

GS/FX

Three stunning new ways to look at graphics on the IIGS

The new Super Hi-Res graphics mode on the Apple IIGS brings a dramatic improvement in color and resolution to the graphics modes available on the older Apple II's. Because of this advanced graphics mode, you can add all kinds of special effects to your programs and GS/FX will help you do so. You can, for example, load a Super Hi-Res picture, present the viewer with a black screen, and slowly have your picture fade into view, color by color. You can also cause Super Hi-Res pictures to fade out to black or "fold down" the screen color by color (see Figure 1). The program will even turn the Super Hi-Res display mode on for you. All this from a BASIC program!

USING GS/FX

GS/FX is a machine language utility that, once loaded, can be CALLED from BASIC. It is relocatable, so it will work at any address. To use the program, just BLOAD it at the required address. You need to load the program only once, provided it is not overwritten, but you can CALL it as many times as you want.

The default location for BLOADing GS/FX is \$2000 (8192). Before you CALL GS/FX, use a POKE to determine which of the three special effects you wish GS/FX to perform. POKE 26,0 causes the picture to fade in from black, one color at a time; POKE 26,1 causes the picture to fade out to black, one color at a time; and POKE 26,2 will give you the scroll in effect, similar to the fade-in but the fade rolls down from the top of the screen to the bottom. You can also set the speed of the effect. Use the command POKE 27, speed where 0 is the fastest and 255 is the slowest speed. A speed of 80 is pretty fast; a speed less than 20 or so is too fast for normal viewing.

For example, with a Super Hi-Res picture in memory, use the following commands:

```
BLOAD GS.FX, A8192
POKE 26, 0
POKE 27, 80
CALL 8192
```

Note that CALLing GS/FX displays the Super Hi-Res screen only while the effect is in progress. Use the following command to view the Super Hi-Res screen:

```
POKE 49193, 193
```

Use this to turn off the Super Hi-Res screen:

```
POKE 49193, 65
```

WARNING: Do not POKE any other values into the above location, 49193. Doing so can have disastrous results.

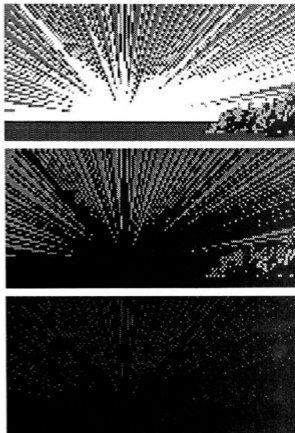


Figure 1: GS/FX Fade Out to Black

The program GS.FX.DEMO (Listing 3) is a short Applesoft program to show you how to use GS/FX. If you do not have a Super Hi-Res picture handy, the demo will use PIC.CREATE (Listing 4) to make one for you.

ENTERING THE PROGRAMS

If you have an assembler, enter the source code from Listing 1 and save the object code as GS.FX. If you don't have an assembler, use the hex codes in Listing 2 and save the file with

```
BSAVE GS.FX, A$2000, L$1B4
```

Enter the Applesoft program in Listing 3 and save it with

```
SAVE GS.FX.DEMO
```

If you have an assembler, enter the source code from Listing 4 and save the object code as PIC.CREATE. If you don't have an assembler, enter the hex codes from Listing 5 and save with

```
BSAVE PIC.CREATE, A$300, L$52
```

For help with entering *Nibble* listings, see the Typing Tips section.

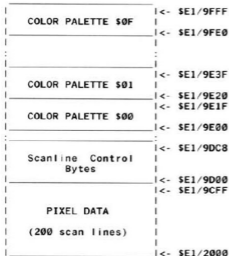


Figure 2: Super Hi-Res Graphics Display Buffer

HOW THE PROGRAM WORKS

There are three major components of a Super Hi-Res picture. This layout is illustrated in Figure 2.

