left margin.

## Part I: Basic Supergraphics

Your first step in getting to know Supergraphics should be the demonstration program "SGDEMO" provided on the system diskette (a listing of "SGDEMO" is included in these instructions). This demonstration shows the main features of the program and some typical displays. If you are familiar with the Atari, you will take note of the fact that text and high-resolution images are combined, one of the features of Supergraphics.

### >>> Source of images

NOTE: The system diskette program "GRFBAS", described in Part III, will create images for you using a display and text editor format. The following section describes the image format for the benefit of programmers. It is included for reference, and may contain more information than needed by normal users.

One of the principal purposes of Supergraphics is to project three-dimensional images. The source for these images is a table of coordinate numbers that define lines to be drawn.

### >>> Cartesian Coordinate System

For each line ending in the image, three coordinate numbers are required, one for each dimension of space. The three numbers are assigned the letters X, Y, and Z.

X (by convention) defines a point along the horizontal axis. Y defines the vertical axis and Z defines a point in depth. This system of numbers is called the "Cartesian Coordinate System", developed by Renee Descartes.

Imagine that you are in a large city. Your friend is trying to explain the route to his house. He lives three blocks east and seven blocks north of

the bus station and, finally, on the eleventh floor of his building. If he and you knew the Cartesian Coordinate System, he might say "Walk along X until it is equal to 3. Now walk along Z until it is equal to 7. Now move along Y until it is equal to 11. Knock on my door".

Supergraphics accepts numbers in just this way. A table of numbers is created, describing a house or an airplane or anything else. Then Supergraphics is told where the numbers are located in Atari's memory.

#### >>> Allowable Image number range

The value range of the numbers that will be accepted by Supergraphics is (integer) -128 to 127 in the three dimensions X, Y, and Z. This means that there are about 16 million definable points in the mathematical space of Supergraphics.

Image coordinate numbers are in sets of four. The first number is a command that carries information about color and whether to draw a line or simply move. The remaining three numbers are the X, Y, and Z Cartesian coordinates. Here is a description of the number set:

#### >>> Command Number Format

Number 1: Command. If this number is less than 128, Supergraphics will move, not draw a line, to the coordinates that follow. If the number is 128 or greater, a line will be drawn. If the number equals 255, this signals Supergraphics that the end of the image table has been reached.

#### >>> Color Entries

A color number may be added to the command number if desired. Here are the colors and their numbers:

Number	Color
1	Orange
2	Blue
3	White

These colors depend upon the locations of the dots (in odd, even, or both columns), and are not specifically discussed in the Atari Manuals.

To create a command number, select a color if desired (this should only be done if the color must be changed), then add 128 to the color number if a line is to be drawn. If there is no color number added to the command value 0/128, the previous color will be used. Examples:

Number	Means
131	Draw a line using color 3 (131=128+3)
1	Change to color 1 and move to the given coordinates
128	Draw a line using the previously entered color

>>> Coordinate Number Format

Numbers 2 - 4: These entry numbers are the X, Y, and Z coordinates for a point. They are interpreted by Supergraphics as lying in the range -128 to 127, although BASIC will not place negative numbers directly into memory. To prepare the numbers for placement into memory, simply add 256 to those numbers that are less than 0. A BASIC example:

```
100 N = N + 256 * (N < 0)
110 POKE A , N
```

>>> BASIC DATA Table format

A simple way to make an image available (not as simple as using the program "GRFBAS") is to enter the numbers into BASIC DATA statements, then use a loop to place them into memory. Here is an example

program to do this:

```

10 FOR ADDRESS = START TO HIGHNUMBER
20 READ NUMBER
30 IF NUMBER = 255 THEN 70
40 NUMBER = NUMBER + 256 * (NUMBER < 0)
50 POKE ADDRESS,NUMBER
60 NEXT ADDRESS
70 POKE ADDRESS,NUMBER
80 REM THE REST OF THE PROGRAM
,
,
,
999 REM IMAGE DATA STATEMENTS
1000 DATA 3 , -127, -127, 0
1010 DATA 128, 127, -127, 0
1020 DATA 128, 127, 127, 0
1030 DATA 128, -127, 127, 0
1040 DATA 128, -127, -127, 0
1050 DATA 255

```

The DATA table provided will draw a square in the X, Y plane.

### >>> Enabling Supergraphics

To enable Supergraphics, first load it into memory:

- A. Using the DOS command 'L', then SUPERG.  
Enable it with:

```

90 X = USR(22016)
100 OPEN #1,12,0,"G:"

```

- B. Or, use the program listed in Appendix C.

After initializing Supergraphics, it must be allowed to control all of the display functions on the Atari. All text printing and graphics display must be performed by Supergraphics. This means that all printing must be done to the channel opened to 'G:' (#1 in the above example). Any regular print commands will cause an abrupt exit from Supergraphics. All input must also be done through

the same channel. Example:

```
200 INPUT #1;X$
```

>>> Image Table Identification

At this point, Supergraphics will print any text provided, but it won't display any graphic images. To make Supergraphics display an image, we must tell it where the image is located in memory. Let's assume that an image has been loaded into memory starting at decimal address 19712. The command to enable graphics is:

```
120 PRINT #1; "%SH 19712"
```

Remember: Each command to Supergraphics must be printed to the channel opened to 'G:', and must begin with a percent sign (%). The command "%SH" means "shape" and it is followed by the beginning address of the image table. After this command has been printed by BASIC (like all Supergraphics commands, it won't appear on the screen), the image will appear.

>>> Example Images

To speed up the learning process, you may want to load the demonstration images from the system diskette rather than create your own. To do so, use the DOS 'L' command, and IMAGES.

Now the following images will be accessed by the "%SH" addresses listed:

Command	Image
"%SH 19712"	Fighter
"%SH 19776"	Atari Logo Sign
"%SH 19904"	X, Y, Z Coordinate arrows
"%SH 19968"	Hang glider
"%SH 20057"	Butterfly (wings down)
"%SH 20129"	Butterfly (wings up)

## &gt;&gt;&gt; Display Commands

NOTE: Be sure to see Appendix A: Short-form listing of Supergraphics Commands.

Proper Supergraphic command format within a BASIC program requires that a carriage return precede the command line. Here are examples:

This example will not work:

```
10 PRINT #1; "HELLO"; : REM NO CARRIAGE RETURN
20 PRINT #1; "%RX 20"
```

This example will work:

```
10 PRINT #1
20 PRINT #1; "%RX 20"
```

It is not usually necessary to PRINT a carriage return, as in the example above. If programming problems arise, or a Supergraphics command is printed instead of being executed, this may be the problem.

