SECOND FEATURE

HPLOT GS

Draw pictures on the IIGS' Super-Res screen using commands familiar from the Hi-Res environment.

Get the graphics the IIGS was born to run. You don't need a commercial program to take advantage of the IIGS' exceptional Super Hi-Res capability and colors. This program lets you create Super Hi-Res graphics in 16 colors, which can be changed to any of the 4096 available colors. Once you put the program SUPER.HIRES into memory, you can use easy ampersand (&) commands to create the kind of graphics you want.

The program commands let you switch from Super Hi-Res to text and back to Super Hi-Res without erasing any picture data. Or you can clear the Super Hi-Res screen and start off with a default palette of colors just by issuing an HGR command. In addition, you can even save and reload pictures you have created.

&STORE defines a color by setting the amount of its red, blue, and green components.

THE PROGRAM COMMANDS

Since all the functions you need in SUPER.HIRES (Listing 1) are called using the Applesoft ampersand (&) command, adding Super Hi-Res graphics to your own programs is simple. To install SUPER.HIRES, just BRUN it. A summary of commands available with SUPER.HIRES appears in Table 1.

To begin, you should first clear the Super Hi-Res screen and start off with a fresh palette of colors. This is accomplished by issuing the following command:

& HGR

& HGR will display the Super Hi-Res screen, erase it, and put in a default palette of colors. This palette is the same as the default palette in Paintworks Plus from Activision. Any picture currently in memory will be permanently erased, so be sure you've SAVED everything you want to keep before executing the & HGR command.

& GR

To be sure you don't already have a picture in memory that you want to save, you can first issue an & GR. This command will display the Super Hi-Res screen without clearing any picture data. It does the equivalent of POKE 49193,163.

& TEXT

This command will return the display to the previous screen. For example, if you were displaying a standard Hi-Res screen when you switched to Super Hi-Res, issuing this command will return you to the standard Hi-Res screen.

& STORE *X,R,B,G*

This command defines a color by setting the amount of its red, blue, and green components. Sound confusing? It really isn't.

X is a number from 0-15 that tells SUPER.HIRES which of the 16 colors in the palette is about to be defined. *R*, *B*, and *G* represent how much red, blue, and green, respectively, will be in the color. *R*, *B*, and *G* must also range from 0-15. Hence, you can mix 16 different intensities of red, blue, and green.

For example, let's say you want color 0, which is the background color, to be set to the brightest purple available. The settings for that color would be:

& STORE 0,15,15,0

TABLE 1: SUPER.HIRES Commands

Commands	Functions
&HGR	Clears Super Hi-Res screen and puts in a Paintworks Plus palette
&HCOLOR=X	Sets next plot to be plotted using color X
&STORE X,R,B,G	Replaces color X with the color mixture of red, blue, and green R=Red, B=Blue, G=Green (Range: 0-15)
&TEXT	Returns to mode display was in before going into Super Hi-Res
&GR	Goes into Super Hi-Res mode without erasing any screen data
&HPLOT X,Y	Plots a point given in X and Y

This would store a red intensity of 15, a blue intensity of 15, and a green intensity of 0 for color 0. The red and blue would mix at full intensity to produce a bright purple.

& HCOLOR=*X*

This command sets the current plotting color to whichever of the palette's 16 colors you specify. Note that the color you specify points only to a palette value, and the red, blue and green settings for the palette value determine the actual color to be displayed. For example, at one time, & HCOLOR=7 might plot as green, but at another time, it may plot as gray or blue. It just depends on how you have defined color 7.

Again, *X* must be in the range of 0-15 since there are only 16 colors in the palette. I know what you're thinking; the IIGS can display more than 16 colors! The Super Hi-Res screen is able to display colors from any of 16 palettes. And with 16 colors in each palette, it can display 256 colors on the screen. But SUPER.HIRES can't use all 16 palettes; for simplicity's sake, it only uses one. Thus, with only one palette, SUPER.HIRES handles only 16 colors.

