Fudge It!

by Don Fudge

Create Gorgeous Color-Filled Scenes!

In this column I'll discuss color-filling algorithms and present a machine language program called FILL4 that color-fills line drawings made by white lines on a black background. The program enables you to create scenes easily and is fun to use. It does, however, require prior creation of a line drawing picture. You may use HPLOT shape-drawing routines that I have presented in earlier columns, or write your own. You might want to create an etch-a-sketch screen drawing program by having the computer HPLOT lines connecting various coordinates determined by game paddle settings (as you move the paddles and push a paddle button to cause hplotting from your earlier hi-res screen position to your newest hi-res screen position). Or simply do HGR: POKE-16302,0:HCOLOR = 3:HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO X4,Y4------.

Once that line drawing is saved (BSAVE line drawing, A$2000,L$1FF8) you are ready to color it. When you key in the enclosed programs, you will have some great scene creation utilities (including the fastest color-fill algorithm available anywhere). If you haven't the time to type programs, I would suggest Avant-Garde's Paint Master Scene Utility as an inexpensive scene creator.

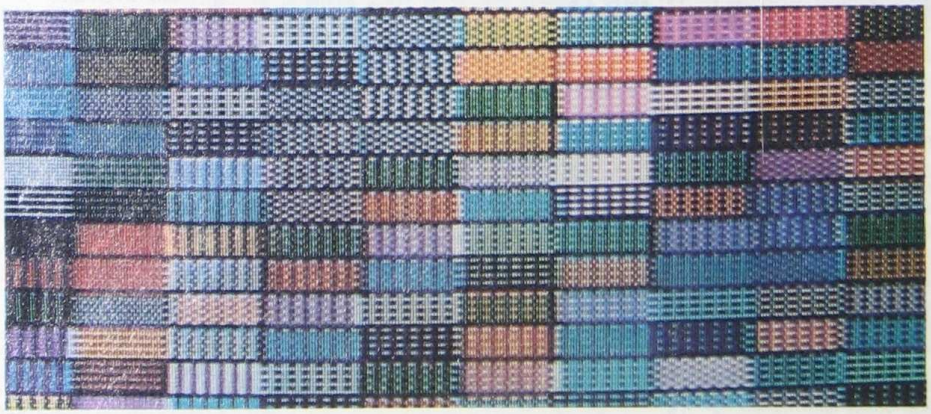
Scene painting is filling enclosed spaces with colors and/or patterns. The palette included here contains hundreds of colors/patterns to choose from. Or use Listing 4 (with line 10 modified to 10 HGR2) from my March *inCider* column to create a color palette (see photo), and then hit control-reset and type BSAVE PATRN, A$4000, L$1FF8. Don't forget to use POKE 103,1: POKE 104,96: POKE 24576,0 *before* loading or running that listing.

Start-of-Program Pokes

A note about the above pokes. Several people have written saying that they've had trouble making programs work after typing them from *inCider* columns. The problem in all cases was either that they did not carefully read the column and skipped the section about the necessary pokes, or they decided to put the pokes at the beginning of their graphics programs. *This will not work!*

Here is what happens. The Apple normally loads all Applesoft programs at $800. But to avoid having graphics and string or variable storage clashing in memory, you should begin graphics programs at $4000 if they use hi-res page 1, and $6000 if they use hi-res page 2. This enables you to ignore HIMEM and LOMEM and to use $800–$1FFF for subroutine and data table storage. Further, it al-

Address correspondence, including inquiries about additional routines and programs, to Don Fudge at Avant-Garde Creations, PO Box 30160, Eugene, OR 97403.



Some of the colors in a scene painting palette.

lows Basic programs to be $5600 long, including variable and string storage.

POKE 103,1: POKE 104,64: POKE 16384,0 permits use of page 1 graphics and starts your Basic programs at $4000. POKE 103,1: POKE 104,96: POKE 24576,0 permits use of page 1 and/or page 2 graphics and starts your Basic programs at $6000. Page 1 is $2000–$3FFF and page 2 is $4000–$5FFF.

Now, it is impossible to move a loaded program by use of these pokes; if you put the pokes in line 1, your program will bomb. Line 1 is not run until a program is entirely loaded and by then it is too late. So you must do these pokes either in IMMEDIATE mode (no line number) or in the Hello program that boots when you turn on your computer and your drive runs. It is okay to have a line like 90?CHR$(4)"RUN PALETTE" *after* a line such as 50 POKE 103,1: POKE 104,96: POKE 24576,0.

Getting Ready

For Listing 1 (FILL4) you need only do CALL-151 and 9000:, and then start typing code, hitting return after six lines are full and updating your address. PATRNMAKER (Listing 2) needs only to be keyed in, saved and run, from Basic. Again, saving the March issue's color palette, as previously advised, is also viable here, but you get no patterns—just colors.

Once FILL4 is keyed in, use BSAVE FILL4,A$9000,L$400. Then, after you're done running PATRNMAKER (POKE 103,1: POKE 104,96: POKE 24576,0 *first*) a file called PATRN will have been saved on your disk. This is the palette (a binary picture) your PALETTE program will be looking for when you choose your very first color.

PALETTE, Listing 3, is another Basic program to key in. Once FILL4, PATRN, and PALETTE are all saved on your disk, do POKE 103,1: POKE 104,96: POKE 24576,0 and run PALLETTE.

