PLAYER MAKER

ICON Software
925 Waverley St. #102
Palo Alto, CA. 94301

CONTENTS

INTRODUCTION

OVERVIEW

PLAYER MAKER is a BASIC utility program which makes it easy to use Atari's powerful animation feature called Player-Missile Graphics. PLAYER MAKER allows you to define the shape of each of your players and to view your players in different resolutions, widths, and colors. You can independently create up to four different players, but can work on constructing two players on the screen at any one time enabling you to make comparisons.

PLAYER MAKER generates a BASIC subroutine which you enter into your own program. You do not need any knowledge of the workings of Atari's Player-Missile Graphics capabilities or memory requirements to successfully utilize the power of this program. The design and implementation of the players and the necessary hooks for your program to put them on the screen and move them is totally handled by PLAYER MAKER.

REQUIRED ACCESSORIES

ATARI BASIC Language Cartridge
48K RAM
ATARI 810 Disk Drive

OPTIONAL ACCESSORIES

One ATARI Joystick Controller

CONTACTING THE AUTHOR

Users wishing to contact the author about PLAYER MAKER may write to:

Wayne Harvey
925 Waverley St. #102
Palo Alto, CA. 94301

GETTING STARTED

LOADING PLAYER MAKER INTO MEMORY

1. Be sure a BASIC Language Cartridge is in the left cartridge slot of your computer console and that you have 48K or RAM.

2. Have your computer turned off.

3. Turn on your disk drive.

4. When the busy light goes out, open the disk drive door and insert the PLAYER MAKER diskette with the label in the lower right-hand corner nearest you. (Use disk drive one if you have more than one drive.)

5. Turn on your computer and your TV set. The program will load into computer memory and start automatically.

6. Finally, remove the PLAYER MAKER master diskette once it is loaded and insert into your disk drive a blank formatted diskette for storing the subroutine created by this program.

THE FIRST DISPLAY SCREEN

If you properly loaded PLAYER MAKER, you will see:

```
---------------------------
|                         |
|     PLAYER MAKER        |
|                         |
|      (c) 1983           |
|                         |
|     WAYNE HARVEY        |
|                         |
|                         |
|  PLEASE WAIT A MOMENT   |
L-------------------------L
```

After a minute, a menu will appear at the top of your screen. The area in which you define your players will appear below this (see THE PLAYER MAKER SCREEN).
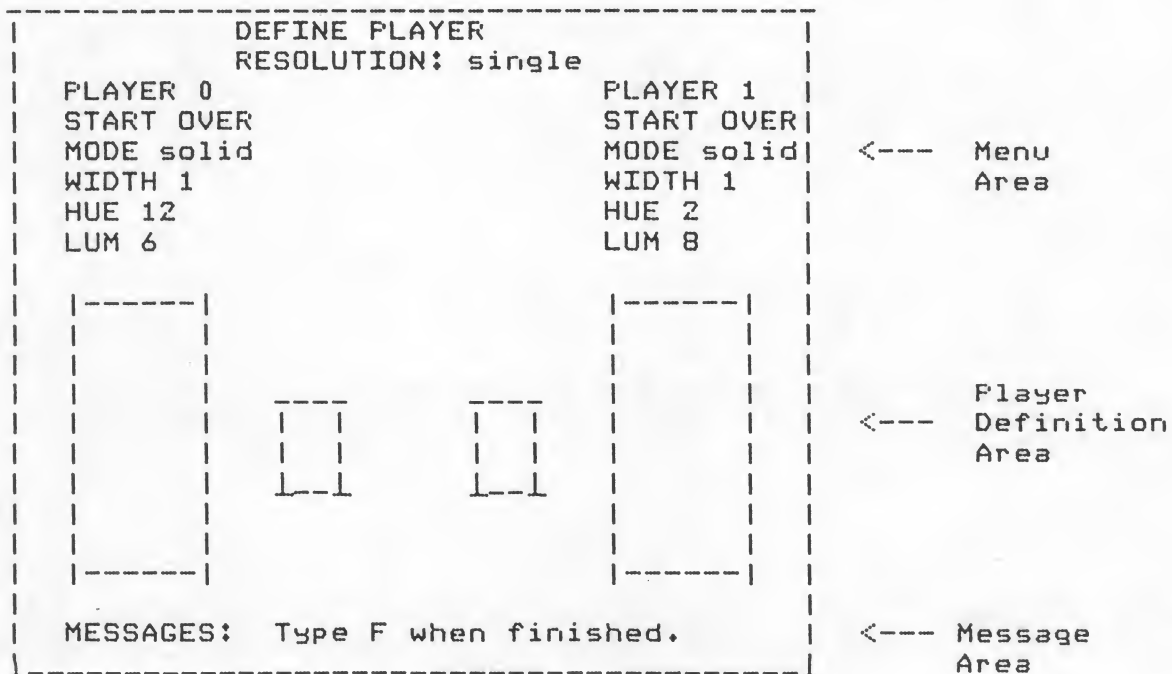
USING PLAYER MAKER