& HPLOT X,Y

This command will plot a point on the screen at the given coordinates. X ranges from 0-319 and Y ranges from 0-199. Error messages will be produced if you exceed these limits. In most cases, the screen returns to its original state before displaying the error message. However, in some cases, the errors are caught by the IIGS's ROM routines and the screen will remain in Super Hi-Res. To get back to the text screen, just type & TEXT and Return (even though you won't see what you type).

&HPLOT will plot a point on the screen at the given coordinates.

The Super Hi-Res screen in the IIGS does have one peculiarity. If you try to plot a point or change a color while the screen is not being displayed, strange things may occur. For example, if you try to change the color without the screen display, the palette will appear at the bottom of the Super Hi-Res screen. I made one attempt to plot some points without the screen display, and ended up with magnified splotches on the screen.

To run the demonstration (**Listing 2**) enter the IIGS Control Panel with Control-Open-Apple-Escape and set the System Speed to Fast. Then RUN SUPER.SHOW.

LOAD.SAVE (**Listing 3**) is a stand-alone program that can be used to load, save and display Super Hi-Res screens created by SUPER.HIRES and other programs. To use it, just run it under ProDOS. The menu offers three choices: load, save and quit. When you elect to save a screen, the current Super Hi-Res screen will be displayed. When you press Return, a prompt on the text screen obtains the pathname of the disk file where the graphic data is to be stored. When you select the load option, you are first prompted for the pathname of the disk file where the graphic was stored. After you press Return, the Super Hi-Res screen is displayed and you can watch the graphic pixel data load first, followed by the scanline control bytes and the palette. Press Return and the text screen is restored.

ENTERING THE PROGRAMS

If you do not have an assembler, enter the Monitor with CALL -151, and type in the code to the left of the line numbers in **Listing 1**. Save it to disk with the command:

```
BSAVE SUPER.HIRES,A$6000,L$158
```

If you do have an assembler, enter the source code to the right of the line numbers in **Listing 1** and assemble it, using SUPER.HIRES as the object file name. Note that you do not need an assembler that can handle 65816 opcodes.

Next, enter the Applesoft demonstration program SUPER.SHOW from **Listing 2** and save it with:

```
SAVE SUPER.SHOW
```

Finally, enter **Listing 3**, LOAD.SAVE, and save it with:

```
SAVE LOAD.SAVE
```

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

HOW THE PROGRAM WORKS

Let's start by examining how the IIGS Super Hi-Res screen is set up. The entire Super Hi-Res screen is broken into three parts. The first 32,000 bytes are where the bit map is stored. In 320 mode (320 x 200-pixel resolution), each byte in the bit map represents 2 pixels on the screen, with each pixel represented by 4 bits. These bits set which color is chosen from the palette.

The next 256 bytes are used for what are called Scanline Control Bytes (SCBs). There is one SCB for each line on the display. With only 200 lines on a screen, 56 of the bytes are not used. Each bit of an SCB controls a different screen parameter for that line. Four bits control which palette will be used for that line, one bit indicates the resolution (0=320, 1=640) of the line, one bit determines whether the horizontal interrupt is on or off, and one bit controls whether fill mode is on or off. Those of you who are counting will see that only 7 bits are described; the other bit is not used.

The last 512 bytes are used to define the color palettes. Each of the 16 palettes contains information for 16 colors. Each color is defined by two bytes; one byte defines the red and one defines the green/blue intensities for that color. Only four bits are used to define the amount of each color (red, blue, or green). The red amount is defined by an entire byte. The high nibble of the red byte is unused. The four bits of the green amount go into the high nibble of the second color byte and the four bits of the blue amount go into the lower nibble of that same byte.

Simple addition tells you that a Super Hi-Res picture takes up 32,768 bytes or 32K of memory. Of course, there is not nearly enough space available in main memory to put it there, so the IIGS has reserved a whole different 64K bank of memory for display of the Super Hi-Res screen. This bank of memory is located starting at $E10000. Since the standard 6502 commands do not allow for memory management of any address out of main memory (or bank zero) the 65816 must have commands to do this. SUPER.HIRES uses the LDA and STA forms of these commands to access the Super Hi-Res page.