Using the PALETTE Program

Here are a few details you will need to know:

1. If you will be loading any HPLOT shapes, TEST 0 (CALL2048) must be on the disk. This file was presented in my March column as Listing 2. If you have no such file, do not worry; just avoid the HPLOT shape part of option 9.

2. If you will be loading block shapes into your scene (also in option 9) you will need TESTTB, another routine presented in March. If you have no TESTTB, all is well—simply avoid using block shapes as additions to the screen.

3. If you will want to use option 13 (GO TO SCAN & SAVE SHAPE) you will need my SCANA program from the April *inCider*. It turns any part of the screen into a table-ized block shape. If you have no such file, no problem—just don't try to create block shapes with sections of the screen in the hi-res scenes you create with PALETTE.

Incidentally, do you notice how I'm tying all the programs I've presented in this column together to create a *system* with which you can perform any general graphics feat you

```
*9000.93FF

9000- A9 00 85 1F 85 FA 85 1B
9008- 85 FB 85 1D 85 FC 85 1A
9010- 85 EC 85 EE 85 CF 85 CE
9018- 85 E3 A5 26 85 FD A5 27
9020- 85 FE A4 FF 84 EF B1 26
9028- C9 7F D0 01 60 C9 FF D0
9030- 01 60 A5 EF C9 04 90 0B
9038- 4A 4A 0A 0A 85 D3 A5 EF
9040- 38 E5 D3 85 FC A5 FC D0
9048- 07 A5 06 A2 06 4C 6A 90
9050- C9 01 D0 07 A5 07 A2 07
9058- 4C 6A 90 C9 02 D0 07 A5
9060- 08 A2 08 4C 6A 90 A5 09
9068- A2 09 86 1E 85 D3 B1 26
9070- 0A D0 07 A5 D3 91 26 4C
9078- 3E 91 B1 26 29 7F 85 19
9080- A5 ED 29 7F C5 19 B0 06
9088- 20 E5 90 4C 31 91 20 94
9090- 90 4C 31 91 A9 00 85 FB
9098- 85 EB 85 CF B1 26 0A E6
90A0- EB 0A 90 FB A5 EB C9 02
90A8- B0 03 E6 CF 60 A2 00 A1
90B0- 1E 85 19 C6 EB F0 29 A9
90B8- 07 38 E5 EB 85 EB 85 EC
90C0- A5 19 C9 80 90 02 E6 FB
90C8- 4A C6 EB D0 FB 0A C6 EC
90D0- D0 FB 85 19 B1 26 05 19
90D8- A6 FB F0 02 09 80 91 26
90E0- E6 1A E6 E3 60 A9 00 85
90E8- FB 85 EB 85 CF B1 26 E6
90F0- EB 4A 90 FB A5 EB C9 02
90F8- B0 03 E6 CF 60 A2 00 A1
9100- 1E 85 19 C6 EB F0 29 A9
9108- 07 38 E5 EB 85 EB 85 EC
9110- A5 19 C9 80 90 02 E6 FB
9118- 0A C6 EB D0 FB 4A C6 EC
9120- D0 FB 85 19 B1 26 05 19
9128- A6 FB F0 02 09 80 91 26
9130- 60 A5 1A D0 09 A9 00 85
9138- FA 85 1A 4C 81 91 A9 00
9140- 85 1A 85 FA C0 27 F0 ED
9148- C8 E6 EF E6 1E A5 1E C9
9150- 0A D0 07 A2 06 86 1E 4C
9158- 62 91 C9 E0 D0 04 A2 DC
9160- 86 1E B1 26 C9 7F D0 02
9168- F0 CB C9 FF D0 02 F0 C5
9170- B1 26 0A D0 09 A2 00 A1
9178- 1E 91 26 4C 3E 91 20 E5
9180- 90 A6 E3 F0 07 A9 00 85
9188- E3 4C 1F 92 C0 00 F0 F9
9190- E6 FA A5 FA C9 02 B0 4E
9198- A4 FF A5 1E C9 0A B0 24
91A0- A5 FC D0 06 A9 0A 85 1E
91A8- D0 3C C9 01 D0 06 A9 07
91B0- 85 1E D0 32 C9 02 D0 06
91B8- A9 08 85 1E D0 28 A9 09
91C0- 85 1E D0 22 A5 FC D0 06
91C8- A9 E0 85 1E D0 18 C9 01
91D0- D0 06 A9 DD 85 1E D0 0E
91D8- C9 02 D0 06 A9 DE 85 1E
91E0- D0 04 A9 DF 85 1E 88 84
91E8- EF C6 1E A5 1E C9 05 D0
91F0- 07 A2 09 86 1E 4C 00 92
91F8- C9 DB D0 04 A2 DF 86 1E
9200- B1 26 C9 7F D0 02 F0 17
9208- C9 FF D0 02 F0 11 B1 26
9210- 0A D0 09 A2 00 A1 1E 91
9218- 26 4C 81 91 20 94 90 A9
9220- 00 85 E3 A5 1D F0 03 4C
9228- EE 92 A5 1E C9 0A B0 24
9230- A5 FC D0 06 A9 DC 85 1E
9238- D0 3C C9 01 D0 06 A9 DD
9240- 85 1E D0 32 C9 02 D0 06
9248- A9 DE 85 1E D0 28 A9 DF
9250- 85 1E D0 22 A5 FC D0 06
9258- A9 06 85 1E D0 18 C9 01
9260- D0 06 A9 07 85 1E D0 0E
9268- C9 02 D0 06 A9 08 85 1E
9270- D0 04 A9 09 85 1E 20 D5
9278- F4 A4 FF 84 EF A5 27 C9
9280- 3F 90 0C A5 26 C9 D0 90
9288- 06 20 04 F5 4C EE 92 B1
9290- 26 C9 7F D0 02 F0 57 C9
9298- FF D0 02 F0 51 B1 26 0A
92A0- F0 43 A5 EE D0 1F B1 26
92A8- 29 7F 85 19 A5 ED 29 7F
92B0- C5 19 B0 0D 18 A5 19 4A
92B8- B0 07 A9 02 85 EE 4C C5
92C0- 92 A9 01 85 EE A5 EE C9
92C8- 02 F0 0C 20 94 90 A5 CF
92D0- D0 1C E6 E3 4C 3E 91 20
92D8- E5 90 A5 CF D0 10 A9 02
92E0- 85 FA 4C 81 91 A2 00 A1
92E8- 1E 91 26 4C 3E 91 E6 1B
92F0- A5 1B C9 02 B0 2D A5 FC
92F8- D0 06 A9 DC 85 1E D0 18
9300- C9 01 D0 06 A9 DD 85 1E
9308- D0 0E C9 02 D0 06 A9 DE
9310- 85 1E D0 04 A9 DF 85 1E
9318- A5 FD 85 26 A5 FE 85 27
9320- 4C 6F 93 A5 1E C9 0A B0
9328- 24 A5 FC D0 06 A9 DC 85
9330- 1E D0 3C C9 01 D0 06 A9
9338- DD 85 1E D0 32 C9 02 D0
9340- 06 A9 DE 85 1E D0 28 A9
9348- DF 85 1E D0 22 A5 FC D0
9350- 06 A9 06 85 1E D0 18 C9
9358- 01 D0 06 A9 07 85 1E D0
9360- 0E C9 02 D0 06 A9 08 85
9368- 1E D0 04 A9 09 85 1E 20
9370- 04 F5 A4 FF 84 EF A5 27
9378- C9 21 B0 07 A5 26 C9 28
9380- B0 01 60 B1 26 C9 7F D0
9388- 02 F0 F7 C9 FF D0 02 F0
9390- F1 B1 26 0A F0 46 E6 1D
9398- A5 CE D0 1F B1 26 29 7F
93A0- 85 19 A5 ED 29 7F C5 19
93A8- B0 0D 18 A5 19 4A B0 07
93B0- A9 02 85 CE 4C BB 93 A9
93B8- 01 85 CE A5 CE C9 02 F0
93C0- 0C 20 94 90 A5 CF D0 13
93C8- E6 E3 4C 3E 91 20 E5 90
93D0- A5 CF D0 07 A9 02 85 FA
93D8- 4C 81 91 60 A2 00 A1 1E
93E0- 91 26 E6 1D 4C 3E 91 00
93E8- 46 42 46 42 46 46 46 46
93F0- 40 60 40 60 40 40 40 40
93F8- 44 40 44 40 44 44 04 44
```