USING THE JOYSTICK OR THE KEYBOARD

Much of the use of PLAYER MAKER involves moving around a flashing cursor on the screen. At times you move the cursor within the menu which is located in the upper portion of the screen. At other times you move the cursor within a block eight units wide which is your player definition area. In either case, you have a choice of one of two methods to move the cursor -- either you can use a joystick or the four arrow keys on the keyboard.

Once you have positioned the cursor on a menu item that interests you, you have a choice once again of using either of two methods to tell PLAYER MAKER that you wish to select that item. You can either press the joystick button or you can hit the space bar on the keyboard. You also have the choice of using the joystick button or the space bar to switch on or switch off pixels when defining your player.

For the purpose of discussion throughout this document, I will assume you are using a joystick and therefore I will refer to moving the stick left, right, up, or down. I also will refer to pressing the joystick button. If you choose not to use a joystick, then use the keyboard arrow keys in place of the stick and press the space bar instead of pressing the joystick button.

THE PLAYER MAKER SCREEN

```
 ------------------------------------------
|          DEFINE PLAYER                   |
|          RESOLUTION: single              |
|   PLAYER 0                 PLAYER 1      |
|   START OVER               START OVER|
|   MODE solid               MODE solid|   <---  Menu
|   WIDTH 1                  WIDTH 1      |        Area
|   HUE 12                   HUE 2        |
|   LUM 6                    LUM 8        |
|                                          |
|   |------|                 |------|   |
|   |      |                 |      |   |
|   |      |                 |      |   |
|   |      |    ----   ----   |      |   |        Player
|   |      |   |  |   |  |    |      |   |   <---  Definition
|   |      |   |  |   |  |    |      |   |        Area
|   |      |   |__|   |__|    |      |   |
|   |      |                 |      |   |
|   |      |                 |      |   |
|   |------|                 |------|   |
|                                          |
|   MESSAGES:  Type F when finished.      |   <---  Message
 ------------------------------------------         Area
```

Most of the time you are using PLAYER MAKER, your screen looks similar to how it appears in the illustration above. The first eight lines of the screen constitute the menu area. The next portion of the screen constitutes the player definition area. Messages are printed in the last two lines. The screen always has a left half and a right half. Each half contains information corresponding to a different player.

THE PLAYER MAKER MENU

The PLAYER MAKER program begins with a flashing cursor at the top of the menu area. If you select the first menu item (DEFINE PLAYER) by pressing the joystick button while the cursor is flashing on that item, you leave the menu and enter the player definition area (see DEFINING A PLAYER). You also can accomplish this by typing a D on the keyboard.

The second menu item (RESOLUTION) cannot be set individually for each player. The Atari Player-Missile Graphics feature requires that all players be the same resolution. To change the resolution of your players (from single to double or vice versa), position the cursor on RESOLUTION and press the joystick button. The change appears instantly on the screen. Double resolution players are twice as tall vertically as single resolution players.

The cursor can be moved to the left or right side of the screen in order to alter items corresponding to the player on either side of the screen. Each half of the screen has six menu items that are described below.

