

GRAPHICS ANIMATION

*A system for creating and displaying
graphics and effecting animation
on the Sorcerer computer*

By Lee Anders

A
product
of



QUALITY SOFTWARE

6660 Reseda Blvd., Suite 105, Reseda, CA 91335
(213) 344-6599

Published by Quality Software
©1980 by Quality Software. All rights reserved.
No part of this publication may be reproduced
without prior written consent.

GRAPHICS ANIMATION

Your GRAPHICS ANIMATION package consists of two independent programs. The first program, which is named SYSTM, is a set of machine language subroutines including a data storage area and is approximately 3000 bytes long. The entire SYSTM program is "hidden" in lines 1 and 2 of a BASIC program. When the program is RUN, control jumps over lines 1 and 2 and all BASIC lines greater than 2 are available for the user's program. The user's program calls SYSTM subroutines using theUSR function. When the user's program is saved, the SYSTM program and all special graphics are saved right along with it.

Also included in the GRAPHICS ANIMATION package is a CHARACTER MAKER program to aid you in creating your own graphics characters and saving them for inclusion in a BASIC program. Special graphics layout paper has been provided that is also helpful in designing graphics shapes.

CONTENTS

	PAGE
I LOADING INSTRUCTIONS	4
II DEMONSTRATION PROGRAMS	5
III USING SYSTM SUBROUTINES	10
IV GRAPHIC IMAGE HANDLING	11
Graphic Character Set Save & Restore (SYSTM Subroutines 8 and 9)	
Screen Addressing (Variables GX and GY)	
Character Display and Read (SYSTM Subroutines 3 and 2, Variable GC)	
Graphic Image Definition	
Image Display (SYSTM Subroutine 5, Variable GB\$)	
Move an Image (SYSTM Subroutine 6)	
Erase an Image (SYSTM Subroutine 7)	
Null Character	
Set Up an Array (SYSTM Subroutine 4)	
Clear Screen (SYSTM Subroutine 1)	
V MEDIUM RESOLUTION GRAPHICS	17
Medium Resolution Enable (SYSTM Subroutine 10)	
Point On/Off (SYSTM Subroutines 12 and 13)	
Image Definition	
Medium Resolution Graphic Characters Image Display, Move, Erase	
VI KEYBOARD INTERFACE	20
(SYSTM Subroutine 20)	
VII CHARACTER MAKER PROGRAM	21
VIII CHARACTER CODES	26
IX GRAPHICS LAYOUT SHEETS	27
X TABLE OF SYSTM SUBROUTINES	28

I. LOADING INSTRUCTIONS

There are three programs on your GRAPHICS ANIMATION cassette. The first program on the cassette is a copy of the SYSTM program part of your GRAPHICS ANIMATION package. Following this is a DEMO program that uses the SYSTM program. Following the DEMO program are two copies of the CHARACTER MAKER program that you will find convenient to use in designing and making graphic shapes.

The first two programs load with a CLOAD command. To load SYSTM, be sure that the BASIC ROMPAC is inserted before you turn on your Sorcerer. Turn on the Sorcerer and with the cassette in the recorder, follow these directions:

```
(RESET the Sorcerer)
READY
CLOAD SYSTM
```

At this point, SYSTM will be loaded and will reside within your BASIC program hidden in lines 1 and 2. You will be able to type in your program at that point. Do not use line numbers less than 3.

Following the copy of SYSTM there is a DEMO program written in BASIC that contains a copy of SYSTM within it. Should the copy of SYSTM fail to load, you may recover it by loading the DEMO program (discussed in the next section) and deleting all line numbers greater than 2. DEMO is loaded in the same way as SYSTM except that CLOAD SYSTM should be replaced by CLOAD DEMO.

Following the DEMO program are two copies of the CHARACTER MAKER program. This is a machine language program for assisting in the construction of graphic images needed in your BASIC program. To load and run this program, follow these instructions:

```
BYE (to enter the Sorcerer Monitor)
LOG CHAR
```

After the program is loaded, you will be able to begin defining a set of graphic characters and images. To operate this program see the section of this booklet entitled CHARACTER MAKER.

II. DEMONSTRATION PROGRAMS

The DEMO program is written in BASIC and a listing of all lines except 1 and 2 is provided below. Within the DEMO program there are three demonstration programs. To run the first demonstration program type RUN<CR> where <CR> stands for the RETURN key. The man may be slowed by pressing the 'Z' key or speeded up by pressing the '/' key. The second and third demonstration programs are medium resolution graphics programs. The second program may be run by typing RUN 500<CR> while the third may be run by typing RUN 800<CR>. Any of these programs may be stopped by typing CTRL C which returns the user to BASIC.

Before running the DEMO programs, it is a good idea to place some text on the screen. This will help to illustrate the power of the SYSTM commands for display, erase and move. The text reappears each time the moving object passes by it, and the programmer does not need to keep track of the background information. To place some text on the screen, try typing LIST 500<CR> before running the demonstration programs.