Listing 1. FILL4.

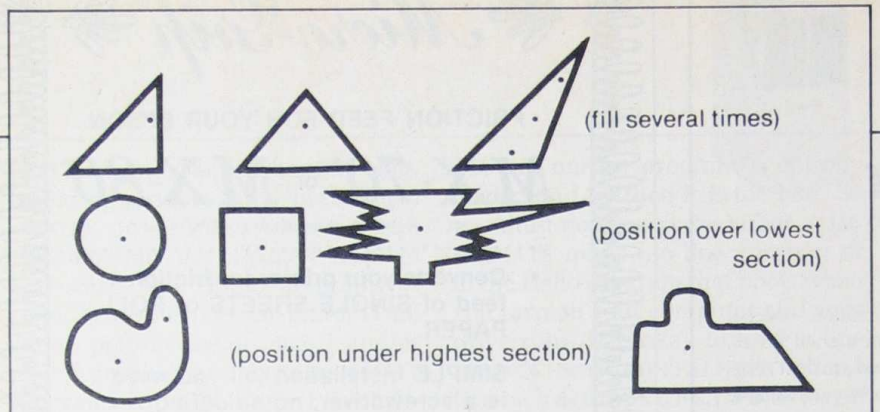


Figure 1. Sample figures to be filled.

desire with ease and convenience? This is no accident. It parallels the way I have written my major graphics utilities. The only problem this might create for you is that there will be so many possibilities for graphics creativity, you may have trouble deciding what to do next. Such is life.

4. Pictures you save with PALETTE will be hi-res page 1 pictures ($2000). If you need them in your own programs at $4000, simply BLOAD them at that address when using them. If you need to save screen pictures in only a few sectors and to retrieve and draw them in only a few seconds, you'll need to go beyond PALETTE to something like Paint Master Scene Utility.

5. When creating line drawings, frame them. They will look better and "color wraparound" will be precluded. HPLOT 0,0 TO 0,191 TO 279,191 TO 0,0 will do it.

6. It is best, especially on unframed pictures, to start filling near the right side of the screen and work left.

7. To quiet the cursor clicks, hit A.

8. When filling with color, hit the space bar to produce an extra large paintbrush.

9. To see screen coordinates when filling, hit C.

10. During filling, if you hit 1–9 you will be asking for the PAINTBRUSH feature (which has nothing to do with color-filling). To move the paintbrush without affecting the picture, hold down paddle button #1. To paint use the paddle #0 knob and work sideways. To paint higher or lower on the screen use the paddle #1 knob to move vertically, with or without holding down the paddle #1 button (depending upon whether or not you want the picture to be affected). Move the paddle #0 knob fairly slowly as you paint. Choose different heights of brush and different colors by hitting the space bar to stop painting, then P for PAINTBRUSH mode and 1–9 for height and 0–7 for color. The colors are black = 0 or 4, white = 3 or 7, green = 1, violet = 2, orange = 5, and blue = 6.