When first called, SUPER.HIRES sets up the ampersand vector to point to the main program. Thereafter, when an ampersand command is encountered, the program will compare your command to see if it's a supported command. If it's valid, the program branches to the code that implements the appropriate function.

The &HGR command is very simple: It stores a zero in every location of the Super Hi-Res screen and then moves the standard palette into palette 0.

The &HPLOT command uses an Applesoft ROM routine, FRMNUM, to get the X-coordinate. FRMNUM evaluates the expression pointed to by Applesoft's pointer, TXTPRT, and stores the result in Applesoft's main floating point accumulator, FAC. Then it calls GETADR, which will convert FAC into a 2-byte integer, and stores it in LINNUM ($50 and $51). Next, it will store the X-coordinate and get the Y-coordinate. It then goes to convert the coordinates into a screen address. Fortunately, this is an easy task, since the screen is set up very simply, unlike (as many of you know) the standard Hi-Res screen. There are much faster methods of figuring the address, but none so simple.

The LOAD.SAVE Program

Since the Super Hi-Res screen is not in main memory, it can't be saved with a simple BSAVE command. Instead, the picture is transferred into main memory one piece at a time, and each piece is saved. Then, to load the picture, you load sections into main memory and move them to screen memory. That's exactly what the LOAD.SAVE program does.

Pictures saved with this program can also be loaded, edited, and printed using Paintworks Plus. If they are properly saved, pictures created with Paintworks can be loaded and then drawn on the screen. Paintworks Plus uses three file types; $C0 for Paintworks format picture, $C1 for screen type, and $C2 for animation files. In order to load a Paintworks Plus picture, you must first save it as a "screen" file from Paintworks Plus. Then you must change the first portion of line 80 of the LOAD.SAVE program (Listing 3) to read:

```
80 T$="$C1":
```

If you don't do this, the program will print a FILE TYPE MISMATCH error when it tries to load the picture.

CUSTOMIZATION

There are several features and functions that this program does not use or does not do that you may want to change. You might want to make such major changes as including an HPLOT TO function. You might also want to access all available palettes.

