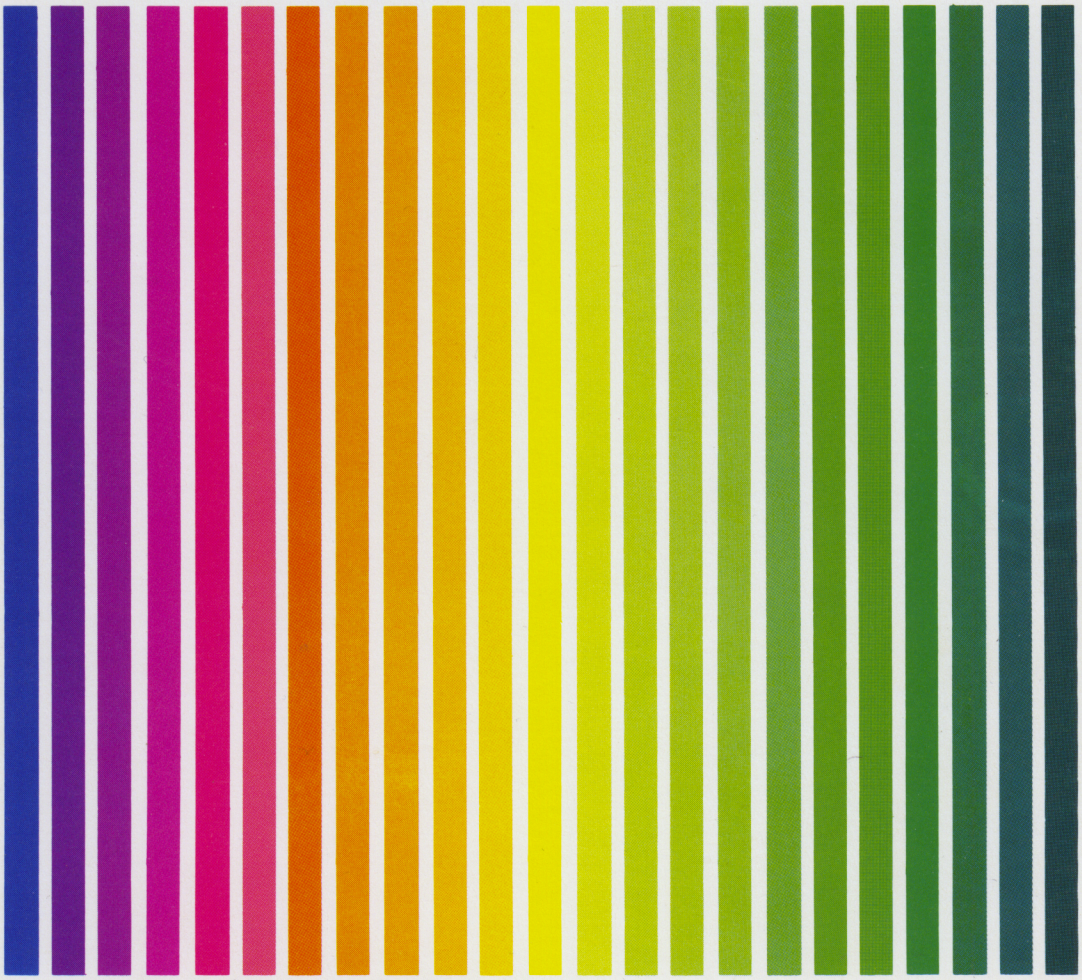
APX ATARI® PROGRAM EXCHANGE



MARCH 1982

GTIA DEMONSTRATION DISKETTE

DISKETTE (APX-20104)
REQUIRES: 24K RAM

User-Written Software for ATARI Home Computers

MARCH 1982

GTIA DEMONSTRATION DISKETTE

DISKETTE (APX-20104)
REQUIRES: 24K RAM

GTIA DEMONSTRATION DISKETTE

TRADEMARKS OF ATARI

The following are trademarks of Atari, Inc.