11. To color-fill use option 11, but make sure you have first picked a color via option 10 or 17. Move the cursor around the screen with the paddles, using the #0 button to fill and the #1 button to cease filling.

12. Options 2–8 all relate to vector shape use. Use option 9 to load in your vector shape table first and then view the entire table with option 2 (control-C for early exit), or specify DRAW/XDRAW, ROT (rotation), HCOLOR, X-Y coordinates, SCALE, or background color via options 2–8.

13. After loading in a block or vector shape, with option 9 you can move the shape around on the screen (as a cursor) and print it by hitting button #0. Then hit any key and choose to have more shapes, if desired. Shape tables of the vector type must have standard indexes to work; see your *Applesoft Manual*. Rotate (ROT) 16 units for every 90 degree rotation desired (option 4). Upside-down is a rotation of 32, for example.

14. SEE COLOR BYTE #S, option 16, is only for the more advanced. If you understand the way color bytes work (see my March *inCider* column) then it might be useful to see what color bytes you are color-filling with. Hitting C during filling gets you color bytes as well as X-Y coordinates.

15. MYSTERY COLOR (option 17) merely gives random color bytes.

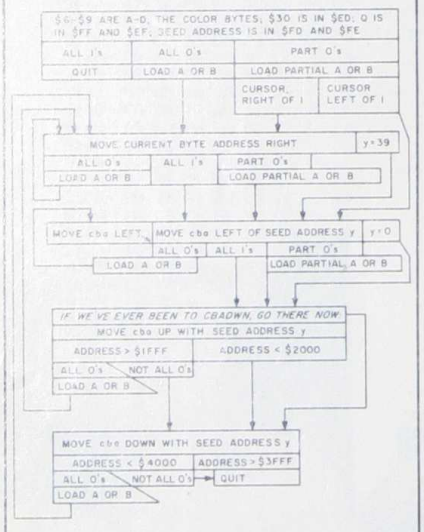


Figure 2. Fill routine flow chart.

"There will be so many possibilities for graphics creativity, you may have trouble deciding what to do next."

A Dry Run

Let's say you're running PALETTE now. Hit return when you see the notice about not erasing the screen if you hit the space bar, unless you have a line drawing already on hi-res page 1. Use option 10 to choose a color and the paddles and button #0 to select a color or pattern. When choosing, have both dots on the palette color chosen as centered as possible. Back in the menu, select option 9 if you have a binary picture line drawing to load in and color. Remember that FILL4 fills color on

black backgrounds only and that lines must be white.

Now use option 11 to fill color, with button #0 as the fill button and button #1 as the quit button. If there is nothing on the screen to fill, choose white (rightmost column) in the palette. Then when in the fill subroutine, hit button #0 to fill the screen with white, followed by the space bar, and then hit 0 to designate 0 (black) for paint color. Create lots of black rectangles to color-fill by using paddle #0 to paint, and both paddles and button #1 (held down) to relocate. Hit the space space bar to exit painting and E to exit back to menu. Use option 10 to choose a color and option 11 to go to your picture and color-fill the rectangles.

If you're not having a great time, perhaps you need a good line drawing to color-fill with. Take care of that problem in the ways I have already mentioned, or you can ask Avant-Garde for their Dot and Draw program, which has dozens of line-drawings you can fill.

How to Color-Fill

Do not fill too close to lines. Consider that the bottom of the cursor is where the filling commences. Use picture frames to avoid wraparound. If you are filling with a color that has color bytes whose values exceed 127, then you have a color-bit-on color. If the values are less than 128, then you have a color-bit-off color. Color clash (a function of Apple graphics, not Fudge programming) can happen if color-bit-on and color-bit-off colors are horizontally adjacent. Don't forget to hit C to see color bytes of the current color or X-Y coordinate positions, if necessary.

Finally, fill as cleanly as possible. This means filling at the most appropriate places in an enclosure. Refer to Figure 1. The hardest types of figures to fill are tiny ones and those shaped like stars.

The FILL4 routine in PALETTE is made with speed as priority one, variety of colors/patterns as priority two, and completeness of color-fill of a complex figure as priority three. In my opinion an adventure game whose