PLAYER: Each half of the screen is used to define a particular player. This menu item shows the player's number which can be a 0, 1, 2, or 3. Press the joystick button while the cursor is positioned on PLAYER to retrieve the player you wish to work on. All current information corresponding to the new player automatically appears on the screen. Each time you press the button, the player on that half of the screen changes. Since the player on the other side of the screen will not be duplicated, pressing the button three times will bring you back to the first player displayed.

START OVER: Use this command cautiously. When the button is pressed, the player that appears below is reinitialized to a clean slate on which you can redefine that player. Thus, if you do not like the design of any given player, you can start over with this command. Before your player is destroyed, PLAYER MAKER asks (in the message window) if you are sure you want to reconstruct this player. If you wish to clear the player then type Y for YES. (See MODE).

MODE: A player image is created by either illuminating or turning off pixels of light (little blocks of light) to make an object of a desired shape. For some images it is easiest to create the player by starting with a blank background and turning on some blocks of light. This mode of operation is called the "clear" mode. In most cases, however, it is easier to start with a block of light and turn off certain squares, leaving the desired shape behind. To use this technique, select the "solid" mode. By default, all players begin in the "solid" mode as a rectangular block of light 8 pixels

wide and 10 pixels tall. If you wish to start with a blank slate, change the mode to "clear" (using the joystick button) and then select the START OVER command. The mode can be changed at any time without affecting any work you have done on your player. However, as you extend your player in the vertical direction, while in the solid mode, you will gain solid rows of light; if you are in the clear mode, your added rows will be blank. (See DEFINING A PLAYER).

WIDTH: The width of each player can be changed at any time. You always have only 8 blocks of light (pixels) spanning the width of any player, but these pixels can be made thinner or wider. There are three possible widths: the default of 1, double width (2), and quadruple width (4). Position the cursor on WIDTH and press the joystick button, and the width will change from 1 to 2 to 4 and back to 1.

HUE: The hue (color) of your player is changed by pressing the joystick button. Hues range from 0 to 15. If you press the button while a player's hue is 15, it changes back to 0.

LUM: The luminance (brightness) of your player can be increased in increments of two from 0 to 14 and back to 0 by pressing the joystick button.

NOTE: The different colors of the players may be more or less pleasing depending on the color of the background. You can change the background color in the FINISHING UP mode (see page 6).

DEFINING A PLAYER

A player cannot be more than 8 units wide, but it can span up to the entire height of the screen. You are provided with a window 8 units wide which corresponds to an enlarged segment of the player. The actual player appears to the side of the window. Using the joystick you can move a cursor within this window, and by pressing the joystick button you can either illuminate or turn off blocks of light in the window. The corresponding pixel of light on your actual player is then simultaneously turned on or off.

The window always remains 10 units high, but a player can grow vertically from its starting height of 10 units. Press the cursor against the top of the window to make a player taller. If you are in solid mode, you will see a solid line of light added to the top of the player. In clear mode, empty space is added to the top of your player. Your window will not stretch, but what you see in the window will scroll upwards. At any one time you will always see 10 rows of the player in the window, but by pressing the cursor against the window top or bottom, you can scroll the 10 rows you see in the window up or down.

The window will no longer scroll up (or down) when your player's top (or bottom) has reached its limit. These limits are above the top and below the bottom of the screen.

You can jump back and forth between defining the player on the left and right half of your screen by moving your cursor past the right or left edge of each corresponding window. At any time you can move the cursor into the Menu Area by typing the letter M on the keyboard. You can move it back to the Player Definition Area by typing the letter D on the keyboard.

FINISHING UP

Once you have defined all the players you wish to create, type F on the keyboard. All the players you have defined, and their corresponding widths, hues, and luminances, will appear on one screen. By placing the cursor on any of these menu items, you can make any final changes you wish in these attributes. If you are still not satisfied with the shape of a player, you can return to the Define Player mode. You also have the option of changing the color of the background. Once you have set the background to a new color, it will remain that color even if you return to the Define Player mode. You may also choose to save the work you did and end the program. If you choose to end, you can request PLAYER MAKER to print an entire subroutine to disk (which makes it easy to use your players in a BASIC program). If you wish to incorporate your player data into memory with your own routines, you may have only DATA statements printed to disk (these are included in the subroutine).