ATARI®
ATARI 400™ Home Computer
ATARI 800™ Home Computer
ATARI 410™ Program Recorder
ATARI 810™ Disk Drive
ATARI 820™ 40-Column Printer
ATARI 822™ Thermal Printer
ATARI 825™ 80-Column Printer
ATARI 830™ Acoustic Modem
ATARI 850™ Interface Module

****

Distributed by

The ATARI Program Exchange
P. O. Box 427
155 Moffett Park Drive, B-1
Sunnyvale, CA 94086

To request an APX Software Catalog, write to the address above, or call toll-free:

800/538-1862 (outside California)
800/672-1850 (within California)

Or call our Sales number, 408/745-5535.

****

CONTENTS

INTRODUCTION . . . 1

Overview . . . 1
Required accessories . . . 1

GETTING STARTED . . . 2

Loading the program into computer memory . . . 2
Files on the diskette . . . 2

DESCRIPTION OF THE GTIA CHIP . . . 3

Graphics mode 9 . . . 4
Graphics mode 11 . . . 5
Graphics mode 10 . . . 6
CTIA/GTIA compatibility . . . 7

HOW TO USE THE GTIA . . . 9

Resources . . . 9
Program explanations . . . 9
Grahics mode 9--Program: BRASS . . . 9
Graphics mode 11--Program: GZAP . . . 11
Graphics mode 10--Program: LOGO . . . 13

GETTING STARTED

LOADING THE PROGRAM INTO COMPUTER MEMORY

1. Insert the ATARI BASIC Language Cartridge in the cartridge slot of your computer.

2. Turn on your disk drive.

3. When the BUSY light goes out, open the disk drive door and insert the program diskette with the label in the lower right-hand corner nearest to you. Close the door.

c. Turn on your computer and TV set.

d. When the READY prompt displays on your TV screen, type RUN "D:<filename>" and press the RETURN key. (Replace <filename> with the name of the program you want to run. See the list of filenames below.) If you're using more than one disk drive, remember to follow the device initial (D) with the number of the drive containing the program diskette (e.g., RUN "D2:<filename>" for disk drive two). The program will load into computer memory and start.

FILES ON THE DISKETTE

The diskette contains the following demonstration program files written in ATARI BASIC:

ROTATE7.LST	CORRIDOR
LOGO	SINK
ROTATE8.LST	FUNAL
POLES	WHIRL
STRIPE	GRENHOLE
DIZZY	MELON
HYPNO	MELONS
GZAP	SMELONS
ROLL	SAS
ESCAPE	BALL
BRASS	RING
GVEIL	GTIATST
GCONE	BARS

DESCRIPTION OF THE GTIA CHIP

Note. The following discussion is Appendix E of De Re ATARI (APX-90008, $19.95)

The GTIA is a new television interface adaptor chip. It is a CTIA display chip with a few more features. It provides three additional modes of interpreting information coming from the ANTIC chip, which is dedicated to the television display. ANTIC doesn't require a new mode to talk to GTIA; instead, it uses a high-resolution mode, $F. GTIA is completely upward compatible with the CTIA. A brief summary of CTIA's features follows so that the differences between CTIA and GTIA can be presented.

The CTIA is designed to display data on the television screen. It displays the playfield, players, and missiles, and detects any overlaps or collisions between objects on the screen. CTIA interprets the data supplied by ANTIC according to six text modes and eight graphics modes. In a static display, it uses the data from ANTIC to display hue and luminance as defined in one of four color registers. The GTIA expands this capability to use all nine color registers or sixteen hues with one luminance, or sixteen luminances of one hue in a static display.

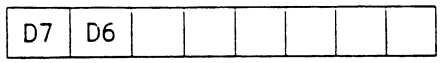
The three graphics modes of GTIA are three new interpretations of ANTIC mode $F, a hi-res mode. All three modes affect the playfield only. Players and missiles can still be added to introduce new hues or luminances or to use the same colors and luminances in more than one way. All displays of hues and luminances can still be changed on the fly with display list interrupts. The GTIA uses four bits of data from ANTIC for each pixel, called the pixel data. Each pixel is two color clocks wide and one scan line high. Thus, the pixels are roughly four times wider than their height. The display has a resolution of 80 pixels across by 192 down. Each line then requires 320 bits of 40 bytes of memory, the same number of bytes used in ANTIC mode $F. Therefore, for a program to run the GTIA modes, it must have at least 8K of free RAM for the display.