Listing 2. PATRNMAKER.

```
2 C(1) = 0:C(2) = 42:C(3) = 85:C(4) = 127:C(5) = 170:C(6) = 217
10  HGR2
12  GOSUB 15: GOTO 81
15 X = 16384:Q = 0:Z = 1024:E = 0:U = 2:XX = 16384
20  FOR A = 1 TO 6: FOR B = 1 TO 6: IF A - B = 0 THEN 62
30  FOR C = 1 TO 6: IF (A - C) * (B - C) = 0 THEN 61
40  FOR D = 1 TO 6: IF (A - D) * (B - D) * (C - D) = 0 THEN 60
50 N = N + 1: IF N > 480 THEN  CALL 54915: RETURN
52  POKE X,C(A): POKE X + 1,C(B): POKE X + Z,C(C): POKE Z + X + 1,C(D):X =
    X + 2048:Q = Q + 1: IF Q < 4 THEN 52
53 Q = 0:X = XX + U + E:U = U + 2: IF U < 42 THEN 60
54 U = 2:E = E + 128:X = XX + E: IF E < 1024 THEN 60
55 X = X - 984:E = 128: RETURN
60  NEXT
61  NEXT
62  NEXT
63  NEXT
81  DATA  8,17,34,68,136,145,162,196,17,34,68,8,145,162,196,136,119,110,93
    ,59,247,238,221,187,110,93,59,119,238,221,187,247,25,51,102,76,153,17
    9,230,204,51,102,76,25,179,230,204,153,42,85,42,85,170,213,170,213,85
    ,42,85,42,213,170,213,170
82  DATA  127,127,127,127,255,255,255,255,0,0,0,0,128,128,128,128,34,68,8,
    17,162,196,136,145,68,8,17,34,196,136,145,162,93,59,119,110,221,187,2
    47,238,59,119,110,93,187,247,238,221,102,76,25,51,230,204,153,179,76,
    25,51,102,204,153,179,230
83  DATA  1,21,1,22,2,21,2,22,1,5,1,6,2,5,2,6,1,13,1,14,1,15,1,16,1,17,1,1
    9,2,13,2,14,2,15,2,16,2,17,2,19,3,23,3,24,4,23,4,24,3,7,3,8,4,7,4,8,3
    ,13,3,14,3,15,3,16,3,17,3,19
84  DATA  4,13,4,14,4,15,4,16,4,17,4,19,5,25,5,26,6,25,6,26,5,13,5,14,5,15
    ,5,16,5,17,5,19,6,13,6,14,6,15,6,16,6,17,6,19,7,27,7,28,8,27,8,28,7,1
    3,7,14,7,15,7,16,7,17,7,19,8,13,8,14
85  DATA  8,15,8,16,8,17,8,19,9,29,9,30,10,29,10,30,9,13,9,14,9,15,9,16,9,
    17,9,19,10,13,10,14,10,15,10,16,10,17,10,19,11,31,11,32,12,31,12,32,1
    1,13,11,14,11,15,11,16,11,17,11,19,12,13,12,14,12,15,12,16,12,17,12,1
    9
86  DATA   5,6,6,7,7,8,5,7,5,8,6,8,5,5,6,6,7,7,8,8,13,13,13,14,13,15,13,16
    ,13,17,13,19,14,14,14,15,14,16,14,18,14,20, 15,15,15,16,15,17,15,18,1
    6,16,16,18,16,20,17,17,17,19,19,19,2,2
87  DATA  1,1,3,3,4,4,28,27
88  DIM C1(33),C2(33),C3(33),C4(33)
90  FOR A = 1 TO 32: READ C1(A),C2(A),C3(A),C4(A): NEXT
100  DIM B1(141),B2(141)
110  FOR A = 1 TO 140: READ B1(A),B2(A): NEXT
115  HCOLOR= 0
116 N = 1
117  FOR X = 0 TO 36 STEP 4
120  FOR Y = 64 TO 168 STEP 8
130  FOR A = 0 TO 6 STEP 2
135 I = B1(N):J = B2(N)
140  HPLOT 279,Y + A:AD =  PEEK (38) +  PEEK (39) * 256 + X
160  POKE AD,C1(I): POKE AD + 1,C2(I): POKE AD + 2,C3(I): POKE AD + 3,C4(I
     )
162  HPLOT 279,Y + A + 1:AD =  PEEK (38) +  PEEK (39) * 256 + X
165  POKE AD,C1(J): POKE AD + 1,C2(J): POKE AD + 2,C3(J): POKE AD + 3,C4(J
     )
168  NEXT
170 N = N + 1
180  NEXT : NEXT
190  PRINT  CHR$ (4)"BSAVEPATRN,A$4000,L$1FF8"
```

Listing 3. PALETTE.

```
0  CLEAR : HOME : TEXT : VTAB 9: HIMEM: 36864
1  ONERR  GOTO 63990
2 C(1) = 0:C(2) = 42:C(3) = 85:C(4) = 127:C(5) = 170:C(6) = 217
3  PRINT : FLASH : PRINT "IF YOU DON'T WANT SCREEN ERASED NOW, HITSPACE BA
     R; ANY OTHER KEY WILL ERASE THE SCREEN.": NORMAL
4 PP =  PEEK ( - 16384): IF PP > 127 THEN  POKE  - 16368,0: IF PP <  > 160
     THEN  HGR : GOTO 7
5  IF PP > 127 THEN 7
6  GOTO 4
7 C = 3:S = 1:R = 64:X = 139:Y = 79:D$ =  CHR$ (4): HOME :B = 0:X$ = "D": GOTO
    89
8  IF Z9 = 0 THEN Z9 = 1: HGR2 : GOSUB 15: GOSUB 300: GOTO 10
9  POKE  - 16299,0: POKE  - 16304,0: POKE  - 16297,0
10  GOSUB 190:X = 139:C = 3:B = 0: GOTO 89
15  REM
88  RETURN
89  HOME : POKE  - 16303,0: POKE  - 16298,0: HCOLOR= C: SCALE= S: ROT= R
90  PRINT : INVERSE : VTAB 1: HTAB 5: PRINT "DO YOU WANT TO:": NORMAL : PRINT
     "(0)GO TO MAIN MENU": PRINT "(1)VIEW SCREEN": PRINT "(2)SEE THEM ALL
     (VECTOR), ONE AFTER        ANOTHER": PRINT "(3)SPECIFY DRAW OR XDRAW"
     : PRINT "(4)SPECIFY ROTATION"
91  PRINT "(5)SPECIFY COLOR": PRINT "(6)SPECIFY COORDINATES": PRINT "(7)SP
    ECIFY SCALE": PRINT "(8)SPECIFY BACKGROUND": PRINT "(9)LOAD A SHAPE O
    R PICTURE": PRINT "(10)CHOOSE A PALETTE COLOR": PRINT "(11)FILL SHAPE
    S OR PICTURES"
92  PRINT "(12)MIXED-SCREEN TO FULL-SCREEN GRAPHICS(13)GO TO SCAN & SAVE S
    HAPE": PRINT "(14)SAVE 34 SECTOR SCREEN PICTURE": PRINT "(15)ERASE SC
    REEN": PRINT "(16)SEE COLOR BYTE #S": PRINT "(17)MYSTERY COLOR"
```