NOTE: Use of the "%" sign in BASIC printing is permitted, in positions other than at the line beginning. Example:

```
10 PRINT #1; "100 % SCORE!"
```

The numeric entries to these commands may be in the range (+,-) 65536. The entries should be integers. Floating point numbers will cause difficulties. Example:

```
100 X = INT (X)
110 PRINT #1; "%RX ";X
```

## &gt;&gt;&gt; Display Scale

Scale numbers in the range -31 to +31 may be entered to control the overall image size. Negative scale numbers will create an inverted image. Example:



"%SC 20" (Scale)

Scale numbers may also be used to specifically control the scale of the X and Y dimensions. These commands stretch the image in the specified dimension. Examples:

"%SX 20" (Scale X)  
"%SY -5"

>>> X Y, Z Rotation

Here are the Supergraphics commands to rotate the image about the axes X, Y, and Z:

"%RX 10" (Rotate X)  
"%RY 20"  
"%RZ -15"

The number values are in angular units. Each angular unit is equal to 1.4 degrees (360/256).

>>> Translation

"Translation" means movement along the X, Y, and Z axes. If the image is large, the range of translation will be correspondingly smaller, because the limits of the mathematical space are (+,-) 127 in each dimension. If the image values plus the translation value exceed this range, the image will "wrap around" within the defined space. Example:

"%TX 20" (Translate X)  
"%TY -50"  
"%TZ 12"

>>> Position

This command displaces the image on the Atari graphics screen. Normally the center of the image is at X = 128, Y = 96. This value is modified by the entered number. Example:

"%PX -30" moves the X screen image position from 128 to 98 "%PY 20" moves the Y screen image position from 96 to 116

NOTE: Unlike Atari graphics, Supergraphics obeys the mathematical convention that the lower left of the screen is  $X = 0$ ,  $Y = 0$ .

A short tutorial on using the above three commands (rotation, translation and position) is at the end of this section (Part I).

### >>> Default Graphics Color

If an image file has no color specifications, it will be drawn in the default color (3). This default color may itself be changed using this command:

"%CG 1" (Color Graphic = 1)

This color will be used until (if) a color specification appears in the image file. This makes it possible to change the color of an image in sequential drawings.

### >>> Erase screen

This command erases the text and graphics display. There is no numeric entry. Example:

"%NW" (New)

### >>> Enable, disable drawing

These commands enable and disable graphic image drawing, while permitting normal text displays on the graphic screen. Examples:

"%ND" (No Draw)

"%YD" (Yes Draw)

### >>> Text Color

This command selects the color in which text will be printed. Example:

```
"%CT 2" (Color Text = 2)
```

NOTE: Because of the way by which Atari generates colors, there will be some distortion of the printed text when colors other than white (3) are used. Special effects may be obtained through experimentation. Color 3 is the normal (default) text color.

NOTE: To print inverted text (black on white), simply use the Atari reverse key (the one with the Atari logo on it).

```
>>> Quit Supergraphics
```

To disable Supergraphics and return to normal Atari operation, use this command:

```
"%QT" (Quit)
```

### General discussion

This section is intended to clarify the concepts of rotation, translation, and position as used by Supergraphics.

In order to understand the interaction and behavior of the Position, Translate, and Rotate commands, think of the drawing area as a box. This box is divided into 255 segments for each of the three dimensions (X, Y, and Z). Anything we draw will be contained in this box. The size of the box (relative to the TV screen) is set by the scale command (%SC). Each direction can be scaled separately by the %SX, %SY or %SZ commands. Regardless of how the box is scaled, it is always divided into 255 segments along each direction, even though these will generally not all be resolved in the image. If you draw a line from X=-127 to X=127, it will go from one side of the box to the other. If the box is small, the line will be small, and

vice versa. When you translate the image, you move it within the box. If you translate it so that any lines go beyond  $\pm 127$  in any direction, those lines will wrap around.

When you rotate the image, it rotates around the center of the box. If you translate a centered image from the center of the box, the image will no longer rotate around its own center, but will move around the box center like an object off center on a wheel. If you want to move an image, but have it rotate around its own center, you use the Position commands (%PX, %PY). These do not move the image within the box; they move the box around on the TV screen but keep the image in the same location in the box. Any lines that go off the viewing area will wrap around.

Let's work a little with GRFBAS to see the above concepts in action. Load GRFBAS; POKE 16384,255 (this starts a new file), and then RUN <RET>. Press "D" for DRAW on the menu. We will draw a single line and watch what happens with the various SUPERG commands. Press "M" <RET>; you will be prompted for the X-coordinate. Type "-32" <RET>. This moves the invisible cursor to -32 on the X-coordinate. You are then prompted for the Y and Z coordinate locations. Enter 0 (zero) for both. Now type "D", and set X to 32; you can just press <RET> for Y and Z to keep them as they were (0 in this case). A short line will be drawn on the TV. Now press <RET>, and then "S" for SHOW from the menu. This mode allows you to see what happens with the image with the SG commands. Press "C"; you will see "ENTER SUPERGRAPHICS COMMAND : ". You can now enter any SG command and see what happens. Try "SC8" <RET>; the TV goes black, but nothing happens to the line. This is because the default scale size is 8. Try "SC16" <RET>; the line doubles in length. Press "R"; you will see "BASIC COMMAND : R (X,Y,Z)?" Press Z and hold it down; the line will rotate around its center. Bring Z back to 0 and press "T". Then press X; the line is being translated (within the box). Press X a couple times more. Now type "R", and then hold Z down. Note that the line now

goes around the box center. Put the line back to the center by typing "C" and then "TX" <RET>. Now type "P", and then press X two or three times. The line moves as before, but this time the box is moving with it. Try "R", then hold down the Z key. The line still rotates around its own center, since it is still at the center of the (invisible) box. Now try experimenting on your own.

## Part II: Advanced Supergraphics

This section describes advanced instructions, combinations of instructions and methods.

## &gt;&gt;&gt; Command Loops

In a BASIC program one would rotate an image this way:

```
10 FOR A = 0 TO 256 STEP 4
20 PRINT #1; "%RX";A
30 NEXT X
```

This method has the significant drawback that Atari BASIC is more than 100 times slower than Supergraphics. Therefore Supergraphics has its own control loop capability. Here is an example that duplicates the function of the above BASIC segment:

```
10 PRINT #1; "%RX 0 T 256 S 4"
```

NOTE: When using the loop capability, remember that, unlike BASIC, the controlled value must equal the ending value at termination. This example:

```
10 PRINT #1; "%RX 0 T 255 S 4"
```

will loop endlessly, since the controlled value will not equal 255 during operation.

Negative numbers may be used in loop commands:

```
10 PRINT #1; "%RX 0 T -256 S -4"
```

NOTE: Pressing any keyboard key will exit a loop command. This is a useful feature in game programs, in which constant animated displays may be controlled by the keyboard. This means that a loop can be created to rotate an image while a keyboard entry is awaited. Example:

```
10 PRINT #1; "%RX 0 T 255 S 4" : REM ENDLESS LOOP
20 INPUT #5; N$
```

## &gt;&gt;&gt; Multiple command lines

Each time a command line is printed, the image is updated. If it is desired to change more than one display constant before displaying the image, simply enter all values to be changed on a single line. Example:

```
10 PRINT #1; "%RX ";X;" ,RY ";Y;" ,RZ";Z
```

Each command is separated in the line by commas. Multiple loops may be entered:

```
10 PRINT #1; "%RX 0 T 256 S 4,RY 0 T 256 S 4,RZ 0 T 256 S 4"
```

These loops will be executed sequentially.