The following is an annotated listing of the DEMO program. Most of the SYSTM commands are illustrated in this listing.

DEMO PROGRAM LISTING

```
2 REM (C) COPYRIGHT 1980 BY QUALITY SOFTWARE
5 REM SET UP USR ADDRESS FOR SYSTM SUBROUTINES
10 POKE 260,222
20 POKE 261,1
30 REM LIST THE CHARACTER CODES DEFINING 7 MEN
40 DATA 3,3,192,193,3,3,1
42 DATA 3,3,194,195,3,3,1
44 DATA 3,3,196,197,198,199,1
45 DATA 3,214,200,201,205,3,1
47 DATA 3,3,202,203,204,3,1
49 DATA 3,206,207,208,209,210,2
55 DATA 3,3,212,213,3,3,1
57 DATA 215,216,217,218,3,3,1
59 DATA 219,220,221,222,3,3,2
65 DATA 3,3,212,224,226,3,1
```

```

67 DATA 3,3,202,225,204,3,1
69 DATA 3,206,207,208,209,210,2
75 DATA 3,3,235,236,3,3,1
77 DATA 215,232,233,234,3,3,1
79 DATA 219,220,221,222,3,3,2
85 DATA 3,3,212,213,3,3,1
87 DATA 3,223,227,228,3,3,1
89 DATA 242,243,244,231,237,3,2
95 DATA 3,3,235,213,3,3,1
97 DATA 3,3,239,240,3,3,1
99 DATA 3,229,230,238,3,3,2
105 DATA 3,3,235,213,3,3,1
107 DATA 3,3,245,246,3,3,1
109 DATA 3,3,247,248,3,3,2
110 REM DIMENSION 7 ARRAYS FOR THE MEN
115 DIM A1(42),A2(42),A3(42),A4(42),A5(42),A6(42),A7(42)
116 REM FILL THE 7 ARRAYS
117 REM NOTE THAT THE TOP 21 CHARACTER CODES ARE
118 REM     THE SAME FOR EACH MAN
120 FOR I=0 TO 41 : READ A1(I): NEXT I
125 FOR I=0 TO 20
130 A2(I)=A1(I)
135 A3(I)=A1(I)
140 A4(I)=A1(I)
145 A5(I)=A1(I)
150 A6(I)=A1(I)
155 A7(I)=A1(I)
160 NEXT I
165 FOR I=21 TO 41: READ A2(I): NEXT I
170 FOR I=21 TO 41: READ A3(I): NEXT I
175 FOR I=21 TO 41: READ A4(I): NEXT I
180 FOR I=21 TO 41: READ A5(I): NEXT I
185 FOR I=21 TO 41: READ A6(I): NEXT I
190 FOR I=21 TO 41: READ A7(I): NEXT I
195 A1(15)=211
200 DL= 59
202 REM RESTORE GRAPHIC CHARACTERS
205 X1=USR(9)
207 REM SET BEGINNING ADDRESS OF FIRST MAN
210 GX=1: GY=15: GB$="A1"
212 REM LOOP FROM 215 TO 285 INDEFINITELY

```

```

213 REM SUCCESSIVELY DISPLAY SIX MEN BY CALLING
214 REM     SUBROUTINE AT 400
215 GOSUB 400
230 GB$="A2"
235 GOSUB 400
240 GB$="A6"
245 GOSUB 400
250 GB$="A3"
255 GOSUB 400
260 GB$="A4"
265 GOSUB 400
270 GB$="A7"
275 GOSUB 400
280 GB$="A1"
285 GOTO 215
300 REM SUBROUTINE AT 400 DOES THE FOLLOWING:
305 REM     DISPLAY MAN
310 REM     READ THE KEYBOARD; IF 'Z' IS PRESSED,
315 REM     INCREASE THE DELAY, IF '/' IS PRESSED,
320 REM     DECREASE THE DELAY
330 REM     EXECUTE THE DELAY LOOP,
340 REM     INCREMENT THE MAN'S POSITION
350 REM     ERASE THE MAN
360 REM     RETURN
400 X1=USR(5)
405 X1=USR(20)
415 IF X1=72 THEN DL=DL-5
420 IF X1=17 THEN DL=DL+5
425 IF DL>400 THEN DL=400
430 IF DL<2 THEN DL=2
435 FOR I=1 TO DL: NEXT I
440 GX=GX+1
445 IF GX=64 THEN GX=1
450 X1=USR(7)
455 RETURN

```

```

470 REM MEDIUM RESOLUTION FUNCTION PLOT
490 REM CLEAR THE SCREEN WITHOUT
495 REM     RESETTNG GRAPHICS CHARACTERS
500 X1=USR(1)
502 REM ENABLE MEDIUM RESOLUTION GRAPHICS MODE
505 X1=USR(10)
507 REM INITIAL Y COORDINATE
510 GY=45
512 REM LOOP PLOTTING HORIZONTAL LINE
515 FOR GX=1 TO 128
520 X1=USR(12)
525 NEXT GX
527 REM LOOP TO DEFINE AND PLOT MEDIUM RESOLUTION CURVE
530 FOR I=1 TO 128 STEP .2
535 GY=SIN(I/5)
540 GY=45+INT(GY*((129-I)/4))
545 GX=INT(I)
550 X1=USR(12)
555 NEXT I
560 END