In its standard mode, each Super Hi-Res screen is made up of 200 horizontal lines of dots (pixels) called scan lines. Each of these scan lines has an associated scan line control byte (SCB) which, among other things, identifies which of 16 color palettes is used for the pixels in that line. By changing one byte in a given color palette, you effectively change the color of every screen pixel that uses that palette color. This makes for very fast color changes and is the key to using this program.

Before beginning a discussion of the program itself, it is worth mentioning a new opcode available on the 65816 chip in the IIGS. The PER opcode puts a specified address on the stack. It is possible to use this opcode to set the stack up in the same manner as a JSR would. I have used this opcode to simulate JSR's so that the program will run successfully at any address. This useful technique is discussed in greater detail by Sandy Mossberg in the April, 1987, issue of *Apple Assembly Line* (pp. 14-18).

Now for a discussion of the program itself. The fade in portion is described here. It is assumed that the picture to be faded uses palette zero (the first palette). This is generally the case and so no problems should occur. Palette \$0F (the last palette) is used as the effects palette by the program.

All colors in palette \$0F are set to black; then all SCB's are pointed at this all black palette. The blue and green components of each color are increased until they reach the maximum level for that color. Each time through this loop, the pause routine is executed. After the blue and green color components are at their maximum value, the program increases the red components of each color in a similar manner until all colors have been faded in.

After all colors have faded in from black, the program restores the original screen display state and exits.

LISTING 1: GS.FX Source Code

```

1 *
2 * GS.FX Source Code
3 * By Peter Stubbs
4 * Copyright(C) 1988
5 * MicroSPARC, Inc.
6 * Concord, MA 01742
7 *
8 * Assembler: MERLIN 14
9 *
10
11         TR      ADR
12         XC
13         XC
14         ORG     $2000      ;Note: program as written
15                          ;is relocatable and will
16                          ;run at any address
17
18 NEWVIDEO EQU $C029      ;SuperRes/linear softswitch
19 WAIT     EQU $FCAB      ;ROM time delay routine
20 PALPTR   EQU $0         ;3 byte pointer to palette
21 PALZERO  EQU $3         ;pointer to main palette
22 MAXBLUE  EQU $6         ;max. value of blue
23 ENDVAL1  EQU $7         ;max. value of blue/green
24 ENDVAL2  EQU $8         ;max. value of red
25 YIDSAVE  EQU $9         ;save video register
26 CODE    EQU $1A        ;function code poked here
27 DELAY    EQU $1B        ;delay value
28
29 START    SEC
30         XCE              ;turn emulation mode on
31         SEP     $130
32
33 *
34 * common code for all three functions
35 *
36
37         LDA     NEWVIDEO  ;get setting of NEWVIDEO
38         STA     YIDSAVE   ;save for later
39         ORA     #$01000000 ;turn on linear addressing
40         STA     NEWVIDEO  ;by setting bit 6
41
42 * Set up the pointers to our color palettes
43
44         LDA     < $E19F00 ;Set up PALPTR to point at
45         STA     PALPTR    ;location of palette F
46         LDA     > $E19F00
47         STA     PALPTR+1
48         LDA     #' $E19F00
49         STA     PALPTR+2
50         LDA     < $E19E00 ;Set up PALZERO to point at
51         STA     PALZERO   ;location of palette zero