The GTIA modes are selected by the priority register, PRIOR. PRIOR is shadowed at location $26F hex by the Operating System and is located at D01B hex in the chip. Bits D6 and D7 are the controlling bits. When neither is set, there are no GTIA modes and GTIA operates just like CTIA. When D7 is 0 and D6 is 1, graphics mode 9 is specified, which allows sixteen different luminances of the same hue. Remember the pixel data supplied by ANTIC is four bits wide, which means sixteen different values can be represented. Players and missiles can be used in this mode to introduce additional hues. When D7 is 1 and D6 is 0, graphics mode 10 is specified. This mode gives nine colors in the display by using the four playfield color registers, plus the four player-missile color registers, plus the one background color register. When players are used in this mode, the four player-missile color registers are used for them also. When D7 is 1 and D6 is 1, graphics mode 11 is specified. This mode gives 16 hues with the same luminance again because sixteen different values can be represented by four bits. Players and missiles can be used in this mode to introduce difference luminances. (Note. A later section of this manual contains line-by-line explanations of three demonstration program, one illustrating each of these graphics modes--9, 10, and 11.)

Setting up the new GTIA modes is as simple as setting up the present modes in CTIA. To implement the modes from ATARI BASIC, use the commands GRAPHICS 9, GRAPHICS 10, and GRAPHICS 11 for modes 9, 10, and 11. In assembly language, selecting one of these modes is identical to opening the screen for any of the other modes. If you're building your

own display list, then PRIOR must be set to select the correct modes, as shown in Figure 1.

PRIOR



D7	D6	OPTION	
0	0	No GTIA modes (CTIA operation)	(Modes 0-8)
0	1	1 Hue, 16 Luminances	(Mode 9)
1	0	9 Hues/Luminances	(Mode 10)
1	1	16 Hues, 1 Luminances	(Mode 11)

Figure 1
Bit Pattern In PRIOR selects GTIA

GRAPHICS MODE 9

Mode 9 produces up to sixteen difference luminances of the same hue. ANTIC provides the pixel data that selects one of sixteen different luminances. The background color register provides the hue. In ATARI BASIC, this is done using the SETCOLOR command to set the hue value in the upper nybble of the background color register, and to set the luminance value in the lower nubble to all zeros. The format of the command is:

```
SETCOLOR 4,hue value,0
```

where 4 specifies the background color register, "hue value" sets the hue and can be anything from 0 to 15, and 0 sets the luminance part of the register to zero. This has to be done because the pixel data from ANTIC is then logically OR'ed with the lower nybble of the background color register to set the luminance that appears on the screen. The COLOR command is then used to select luminances for drawing on the screen by using values from 0 to 15 as its parameter. So, an ATARI BASIC program will include at least the following statements to use graphics mode 9:

```
GRAPHICS 9          to specify mode 9
SETCOLOR 4,12,0     to initialize the background color
                    register to some hue--green here
FOR I=0 TO 15       some method where the
COLOR I             COLOR command is used
PLOT 4,I+10         to vary luminance
NEXT I
```

In assembly language, use the OS shadow for the background color register $2C8 to set the hue in the upper four bits with hex values from $0 to $F. If CIO calls are used, store

the pixel data into the OS register ATACHR, located at $2FB. This selects the luminance with hex values from $0 to $F. If you're maintaining your own display data, then the pixel data goes directly into the left or right half of the display RAM byte.