```

```

700 REM MEDIUM RESOLUTION BOAT
710 REM 799 TO 825 LIST THE CHARACTER CODES FOR THE BOAT
799 DATA 4
800 DATA 3,3,3,3,3,3,193,207,237,223,193,194,1
810 DATA 3,3,199,203,223,237,223,247,243,255,247,255
815 DATA 223,237,195,195,1
820 DATA 208,253,255,255,255,255,255,255,255,255,255,255
825 DATA 255,255,254,224,2
830 REM DIMENSION THE ARRAY THAT WILL BE DISPLAYED
850 DIM B1(46)
851 REM RESTORE THE DATA STATEMENTS TO BE READ FROM THE
852 REM     TOP OF THE PROGRAM
853 RESTORE
854 REM READ DATA STATEMENTS UNTIL FIND THE CODE STORED
855 REM     AT LINE 799
856 READ I
857 IF I<>4 GOTO 856

```

```
858 REM FILL ARRAY FOR BOAT
860 FOR I=0TO46: READ B1(I): NEXT I
865 REM ENABLE MEDIUM RESOLUTION GRAPHICS
875 X1=USR(10)
877 REM INITIALIZE BOAT POSITION
880 GX=1: GY=25: GB$="B1"
885 REM DISPLAY BOAT
890 X1=USR(5)
895 REM LOOP TO DISPLAY BOAT AND DELAY
900 FOR GX=1 TO 49
910 X1=USR(6)
912 REM DELAY
915 FOR I=1 TO 90: NEXT I
920 NEXT GX
930 REM LOOP TO MOVE THE BOAT BACKWARDS
935 IF X1=1 THEN END
940 FOR GX=48 TO 1STEP -1
950 X1=USR(6)
955 FOR I=1 TO 90 : NEXT I
960 NEXT GX
970 GOTO 900
READY
```

III. USING SYSTM SUBROUTINES

Communication between your BASIC program and the subroutines within SYSTM is accomplished through the use of the USR function and four BASIC reserved variable names (GX,GY,GC,GB\$). It is good programming practice to avoid the use of these variable names except in connection with calling SYSTM subroutines. The uses of the four variable names are as follows:

GX,GY define the starting location of an image to be placed on the video screen.

GB\$ is used to pass the name of an array containing integer values defined within your BASIC program to SYSTM.

GC is used to pass a single character to or from a subroutine in SYSTM.

After GX, GY and GB\$ or GC are assigned values within your BASIC program, the SYSTM subroutine is called by an X=USR(code) BASIC instruction where "code" is a number determining the appropriate SYSTM subroutine to be called.

When Sorcerer BASIC executes the command X=USR(Y), it jumps to the address at the memory locations 260,261 (decimal). To use the subroutines within SYSTM, this address must be set to the address of the SYSTM subroutine package. The following two commands must be executed before the SYSTM subroutines become effective.

```
POKE 260,222  
POKE 261,1
```

Ordinarily these two commands will be located near the beginning of the user's program, e.g., at lines 10 and 20. When executed, these two commands set the USR address to the beginning of the SYSTM subroutines. After this is done, any time the command X=USR(Y) is executed, the value Y is passed to SYSTM. This value is used by SYSTM to determine which subroutine to execute.

The remainder of this booklet describes the subroutines available to the user with the SYSTM program. On the back page of the booklet is a table summarizing the available commands.

IV. GRAPHIC IMAGE HANDLING

Graphic Character Set Save and Restore (SYSTEM Subroutines 8 and 9)

A buffer (save area) of 1024 bytes is provided within SYSTEM in which you may store 128 graphic characters. The stored graphic characters will then be automatically saved on tape and reloaded along with SYSTEM and your BASIC program each time you CSAVE and CLOAD your program.

A subroutine 8 is provided to move all 128 programmable graphic characters into the buffer. Subroutine 9 is provided to restore the graphics set from the buffer for use by your program.

If you have generated a set of graphic characters and have loaded them into the user defined graphics area FC00-FFFF, you may move them into the buffer area of the SYSTEM program by issuing the following three direct commands (i.e. BASIC commands that do not have line numbers):

```
POKE 260,222  
POKE 261,1  
X1=USR(8)
```

The first two of these commands POKE into the Sorcerer's memory the address of the SYSTEM subroutine package. The last command executes the SYSTEM subroutine that stores the graphics definitions into the buffer in SYSTEM. Following these direct commands, the graphic character set will be saved each time you CSAVE your BASIC program.