LISTING 1: SUPER.HIRES

```
                  * SUPER HIRES
                  * BY TOM DORRIS
                  * COPYRIGHT (C) 1987
                  * BY MICROSPARC, INC
                  * CONCORD, MA  01742
                  *
                  * MERLIN ASSEMBLER
                  *
                  CHKCON   EQU   $DEBE
                  SYNCHR   EQU   $DEC0      ;PRINT SYNTAX ERROR
                  FRMNUM   EQU   $DD67
                  ILLQTY   EQU   $E199      ;PRINT ILLEGAL QUANTITY ERROR
                  GETADR   EQU   $E752
                  GETBYTE  EQU   $E6F8
                  DISPLY   EQU   $C029
                  ADRLO    EQU   $FD
                  ADRHI    EQU   $FE
                  ADRBNK   EQU   $FF
                  CLRMSK   EQU   $FC
                  XHI      EQU   $6
                  XLO      EQU   $7
                  YCOR     EQU   $8
                  COLOR    EQU   $9
                  *
                           ORG   $6000
                  *
6000: A9 0B       SETAMP   LDA   #BEGIN
6002: 8D F6 03             STA   $3F6
6005: A9 60                LDA   #>BEGIN
6007: 8D F7 03             STA   $3F7
600A: 60                   RTS
                  *
600B: 85 06       BEGIN    STA   XHI
600D: 20 B1 00             JSR   $00B1
6010: A5 06                LDA   XHI
6012: C9 89                CMP   #137       ;IS IT TEXT?
6014: F0 17                BEQ   TEXT       ;YES, DISPLAY TEXT
6016: C9 91                CMP   #145       ;HGR?
6018: F0 28                BEQ   HGR        ;YES, DO AN HGR
601A: C9 92                CMP   #146       ;HCOLOR=?
601C: F0 15                BEQ   HCL        ;YES, SET COLOR
601E: C9 93                CMP   #147       ;HPLOT?
6020: F0 14                BEQ   HPL        ;YES, PLOT POINT
6022: C9 A8                CMP   #168       ;STORE?
6024: F0 13                BEQ   STR        ;YES, CREATE COLOR
6026: C9 88                CMP   #136       ;GR?
6028: F0 12                BEQ   GR         ;YES, DISPLAY SUPERHIRES
602A: 4C 50 61             JMP   PRSYNERR   ;NOTHING VALID, PRINT ERROR
602D: A9 01       TEXT     LDA   #$1
602F: 8D 29 C0             STA   DISPLY     ;DISPLAY TEXT SCREEN
6032: 60                   RTS
6033: 4C 85 60    HCL      JMP   HCOLOR
6036: 4C C1 60    HPL      JMP   HPLOT
6039: 4C 8E 60    STR      JMP   STORE
603C: A9 A3       GR       LDA   #163
603E: 8D 29 C0             STA   DISPLY     ;DISPLAY SUPERHIRES SCREEN
6041: 60                   RTS
6042: 20 3C 60    HGR      JSR   GR         ;DISPLAY SUPERHIRES
6045: 20 3D 61             JSR   SETUP      ;SETUP ADDRESS POINTERS
6048: 87 FD       MORE     HEX   87FD       ;GS COMMAND STA [FD]
604A: E6 FD                INC   ADRLO
604C: D0 FA                BNE   MORE
604E: E6 FE                INC   ADRHI
6050: A4 FE                LDY   ADRHI
6052: C0 A0                CPY   #$A0       ;END OF HIRES DATA?
6054: D0 F2                BNE   MORE       ;NO, ERASE ANOTHER PAGE
6056: A2 00                LDX   #0
6058: BD 65 60    CLR      LDA   CLRS,X
605B: 9F 00 9E             HEX   9F009EE1   ;GS COMMAND STA E19E00,X
605E: E1
605F: E8                   INX
6060: E0 20                CPX   #$20       ;LAST COLOR?
6062: D0 F4                BNE   CLR        ;NO, STORE ANOTHER
6064: 60                   RTS
                  * PAINTWORKS PLUS PALETTE
6065: 00 00 77    CLRS     HEX   0000770741082C070F00
6068: 07 41 08 2C 07 0F 00
606F: 80 00 70             HEX   8000700F000DA90FF00F
6072: 0F 00 0D A9 0F F0 0F
6079: E0 00 DF             HEX   E000DF04AF0D8F07CC0C
607C: 04 AF 0D 8F 07 CC 0C
6083: FF 0F                HEX   FF0F
6085: 20 F8 E6    HCOLOR   JSR   GETBYTE    ;GET REQUESTED COLOR
6088: 8A                   TXA
6089: 29 0F                AND   #$0F       ;MASK OUT UNNEED BITS
608B: 85 09                STA   COLOR
608D: 60                   RTS
608E: 20 F8 E6    STORE    JSR   GETBYTE    ;GET COLOR TO CHANGE
6091: E0 10                CPX   #$10
6093: 90 01                BCC   OK
6095: 60                   RTS
6096: 86 06       OK       STX   XHI        ;STORE COLOR IN UNUSED LOC
6098: 20 37 61             JSR   GETBT      ;GET RED COLOR
609B: 86 07                STX   XLO
609D: 20 37 61             JSR   GETBT      ;GET BLUE COLOR
60A0: 8A                   TXA
60A1: 29 0F                AND   #$0F       ;MASK OUT UNNEED BITS
60A3: 85 08                STA   YCOR
60A5: 20 37 61             JSR   GETBT      ;GET GREEN COLOR
60A8: 86 FC                STX   CLRMSK
60AA: A5 06                LDA   XHI        ;GET COLOR TO CHANGE
60AC: 0A                   ASL              ;MULITPLY BY 2
60AD: AA                   TAX              ;USE AS POINTER
60AE: A5 FC                LDA   CLRMSK     ;GET GREEN COLOR
60B0: 0A                   ASL              ;MOVE INTO HIGH NIBBLE
60B1: 0A                   ASL
60B2: 0A                   ASL
60B3: 0A                   ASL
60B4: 65 08                ADC   YCOR       ;ADD BLUE VALUE
60B6: 9F 00 9E             HEX   9F009EE1   ;GS COMMAND STA E19E00,X
60B9: E1
60BA: A5 07                LDA   XLO        ;GET RED COLOR
60BC: 9F 01 9E             HEX   9F019EE1   ;GS COMMAND STA E19E01,X
60BF: E1
60C0: 60                   RTS
60C1: 20 67 DD    HPLOT    JSR   FRMNUM     ;GET X COOR
60C4: 20 52 E7             JSR   GETADR     ;EVALUATE AND STORE IN LINNUM
60C7: A5 50                LDA   $50
60C9: 85 07                STA   XLO
60CB: A5 51                LDA   $51
60CD: 85 06                STA   XHI
60CF: F0 0D                BEQ   XOK
60D1: C9 02                CMP   #2
60D3: B0 06                BCS   ERROUT
60D5: A5 07                LDA   XLO
60D7: C9 40                CMP   #64
60D9: 90 03                BCC   XOK
60DB: 4C 4A 61    ERROUT   JMP   PRILLQTY
60DE: 20 37 61    XOK      JSR   GETBT      ;GET Y COOR
60E1: 86 08                STX   YCOR
60E3: E0 C8                CPX   #200
60E5: 90 03                BCC   YOK
60E7: 4C 4A 61             JMP   PRILLQTY
60EA: 20 3D 61    YOK      JSR   SETUP      ;SETUP ADDRESS POINTERS
60ED: A5 07                LDA   XLO
60EF: 18                   CLC              ;GET X COOR AND
60F0: 6A                   ROR              ;DIVIDE BY 2
60F1: A6 06                LDX   XHI
60F3: F0 02                BEQ   NODIV
60F5: 09 80                ORA   #$80       ;DIVIDE XHI BY 2
60F7: 85 FD       NODIV    STA   ADRLO
60F9: A6 08                LDX   YCOR       ;GET YCOOR FOR ADDRESS
60FB: E8                   INX
60FC: CA          DOMORE   DEX              ;LAST LINE?
60FD: F0 0E                BEQ   DONE2      ;YES, PLOT POINT
60FF: A9 A0                LDA   #$A0
6101: 18                   CLC
6102: 65 FD                ADC   ADRLO      ;ADD 160 TO ADDRESS FOR
6104: 85 FD                STA   ADRLO      ;EACH LINE
6106: 90 02                BCC   NOOVER
6108: E6 FE                INC   ADRHI
610A: 4C FC 60    NOOVER   JMP   DOMORE
610D: A5 07       DONE2    LDA   XLO
610F: 29 01                AND   #1         ;EVEN COOR?
6111: D0 11                BNE   ODD        ;NO, PLOT PIXEL IN LOW NIBBLE
6113: A5 09                LDA   COLOR      ;YES, MOVE COLOR TO HIGH NIBBL
6115: 0A                   ASL
6116: 0A                   ASL
6117: 0A                   ASL
6118: 0A                   ASL
```