```

```

52         LDA     > $E19E00
53         STA     PALZERO+1
54         LDA     #' $E19E00
55         STA     PALZERO+2
56
57 * Branch here depending on which routine you want
58 JUMPTBL LDA CODE
59         BNE     1
60         BRA     INFX      ;fade in
61 1       CMP     #1
62         BNE     2
63         BRA     OUTFX     ;fade out
64 2       CMP     #2
65         BNE     3
66         BRG     SCROLLFX  ;scroll in
67
68 3       JSR     $FF3A      ;incorrect code, so ring bell
69         BRA     EXIT      ;and exit
70
71 *
72 INFX    = *              ;fade-in code starts here
73 *
74
75 CLRPAL LDA #$0          ;Color to use to clear the
76         TAX              ;palettes (00=black)
77 LOOP0   STAL $E19FE0.X  ;Clears Palette 50F
78         INX
79         CPX     #$20      ;are we at end of palette ?
80         BNE     LOOP0     ;if not, zero next byte
81         LDY     #$1E      ;Start fade from last color
82
83 FADE1T  PER RETURN1-1   ;relocatable version
84         BRG     SETUP2   ;of JSR SETUP2
85 RETURN1 LDA ENDVAL1     ;Get max val of blue/green
86         BEQ     DONE2    ;if = 0 is no need to fade
87         LDX     #$FF     ;so X=0 after next line
88 COLORUP INX            ;turn blue intensity up 1
89         TXA              ;(changes bits 3-0 only)
90         STA     [PALPTR].Y ;Set the new intensity
91                          ;in the color table
92         PER     RETURN3-1 ;relocatable version
93         BRG     PAUSE    ;of JSR PAUSE
94 RETURN3 CPX MAXBLUE     ;at maximum blue value yet ?
95         BNE     COLORUP  ;if not loop back
96         LDA     ENDVAL1   ;else check green component
97         CMP     #$10      ;is this lowest intensity?
98         BLT     DONE2     ;if yes skip add'n for new
99 DONE1   TXA              ;b-mp bits 7-4 (green bits)
100 COLORUP2 CLC           ;up one intensity by adding
101         ADC     #$10      ;#$00010000 ($10)
102         STA     [PALPTR].Y ;save new value
103
104         PER     RETURN4-1 ;relocatable version
105         BRG     PAUSE    ;of JSR PAUSE

```

LISTING 1: GS.FX Source Code

```

100 RETURN4 BNE ENDVAL1 ;at max val for blue/green ?
107 END4 CMP COLORUP2 ;if not cycle again
100 DONE2 LDA #FFF ;else start on red byte
109 COLORUP3 ENK
110 TXA
111 STA ;To point at high byte
112 INY ;zero high byte
113 DEY ;restore X
114
115 PER RETURNS-1 ;relocatable version
116 INK PAUSE ;of JSR PAUSE
117 RETURN3 CPX ENDVAL2 ;at max for ENH color?
118 BNE COLORUP3 ;if not, loop back
119 DEY ;move to next color down
120 DEY ;by moving two bytes down
121 BPL FADEIT ;cycle until color 0 done
122
123 EXIT LDA VIDEAVE ;Restore original NEWVIDEO
124 STA NEWVIDEO
125 HTS ;And exit
126
127 *
128 OUTFX = * ;fade-out effect starts here
129 *
130
131 COPYPAL LDA #00 ;Start at color zero
132 COLORP LDA [PALZER0] Y ;get blue/green byte from
133 TXA ;palette zero and put it in
134 STA SE19F0B.X ;same position in pal. SF
135 INY
136 LDA [PALZER0] Y ;Get the red color byte
137 TXA ;make X = Y
138 STA SE19F0B.X ;and copy red color byte
139 INY
140 CPY #320 ;Are we at the end
141 BNE SLOOP0 ;no
142 LDA #00 ;Start fade from 1st color
143 OFADEIT BRE GRETURN-1 ;relocatable version
144 BRL SETUP2 ;of JSR SETUP2
145 GRETURN1 LDA ENDVAL2 ;Get max val of red byte
146 OCLRDNW3 TXA
147 INY ;To point at high byte
148 STA ;update red color byte
149 DEY ;point X back at low byte
150 BRE GRETURN-1 ;relocatable version
151 BRL PAUSE ;of JSR PAUSE
152 GRETURN2 DEY ;fade red intensity down 1
153 CPX #SF ;wrapped from 0 F
154 BNE OCLRDNW3 ;if not, reloop
155 LDA ENDVAL1 ;get max blue/green byte
156 TXA
157 OCLRDNW2 STA [PALPTR] Y ;update green color byte
158 BRE GRETURN-1 ;relocatable version
159 BRL PAUSE ;of JSR PAUSE
160 OCLRDNW1 CMP #310 ;lowest intensity ?
161 BLT GOONE ;if yes, skip the subtract
162 SEC ;drop bits 7-4 (green) by 1
163 SBC #310 ;by subtracting #00001000
164 BRA OCLRDNW2 ;(cleans blue unchanged)
165 OONE LDA MAXBLUE ;get maximum blue intensity
166 OCLRDNW TXA
167 STA [PALPTR] Y ;update the blue/green
168 BRE GRETURN-1 ;relocatable version
169 BRL PAUSE ;of JSR PAUSE
170 GRETURN4 DEY ;turn blue intensity down 1
171 CMP #SF ;wrapped from 0 F
172 BNE OCLRDNW ;no, so reloop
173 INY ;move to next color up
174 INY ;by moving two bytes up
175 CPY #320 ;at end of the palette ?
176 BNE OFADEIT ;if not, loop back
177
178 ORTS LDA VIDEAVE ;
179 STA NEWVIDEO
180
181 RTS
182 *
183 SCROLLFX = * ;scroll code starts here
184 *
185
186 SCRLPAL LDA #0 ;Color to use to clear the
187 TXA ;palettes (00=black)
188 SLOOP0 STA SE19F0B.X ;Clear Palette SF
189 STA SE19F0C.X ;Clear Palette SE
190 INK
191 CPX #320 ;At end of palette ?
192 BNE SLOOP0 ;if not zero move byte
193 LDA #0 ;Start with palette 00E
194 BRE SRETURN-1 ;relocatable version
195 BRL SCRLS0B ;of JSR CLRS0B
196 SRETURN1 LDA #0 ;in color 00F (at
197 SLOOP1 DEY ;byte 0F in palette & copy