Once you have saved your graphic character set within the buffer in SYSTEM, you will need instructions within your BASIC program to restore them to the user defined graphics area (locations FC00-FFFF in the Sorcerer memory) for use by the BASIC program. The following instructions within your BASIC program will move the character set from the buffer within SYSTEM to the user defined graphics area:

```
10 POKE 260,222  
20 POKE 261,1  
30 X1=USR(9)
```

Lines 10 and 20 set the USR call address to be that of the SYSTM program. Line 30 causes the restore subroutine within SYSTM to be executed. These three statements may well be the first statements in any program using SYSTM.

Screen Addressing (Variables GX and GY)

The Sorcerer video has 1920 addressable character locations arranged 64 wide and 30 high. Using SYSTM, screen addressing is done with the reserved variables GX, GY.

GX may be assigned any integer value from 1 to 64. Any other value will cause unpredictable results. A value of 1 points to the left hand side of the screen while a value of 64 points to the right hand side of the screen.

GY may be assigned any integer value from 1 to 30. Any other value will cause unpredictable results. A value of 1 points to the top of the screen while a value of 30 points to the bottom of the screen.

Character Display and Read (SYSTM Subroutines 3 and 2, Variable GC)

A single character may be displayed on the screen at the location addressed by GX,GY with subroutine 3. Moreover, the character currently stored at location GX,GY may be retrieved without changing it by calling subroutine 2.

To display a character on the screen, assign the character code of the desired character to GC. Then execute subroutine (3). For example,

```
100 GX=32
110 GY=15           :REM POINT TO MIDDLE OF SCREEN
120 GC=65          :REM THIS IS THE CHARACTER CODE FOR "A"
130 X1=USR(3)
```

An 'A' is displayed on the screen and the character code representing the character previously at that location is in X1. You may test X1 to determine what was previously at that location.

If you wish to find out what is currently displayed at a given screen location without changing it, use subroutine 2.

```
150 GX=32
160 GY=15           :REM POINT TO MIDDLE OF SCREEN
170 X1=USR(2)      :REM PUT CHARACTER CODE IN X1
```

X1 has the character code of the character at GX,GY.

Graphic Image Definition

An image larger than one character may be handled with a single subroutine call. Before displaying such an image, an array must be defined that contains the character codes and formatting necessary to display the image. The image may be any number of characters wide or high so long as it will fit on the screen. Each line must begin in the same horizontal position, but the lines may have different lengths.

To define an image that SYSTM can handle, you must determine the character code values that represent the image. For a better understanding of what a character code is, you may refer to the section of this booklet entitled CHARACTER CODES. The image array is usually filled using BASIC DATA statements. The first character code in the array represents the character at the top left hand corner of the image. To define the end of a line of the image, a 1 is entered into the array. To define the end of the last line of the image, a 2 is entered. You also may enter a null character (3) in places where you do not want any character displayed and you want the background image to show through. For more on this, see the section entitled Null Character.

An example of how to construct an array of character code values is given below. Each DATA statement represents a line of the image. A DIMENSION statement is used to create an array large enough to hold all of the DATA values, and a READ statement assigns the DATA values to the array elements.

```
200 DATA 138,127,1
210 DATA 220,32,251,2
220 DIM A1(6)
230 FOR I=0 TO 6
240 READ A1(I)
250 NEXT I
```

At this point the array A1 (any available variable name of two characters may be used) is loaded with character code values, and the image represented by these values may be displayed, moved, and erased.

Image Display (SYSTM Subroutine 5, Variable GB\$)

To display the image array, the screen address variables GX,GY must be assigned values. The screen address determined by GX,GY will be the location of the upper left hand corner of the image. In the example above, the character at that location has character code 138. GB\$ is assigned the name of the array which is A1 in this example. The array name must consist of two characters. Finally subroutine 5 must be called. An example of this sequence is given below.

```
260 GX=20
270 GY=7
280 GB$="A1"
290 X1=USR( 5 )
```

After the return from the subroutine 5 call, the image will be displayed and X1 will equal 1 if any character in the image overlaid any non-blank character previously displayed on the screen. Otherwise after the return, X1 will be 0. If any character is overlaid by the display of "A1", it will be restored if "A1" is moved away or erased.

Move an Image (SYSTM Subroutine 6)

Subroutine 6 is used to move images already displayed. If the value of any of the reserved variables (GX,GY,GB\$) has been set previously to the value currently desired, it does not need to be reset. Therefore, to move the current image one character position to the right, just add 1 to GX and call subroutine 6.

```
300 GX=GX+1
310 X1=USR( 6 )
```

The image is now at the new location. Just as with the display subroutine, if any non-blank character is overlaid by the image at the new location, X1 will be 1 after return from the subroutine call. Otherwise X1 will be 0. Any characters that are "uncovered" by the move will be redisplayed.

Erase an Image (SYSTEM Subroutine 7)

To erase the current image from the screen, call subroutine 7. Again, if GX,GY, and GB\$ have not been changed since the last display or move command, they need not be reset.

```
320 X1=USR(7)
```

The image is no longer displayed and any overlaid characters are redisplayed.