LISTING 1: SUPER.HIRES *(continued)*

```
6119: 85 FC       155            STA   CLRMSK
611B: A7 FD       156            HEX   A7FD        ;GS COMMAND LDA [FD]
611D: 29 0F       157            AND   #$0F        ;MASK OUT OLD PIXEL
611F: 87 FD       158            HEX   87FD        ;GS COMMAND STA [FD]
6121: 4C 30 61    159            JMP   PLT
6124: A5 09       160  ODD       LDA   COLOR
6126: 29 0F       161            AND   #$0F
6128: 85 FC       162            STA   CLRMSK
612A: A7 FD       163            HEX   A7FD        ;GS COMMAND LDA [FD]
612C: 29 F0       164            AND   #$F0        ;MASK OUT OLD PIXEL
612E: 87 FD       165            HEX   87FD        ;GS COMMAND STA [FD]
6130: A7 FD       166  PLT       HEX   A7FD        ;GS COMMAND LDA [FD]
6132: 05 FC       167            ORA   CLRMSK      ;PLOT POINT
6134: 87 FD       168            HEX   87FD        ;GS COMMAND STA [FD]
6136: 60          169            RTS
6137: 20 BE DE    170  GETBT     JSR   CHKCOM      ;FIND COMMA AND
613A: 4C F8 E6    171            JMP   GETBYTE     ;GET NUMBER
613D: A9 E1       172  SETUP     LDA   #$E1
613F: 85 FF       173            STA   ADRBNK
6141: A9 20       174            LDA   #$20
6143: 85 FE       175            STA   ADRHI
6145: A9 00       176            LDA   #$00
6147: 85 FD       177            STA   ADRLO
6149: 60          178            RTS
                  179  *
614A: 20 2D 60    180  PRILLQTY  JSR   TEXT
614D: 4C 99 E1    181            JMP   ILLQTY
                  182  *
6150: 20 2D 60    183  PRSYNERR  JSR   TEXT
6153: A9 FF       184            LDA   #$FF
6155: 4C C0 DE    185            JMP   SYNCHR


--End assembly, 344 bytes, Errors: 0
```