## &gt;&gt;&gt; Multiple images

NOTE: To fully understand the following method, particularly the reference to two display screens, read Part IV: System Technical Description.

Here is the method to draw multiple images or copies of an image. Normally Supergraphics "undraws" the previous image before drawing the new image or image position. To prevent this erasure, simply make a "%SH" shape address entry. The "%SH" command abandons the "undraw" data normally used to erase the previous image. Example:

```
10 FOR X = -64 TO 64 STEP 32
20 PRINT #1; "%SH 4096,PX";X
30 PRINT #1; "%" : REM PLACE IMAGE ON BOTH SCREENS
40 NEXT X
```

Notice the null command "%" in line 30. This command simply redraws the image without changing anything, assuring that both screens carry a copy of the image. The image would otherwise flash as Supergraphics switches between screens.

This example makes five copies of the image on the screen. The described method may be used to repeat an image, or develop many images without erasing the old ones.

NOTE: If difficulties arise while using a combination of rotation and image switching, simply be sure that both screens carry the same image (see line 30 in the above example).

To return to a previous image position, use the above method, specifying the old image position (or rotation, scale etc.), and make two copies of the drawing before moving. Example:

```
10 PRINT #1; "%SH 4096,PX"; OLDX ;",PY"; OLDY
20 PRINT #1; "%"
30 REM OK TO MOVE
```

### >>> Images with moving parts

There is a variation on the method just described. In this variation, the shape table address is changed, but the "undraw" information is retained. This means that you can flap the wings on a butterfly, for example, or other things that require two or more versions of an image.

In the "SGDEMO" program this technique is used to display the butterfly image. Two butterfly images, one with wings up, the other with wings down, are used. So it is necessary to change the image address, but still "undraw" the previous images. Here is an example:

```
10 FOR Q = 0 TO 30
20 PRINT #1; "%SH + ";WINGSUP
30 PRINT #1; "%SH + ";WINGSDOWN
40 NEXTQ
```

The address numbers "WINGSUP" and "WINGSDOWN" point to the beginning addresses of the two versions of the butterfly image.



Notice the "+" sign between "%SH" and the number. This "+" sign, when used with the "%SH" command, preserves the "undraw" information. If the "+" sign is not used, the undraw information is abandoned, as in the previous example.

>>> Getting a value from Supergraphics

It may be necessary to obtain a program value from Supergraphics, rather than transmitting one. To do this, use the following procedure:

```
10 PRINT #1; "%RX ?"
20 RX = PEEK (218)
```

The requested command value will be placed in memory location 218, where it can be recovered with the BASIC "PEEK" feature. In the case of the query:

```
10 PRINT #1; "%SH ?"
```

The present shape address (two bytes) will be provided in locations 218 and 219. To obtain the address value, use this method:

```
20 ADDRESS = PEEK (218) + PEEK (219) * 256
```

>>> Text Clear

The previously described "%NW" command should be used to clear the Supergraphics display and resets the cursor position to the upper left.

>>> Screen position control

The cursor position on the screen can be controlled by the use of the following two commands:

```
"%VT 10" (Line 10)
"%HT 15" (Column 15)
```

### Part III: Use of the "GRFBAS" utility program

This program may be used to create, view and save Supergraphic images for use elsewhere. To operate the program, type 'RUN "D:GRFBAS"' (Atari BASIC is required). A menu will appear.

The menu options permit you to "(S)how", "(D)raw", and "(F)ile" an image. First, let's review "(D)raw". Press "D".

A display of the image (if any) and the last 6 lines of the image table will be displayed, followed by a prompt:

```
(D)RAW, (M)OVE,(1-7) COLOR, (-) DELETE
EXAMPLES: 'D3','M'. ENTER:
```

The "-" command will delete the last entry. The "D"raw command will command that a line be drawn. The "M" command moves without drawing. These commands may be followed by a color number (1-3). Remember that a color number should only be entered if a color change is required. Specifying a color for each line slows the display.

After entering "D" or "M" and the optional color number, a new prompt will appear:

```
'D' X      (Enter X value here)
```

after entering the X value, and RET, you will see:

```
'D' Y      (Enter Y value here); and then:
'D' Z      (Enter Z value here)
```

The numbers should be integers in the range -127, +127. After each entry, press RETURN. The entered numbers will appear in the data table and (if command "D" has been used) will also appear in the graphic display. To quit the "(D)raw" feature, simply press RETURN with no entry from the first of the "(D)raw" prompts.

NOTE: For maximum display speed, make images as much as possible out of line continuations rather than moves. This is because each line ending must be calculated, and calculation time is a significant percentage of display time. Another speed hint is to only specify color in line entries when color is to be changed.

To "(S)how" the image, press "S" from the menu. There are a number of options available to rotate, scale, and otherwise modify the image.

To "R"otate, "T"ranslate, "P"osition, or "S"cale the image, simply press the corresponding letter key. Now press "X", "Y", or "Z" to carry out the command on these dimensions. To change commands, press one of the letters "R,T,P,S" again.

To change the direction of movement while pressing the "X,Y,Z" keys, press "-". Subsequent "X,Y,Z" keys will move in the reverse direction. To change the step size, press "I" and enter the new increment value.

To zero all display constants, press "\*". Rotation, scale, translation and other values will be set to 0 or their default values.

To make a direct Supergraphics "C"ommand, press "C". Example:

```
ENTER SUPERGRAPHICS COMMAND: RX 0 T 256 S 4
```

NOTE: Don't enter the "%" command token before your command in this feature. The token is added automatically.

NOTE: All rotations and color inputs made to "(S)how" are preserved for use elsewhere in GRFBAS. Scale returns to the default value of 8.

To quit the "(S)how" feature, press RETURN with no command entry.

To "(F)ile" your image, press "F" from the menu. A file name will be requested, followed by a specification of "(L)oad or (S)ave". The most recently accessed disk drive will yield or receive the image file.

## Part IV: System Technical Description

Supergraphics uses two Atari high resolution display screens for image display. This produces a flicker-free graphic display, because image erasure and redrawing is conducted on the invisible screen. Therefore it is necessary to make two copies of an image that is to remain on display. Example:

```
10 PRINT #1; "%RX -32" : REM ROTATE X AXIS
20 PRINT #1; "%"      : REM COPY TO OTHER SCREEN
```

This procedure need only be used if another image is to be drawn and it is desired to keep the first on display.

Text display is made to both screens simultaneously, so that graphic screen switching won't disrupt the text display.