Null Character

A special null character (3) has been provided for use in defining an image. Each line of an image you define must begin in the same character position. Therefore if the desired image contains a line of characters that does not start in the left most column, a null character may be inserted as a place holder. The null character can also be used in the center of an image. A space character (32) could also be used, but it will blank out any background character when the image is displayed. The following example illustrates the use of the null character to create an image in which the left most character in a line is a null.

```
350 DATA 3,3,140,141,1
360 DATA 170,171,172,173,2
370 DIM N1(9)
380 FOR I= 0 TO 9
390 READ N1(I)
400 NEXT I
```

The image created in this way has the following shape:

```
   XX
  XXXX
```

Since the top line of the image does not start in the left most column of the image, the two 3's are used as place holders. When the image is displayed, any characters on the screen that reside in the positions covered by the two place holders will remain visible.

Set Up an Array (SYSTM Subroutine 4)

When SYSTM first displays an array by means of subroutine 5, it executes an internal set up subroutine. This internal subroutine reorganizes the data inside the array. The subroutine utilizes the memory space allocated by the DIM statement to store the character code data in its own internal format. A flag is also set in the array the first time subroutine 5 is executed to indicate set up has taken place. Each time the image is redisplayed or moved, SYSTM checks to see if the flag is set, and if it is, SYSTM does not perform another set up operation. This saves considerable processing time.

If the user wishes to change the character code values within an array, he must reload the entire array. Moreover, before displaying the modified array the first time, the user must call subroutine 4. Calling subroutine 4 clears the flag, and informs SYSTM that a set up operation must be performed anew.

In summary, when reloading an array with character code values, subroutine 4 must be called prior to the first call to subroutine 5.

Clear Screen (SYSTM Subroutine 1)

The screen may be cleared at any time by calling subroutine 1. This subroutine does not affect the graphics characters with codes 128 to 191. The use of the BASIC command PRINT CHR\$(12) clears the screen but also resets the Sorcerer graphics with character codes 128 to 191.

```
X1=USR(1)
```

The screen is cleared.

V MEDIUM RESOLUTION GRAPHICS

Medium resolution graphics provide the BASIC programmer with the capability to plot an arbitrary medium resolution graphic image anywhere on the screen. There are six such medium resolution graphic points for each graphic character arranged as follows:

XX
XX
XX

Each of the two medium resolution graphic points in the middle of this rectangle is composed of eight high resolution graphic points arranged in a rectangle 4 wide and 2 high. Each of the other four medium resolution graphic points is composed of 12 high resolution points arranged in a rectangle 4 wide and 3 high.

Medium Resolution Enable (SYSTEM Subroutine 10)

Subroutine 10 is used to enable medium resolution point graphics. When medium resolution is enabled, a special set of 64 graphic characters is generated and stored as graphic characters with character codes 192 to 255. All graphic characters previously there will be overlaid. Because the characters that were there may be displayed on the screen, you must be careful to avoid enabling medium resolution graphics when characters 192 to 255 are being displayed.

```
410 X1=USR(10) :REM ENABLE MEDIUM RESOLUTION GRAPHICS
```

After statement 410 is executed, the 64 characters needed for medium resolution graphics will be stored as graphic characters 192 to 255. Animation graphic commands may still be intermixed with medium resolution graphic commands. When using animation graphics after enabling medium resolution graphics, only character codes 128 to 191 may represent user defined high resolution characters. In summary, subroutine 10 generates medium resolution graphic characters and stores them as characters 192 to 255.

When medium resolution graphics are no longer needed, any graphic characters that were originally in the buffer may be restored using subroutine 9.

Point On/Off (SYSTEM Subroutines 12 and 13)

Subroutine 12 is used to turn a medium resolution point on, and subroutine 13 is used to turn the point off. As discussed before with graphic images, the variables GX and GY are used to address the screen. When turning on or off medium resolution points with subroutines 12 or 13, GX may be any value from 1 to 128, and GY may be any value from 1 to 90. Any other value will cause unpredictable results. To turn on the center point of the screen, load GX with 64 and GY with 45 and call subroutine 12.

```
420 GX=64
430 GY=45
440 X1=USR(12)
```

The center point of the screen is now on. To turn it off, use subroutine 13. If GX,GY have not been altered, one statement is all that is needed.

```
450 X1=USR(13)
```

The center point is now off.

To draw a straight line on the screen in medium resolution graphics, use a FOR-NEXT loop:

```
460 GY=45
470 FOR GX=1 TO 128
480 X1=USR(12)
490 NEXT GX
```

Medium Resolution Animation

The graphic characters used to define medium resolution points may be used with graphic image display, move, and erase subroutines 5, 6, and 7. You may have 64 graphic characters defined in characters 128 to 191 at the same time that the medium resolution graphic characters reside in characters 192 to 255. To use the medium resolution graphic characters, you must know how to address them. The next section tells you how the 64 medium resolution characters may be addressed.