END OF LISTING 1

```
                 KEY PERFECT 5.0
                     RUN ON
                  SUPER.HIRES
   ======================================
     CODE-5.0    ADDR# - ADDR#    CODE-4.0
     --------    -------------    --------
     D82A3259     6000 - 604F       286F
     F3FB01AC     6050 - 609F       28CB
     F6C8FDA0     60A0 - 60EF       2797
     53AF84AF     60F0 - 613F       2789
     9FA2995B     6140 - 6157       0DA4
     AF9071E5 = PROGRAM TOTAL =     0158
```

LISTING 2: SUPER.SHOW

```
10  REM  *********************
20  REM  *     SUPER.SHOW    *
30  REM  *   BY TOM DORRIS   *
40  REM  * COPYRIGHT (C) 1987 *
50  REM  * BY MICROSPARC, INC *
60  REM  * CONCORD, MA  01742 *
70  REM  *********************
80  REM
90  REM  * DRAW OVAL FIGURE
100  ONERR  GOTO 530
110  PRINT  CHR$ (4)"BRUN SUPER.HIRES"
120  POKE 216,0
130  &  HGR : FOR I = 0 TO 15: &  STORE I,I,I
     ,0: NEXT
140  FOR I = 0 TO 319:YP = 96 - ( SIN ((I - 1
     40) / 38) * 30): FOR J = 0 TO 15: &  HCOLOR=
     J: &  HPLOT I,YP + J: NEXT J,I
150 PA = 1.570795:PB = 4.712385:DP = .0174532
     778
160 O = 0:E = 15:XC = 160:YC = 100:R = 50:RY =
     R
170  FOR I = PA TO PB STEP DP
180  FOR R = O TO E
190  &  HCOLOR= R
200 X = XC + R *  COS (I):Y = YC + RY *  SIN
     (I)
210  &  HPLOT X,Y
220  &  HPLOT Y + 60,X - 60
230  &  HPLOT XC + XC - X,Y
240  &  HPLOT Y + 60,YC + (YC - (X - 60))
250  NEXT R
260  NEXT
270  REM  * DRAW COLOR BAR
280  FOR I = 0 TO 179: &  HCOLOR=  INT (I / 1
     1): FOR J = 20 TO 30: &  HPLOT 70 + I,J:
      NEXT J,I
290  REM  * DRAW OUTLINE OF SCREEN
300  FOR I = 0 TO 319: &  HCOLOR=  INT (I / 2
     0): &  HPLOT I,0: &  HPLOT 319 - I,199: NEXT
     I
310  FOR I = 0 TO 199: &  HCOLOR=  INT (I / 1
     2.5): &  HPLOT 0,I: &  HPLOT 319,199 - I
     : NEXT
320  REM  * BEGIN COLOR ANIMATION
330  REM  * FIRST STAGE
340 DL = 100: FOR R = 0 TO 1: FOR G = 0 TO 1:
      FOR B = 0 TO 1: GOSUB 480: NEXT B,G,R
350 R = 0:B = 0:G = 0:DL = 1: GOSUB 480
360  REM  * SECOND STAGE
370 DL = 100
380  FOR R1 = 0 TO 1: FOR B1 = 0 TO 1: FOR G1
      = 0 TO 1: GOSUB 510: NEXT G1,B1,R1
390  REM  * THIRD AND FINAL STAGE
400  FOR R = 0 TO 1: FOR B = 0 TO 1: FOR G =
     0 TO 1
410  FOR I = 0 TO 15: &  STORE I,R * I,B * I,
     G * I
420  FOR J = 1 TO 100: NEXT J: &  STORE I,0,0
     ,0: NEXT I
430  FOR I = 15 TO 0 STEP  - 1: &  STORE I,R *
     I,B * I,G * I: FOR J = 1 TO 100: NEXT J:
      &  STORE I,0,0,0
440  NEXT I,G,B,R
450  FOR I = 0 TO 15: &  STORE I,I,I,I: NEXT

460 C =  INT ( RND (1) * 15) + 1:R =  INT ( RND
     (1) * 15) + 1:B =  INT ( RND (1) * 15) +
     1:G =  INT ( RND (1) * 15) + 1: &  STORE
     C,R,B,G: FOR I = 1 TO 200: NEXT : IF  PEEK
     ( - 16384) < 128 THEN 460
470  &  TEXT : END
480  FOR I = 0 TO 15: &  STORE I,R * I,B * I,
     G * I: FOR J = 1 TO DL: NEXT J,I
490  RETURN
500  REM  * SECOND COLOR LOADING
510  FOR I = 1 TO 15: &  STORE I,R1 * I,B1 *
     I,G1 * I:X = I - 1: &  STORE X,R1 * X,B1
      * X,G1 * X: FOR J = 1 TO DL: NEXT J: &
      STORE I,R * I,B * I,G * I: &  STORE X,R
      * X,B * X,G * X: NEXT I
520  RETURN
530  HOME : PRINT "UNABLE TO LOAD SUPER.HIRES
     ": END
```