Listing continued.

Listing continued.

```
 95  INPUT "(TYPE 0-17):";B$: IF  LEN (B$) = 0 THEN 90
 96  IF  ASC (B$) > 57 OR  ASC (B$) < 48 THEN 90
 97 ZZ =  VAL (B$): IF ZZ < 0 OR ZZ > 17 THEN 90
 98  ON ZZ GOTO 100,110,1100,1200,1300,1600,1700,1800,400,8,900,8000,9000,1
     0000,11000,12000,13000
100  IF  ASC (B$) = 49 THEN  POKE  - 16304,0: POKE  - 16297,0: GOSUB 63000
     : HOME : GOTO 89
110  IF  ASC (B$) = 50 THEN 140
111  REM
113  IF ZZ = 0 THEN  END
140 D =  INT (AD / 256):T = D * 256:SH = AD - T
145 NU =  PEEK (AD)
150  POKE 232,SH: POKE 233,D
155  PRINT : PRINT "YOU'LL SEE FROM SHAPE 1 TO SHAPE "NU".": PRINT "HIT SP
     ACE BAR TO MOVE ON TO NEXT SHAPE.": GOSUB 63000
158  HGR : SCALE= S: ROT= R: HCOLOR= B: HPLOT 0,0: CALL 62454
159  HCOLOR= C
160  FOR QW = 1 TO NU: GOSUB 1000: VTAB 23: PRINT "SHAPE: "QW: GOSUB 63010
     : HGR : SCALE= S: ROT= R: HCOLOR= B: HPLOT 0,0: CALL 62454: HCOLOR= C
     : NEXT
161  HOME : GOTO 89
190  POKE 232,96: POKE 233,3: POKE 864,1: POKE 865,0: POKE 866,4: POKE 867
     ,0: POKE 868,60: POKE 869,54: POKE 870,0: ROT= 0: SCALE= 1: IF VV = 1
      THEN VV = 0: RETURN
195  POKE 230,64
200 P0 = 1.09 *  PDL (0):P1 = .75 *  PDL (1): IF P0 + 18 < 280 THEN  XDRAW
     1 AT P0,P1: XDRAW 1 AT P0 + 18,P1: FOR QW = 1 TO 100: NEXT : XDRAW 1 AT
     P0 + 18,P1: XDRAW 1 AT P0,P1:O =  PEEK ( - 16336)
201  IF P0 + 18 > 279 THEN  GOSUB 62090: GOTO 200
210 PP =  PEEK ( - 16287): IF PP > 127 THEN P =  PEEK (229): IF P / 2 <  >
      INT (P / 2) AND P > 0 THEN P = P - 1
215  IF PP < 128 THEN 200
220 A =  PEEK (P +  PEEK (38) +  PEEK (39) * 256):B =  PEEK (P + 1 +  PEEK
     (38) +  PEEK (39) * 256)
225 C =  PEEK (P + 2 +  PEEK (38) +  PEEK (39) * 256):D =  PEEK (P + 3 +  PEEK
     (38) +  PEEK (39) * 256)
230  XDRAW 1 AT P0,P1 + 1: XDRAW 1 AT P0,P1 + 1:E =  PEEK (P +  PEEK (38) +
      PEEK (39) * 256):F =  PEEK (P + 1 +  PEEK (38) +  PEEK (39) * 256): POKE
     6,A: POKE 7,B: POKE 8,C: POKE 9,D
232 G =  PEEK (P + 2 +  PEEK (38) +  PEEK (39) * 256):H =  PEEK (P + 3 +  PEEK
     (38) +  PEEK (39) * 256)
235  POKE 220,E: POKE 221,F: POKE 222,G: POKE 223,H
290  POKE 230,32: POKE  - 16368,0: POKE  - 16300,0: RETURN
300  PRINT  CHR$ (4)"BLOADPATRN,A$4000": RETURN
400  HOME : VTAB 9: POKE  - 16303,0: POKE  - 16298,0
410  PRINT "WHICH TYPE OF PICTURE DO YOU WISH TO    FILL:": PRINT : PRINT
     "(1)BLOCK OR HPLOT SHAPES": PRINT "(2)VECTOR SHAPES": PRINT "(3)34 SE
     CTOR SCREEN PICTURE": PRINT : INPUT "(1-3):";A: IF A < 1 OR A > 3 THEN
     410
412  IF A = 1 THEN 420
415  PRINT : INVERSE : PRINT "SWITCH TO YOUR SHAPE DISK:": NORMAL : GOSUB
     63000
420  ON A GOTO 500,600,700
500  PRINT : INPUT "SHAPE TABLE NAME: ";ST$: IF  LEN (ST$) = 0 THEN 500
505  PRINT : PRINT "IF HPLOT SHAPE DESIRED, HIT H.";: GET H$: PRINT  CHR$
     (13): IF  LEN (H$) = 0 THEN 509
506  IF  ASC (H$) = 72 THEN  PRINT D$"BLOADTEST 0 (CALL2048)": GOTO 509
508  PRINT D$"BLOADTESTTB"
509  PRINT : INVERSE : PRINT "SWITCH TO YOUR SHAPE DISK:": NORMAL : GOSUB
     63000
510 D$ =  CHR$ (4): PRINT D$"BLOAD";ST$: PRINT "ADDRESS: " PEEK (43634) +
      PEEK (43635) * 256: PRINT "LENGTH: " PEEK (43616) +  PEEK (43617) *
     256
525  PRINT : INPUT "SHAPE #: ";SN: IF SN > 23 OR SN < 1 THEN 525
530  POKE 7,SN
535  IF  ASC (H$) = 72 THEN  CALL 2048: POKE  - 16304,0: POKE  - 16297,0: GOSUB
     63010: GOTO 550
540  PRINT : INPUT "VT: ";VT: INPUT "VB: ";VB: INPUT "HR: ";HR: INPUT "HL:
      ";HL
541 VV = 1: GOSUB 190: POKE  - 16304,0: POKE  - 16297,0
542 P0 = 1.09 *  PDL (0):P1 = .75 *  PDL (1): XDRAW 1 AT P0,P1: FOR QW = 1
     TO 100: NEXT : XDRAW 1 AT P0,P1:O =  PEEK ( - 16336)
543 PP =  PEEK ( - 16287): IF PP > 127 THEN 546
544  IF PP < 128 THEN 542
546 TV =  INT (P1 - (.5 * (VB - VT))):BV = TV + (VB - VT):RH =  INT ((P0 /
     7) + (.5 * (HR - HL))):LH = RH - (HR - HL)
547  IF (TV < 0 OR BV > 191) OR (RH > 39 OR LH < 0) THEN  GOSUB 60000: POP
     : GOTO 542
548  POKE 252,TV: POKE 253,BV: POKE 254,RH: POKE 255,LH
550  CALL 2116: GET A$: PRINT  CHR$ (13): HOME : VTAB 9: POKE  - 16303,0: POKE
      - 16298,0: INPUT "DO YOU WANT ANOTHER? (Y/N): ";A$: IF  LEN (A$) = 0
      THEN 550
560  IF  ASC (A$) = 89 THEN 525
570  GOTO 89
600  PRINT : INPUT "SHAPE TABLE NAME: ";ST$: IF  LEN (ST$) = 0 THEN 600
610 D$ =  CHR$ (4): PRINT D$"BLOAD";ST$: PRINT "ADDRESS: " PEEK (43634) +
      PEEK (43635) * 256: PRINT "LENGTH: " PEEK (43616) +  PEEK (43617) *
     256
611 AD =  PEEK (43634) +  PEEK (43635) * 256:NS =  PEEK (AD)
612  PRINT "# OF SHAPES IN TABLE: "NS
620  PRINT : INPUT "SHAPE #: ";SN: IF SN > NS OR SN < 1 THEN 620
625  POKE  - 16304,0: POKE  - 16297,0
630 HI =  INT (AD / 256):LO = AD - (HI * 256)
640 VV = 1: GOSUB 190
642 P0 = 1.09 *  PDL (0):P1 = .75 *  PDL (1): XDRAW 1 AT P0,P1: FOR QW = 1
     TO 100: NEXT : XDRAW 1 AT P0,P1:O =  PEEK ( - 16336)
644 PP =  PEEK ( - 16287): IF PP > 127 THEN 649
645  IF PP < 128 THEN 642
649  POKE 232,LO: POKE 233,HI: ROT= R: SCALE= S: HCOLOR= C
650  POKE  - 16304,0: POKE  - 16297,0: DRAW SN AT P0,P1: GOSUB 63000: HOME
```