Medium Resolution Graphics Characters

Following medium resolution enable (subroutine 10), the 64 medium resolution graphic characters reside in the characters 192 to 255. Character 192 is a blank (none of the 6 medium resolution points is on), and the character 255 is the white character (all 6 of the medium resolution points is on). The map below provides the code for determining which of the 6 medium resolution points is on within a character:

```
32 16
   8  4
   2  1
```

To select a character with a given collection of medium resolution points on, add the values of the points you want on from the figure above to 192. This will be the character code for that character. For instance, if you wish a character with the upper left hand corner and lower right hand corner points on, add 32 and 1 to 192. Character code 225 will be the character code for the desired character.

Suppose that you wish to construct a large 'I' the size of four characters arranged in a 2 by 2 array. The 'I' consists of 16 medium resolution points arranged as shown below:

```
XXXX
  XX
  XX
  XX
  XX
  XX
XXXX
```

The upper left hand corner is a character that looks like this:

```
XX
 X
 X
```

The character code representing the above character is $32+16+4+1+192=245$. Doing the same thing for the other three characters composing the large 'I', we find that the upper right hand corner has code $32+16+8+2+192=250$. The lower left hand character has code $16+4+2+1+192=215$, and the lower right hand character has code $32+8+2+1+192=235$. We can define an array of character code values that can be used to display, move, and erase the large 'I'.

```

500 DATA 245,250,1
510 DATA 215,235,2
520 DIM I1(5)
530 FOR I=0 TO 5
540 READ I1(I)
575 NEXT I

```

Now "I1" can be displayed, moved, and erased using subroutines 5, 6, and 7. Note that these subroutines display characters, not medium resolution points. Therefore, GX must be a value from 1 to 64, and GY must be a value from 1 to 30.

VI. KEYBOARD INTERFACE (SYSTEM Subroutine 20)

When the BASIC statement INPUT is executed, the computer waits for the RETURN key to be pressed before executing another statement. Many action games can not be halted to wait for keyboard input. A special keyboard subroutine 20 has been included in the SYSTEM program to overcome this problem. Moreover, if more than one key is pressed simultaneously, subroutine 20 can be used to read all of the keys that are pressed.

Calling subroutine 20 will result in returning the keyboard code value for a depressed key, or if no key is down, the value 255 will be returned. If more than one key is down, repeated calls to subroutine 20 will successively return the code values of the depressed keys. When the value 255 is returned, no key is being depressed. Note that the subroutine 20 reads the SHIFT LOCK key, and that if that key is set, the routine will return the key code value 3.

```

600 I=0
610 X1=USR(20)
620 IF X1=255 THEN 700
625 IF X1=3 THEN 700 :REM IGNORE SHIFT LOCK KEY
630 KB(I)=X1
635 IF I=0 THEN 650
640 IF KB(I)=KB(0) THEN 700
650 I=I+1
660 GOTO 610
700 IF I=0 THEN 610
710 FOR J=0 TO I-1
720 PRINT KB(J);
730 NEXT J
740 GOTO 600

```

When executed, the above routine will loop until one or more keys are depressed. When one or more keys are pressed, the keyboard code values of the corresponding keys will be printed on the video.

The following table lists the keyboard codes that are returned when the corresponding key is pressed and subroutine 20 is executed:

KEYBOARD CODES

KEY	CODE	KEY	CODE	KEY	CODE	KEY	CODE
ESC	—	SKIP	11	GRPH	1	CTRL	2
1	28	Q	19	LOCK	3	SHFT	12
2	36	W	27	A	18	Z	17
3	44	E	34	S	26	X	16
4	35	R	33	D	25	C	24
5	52	T	43	F	32	V	41
6	60	Y	51	G	42	B	40
7	68	U	59	H	50	N	49
8	76	I	57	J	58	M	48
9	67	O	66	K	56	,	64
0	84	P	75	L	65	.	73
:	92	[83	;	74	/	72
-	100]	82	@	81	SHFT	12
^	91	LF	90	\	80	RPT	9
		RTN	89	_	88	R/S	—
				CLR	8		

NUMERIC KEYPAD CODES

-	99	/	98	X	97	+	96
7	116	4	106	1	105	0	104
8	107	5	114	2	113	3	4
9	124	6	115	3	4	=	123

VII. CHARACTER MAKER PROGRAM

The CHARACTER MAKER program is a separate machine language program contained on the GRAPHICS ANIMATION tape. It is loaded with an LOG CHAR command from the Sorcerer Monitor. The purpose of the CHARACTER MAKER program is to provide the user with a tool to create graphic

characters and images. The results from using the CHARACTER MAKER program may be saved on cassette and then reloaded when constructing your BASIC program. A later paragraph, "CHARACTER MAKER--SAVE MODE", explains how this is accomplished.