Supergraphics uses the following general procedure for each image update:

1. Prepare access to the screen not presently on display.
2. Undraw the old version of the image, if present, by zeroing all bytes that were used for the image.
3. Compute new three-dimensional matrices (see below).
4. Draw updated image, using matrix transformation.
5. Switch display to the new image screen.

NOTE: An additional version of Supergraphics is available on the system diskette. Named SUPERBW, this version draws only in black and white. It may be used in exactly the same way as described for normal Supergraphics. Because it draws in black and white, the lines drawn are somewhat more precise. Also, this version draws more quickly. Therefore if a black and white monitor is in use, or maximum speed is desired, use SUPERBW.

Supergraphics uses a fast assembly-language matrix

multiplication algorithm to provide the mathematical transformations needed for three-dimensional display. The following 3 by 3 matrices are multiplied together to provide the display matrix:

(X),(Y),(Z) = rotation angles in units (1 unit = 1.4 degrees)

X,Y,Z = Cartesian Scalar values

Matrix 1:

Scale X	0	0
0	Scale Y	0
0	0	Scale Z

Matrix 2:

1	0	0
0	cos (X)	-sin (X)
0	sin (X)	cos (X)

Matrix 3:

cos (Y)	0	-sin (Y)
0	1	0
sin (Y)	0	cos (Y)

Matrix 4:

cos (Z)	-sin (Z)	0
sin (Z)	cos (Z)	0
0	0	1

The preceding matrices are computed once per image. The result matrix of the previous multiplications is then used to transform each line ending in the image, using this final matrix:

X+X Translation	Y+Y translation	Z+Z Translation
0	0	0
0	0	0

The plotting coordinates X' and Y' are then offset

by the coordinates of the screen center:

Plot X = X' + 128 + X Position

Plot Y = Y' - 96 + Y Position

Appendix A: Short-form list of  
Supergraphics commands

The commands listed below become operative after Supergraphics has been loaded and initialized. The initialization is by way of the command "Z =USR(22016)", as in this BASIC program example:

```
10 Z = USR(22016)
```

All entries are decimal integers in the range +-65536. Some of the commands use only the subset +-256 of the entered numbers.

Most of these commands may be used in the loop entry:

```
"%(command) (number) T (number) S (number)"
```

This loop entry is equivalent to the basic statement:

```
FOR (number) TO (number) STEP (number)
```



Command	Means	Default Value	Comment
"RX"	Rotate	X 0	in units of angle
"RY"	Rotate	Y 0	1 unit = 1.4 degrees
"RZ"	Rotate	Z 0	(64 = 90 degrees)
"TX"	Translate	X 0	limited to bounds
"TY"	Translate	Y 0	of +-127 space
"TZ"	Translate	Z 0	including image values
"PX"	Position	X 0	added to 128
"PY"	Position	Y 0	added to 96
"SX"	Scale	X 8	range +-31
"SY"	Scale	Y 8	
"SZ"	Scale	Z 8	
"SC"	Scale X,Y,Z	8	
"CG"	Color Graphics	3	range 1-7
"CT"	Color Text	3	
"VT"	Vertical Tab	(to line #)	
"HT"	Horizontal tab	(to column #)	
"YD"	Yes Draw		
"ND"	No Draw		
"NW"	New (erase)		
"QT"	Quit		
"SH"	Shape (address)	Abandons "undraw" data unless followed by "+"	

Appendix B: Supergraphics Memory Map

\$0000 - \$00FF	6502 Page Zero
\$0100 - \$01FF	6502 Stack
\$0200 - \$05FF	Atari OS use
\$0600 - \$06FF	Free user RAM
\$xxxx - \$4FFF	BASIC Program area (including shape tables) xxxx = \$0700 for Cassette xxxx = \$2A80 for DOS I xxxx = \$1CFC for DOS II
\$5000 - \$55FF	SUPERG buffers and variables
\$5600 - \$5FFF	SUPERG or SUPERBW
\$6000 - \$7FFF	Graphics screen area #1
\$8000 - \$9FFF	Graphics screen area #2

## Appendix C: Getting SUPERG up and running

The disk you received can be booted on the Atari. Check the disk directory; if you have a file "DUP.SYS" you have the newer DOS II. This gives you more space to write a BASIC program. However, you should not use any disks formatted with DOS I when the DOS II system is booted, and vice versa, unless you are thoroughly familiar with both. "SUPERG" can be loaded from the disk in either of two ways:

- A. From Disk Utility, type "L", then "SUPERG"  
Or:
- B. From a BASIC program load. Example:

```

10 OPEN #1,4,0,"D:SUPERG"
20 TRAP 70
30 FOR A = 22010 TO 32000
40 GET #1,X
50 POKE (A) ,X
60 NEXT A
70 CLOSE #1
80 GRAPHICS 24
90 X = USR (22016)
100 OPEN #1,12,0,"G:"
110 PRINT #1;"HELLO SUPERGRAPHICS!"
120 GOTO 120 : REM ENDLESS LOOP
130 REM Your program should start at
140 REM line 120, omitting the loop.

```

Notice that the starting address for the loop is the file beginning - 6. This is because Atari has a special header on binary load files that must be discarded. All printing in the program MUST be to the I/O block (in this example, #1) opened for Supergraphics (G:), otherwise there will be a nasty exit from the Supergraphics features.

Loading from Tape.

Note that the tape version must also have 40K of RAM, and will therefore not run in the Atari 400 unless you have RAM expansion which gives you at least 40K.

Put the cassette in for the proper side, rewind, and advance to 010 (If the loading sound starts too soon, advance to a lower number). Turn the Atari OFF, then ON. Press PLAY.

Side A SGDEMO

CLOAD (RET)	Will buzz once
RET	At READY prompt
RUN (RET)	Will buzz once
RET	Loads SUPERG, buzzes once
RET	Loads IMAGES, and runs.

Side B GRFBAS

CLOAD (RET)	Will buzz once
RET	At READY prompt
RUN (RET)	Will buzz once
RET	Loads SUPERG, and runs.

A file, "TETRA" is next on the tape on side B, and can be loaded with the "F" command in GRFBAS.

To load SUPERG from your own program (as in the program example under loading from a disk), load side A of the tape until you get the READY prompt. Note the counter location and write it down for future reference. Now load your BASIC program, which must open a channel to the cassette rather than to the disk. Put the Superbasic tape in and set it at the noted counter location. Now run your program to load SUPERG from the tape.

## Appendix D: DEMO Program Listing (BASIC)

NOTE: You may not have lines 1630-1910 in the supplied program.