Listing continued.

scenes take a long time to fill gets old really fast.

How Color-Filling Works

In general, color-filling works like this. Bytes on the screen are inspected for on bits. If there is room to stick some or all of a color-byte into the screen-byte, it is done. Once the byte is "filled" the next screen-byte to the right is handled in the same way. If the byte found is not all 0's (black), it tells the routine to quit moving to the right because a line has been encountered. So now bytes are inspected from right to left. Once on bits (a line) are found, this particular horizontal line in the enclosed space being filled

"If there is room to stick some or all of a color-byte into the screen-byte, it is done."

is done, so we go up a line and fill. The same right-first-and-left-second sequence of inspection happens here. Once that line is filled, up again... until you hit a line that says you can no longer fill in the upward direction. Then you start filling in a downward direction, with the same right and left sequence. When you hit the bottom of the enclosed space (too many on bits to allow byte-filling) the routine ceases.

The fill location is where all this starts. But the horizontal byte column number of the fill location is also the "go-back-to" X coordinate. This means that once you have hit a right boundary you go to X equals "go-back-to" minus 1. And when you're done with leftward filling due to an encounter with a line, you go up to the next line (using the "go-back-to"

(minus 1 as Y coordinate) to use "go-back-to" as the X byte coordinate (there are 280 X coordinates at the bit level, but only 40 at the byte level). Once you are done with upward filling, use the original "go-back-to" Y coordinate plus 1 as the next inspection byte for filling.