```

END OF LISTING 1

LISTING 2: GS.FX

Start: 2000 Length: 184

BC 2000:38	FB E2	30 AD	29 C0	85	F9 2040:AA	9F E0	9F E1	E8 E0	20	1A 2008:A0	00 87	03 88	9F E0	9F
F9 2008:09	09 40	8D 29	C0 A9	E0	7E 2048:D0	F7 A0	1E 62	00 82	82	21 200A:01	C8 87	03 88	9F E0	9F
28 2010:85	00 A9	00 85	03 A9	9E	A7 2050:35	01 A5	07 F0	26 A2	FF	27 200B:E1	C8 80	20 D0	EC A0	00
68 2018:85	00 A9	00 85	03 A9	9E	37 2058:E8	84 90	00 62	02 00	82	E5 200C:62	02 00	82 00	88 00	A6 08
28 2018:85	00 A9	00 85	03 A9	9E	17 2060:49	01 E4	06 D0	F2 A5	07	DC 200B:8A	C8 87	D0 81	62 02	00
B1 2020:85	04 A9	E1 85	05 A5	1A	C6 2068:C9	10 90	10 8A	18 69	10	4A 200C:02	F8 00	CA E0	FF D0	F0
01 2028:D0	02 80	12 C9	01 D0	02	03 2070:97	00 62	02 00	82 33	01	5B 200C:A6	07 8A	97 00	62 02	00
65 2030:80	66 C9	02 D0	03 82	BF	9C 2078:C5	07 D0	F1 A2	FF E8	8A	3E 200D:82	D8 00	C9 10	90 05	38
58 2038:00	20 3A	F0 00	54 A9	00	0E 2080:C8	97 00	88 62	02 00	82	E9 200E:E9	10 80	EF A6	06 9A	97
					2F 2088:21	01 E4	08 D0	F0 88	88	81 200E:00	62 02	00 82	C4 00	CA
					36 2090:10	8A A5	09 D0	29 C0	60					