It is a simple matter to integrate the subroutine generated by PLAYER MAKER into your own program. You communicate with the PLAYER MAKER subroutine through four special variables: PLAYER, XSPOT, YSPOT, and MYPLAYER. Your BASIC program should assign values to PLAYER, XSPOT, and YSPOT, and follow this with the statement "GOSUB MYPLAYER". Examples and explanations of how you accomplish this are shown in the demonstration program that follows.

The value in variable PLAYER (0, 1, 2, or 3) tells the subroutine which of your players you want to move. All players originally are placed off the visible screen. By assigning a value to XSPOT and a value to YSPOT and then calling the subroutine MYPLAYER, you can move your player to the corresponding location on the screen. A value of zero in XSPOT corresponds to the leftmost edge of the screen while a value of 160 is the rightmost edge. YSPOT ranges from zero at the screen top to 100 at the screen bottom for double resolution players and 200 at the bottom for single resolution players.

A player is positioned at (XSPOT, YSPOT) by placing its upper leftmost corner at the corresponding screen location. Thus putting a player at (0, 0) places its entire image on the upper leftmost corner of the screen while putting it at (160, 200) places all but the player's corner off the bottom and right edge of the screen. For this reason one will often want to begin by placing a player at say (-20, 0) where it would be just off the screen's left edge. By increasing XSPOT (and calling the MYPLAYER subroutine), the player appears to slowly move onto the screen.

Avoid using variable names which begin with XX in your BASIC program since such names are used by the subroutine. Also, do not use line number 1 or line numbers above 30500. Once you have typed in your program (or retrieved it using the LOAD command), you must merge the PLAYER MAKER subroutine to it. To do this, type the direct command (no line number) ENTER "D:filename" where "filename" is the file in which you told PLAYER MAKER to store your subroutine. You can now resave your entire program including the subroutine by typing SAVE "D:filename". Then, at a later date, you can load the entire program into memory with only one LOAD command.

The following is an example, with explanations, of a program which uses two players. This program, along with the subroutine generated by PLAYER MAKER, is stored in a file called DEMO. To see it run, type: RUN "D1:DEMO".

```
110 REM PLAYER MAKER Demo program.
112 REM By Wayne Harvey
114 REM (c) 1983
116 REM
120 DIM X(1),Y(1),ANS$(1)
130 REM
132 REM Put Player 0 to left of screen and Player 1 to right.
134 REM
140 REM **********************************************************
142 POKE 53248,0:POKE 53249,250
144 REM **********************************************************
146 REM
150 REM Turn screen black and turn off cursor.
152 REM Reset color of Player 0 since program changes it.
154 REM Reset collision registers.
156 REM
160 REM **********************************************************
162 GRAPHICS 0:POKE 710,0:POKE 752,1:PRINT " "
164 POKE 704,198:POKE 53278,0
166 REM **********************************************************
168 REM
170 REM Randomly select both X and Y coordinates for both players.
172 REM
180 REM **********************************************************
182 FOR PLAYER=0 TO 1
184 X(PLAYER)=20+INT(110*RND(0))
186 Y(PLAYER)=10+INT(70*RND(0))
188 NEXT PLAYER
190 REM **********************************************************
192 REM
200 REM Change the X and Y coordinates of both players.
202 REM
210 REM **********************************************************
212 XDIR=SGN(X(0)-X(1)):YDIR=SGN(Y(0)-Y(1))
214 X(0)=X(0)+XDIR:Y(0)=Y(0)+YDIR
216 IF X(0)>130 THEN X(0)=129
218 IF X(0)<20 THEN X(0)=21
220 IF Y(0)>80 THEN Y(0)=79
222 IF Y(0)<10 THEN Y(0)=11
224 X(1)=X(1)+INT(4*RND(0))*XDIR:Y(1)=Y(1)+INT(4*RND(0))*YDIR
226 REM **********************************************************
228 REM
230 REM The following two lines are the key to using MYPLAYER.
232 REM In each line the variables XSPOT, YSPOT, and PLAYER
234 REM are assigned values, and then the MYPLAYER
236 REM subroutine is called.
238 REM
240 REM **********************************************************
242 XSPOT=X(0):YSPOT=Y(0):PLAYER=0:GOSUB MYPLAYER
244 XSPOT=X(1):YSPOT=Y(1):PLAYER=1:GOSUB MYPLAYER
246 REM **********************************************************
248 REM
```

```
250 REM The next two lines give some variety to the direction
252 REM Player 0 takes when Player 0 lines up with Player 1.
254 REM
260 REM **************************************************************
262 IF XDIR=0 THEN X(0)=X(0)+1-INT(3*RND(0))
264 IF YDIR=0 THEN Y(0)=Y(0)+1-INT(3*RND(0))
266 REM **************************************************************
268 REM
270 REM If no collision between players, then continue movement.
272 REM
280 REM **************************************************************
282 IF  NOT PEEK(53260) THEN 212
284 REM **************************************************************
286 REM
290 REM A collision occurred so the color of Player 0
292 REM is changed in the following lines.
294 REM
300 REM **************************************************************
302 FOR COLOR=198 TO 207
304 POKE 704,COLOR
306 FOR PAUSE=1 TO 20:NEXT PAUSE
308 NEXT COLOR
310 FOR COLOR=207 TO 63 STEP -16
312 POKE 704,COLOR
314 FOR PAUSE=1 TO 20:NEXT PAUSE
316 NEXT COLOR
318 FOR COLOR=63 TO 50 STEP -1
320 POKE 704,COLOR
322 FOR PAUSE=1 TO 20:NEXT PAUSE
324 NEXT COLOR
326 REM **************************************************************
328 REM
330 PRINT :PRINT "Watch again ";:INPUT ANS$
332 POSITION 2,1:PRINT "                    "
334 IF ANS$="Y" THEN 142
340 REM
342 REM Move players off the screen before ending program.
344 REM
350 REM **************************************************************
352 POKE 53248,0:POKE 53249,0:GRAPHICS 0:END
354 REM **************************************************************
356 REM
```