END OF LISTING 2

```
                  KEY PERFECT 5.0
                      RUN ON
                    SUPER.SHOW
   ======================================
     CODE-5.0    LINE# - LINE#    CODE-4.0
     --------    -------------    --------
     10CCC6A0       10 -   100      5B61
     9DB6638B      110 -   200      7749
     7043E457      210 -   300      60E8
     B6216681      310 -   400      7F27
     672EA3A9      410 -   500      8E6B
     B578019C      510 -   530      406D
     55A0C3AD = PROGRAM TOTAL =     0572
```

LISTING 3: LOAD.SAVE

```
10  REM  *********************
20  REM  *     LOAD.SAVE     *
30  REM  *   BY TOM DORRIS   *
40  REM  * COPYRIGHT (C) 1987 *
50  REM  * BY MICROSPARC, INC *
60  REM  * CONCORD, MA  01742 *
70  REM  *********************
80 T$ = "BIN": IF  PEEK (48896) <  > 76 THEN
      HOME : PRINT "THIS PROGRAM REQUIRES PRO
     DOS": END
90  TEXT : HOME : VTAB 8: HTAB 16: PRINT "LOA
     D.SAVE": VTAB 10: HTAB 7: PRINT "SUPER H
     IRES LOADER AND SAVER"
100  PRINT : HTAB 14: PRINT "BY TOM DORRIS": PRINT
     : PRINT " COPYRIGHT 1987 BY MICROSPARC,
     INC."
110  PRINT : HTAB 6: PRINT "PRESS RETURN TO C
     ONTINUE...";: GET A$: PRINT
120 D$ =  CHR$ (4): GOSUB 460
130  HOME
140  HTAB 5: VTAB 5: PRINT "CHOOSE YOUR OPTIO
     N:"
```