The CHARACTER MAKER program has the following modes:

CREATION MODE
DISPLAY MODE
SAVE MODE
RESTORE MODE
MERGE MODE

Each of these modes will be described in turn.

CHARACTER MAKER--CREATION MODE

The Creation Mode permits the user to create a graphic character. When the program is loaded with an LOG command, you will be operating in the Creation Mode. In both the Creation Mode and the Display Mode, the screen will consist of three sections. These sections are:

CHARACTER DISPLAY--The 8x8 rectangle at the upper left hand corner of the screen

IMAGE DISPLAY--A large rectangle 40 characters wide by 18 high at the upper right hand portion of the screen
(blank when the program is loaded)

KEY, CHARACTER SET AND POINTER--The three rows of characters at the bottom of the screen

The character displayed in the CHARACTER DISPLAY at the upper left hand corner of the screen will be pointed to by the select pointer in one of the two rows of characters at the bottom of the screen. Note that there are two pointers. The select pointer points to a character in one of the two rows at the bottom of the screen. The row pointer points to a row in the CHARACTER DISPLAY at the upper left portion of the screen. In the Creation Mode you may:

Select a new character
Erase the current character
Modify the current character
Enter the Display Mode, Save, Restore, or Merge

SELECT. To select another character, you may move the select pointer by pressing the SHIFT ARROW keys on the numeric keypad. Holding down the SHIFT key and pressing the right arrow key on the numeric keypad will move the select pointer to the right and display the new character in the CHARACTER DISPLAY at the upper left. You may use the REPEAT key to speed the movement of the select pointer. Similar action occurs when the SHIFT key is held down and the left arrow key is pressed.

The select pointer may be moved directly while in the Creation Mode by holding down the GRAPHICS key or both the SHIFT key and the GRAPHICS key and pressing any other key. The select pointer will move to the position indicated by the row of characters that are printed above the two rows of graphic characters at the bottom of the screen. For example, if you hold down the SHIFT and GRAPHICS keys and press the '1' key, the select pointer will move to the first character position on the bottom row.

You will also notice as you select characters that the character code for the character is printed on the screen below the CHARACTER DISPLAY. The character code is given both in the hexadecimal and decimal number systems.

ERASE. The current character, as seen at the upper left portion of the screen, can be erased by pressing the 'K' key.

MODIFY. The current character may be modified by using the keys 1 to 8 and by moving the row pointer with the 'X' and '+' keys on the numeric keypad. Pressing a key from 1 to 8 causes the corresponding pixel to be turned off if it is currently on and on if it is currently off. To modify a different row of the current character, you need to move the row pointer. Pressing '+' on the numeric keypad causes the row pointer to move down one row. Pressing the 'X' key on the numeric keypad causes the row pointer to move up one row.

OTHER MODES. To enter the Display Mode from the Creation mode press the 'D' key. Similarly you may enter the SAVE, RESTORE, or MERGE modes by pressing the 'S', 'R', or 'M' keys respectively.

CHARACTER MAKER--DISPLAY MODE

The Display Mode is entered by pressing the 'D' key while in the Creation Mode. The user may recognize that he is in the Display Mode by the presence of the hyphen (-) cursor in the IMAGE DISPLAY section of the screen. In the Display Mode the cursor may be moved by holding down

the SHIFT key and pressing one of the four arrow keys on the numeric keypad. The IMAGE DISPLAY portion of the screen is an area 40 characters wide and 18 high located at the upper right.

To display a character in the IMAGE DISPLAY, move the cursor to the desired location and, holding down the GRAPHICS key or both the SHIFT and GRAPHICS keys, press the appropriate data key to display one of the 128 characters from the CHARACTER SET displayed at the bottom of the screen. It is convenient to use the KEY row (the third row of characters from the bottom) to identify which data key to press. To display a character from the bottom row, both the SHIFT and GRAPHICS keys must be held down while the appropriate data key is pressed. To display a character from the row next to the bottom, only the GRAPHICS key should be held down while the appropriate data key is pressed.

To exit the Display Mode and return to the Creation Mode, press the 'C' key.

CHARACTER MAKER--SAVE MODE

To save a character set on tape press the 'S' key when in the Creation Mode. This will place you in the Sorcerer Monitor and print out directions on what command to use to save the character set on tape. Follow the directions if you wish to save the character set on tape. You may return to the CHARACTER MAKER program by typing

GO 100<CR>

The saved character set can be reloaded at the time you are ready to write a BASIC program. The BASIC ROMPAC must be inserted. The steps are:

1. If in BASIC, type BYE to exit BASIC and enter the Sorcerer Monitor.
2. Load the character set with a LOAD command.
3. Type PP to return to BASIC.
4. Load the SYSTM program.
5. Execute the following direct BASIC commands:

```
POKE 260,222
```

```
POKE 261,1
```

```
X1=USR(8)
```

At this point the character set you saved will be stored in the save area of the SYSTM program and can be saved along with your BASIC program with a CSAVE command.