That part of the demo needs a shape at 20200.

```

2 GRAPHICS 24
5 SETCOLOR 2,0,0:SETCOLOR 1,0,15
10 GOTO 200
20 REM
30 REM DRAWS FIGHTERS
40 REM
50 PRINT #5;"%ND,SH19712,SCO,CG";C;" ,RY";-20*S;" ,YD"
60 FOR X=0 TO 20*T STEP 2*T
70 PRINT #5;"%SC";X;" ,RZ";X*12;" ,PX";X*5*S;" ,PY";-X
80 NEXT X
90 PRINT #5;"%":PRINT #5;"%":PRINT #5;"%"
100 RETURN
110 REM
120 REM KEYIN ROUTINES
130 REM
140 KF=1
150 POKE 764,255
160 PRINT #5;"%VT22":PRINT #5;" ( Press any key to
continue ) "
170 FOR Z=0 TO WT
180 IF PEEK(764)+KF=255 THEN NEXT Z
190 PRINT #5;"%VTO":PRINT #5:KF=0:RETURN
200 REM
210 REM ENTRY POINT:SG LOADED?
211 REM
220 IF PEEK(22016)=162 THEN 245
222 OPEN #1,4,0,"D:SUPERG"
224 TRAP 230
226 FOR A=22010 TO 32000
228 GET #1,X:POKE (A),X:NEXT A
230 CLOSE #1:OPEN #1,4,0,"D:IMAGES"
232 TRAP 240:FOR A=19706 TO 22015
234 GET #1,X:POKE (A),X:NEXT A
240 CLOSE #1
245 X=USR(22016):OPEN #5,12,0,"G:"
250 REM
260 REM FIRST DISPLAY
270 REM

```

```

280 PRINT #5;CHR$(125)
290 S=1:C=1:T=1
300 GOSUB 140:W=20
310 PRINT #5;"%VT0":PRINT #5;"%CT";C
320 PRINT #5;"
330 PRINT #5;"<< Supergraphics (C) P. Lutus 1980 >>"
340 PRINT #5;"
350 PRINT #5;"%CT3":PRINT #5
360 PRINT #5;"... An INCREDIBLE Graphics program."
370 C=C+1:IF C=4 THEN C=1
390 S=-S
400 IF S=1 THEN T=-T
410 GOSUB 50
420 IF PEEK(764)<>255 THEN 440
430 W=W-1:IF W>0 THEN 310
440 REM
450 REM ATARI SIGN DISPLAY
460 REM
465 POKE 764,255
470 PRINT #5;"%ND,NW,SH19776,RZ":PRINT #5;"%VT10"
480 PRINT #5;"Supergraphics is an Atari enhancer."
500 PRINT #5;"It will do things Atari can't."
510 PRINT #5:PRINT #5;"Supergraphics can:":PRINT #5
520 PRINT #5;"1. Draw three-dimensional color
objects"
530 PRINT #5;" at rates up to 30 per second."
540 PRINT #5
550 PRINT #5;"%CT2"
560 PRINT #5;"2. Display 3 color and inverse text."
590 PRINT #5;"%CT3"
600 PRINT #5
610 PRINT #5;"3. Allow you control from BASIC using"
612 PRINT #5;" simple printed commands."
640 GOSUB 140
650 PRINT #5;"%PX,PY52,SC20,RX10,YD,RYOT2048S8"
660 REM
670 REM X,Y,Z FRAME DISPLAY
680 REM
685 POKE 764,255
690 PRINT
#5;"%ND,NW,SH19904,RX10,RY10,PX64,PY48,SC20"
700 PRINT #5;"Images are provided to Supergraphics"
710 PRINT #5;"as sets of three numbers: X, Y, and
Z."

```

```
720 PRINT #5:PRINT #5;"X ( ) is horizontal
position."
730 PRINT #5;"Y ( ) is vertical position."
740 PRINT #5;"Z ( ) is depth."
750 PRINT #5:PRINT #5;"With these numbers, any point
in space"
760 PRINT #5;"may be defined."
770 PRINT #5:PRINT #5;"While drawing the image,
Supergraphics"
780 PRINT #5;"may be commanded to rotate around"
790 PRINT #5;"the X, Y, or Z axes, or move along
the"
800 PRINT #5;"axes. The image may be stretched
along"
810 PRINT #5;"the axes as well."
820 GOSUB 140
830 FOR Q=0 TO 2
840 PRINT #5;"%CT";Q+1
850 PRINT #5;"%VT";3+Q:PRINT #5;"%HT4"
855 PRINT #5;" "
860 NEXT Q
862 POKE 764,255
870 PRINT #5;"%YD,PX64,PY-50,SC15,RX-20,RY8T4096S8"
880 PRINT #5;"%NW,ND,PX,PY,RX-20,RY-20,SC30"
885 POKE 764,255
890 PRINT #5;"Here are examples of Supergraphics"
900 PRINT #5;"image controls:"
910 PRINT #5;"%VT22":PRINT #5;"X axis rotation"
920 PRINT #5;"%YD,RX-12T236S8"
930 PRINT #5;"%VT22":PRINT #5;"Y"
940 PRINT #5;"%RY-12T236S8"
950 PRINT #5;"%VT22":PRINT #5;"Z"
960 PRINT #5;"%RZ8T256S8"
970 PRINT #5;"%VT22":PRINT #5;"X axis translation"
980 PRINT #5;"%TX-32T32S4,TX"
990 PRINT #5;"%VT22":PRINT #5;"Y"
1000 PRINT #5;"%TY-32T32S4,TY"
1010 PRINT #5;"%VT22":PRINT #5;"Z"
1020 PRINT #5;"%TZ-32T32S4,TZ"
1030 PRINT #5;"%VT22":PRINT #5;"X axis scaling
"
1040 PRINT #5;"%SX30T-30S-3,SX-30T30S3"
1050 PRINT #5;"%VT22":PRINT #5;"Y"
1060 PRINT #5;"%SY30T-30S-3,SY-30T30S3"
```

```
1070 WT=500:GOSUB 150
1080 REM
1090 REM HANG GLIDER DISPLAY
1100 REM
1105 POKE 764,255
1110 PRINT #5;"%NW,ND,SH19968,SC8"
1120 PRINT #5;"%VT11":PRINT #5;"Amazing displays can
be developed using"
1130 PRINT #5;"the features of Supergraphics."
1140 PRINT #5
1150 PRINT #5;"Game programs can be written in ways"
1160 PRINT #5;"never before possible."
1170 PRINT #5:PRINT #5;"Remember: All Supergraphics
features"
1180 PRINT #5;"are available from BASIC with a very"
1190 PRINT #5;"simple command format."
1210 GOSUB 140
1230 C=0:S=1:W=8
1240 S=-S
1250 RX=0:RY=64*S:RZ=32:PX=0:PY=48
1260 C=C+1:IF C=3 THEN C=1
1280 PRINT #5;"%CG";C
1290 FOR X=-92*S TO 92*S STEP 6*S
1300 IF PEEK(764)<>255 THEN 1382
1305 SZ=ABS(INT(5*X/92))+3
1310 RX=INT(RX-5+(RND(1)*10))
1320 RY=INT(RY-5+(RND(1)*10))
1330 RZ=INT(RZ-5+(RND(1)*10))
1340 PY=INT(PY-5+(RND(1)*10))
1350 IF PY<48 THEN PY=PY+4
1360 PRINT #5;"%YD,PX";X;" ,PY";PY;" ,RX";RX;
",RY";RY;" ,RZ";RZ;" ,SC";SZ
1370 NEXT X
1380 PRINT #5;"%ND":W=W-1:IF W>0 THEN 1240
1382 REM
1384 REM BUTTERFLY
1386 REM
1388 POKE 764,255
1390 PRINT #5;"%ND,NW,SH20057,PY65,RX30,RZ,SC8"
1400 CG=1:S=1:W=16
1410 N1=1:N3=3:N5=5:N7=7
1420 PRINT #5;"%VT9"
1430 PRINT #5;"Supergraphics commands are very
simple."
```



```

1440 PRINT #5:PRINT #5;"Example: 10 PRINT #5; ";
CHR$(34);"%RX 10";CHR$(34):PRINT #5
1450 PRINT #5;"This command means 'rotate the X
axis"
1460 PRINT #5;"by 10 units'."
1470 PRINT #5:PRINT #5;"There are no
hard-to-remember 'PEEKs'"
1480 PRINT #5;"and 'POKES' in Supergraphics."
1490 PRINT #5:PRINT #5;"Supergraphics is designed to
be very"
1500 PRINT #5;"simple to use."
1510 GOSUB 140
1520 PRINT #5;"%ND,CG";CG
1530 FOR PX=-96*S TO 96*S STEP 8*S
1540 RY=PX*N3
1550 PRINT #5;"%YD,SH+20057,PX";PX;" ,RY";RY
1560 IF PEEK(764)<>255 THEN 1630
1570 PRINT #5;"%SH+20129"
1580 NEXT PX
1590 S=-S
1600 CG=CG+N1:IF CG=N3 THEN CG=N1
1620 W=W-1:IF W>0 THEN 1520
1630 POKE 764,255
1640 A1=20200:A2=19712:A3=19968:A4=19904:S1=8:
S2=14:S3=6:S4=10:GOSUB 1710
1680 PRINT #5;"%ND,RX,RY,RZ,PX,PY"
1690 GOTO 250
1710 C=0:? #5:? #5;"%NW,ND"
1730 ? #5;"%VT19":? #5;"Any number of images can be
on"
1740 ? #5;"display under individual program
control.":GOSUB 140
1750 FOR Y=-10 TO 48 STEP 58
1760 FOR X=-64 TO 64 STEP 128
1765 A0=A1:A1=A2:A2=A3:A3=A4:S0=S1:S1=S2:S2=S3:S3=S4
1770 ? #5;"%":?
#5;"%ND,SH";A0;" ,PX";X;" ,PY";Y;" ,CG";C
1780 Z=X+Y-13:B=Z+256
1790 ? #5;"%YD,RX";Z;" ,RY";Z;" ,RX";Z;" ,SC";S0
1800 C=C+1:IF C=4 THEN C=1
1820 ? #5;"%RX";Z;"T";B;"S8"
1830 GOSUB 1900
1840 ? #5;"%RY";Z;"T";B;"S8"
1845 GOSUB 1900