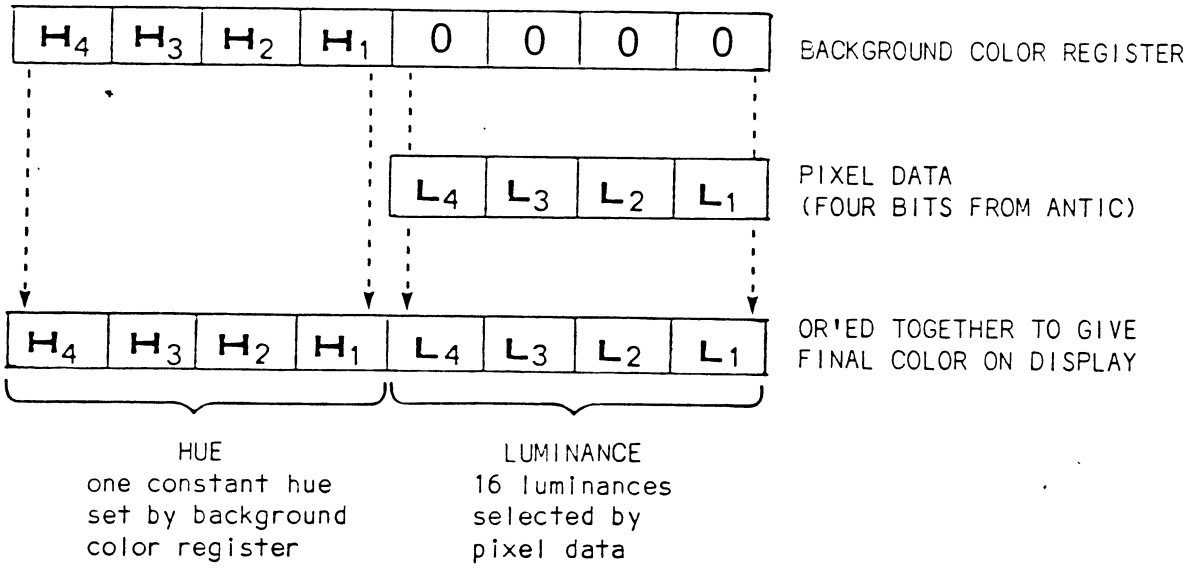


Figure 2
Background Color Register OR'ed with pixel data to give final color.

GRAPHICS MODE 11

Mode 11 is similar to mode 9 except that it provides sixteen different hues, all with the same luminance. Again, ANTIC provides the pixel data to select one of sixteen different hues. In ATARI BASIC, the SETCOLOR command is used to set up the single luminance value in the lower nybble of the background color register, and in the upper nybble, the hue value will be set to all zeros. The format of the command is:

```
SETCOLOR 4,0,luminance value
```

where 4 specifies the background color register, 0 sets the upper nybble to zero, and "luminance value" sets the value of the luminance and can range from 0 to 15. As with the other graphics modes (except mode 9), the first bit of the luminance isn't used, so effectively, only even numbers result in distinct luminances, which gives eight different possible luminances in this mode. The COLOR command is used in this mode to select the various hues by using values from 0 to 15 in its parameter. The pixel data from ANTIC will be logically OR'ed with the upper nybble of the background color register to set the hue part of the value that ultimately generates the color on the screen. So, an ATARI BASIC program using mode 11 will include at least the following statements:

```
GRAPHICS 11       to specify mode 11
SETCOLOR 4,0,12   to initialize the background
                  color register to some luminance,
                  very bright in this case
```

```
FOR I=0 TO 15      some method where the
COLOR I            COLOR command is used
PLOT 4,I+10        to vary the hue
NEXT I
```

In assembly language, use the OS shadow for the background color register $2C8 to set the luminance in the lower four bits with hex values from $0 to $F. If CIO calls are used, store the pixel data into ATACHR, located at $2FB. This selects the hue with hex values from $0 to $F. If you're maintaining your own display data, then the pixel data goes directly into the left or right half of the display RAM byte.