CHARACTER MAKER--RESTORE MODE

To restore a character set from tape that has been created during a previous working session, enter the RESTORE mode by pressing 'R' while in the Creation Mode. Follow the directions printed on the screen to load a character set from tape. Once the character set has been restored, you may return to the CHARACTER MAKER program by typing

GO 100<CR>

CHARACTER MAKER--MERGE MODE

To merge a character set previously saved on cassette tape with the current character set (we shall call it the original character set for the remainder of this section), press the 'M' key to enter the MERGE mode. Enter the Monitor LOAD command as instructed. After the character set is loaded, return to the CHARACTER MAKER program by typing

GO 103<CR>

On return to the CHARACTER MAKER program the loaded character set will be displayed, and you will be able to merge selected characters from the loaded character set into the original character set. To do this, you must know the keypress for each character from the original character set that you wish to replace. To merge a character from the loaded character set into the original character set, first select the character from the loaded character set that you wish to save into the original character set. You may select that character by moving the select pointer with the SHIFT arrow keys. Then press the SPACE BAR. Select the character from the original character set that you wish to overlay by holding down the GRAPHICS or SHIFT/GRAPHICS keys and pressing the desired data key. This will merge the character from the loaded character set with the original character set. Repeat this process for as many characters as you wish to merge with the original character set. To return to the Creation Mode and to redisplay the merged character set press the 'C' key.

The MERGE mode may also be used to move graphics characters from one key to another. Simply follow the above directions with the exception that no character set is loaded from tape. Upon returning to the CHARACTER MAKER program from the Sorcerer Monitor with the GO 103<CR> command, the steps given above will cause rewriting one or more characters in new locations.

VIII. CHARACTER CODES

For those users unfamiliar with use of character codes for the Sorcerer, this section is intended to be a brief primer on their meaning. A character code, when placed in the Sorcerer's video map (F080-F7FF hex, -3968 to -2049 decimal), designates the character to be placed on the screen. A character code is any integer from 0 to 255 representing the possible values of one byte of information.

The first 128 character codes represent fixed characters for the Sorcerer. The programmer cannot change the character that is printed on the video when one of these character codes is sent to the video driver. The character codes from 128 to 255, however, represent characters that may be changed by the programmer. The CHARACTER MAKER program is designed to permit the user to design these characters for himself.

The BASIC programmer should be aware that the characters with character code values from 128 to 191 are reset by the Sorcerer each time the CLEAR key is pressed or the instruction PRINT CHR\$(12) is executed in a BASIC program. Therefore, if possible, it is desirable to avoid the use of modified characters with these character codes.

Character codes 65 to 90 represent the capitalized alphabetic characters while character codes 97 to 122 represent the lower case alphabetic characters. To display the complete set of characters, use the direct BASIC statement

```
FOR I=0 TO 255:POKE -3968+3*I,I:NEXT I
```

In this notation, I represents the character code.

IX. GRAPHICS LAYOUT SHEETS

Two graphics layout sheets have been included, stapled to the center of the booklet. Each sheet represents a screen area six characters high and 20 characters wide laid out in the same proportions as they are displayed on the Sorcerer video. Each character is eight dots wide and eight dots high. The recommended use of the layout sheets is as follows:

1. Decide what size you want the image to be and sketch the general shape on the layout sheet. Arrange your sketch so that you get the size you want while using as few characters as possible.
2. Starting with the outline of your sketch, fill in the dots to form the shape as it will appear on the screen.
3. The area at the bottom of the layout sheet has two boxes for each character you create. In one box put the keypress you want to assign to the character (e.g. GS-1 for GRAPHICS SHIFT 1). In the other box put the associated character code (e.g. 192).
4. Use the CHARACTER MAKER program to create the characters as you formed them on paper. Display the entire image using the CHARACTER MAKER program.

Pads of 50 layout sheets are available from Quality Software at a cost of \$3 per pad. Add \$1 per pad for Fourth Class shipping and handling or \$2 per pad for First Class shipping and handling. Prices subject to change without notice.

X. TABLE OF SYSTM SUBROUTINES

SUBROUTINE	VARIABLES	DESCRIPTION
1	—	CLEAR SCREEN
2	GX, GY	GET SCREEN CHARACTER
3	GX, GY, GC	DISPLAY CHARACTER
4	GX, GY, GB\$	SET UP, DISPLAY IMAGE
5	GX, GY, GB\$	DISPLAY IMAGE
6	GX, GY, GB\$	MOVE IMAGE
7	GX, GY, GB\$	ERASE IMAGE
8	—	SAVE CHARACTER SET
9	—	RESTORE CHARACTER SET
10	—	ENABLE MED. RES. GRAPHICS
12	GX, GY	MED. RES. POINT ON
13	GX, GY	MED. RES. POINT OFF
20		KEYBOARD INPUT



QUALITY SOFTWARE

6660 Reseda Blvd., Suite 105, Reseda, CA 91335

(213) 344-6599