```

```
1850 ? #5;"%RZ";Z;"T";B;"S8"  
1860 GOSUB 1900  
1870 NEXT X:NEXT Y  
1885 KF=0:WT=500:GOSUB 150  
1890 POKE 764,255:RETURN  
1900 IF PEEK(764)=255 THEN RETURN  
1910 POP :GOTO 1890
```

## Appendix E: GRFBAS Program Listing (BASIC)

NOTE: This is an extended version from that on the disk or tape. It provides a better capability for manipulating shape tables.

```

2 OPEN #4,4,0,"K:"
5 GRAPHICS 24:SETCOLOR 2,0,0:SETCOLOR 1,0,15
10 DIM C$(20):DIM L$(20):DIM Q$(20):DIM M$(30)
20 DIM X$(20):DIM Y$(20):DIM Z$(20)
✓30 SUP=22016:START=16384:FILE=START:TOP=20480:GOSUB
  50:GOTO 440
✓50 FOR AD=FILE TO TOP STEP 4
✓55 IF PEEK(AD)<>255 THEN NEXT AD
✓60 RETURN
80 PRINT #5;"%VTO":PRINT #5:F=1:A=0
120 A=A+1:IF A=3 THEN A=5
160 IF A>7 THEN A=1
200 Z=Y:Y=X:X=A
240 GOSUB 840
✓250 FOR J=1 TO 100:C=PEEK(764):IF C=255 THEN NEXT J
✓280 IF C=255 THEN 120
320 PRINT #5;"%CT3":F=0:GOTO 400
360 C=PEEK(764):IF C=255 THEN 360
400 GET #4,C:C$=CHR$(C):RETURN
✓440 IF PEEK(SUP)=162 THEN 560
450 OPEN #1,4,0,"D:SUPERG"
460 TRAP 520
✓480 FOR A=SUP-6 TO 25672
482 GET #1,X:POKE (A),X:NEXT A
✓520 CLOSE #1:POKE FILE,255
560 I=16
✓600 X=USR(SUP):OPEN #5,12,0,"G:"
✓640 PRINT #5;"%ND"
✓680 PRINT #5;"%SH";FILE
✓720 PRINT #5;"%VTO":PRINT #5
760 PRINT #5;"%SC?":DS=PEEK(218)
800 PRINT #5;"%NW,PX64,PY-48 ,SC8,YD"
840 PRINT #5;"%VT,HT"
880 PRINT #5;"%CT";Z
920 PRINT #5;"
"
960 PRINT #5;"%CT";Y
1000 PRINT #5;" << Supergraphics >> (C) P. Lutus

```

```

1980 "
1040 PRINT #5;"%CT";X
1080 PRINT #5;"
"
1120 IF F THEN RETURN
1160 PRINT #5;"%CT3"
1200 PRINT #5
1240 PRINT #5;"OPTIONS":PRINT #5
1280 PRINT #5;"(S)HOW: (R)OT (T)RAN (P)OS (S)CALE"
1320 PRINT #5;"      :      OF (X),(Y),(Z)."
1360 PRINT #5;"      : (C)OMMAND (*)ZERO"
1400 PRINT #5;"      : (-) CHANGE DIRECTION"
1440 PRINT #5;"      : (I)NCREMENT SIZE (Q)UIT"
1480 IF F THEN RETURN
1520 PRINT #5:PRINT #5
✓1560 PRINT #5;"(D)RAW IMAGE      (N)EW FILE":PRINT
#5
1600 PRINT #5;"(F)ILE SAVE/LOAD"
1640 PRINT #5:PRINT #5:PRINT #5;"IMAGE --->"
1680 PRINT #5;"%VT22"
1720 PRINT #5;"SELECT (LETTER) :  "
1760 GOSUB 1800:GOTO 720
1800 GOSUB 80:PRINT #5;"%NW"
1840 IF C$="" THEN RETURN
✓1880 IF C$="Q" THEN PRINT #5;"%QT":TRAP 40000:END
✓1890 IF C$<>"N" THEN 1920
✓1900 ? #5;"(S)TART FILE":? #5;"(G)O TO FILE":?
#5;"(C)LEAR"
✓1905 GOSUB 360:IF C$="S" OR C$="G" THEN 6100
✓1910 IF C$<>"C" THEN RETURN
✓1915 GOTO 3180
1920 IF C$="D" THEN 3120
1960 IF C$="S" THEN 2120
2000 IF C$="I" THEN 4520
2040 IF C$="F" THEN 4640
2080 RETURN
2120 PRINT #5;"%NW,PX,PY,SC";DS:F=1:GOSUB 1280:F=0
2160 GOSUB 360:PRINT #5;"%NW":GOTO 2240
2200 PRINT #5;"%VT,HT":GOSUB 360
2240 IF C$="Q" THEN RETURN
2280 IF C$="I" THEN GOSUB 4520:GOTO 2120
2320 IF C$<>"C" THEN 2520
2360 PRINT #5;"%VTO,HTO":PRINT #5;"ENTER
SUPERGRAPHICS      "