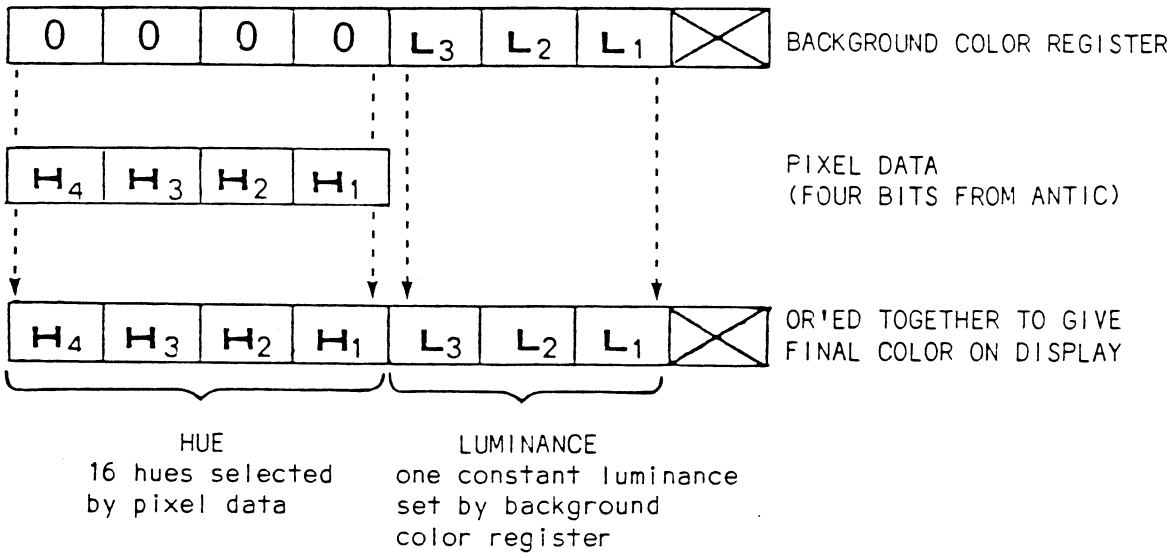


Figure 3
Background Color Register OR'ed with pixel data to give final color.

GRAPHICS MODE 10

Mode 10 allows nine color registers to be used in the playfield at one time. Each color register to be used must be set to some combination of hue and luminance. The pixel data from ANTIC is used in this mode to select one of the color registers for display. In ATARI BASIC, the SETCOLOR command can be used as described in the BASIC Reference Manual to set the colors in the normal background and the four normal playfield registers. (Note. In graphics mode 10, location 704 defines the background color and locations 705-712 define the eight playfield colors.) These can also be set by using the POKE instruction to addresses 708-712, where the four normal playfield registers and the normal background register are located. The POKE instruction must be used to set the four player-missile color registers at locations 704-707. The COLOR command is used to select the color register desired. The only meaingful values for its agrument are 0 to 8. A problem arises with this mode. ANTIC supplies four bits of data per pixel, as it does with graphics modes 9 and 11. This allows for selecting sixteen color registers. However, only nine color registers exist in the hardware. An illegal data value between 9 and 15 will select one of the lower-value color registers. An ATARI BASIC program using mode 10

will include:

1) a GRAPHICS 10 command to specify mode 10;
2a) a set of POKE instructions to put hues and luminances into color registers, or
2b) a combination of SETCOLOR commands and POKE instructions to do that;
3) a COLOR command to select the desired color register.

In assembly language, store the pixel data in ATACHR ($2FB) or directly into the display RAM byte, as in modes 9 and 11. In this mode, the pixel data can range from 0 to 8 and it selects one of the nine color registers, as shown in Figure 4.

COLOR STATEMENT VALUE	COLOR REGISTER USED	OS SHADOW
0	D012	2C0
1	D013	2C1
2	D014	2C2
3	D015	2C3
4	D016	2C4
5	D017	2C5
6	D018	2C6
7	D019	2C7
8	D01A	2C8

Figure 4
Color Register Numbers and Locations
and COLOR Command

CTIA/GTIA COMPATIBILITY

An important question arises in conjunction with GTIA concerning compatibility. GTIA is fully upward compatible with the CTIA and all software that runs on a CTIA system will run the same way on a system with GTIA. This compatibility means you still have the full use of players and missiles, still have collision and overlap detection and display list interrupts. The GTIA graphics modes are fully supported by the OS and all graphics commands and utilities that run in the CTIA modes can be used in GTIA modes.

More colors are available to display at one time on the screen. Sixteen color changes can occur on one line totally independent of processor intervention. This is actually better than what could be done with display list interrupts, which could give at most only twelve color changes per line. Much finer contour and depth can be represented using the shading available in mode 9. This improvement means three-dimensional graphics can be realistically displayed.

On the other hand, there are some disadvantages. GTIA modes are map modes; there can be no text displayed in these modes. A custom display list must be used to switch to a mode that supports character displays. The GTIA pixel is a long, skinny horizontal rectangle (4:1 width to height) and doesn't represent curved lines well. Because each pixel uses

four bits of information, GTIA requires nearly 8K of free memory to operate. Although it is upward compatible, it is not downward compatible. Thus, programs using GTIA modes will not produce correct displays on computers having CTIA chips. They may well be recognizable, but they won't be colorful. There is no way currently for a program to determine whether or not a GTIA is present in a system. Finally, color artifacts produced by a GTIA-chip system won't be identical to the color artifacts produced on the same television with a CTIA-chip system.