LISTING 3: LOAD.SAVE *continued*

```
150  HTAB 5: VTAB 7: PRINT "1. LOAD PICTURE":
      HTAB 5: PRINT "2. SAVE PICTURE": HTAB 5
     : PRINT "3. QUIT"
160  HTAB 5: VTAB 11: PRINT "-->";: GET A$: IF
     A$ < "1" OR A$ > "3" THEN 160
170  PRINT A$
180  IF A$ = "3" THEN  HOME : END
190  IF A$ = "2" THEN 240
200  HOME : HTAB 5: VTAB 5: PRINT "ENTER THE
     NAME OF THE FILE TO": HTAB 5: PRINT "LOA
     D.  INCLUDE ANY PREFIX NEEDED."
210  HTAB 5: VTAB 9: INPUT "-->";A$: IF  NOT
      LEN (A$) THEN 13Ø
220  HOME : HTAB 5: VTAB 5: PRINT "INSERT DIS
     K CONTAINING PICTURE.": HTAB 5: VTAB 19:
      PRINT "PRESS RETURN TO LOAD...";: GET B
     $: PRINT
230  POKE 49193,163: GOSUB 380: GET A$: POKE
     49193,1: GOTO 13Ø
240  POKE 49193,163: GET A$: POKE 49193,1
250  HOME : HTAB 5: VTAB 5: PRINT "ENTER THE
     NAME OF THE FILE TO SAVE": HTAB 5: PRINT
     "PICTURE AS.  NAME SHOULD INCLUDE": HTAB
     5: PRINT "ANY PREFIX NEEDED."
260  HTAB 5: VTAB 9: INPUT "-->";A$: IF  NOT
      LEN (A$) THEN 13Ø
270  HOME : HTAB 5: VTAB 5: PRINT "INSERT DIS
     K YOU WISH TO SAVE": HTAB 5: PRINT "PICT
     URE ONTO.": HTAB 5: VTAB 19: PRINT "PRES
     S RETURN TO SAVE...";: GET B$: PRINT
280  POKE 49193,163
290  FOR I = 1 TO 4
300  POKE 771,Ø: POKE 772,I * 32: POKE 773,22
     5
310  POKE 775,Ø: POKE 776,64: POKE 777,Ø: POKE
     791,(I + 1) * 32
320  CALL 768: REM  *** LOAD NEXT PICTURE SE
     GMENT INTO WORKING MEMORY ***
330  PRINT D$"BSAVE "A$",A$4ØØØ,L$2ØØØ,B"(I -
     1) * 8192
340  NEXT : POKE 49193,1: GOTO 13Ø
350  REM  *** LOAD PICTURE ROUTINE ***
360  REM  *** LOAD IN PICTURE IN 4 EQUAL SECT
     IONS ***
370  END
380  FOR QW = 1 TO 4
390  PRINT D$"BLOAD "A$",A$4ØØØ,L$2ØØØ,B"(QW -
     1) * 8192",T"T$
400  POKE 771,Ø: POKE 772,64: POKE 773,Ø
410  POKE 775,Ø: POKE 776,32 * QW: POKE 777,2
     25
420  POKE 791,96
430  CALL 768: REM  *** MOVE LOADED SECTION T
     O DISPLAY MEMORY ***
440  NEXT : REM  *** LOAD NEXT SECTION ***
450  RETURN
460  FOR I = Ø TO 26: READ A: POKE 768 + I,A:
     C = C + A: NEXT
470  IF C <  > 3025 THEN  HOME : PRINT "DATA
     STATEMENT ERROR!": END
480  RETURN
490  DATA  162,Ø,191,Ø,64,Ø,159,Ø,32,225,232,
     2Ø8,245,238,4,3,238,8,3,173,4,3,2Ø1,96,2
     Ø8,232,96
```

END OF LISTING 3

```
                KEY PERFECT 5.0
                    RUN ON
                   LOAD.SAVE
   ==========================================
      CODE-5.Ø     LINE# - LINE#    CODE-4.Ø
      --------     -------------    --------
      3E67A35A       1Ø  -   1ØØ      A492
      4CEB4F83      11Ø  -   2ØØ      8AAA
      8Ø273386      21Ø  -   3ØØ      D9B6
      F121EF11      31Ø  -   4ØØ      955F
      61F4Ø323      41Ø  -   49Ø      8148
      DD99A7CE = PROGRAM TOTAL =      Ø6A4
```