```

```

2400 PRINT #5;"COMMAND : ";:INPUT #5;C$
2410 PRINT #5
2440 PRINT #5;"%";C$
2480 PRINT #5;"%VTO,HTO":PRINT #5;"
2520 IF C$<>"*" THEN 2630
2560 PRINT #5;"%NW,PX,PY,RX,RY,RZ,TX,TY,TZ,SC8"
2600 GOTO 2200
2630 IF C$="-" THEN I=-I:GOTO 2760
2640 IF C$<>"R" AND C$<>"T" AND C$<>"S" AND C$<>"P"
THEN RETURN
2680 L$=C$
2720 PRINT #5;"%VTO,HTO":PRINT #5;"BASIC COMMAND :
";L$;
✓2725 IF L$<>"P" THEN ? #5;" (X,Y,Z)? " :GOTO 2760
✓2730 ? #5;" (X,Y)? "
2760 GOSUB 360
2800 IF C$="-" THEN I=-I:GOTO 2760
2840 M$=L$:M$(LEN(M$)+1)=C$
2880 IF C$<>"X" AND C$<>"Y" AND C$<>"Z" THEN GOTO
2240
2920 PRINT #5;"%";M$;"?":Q=PEEK(218)+I
2960 PRINT #5;"%";M$;Q
3000 IF Q>127 THEN Q=Q-256
3040 PRINT #5;"%VTO,HTO":PRINT #5;"BASIC COMMAND :
";M$;" = ";Q;" "
3080 GOTO 2760
3120 GOSUB 50
✓3160 IF AD<TOP THEN 3190
✓3170 ? #5;"NO FILE END FOUND":? #5
✓3180 AD=START:? #5:? #5;"AREA CLEARED":? #5:FOR A=AD
TO AD+256:POKE A,0:NEXT A
✓3185 POKE AD,255:FILE=AD:? #5;"%SH";AD
✓3190 X=0:Y=0:Z=0:M$=""
3200 PRINT #5;"%NW,PX,PY40,SC";DS
✓3240 P=AD-24:IF P<FILE THEN P=FILE
3280 PRINT #5;"%VT14"
3320 FOR D=P TO AD-4 STEP 4
3360 B1=PEEK(D)
3400 IF B1=255 THEN 3760
3440 L$="M":IF B1>127 THEN L$="D":B1=B1-128
3480 C$=""
3520 IF B1<>0 THEN C$=CHR$(B1+48)
3560 L$(LEN(L$)+1)=C$
3600 A=PEEK(D+1):B=PEEK(D+2):C=PEEK(D+3)

```

```

3640 A=A-256*(A>127):B=B-256*(B>127):C=C-256*(C>127)
3680 PRINT #5;L$;"      ";;:PRINT #5;"X = ";A;"
";:PRINT #5;"Y = ";B;"      ";;:
PRINT #5;"Z = ";C
3720 NEXT D
3760 PRINT #5;"%VT20":PRINT #5;"(D)RAW,(M)OVE,(1-7)
COLOR, (-) DELETE"
3800 PRINT #5;"EXAMPLES: 'D3','M'. ENTER : ";;:INPUT
#5;L$
3840 IF L$="" OR L$="Q" THEN RETURN
3880 IF L$<>"-" THEN 4000
√3920 AD=AD-4:IF AD<FILE THEN AD=FILE
3960 POKE AD,255:GOTO 3200
4000 Q$=L$(1,1)
4040 IF Q$<>"M" AND Q$<>"D" THEN 3760
4080 Q=128*(Q$="D")
4120 IF LEN(L$)<2 THEN 4200
4160 Q=Q+ASC(L$(2))-48
4200 PRINT #5;"%VT20,HTO"
4240 PRINT #5;"
"
4280 PRINT #5;"%VT20,HTO"
4290 TRAP 4240
4320 PRINT #5;"";L$;"";M$;" X :";:INPUT #5;X$
4330 PRINT #5;"";L$;"";M$;" Y :";:INPUT #5;Y$
4340 PRINT #5;"";L$;"";M$;" Z :";:INPUT #5;Z$
4342 IF X$<>" " THEN X=VAL(X$):X=X+256*(X<0)
4344 IF Y$<>" " THEN Y=VAL(Y$):Y=Y+256*(Y<0)
4346 IF Z$<>" " THEN Z=VAL(Z$):Z=Z+256*(Z<0)
4400 POKE AD,Q:POKE AD+1,X:POKE AD+2,Y:POKE AD+3,Z
4440 AD=AD+4:POKE AD,255
4450 M$="(NO ENTRY=LAST VALUE)"
4480 GOTO 3200
4520 PRINT #5;"ENTER INCREMENT SIZE (NOW ";I;") : ";
4580 TRAP 4610
4600 INPUT #5;I
4610 RETURN
4640 PRINT #5;"%NW,PX,PY"
√4680 FOR AD=FILE TO SUP STEP 4
√4720 IF PEEK(AD)<>255 THEN NEXT AD
√4760 PRINT #5;"FILE SIZE : ";AD-FILE;" BYTES":PRINT
#5
√4800 PRINT #5;"ENTER FILE NAME : ";
√4840 INPUT #5;L$