LISTING 2: GS.FX

```

16 20E8:E0 FF D0 F2 C8 C0 20
96 20F9:D0 BE A5 09 8D 29 C0 60
C9 20F8:A9 00 AA 9F E0 9F E1 9F
7D 2100:C0 9F E1 F8 F0 20 D0 F3
C5 2108:A9 0E 62 02 00 82 C0 00
E7 2110:A0 20 88 B7 03 88 9F E0
32 2118:9F E1 88 B7 03 88 9F E0
78 2120:9F E1 A9 0F 62 02 00 82
0F 2128:42 00 C8 B7 03 88 9F C0
8F 2130:9F E1 88 B7 03 88 9F C0
5C 2138:9F E1 88 B7 03 88 9F C0
91 2140:9F E1 88 B7 03 88 9F C0
6E 2148:9F E1 A9 0E 62 02 00 82
89 2150:45 00 C8 B7 03 88 9F E0
E2 2158:9F E1 88 B7 03 88 9F E0
C7 2160:9F E1 C0 00 D0 AC A5 09
C3 2168:8D 29 C0 60 A2 80 9F 00
56 2170:9D E1 48 A5 1B 20 A8 FC
86 2178:68 F8 E0 C8 D0 F0 AD 29
0F 2180:C0 09 C0 8D 29 C0 60 B7
DB 2188:93 85 07 29 0F 85 06 C8
58 2190:B7 03 85 08 88 A9 0F A2
DA 2198:00 9F 00 9D E1 E8 E0 C8
83 21A0:D0 F7 AD 29 C0 09 C0 8D
02 21A8:29 C0 60 48 A5 1B 20 A8
3B 21B0:FC 68 60 C5

```

TOTAL: 719C

END OF LISTING 2

LISTING 3: GS.FX.DEMO

```

37 10 REM =
C0 20 REM = GS.FX.DEMO
89 30 REM = BY PETER STUBBS
AE 40 REM = COPYRIGHT (C) 1988
CB 50 REM = MICROSPARC, INC.
24 60 REM = CONCORD, MA 01742
45 70 REM =
D1 80 FX = 8192:SPD = 27:EFFECT = 26:VIDEO = 49193
   :SUPER = 193:TXT = 65
24 90 PRINT CHR$(4):"PR#3": HOME : HTAB 35:
   PRINT "GS/FX Demo": PRINT : HTAB 31: PRINT
   "COPYRIGHT (C) 1988": PRINT : HTAB 32:
   PRINT "MICROSPARC, INC."
9E 100 ONERR GOTO 220
52 110 PRINT CHR$(2):"BLOAD GS.FX,A52000": POKE
   216,0
18 120 POKE SPD,80: PRINT : PRINT : PRINT "Which
   speed setting would you like?(S/M/F)":
   POKE - 16368,0: GET AN$: PRINT AN$: IF A
   N$ = "S" OR AN$ = "M" THEN POKE SPD,120
41 130 IF AN$ = "F" OR AN$ = "1" THEN POKE SPD,4
   0
06 140 PRINT : PRINT : PRINT "Is there a picture
   currently in memory?(Y/N)": POKE - 16368
   ,0: GET AN$: PRINT AN$: IF AN$ = "Y" OR AN
   $ = "y" THEN 170
42 150 ONERR GOTO 210
21 160 PRINT CHR$(4):"BRUN PIC.CREATE": POKE 21
   6,0
46 170 HOME : VTAB 12: HTAB 37: PRINT "FADE IN":
   FOR T = 1 TO 2000: NEXT : POKE EFFECT,0:
   CALL FX
81 180 POKE VIDEO,SUPER: FOR T = 1 TO 1000: NEXT
   : POKE VIDEO,TXT: HOME : VTAB 12: HTAB 37:
   PRINT "FADE OUT": FOR T = 1 TO 2000: NEXT
   : POKE EFFECT,1: CALL FX
9C 190 POKE VIDEO,SUPER: FOR T = 1 TO 1000: NEXT
   : POKE VIDEO,TXT: HOME : VTAB 12: HTAB 36:
   PRINT "SCROLL IN": FOR T = 1 TO 2000: NEXT
   : POKE EFFECT,2: POKE SPD,( PEEK (SPD) / 1
   .5): CALL FX
1E 200 POKE VIDEO,SUPER: FOR T = 1 TO 1000: NEXT
   : POKE VIDEO,TXT: HOME : VTAB 23: END
2A 210 POKE 216,0: PRINT "ERROR LOADING PIC.CREAT
   E": VTAB 23: END
E9 220 POKE 216,0: PRINT "ERROR LOADING GS.FX":
   VTAB 23: END