ADVANCED TECHNICAL INFORMATION

There are some special memory locations that enable users to more fully utilize Player-Missile Graphics. You may wish to experiment with some of the following locations.

Each player has its own color which can be changed by POKING the appropriate location with a value equal to 16*HUE+LUMINANCE. Thus, to give Player 1 a hue of 2 and a luminance of 4, use the BASIC statement POKE 705, 36.

```
704      Color of Player 0.
705      Color of Player 1.
706      Color of Player 2.
707      Color of Player 3.
```

Each player can be normal width, double width, or quadruple width. POKE the appropriate location listed below with a 0, 1, or 3 respectively to get these three different widths. Thus, to make Player 2 quadruple width, use the statement POKE 53258, 3.

```
53256      Width of Player 0.
53257      Width of Player 1.
53258      Width of Player 2.
53259      Width of Player 3.
```

Each player has a memory location which tells the computer where to place the player along the horizontal axis (left to right). Each location can be POKED with a number from 0 to 255 where a 47 would put the player at the left edge of the screen and a 208 would put it at the right edge. There is no need to deal with these position registers if you use the subroutine generated by PLAYER MAKER to put your players on the screen. The subroutine uses the variable XSPOT to change your player's horizontal position.

```
53248      Horizontal position of Player 0.
53249      Horizontal position of Player 1.
53250      Horizontal position of Player 2.
53251      Horizontal position of Player 3.
```

There are a number of registers used to detect a collision on the screen, that is, if a player touched another player or touched an object drawn on the screen. Any of the following locations can be checked in BASIC by seeing if PEEK(location) <> 0. If this comparison is true, then a collision occurred. These collision registers cannot be reset to zero by POKING a zero to them. Rather, you reset them all with the one statement: POKE 53278, 0.

```
53252    Player 0 to Playfield collisions.
53253    Player 1 to Playfield collisions.
53254    Player 2 to Playfield collisions.
53255    Player 3 to Playfield collisions.

53260    Player 0 to Player collisions.
53261    Player 1 to Player collisions.
53262    Player 2 to Player collisions.
53263    Player 3 to Player collisions.
```