```

```
✓4880 IF L$="" THEN RETURN
4920 PRINT #5
- 4960 PRINT #5;"(L)OAD OR (S)AVE : ";:INPUT #5;Q$
5000 IF Q$="" THEN RETURN
5040 IF Q$="L" THEN 6000
5080 IF Q$<>"S" THEN RETURN
✓5090 Q$="D":Q$(3)=L$
✓5120 OPEN #1,8,0,Q$
5125 TRAP 5200
✓5130 FOR A=FILE TO TOP
5140 C=PEEK(A):PUT #1,C
5150 IF C<>255 THEN NEXT A
5200 CLOSE #1:RETURN
6000 Q$="D":Q$(3)=L$
6010 OPEN #1,4,0,Q$
6020 TRAP 5200
6030 FOR A=FILE TO TOP
6040 GET #1,C
6050 POKE (A),C
6060 IF C<>255 THEN NEXT A
6070 CLOSE #1:RETURN
✓6100 REM GET NEW FILE ADDRESS
6110 ? #5;"CURRENT FILE ENDS AT ";AD:? #5
6120 Q$="":L$=C$:? #5:? #5;"ENTER ADDRESS: ";
6130 FOR J=1 TO 6:GOSUB 360:IF C=155 OR J=6 THEN
6150
6140 Q$(J)=C$:? #5;C$;:NEXT J
6150 FILE=VAL(Q$):IF L$="S" THEN POKE FILE,255
6160 ? #5:? #5;"%NW,ND,SH";FILE;":YD":RETURN
```

Appendix F: Shape editing

GRFBAS has a very limited editor for generating shape tables for image files. If you have Atari World, Supergraphic's sister product also from United Software of America, you can use the more extensive editor in it to generate SH files for SUPERG.



## Appendix G: Creating images

This is a sample on how to make Supergraphics shape tables. We will make a pyramid (which should be put at SH20200 for the extended demo in Appendix D) and the fighter at the beginning of the demo (SH19712).

The first thing to do is graph out the figure on a sheet of paper (see fig. A). Then determine the coordinates. For the pyramid the four points are:

X	Y	Z	Point
-1	-1	0	a
0	1	1	b
0	1	-1	c
1	-1	0	d

Determine the easiest path that will cover all of the lines you wish to draw. Try to make as many of the lines continuations, rather than redrawing new lines. For the pyramid you need 2 sets of lines.

Set one: Connecting points a-b-c-a-d-b

Set two: Connecting points c-d

Set up the following table:

Move/color	X	Y	Z
0	-1	-1	0
	0	1	1
	0	1	-1
	-1	-1	0
	1	-1	0
	0	1	1
0	0	1	-1
	1	-1	0

These basic points should be multiplied by a common factor (e.g. 127) for a better image. Now enter the table at 20200 with the extended GRFBAS program in Appendix E, or poke it into memory at that location.

NOTE: In order to better show the points, these images are not to scale.

Figure A. The pyramid

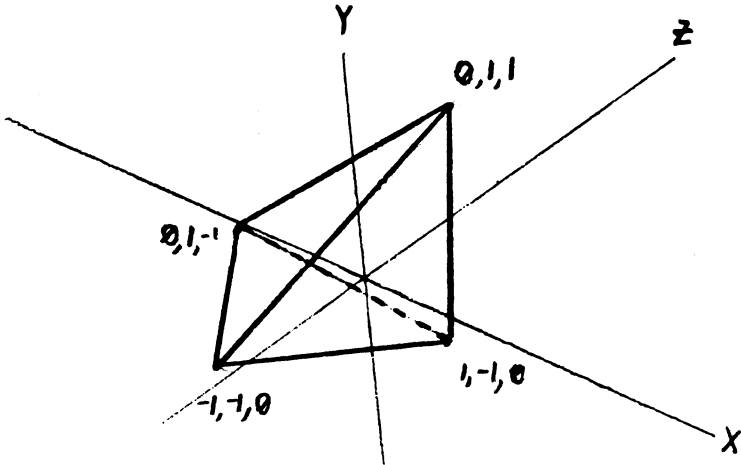
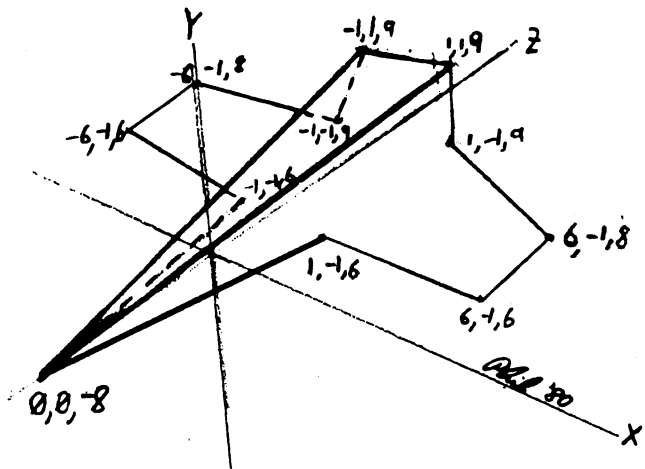


Figure B. The fighter



The Fighter

For the fighter, there are eleven points:

X	Y	Z	Point
0	0	-8	a
1	-1	6	b
6	-1	6	c
6	-1	8	d
1	-1	9	e
1	1	9	f
-1	1	9	g
-1	-1	9	h
-6	-1	8	i
-6	-1	6	j
-1	-1	6	k

This image can also be drawn with two lines:

- Line one: Connecting points  
a-b-c-d-e-f-a-g-h-i-j-k-a
- Line two: Connecting points f-g

In order to make the image finer quality, multiply each point by eight, making the image larger and with better detail.

Appendix H

The following addresses are given for those programmers who wish to program Supergraphics directly in 6502 machine language. The information is given as-is. The source listings are proprietary, and will not be given out.

HEX	LABEL	HEX	LABEL
----	-----	----	-----
5600	INIT	5AA6	DRAWIT
5688	ERASE	5838	SC
56F9	READER	5848	CT
5731	READ3	5B6F	PLOT
598A	PRCHAR	5B9E	LINE
57CE	QUIT	5A40	SCROLL
5903	GETDIG	5E7C	SETCOL
586C	CONLOOP		

The following are the HEX codes used for each command:

Code	Command	Code	Command
0	RX	B	CG
1	RY	C	SH
2	RZ	D	VT
3	TX	E	HT
4	TY	F	NW
5	TZ	10	QT
6	PX	11	SC
7	PY	12	CT
8	SX	13	YD
9	SY	14	ND
A	SZ		

Variable table

Locations	Variables
-----	-----
53DE	XROT
53DF	YROT
53E0	ZROT
53E1	X TRAN
53E2	Y TRAN
53E3	Z TRAN
53E4	X POS
53E5	Y POS
53E6	SCAL X
53E7	SCAL Y
53E8	SCAL Z
53E9	GRAPHICS COLOR
53EA	SHAPE ADDRESS LO
53EB	SHAPE ADDRESS HI
DC	OLDX
DD	OLDY
DA	PLOTX
DB	PLOTY
53C4	NUMBER LO
53C5	NUMBER HI
53C6	LOOPFROM LO
53C7	LOOPFROM HI
53C8	LOOPTO LO
53C9	LOOPTO HI
53CA	LOOPSTEP
53BB	SCBAS (\$60 OR \$80)
53BD	BUFPTR
5385	BUFPOS
53BA	SCREEN (0 OR 3)
53B9	PLOTFLAG
53C2	LASTIN
53C3	COMMAND