```

TOTAL: 114F

END OF LISTING 3

LISTING 4: PIC.CREATE Source Code

```

1 -
2 - PIC.CREATE Source Code
3 - By Peter Stubbs
4 - Copyright(c) 1988
5 - MicroSPARC, Inc
6 - Concord, MA 01742
7 -
8 - Merlin 16
9 -
10 -
11 -   TR   ADR
12 -   KC
13 -   ORG  $300
14 -
15 - VIDSAVE EQU  $0
16 - PIC   EQU  $E12000 ;Start of Super res screen
17 - SCB   EQU  $E18000 ;Start of SCB's
18 - PALZERO EQU  $E18000 ;Location of palette 0
19 - NEWVIDEO EQU  $C029 ;Super Res/linear softswitch
20 -
21 -   SEC          ;EMULATION MODE ON
22 -   KCE
23 -   MK  $11
24 -   LDA  NEWVIDEO
25 -   STA  VIDSAVE ;Save for later
26 -   ORA  $F0100000 ;Turn on linear addressing
27 -   STA  NEWVIDEO
28 - ; set up the SCB's to all point at palette zero
29 -   LDK  $F00
30 -   TKA
31 - LOOP1: STAL  SCB,x
32 -   INK
33 -   CPK  $F0C
34 -   SEQ  MAKE_PAL
35 -   BRA  LOOP1
36 - MAKE_PAL CLC ;Native mode on
37 -   KCE
38 -   KE  $00
39 -   WRP  $F30
40 - ; Create a palette of the 15 intensities of pink
41 -   LDA  $F0000
42 -   TAK
43 - LOOP2: STAL  PALZERO,X
44 -   CLC
45 -   ADC  $F0101 ;Increase intensity by 1
46 -   INK
47 -   INK
48 -   CPK  $F0020
49 -   BNE  LOOP2
50 - ; Create a picture of shaded columns
51 -   LDA  $F0000
52 -   TAK
53 - LOOP 3: STAL  PIC,X
54 -   INK
55 -   INK
56 -   CLC
57 -   ADC  $F1111
58 -   BCC  $K1P
59 -   LDA  $F0000
60 -   CPK  $F0700
61 -   BNE  LOOP
62 -   SEC          ;BACK to emulation mode
63 -   KCE
64 -   LDA  VIDSAVE ;Restore original value
65 -   STA  NEWVIDEO
66 -   RTS
67 - END          ;MERLIN chk = 14

```

END OF LISTING 4

LISTING 5: PIC.CREATE

Start: 300 Length: 52

```

C0 0300:38 FB AD 29 C0 85 09 09
C5 0308:40 8D 29 C0 A2 00 8A 9F
91 0310:00 9D E1 E8 E0 C8 F0 02
F3 0318:80 F5 18 FB C2 30 A9 00
53 0320:00 AA 9F 00 9E E1 18 69
CC 0328:01 01 E8 E8 E0 20 00 D0
C1 0330:F1 A9 00 00 AA 9F 00 20
9E 0338:E1 E8 E8 18 69 11 11 90
96 0340:03 A9 00 00 E0 00 7D D0
46 0348:EC 38 FB A5 09 8D 29 C0
86 0350:60 14

```

TOTAL: 6749

END OF LISTING 5