HOW TO USE THE GTIA

RESOURCES

You can obtain general information about using color and sound in ATARI Home Computers in the following works.

ATARI BASIC Reference Manual (C015307)

De Re ATARI (APX-90008)

ATARI Personal Computer System Operating System User's Manual and Harware Manual (C016555)

PROGRAM EXPLANATIONS

This section examines three of the programs on the demonstration diskette. Each program uses a different graphics mode--modes 9, 10, and 11. Program description include a program listing, an explanation of how the graphics are used to create the effect, and other useful information.

GRAPHICS MODE 9--Program: BRASS
(1 color; 16 luminances)

Use graphics mode 9 for static displays requiring subtle shading. It's especially useful for representing curved surfaces.

Program Listing

```
10 GRAPHICS 9
15 SETCOLOR 4,15,0
20 FOR Y=55 TO 0 STEP -10
30 FOR X=0 TO 24
40 C=X:IF X>11 THEN C=24-X
50 Z=Y+(X)
55 D=INT (SQR(144-(X-12)*(X-12))/2)
57 COLOR 15-C
58 PLOT Z,Y+7-D
60 DRAWTO Z,Y+7+D
70 COLOR C
80 DRAWTO Z,180-Y+D
180 NEXT X
190 NEXT Y
200 GOTO 200
```

Explanation of listing

Line 10
Set up 1 hue, 16 luminance mode (In graphics mode 9, the background color is in location 712; the value defaults to 0 (black); you can specify hue and luminance of the background using SETCOLOR 4)

Line 15
Set hue of entire display to 15 (orange)

Lines 20-55
Determine position of colored strip to be drawn

Lines 57-60
Draw top of strip in one luminance

Lines 70-80
Draw bottom of strip in different luminance

Lines 180-190
Draw another strip

Line 200
Endless loop executes to keep picture on screen

PROGRAMS USING GRAPHICS MODE 9

BRASS
GVEIL
GCONE
CORRIDOR
GTIATST (uses graphics modes 9, 10, and 11)

GRAPHICS MODE 11--Program: GZAP
(16 colors, 1 luminance)

Use graphics mode 11 to get many different colors in a static display. You could add delays to this program to see the sequence of events more easily.

Program Listing

```
10 GRAPHICS 11
20 OLDX=0
30 OLDY=0
50 X=INT(RND(0)*80)
60 Y=INT(RND(0)*192)
70 C=INT(RND(0)*15+1)
75 COLOR C
77 PLOT OLDX,OLDY
80 DRAWTO X,Y
90 COLOR 0
100 PLOT OLDX,OLDY
110 DRAWTO X,Y
120 COLOR C
130 PLOT OLDX,OLDY
140 OLDX=X:OLDY=Y
150 GOTO 50
```

Program Explanation

Line 10
Set up 16 hues with the same luminance (In graphics mode 11, the background color is in location 712; the value defaults to 6, which implies the background will be black and the luminance of all colors will be 6)

Lines 20-30
Initialize position of light beam

Lines 50-60
Generate random destination of light beam

Lines 70-75
Choose color at random

Line 77
Plot a point of color at the beam's old position

Line 80
Draw a line to the new position

Line 90
Switch to the background color (black)

Line 100
Erase the old position by plotting a point in the background color

Line 110
Erase the line by drawing over it with the background color.

Line 120
Switch back to the beam's color

Line 130
Plot a point of color at the beam's old position. This will leave a trail of colored dots as the beam bounces around.

Lines 140 and 150
Make the "new" position the old position (140), and shoot the light beam at a new point (150).

PROGRAMS USING GRAPHICS MODE 11

GZAP
GTIATST (uses graphics modes 9, 10, and 11)

GRAPHICS MODE 10--Program: LOGO
(9 colors, each with its own hue & luminance)

Use graphics mode 10 to show motion by drawing an object, and then rotating colors through the color registers. You can display 9 colors, each with its own hue and luminance. You can use this mode effectively to portray rotating objects (see the program named FUNAL). Note. In graphics mode 10, location 704 holds the background color and luminance, and 712 is used to plot one of the remaining colors.

Program Listing

```
1 TRAP 80
2 DIM A$(30)
3 A$="h",A=BA H AAAAAA"
10 GRAPHICS 10
15 FOR I=0 TO 8:READ A:POKE 704+I,A*16+6:NEXT I
17 POKE 704,0
20 COLR=1:Y=1
30 FOR X=10 TO 69
40 COLOR COLR
50 PLOT X,141-Y:DRAWTO X,191-Y
52 PLOT 79-X,141-Y:DRAWTO 79-X,191-Y
54 FOR Q=36 TO 43
55 PLOT Q,191-Y:DRAWTO Q,141-Y
56 NEXT Q
60 Y=Y*1.23
65 COLR=COLR+1:IF COLR>8 THEN COLR=1
70 NEXT X
80 X=USR(ADR(A$))
90 FOR J=1 TO 12:NEXT J
100 GOTO 80
1000 DATA 0,1,2,3,4,5,6,7,8
```

Program Explanation

Line 1

If any plotting should get out of range, go immediately to the routine that rotates the colors.

Lines 2-3

Dimension a string and define a machine-language routine to rotate the colors in the color registers through locations 705-712 (see ROTATE8.LST) (Note. The string in SMELONS is slightly different. It rotates the colors in the color registers through locations 705-711 (see ROTATE7.LST, which contains a machine-language string you can use in your programs to rotate colors through the color registers). This routine lets one area of the screen (the inside of the melon) remain static while other areas appear to move (the outer melon shell). The disassembled routine to rotate colors through locations 705-712 is as follows:

```
        LDX #$00
        LDY $02C1
LOOP    LDA $02C2,X
```

```
STA $02C1,X
INX
CPX #$08
BCC LOOP
STY $02C8
RTS
```

Line 10
Set up nine colors, each with its own hue and luminance. (The background, location 704, normally defaults to 0--black--if graphics mode 10 is called alone.)

Line 15
Initialize color registers using colors from the DATA statement at line 1000 and a luminance of 6.

Line 17
Re-set background to black. (You'll notice the screen flash gray for a moment when the program is run. This flash occurs because the background, location 704, was originally set to a color of 0, with a luminance of 6.)

Line 20
Initialize the position of color strips and color to draw them with.

Lines 30-56
Draw a portion of the logo.

Line 60
Update the position of the color strips.

Line 65
Change the color.

Line 70
Draw another set of strips.

Line 80
Call the machine-language program in A$ to rotate the colors once.

Line 90
Time delay

Line 100
Rotate the colors again.

Line 1000
Data for initial values of color registers

PROGRAMS USING GRAPHICS MODE 10

LOGO
POLES
STRIPE
DIZZY
HYPNO
ROLL
ESCAPE
SINK
FUNAL
WHIRL
GRENHOLE
MELON
MELONS
SMELONS
SAS
BALL
RING
BARS
GTIATST (uses graphics modes 9, 10, and 11)

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

We're interested in your experiences with APX programs and documentation, both favorable and unfavorable. Many software authors are willing and eager to improve their programs if they know what users want. And, of course, we want to know about any bugs that slipped by us, so that the software author can fix them. We also want to know whether our documentation is meeting your needs. You are our best source for suggesting improvements! Please help us by taking a moment to fill in this review sheet. Fold the sheet in thirds and seal it so that the address on the bottom of the back becomes the envelope front. Thank you for helping us!

1. Name and APX number of program _____

2. If you have problems using the program, please describe them here.

_____

_____

_____

3. What do you especially like about this program?

_____

_____

_____

4. What do you think the program's weaknesses are?

_____

_____

_____

5. How can the catalog description be more accurate and/or comprehensive?

_____

_____

6. On a scale of 1 to 10, 1 being "poor" and 10 being "excellent", please rate the following aspects of this program?

_____ Easy to use
_____ User-oriented (e.g., menus, prompts, clear language)
_____ Enjoyable
_____ Self-instructive
_____ Useful (non-game software)
_____ Imaginative graphics and sound

7. Describe any technical errors you found in the user instructions (please give page numbers).

8. What did you especially like about the user instructions?

9. What revisions or additions would improve these instructions?

10. On a scale of 1 to 10, 1 representing "poor" and 10 representing "excellent", how would you rate the user instructions and why?

11. Other comments about the software or user instructions:

STAMP

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]