This type of algorithm is great for speed, but don't be surprised if complex shapes or enclosures need extra fills in some places. By thinking about the "go-back-to" X coordinate parameter, you can easily visualize how a complex figure might need multiple fills. When a line is encountered while filling upward (still using right-then-left inspection sequencing for the current horizontal line) the routine goes all the way back down to the original fill location (with Y incremented by 1 since lower equals greater for Y) for further inspections. For a hand-like figure with fingers pointing upward, five fills would be needed.

See Figure 2 for a flow chart from the Hi-Res Secrets manual. It relates to the FILL1 routine, which uses only four color bytes. FILL4 not only uses $6–$9, but $DC–$DF as well (8 color bytes).

In the Figure 2 flow chart, details are omitted. "Y" means Y register where the horizontal byte coordinate (0–39) is stored. This is an X, not Y, coordinate. Also,"cba"means current

Listing continued.

```
     : VTAB 9: POKE  - 16303,0: POKE  - 16298,0: INPUT "DO YOU WANT ANOTHE
     R? (Y/N): ";A$: IF  LEN (A$) = 0 THEN 650
660  IF  ASC (A$) = 89 THEN 620
670  GOTO 89
700  PRINT : INPUT "34 SECTOR PICTURE NAME? ";PC$: PRINT D$"BLOAD";PC$: HOME
     : GOTO 89
900 UU = 1: POKE  - 16304,0: POKE  - 16297,0: GOSUB 190
902  IF U8 = 0 THEN U8 = 1: PRINT  CHR$ (4)"BLOADFILL4"
905  POKE  - 16300,0: POKE 230,32
906 Z6 = 0
907  IF U9 = 0 THEN U9 = 1: POKE  - 16303,0: POKE  - 16298,0: HOME : PRINT
     "PDL #0 TO FILL , PDL #1 TO EXIT": GOSUB 62000: POKE  - 16304,0: POKE
      - 16297,0
910 P0 = 1.09 *  PDL (0):P1 = .75 *  PDL (1): GOSUB 1500: XDRAW 1 AT P0,P1
     : FOR QW = 1 TO 100: NEXT : XDRAW 1 AT P0,P1: IF U1 = 0 THEN Z =  PEEK
     ( - 16336)
915 Q =  PEEK (229): POKE 255,0: POKE 239,Q
916 P9 =  PEEK ( - 16384): IF P9 > 127 THEN  POKE  - 16368,0: IF P9 <  > 1
    95 THEN  GOSUB 15000
917  IF P9 = 195 THEN  HOME : GOSUB 16000
918 PP =  PEEK ( - 16286): IF PP > 127 THEN  HOME : GOTO 89
920 PP =  PEEK ( - 16287): IF PP > 127 THEN 950
930  IF PP < 128 THEN 910
950  XDRAW 1 AT P0,P1: POKE 237, PEEK (48): XDRAW 1 AT P0,P1: CALL 36864
980  GOTO 910
1000  IF X$ = "D" THEN  DRAW QW AT X,Y: RETURN
1010  IF X$ = "X" THEN  XDRAW QW AT X,Y: RETURN
1020  RETURN
1100  HOME : VTAB 9: TEXT : INPUT "DO YOU WANT DRAW OR XDRAW? (D OR X):";X
    $: IF X$ <  > "D" AND X$ <  > "X" THEN 1100
1120  HOME : GOTO 89
1200  HOME : VTAB 9: TEXT : INPUT "GIVE DESIRED ROTATION (0-255):";R: IF R
     < 0 OR X > 255 THEN 1200
1220  HOME : GOTO 89
1300  HOME : VTAB 9: TEXT : INPUT "GIVE DESIRED SHAPE COLOR (0-7)-------(0
     OR 4 = BLACK; 3 OR 7 = WHITE: 1 = GREEN;2 = VIOLET; 5 = ORANGE; 6 =
    BLUE  :";C
1305  IF C < 0 OR C > 7 THEN 1300
1320  HOME : GOTO 89
1500  IF  INT ( INT (P1) / 2) <  >  INT (P1) / 2 THEN P1 = P1 - 1
1510  RETURN
1600  HOME : VTAB 9: TEXT : INPUT "GIVE DESIRED X COORD. (0-279):";X: IF X
     < 0 OR X > 279 THEN 1600
1610  HOME : VTAB 9: TEXT : INPUT "GIVE DESIRED Y COORD. (0-191):";Y: IF Y
     < 0 OR Y > 191 THEN 1610
1620  HOME : GOTO 89
1700  HOME : VTAB 9: TEXT : INPUT "GIVE DESIRED SCALE (1-255):";S: IF Y <
    0 OR Y > 255 THEN 1700
1720  HOME : GOTO 89
1800  HOME : VTAB 9: TEXT : INPUT "GIVE DESIRED BACKGROUND COLOR (0-7) (0
     OR 4 = BLACK; 3 OR 7 = WHITE: 1 = GREEN;2 = VIOLET; 5 = ORANGE; 6 =
    BLUE  :";B
1810  IF B < 0 OR B > 7 THEN 1800
1815  HCOLOR= B: HPLOT 0,0: CALL 62454: HCOLOR= C
1820  HOME : GOTO 89
8000  POKE  - 16302,0: GOTO 89
9000  FLASH : PRINT "SWITCH TO YOUR PROGRAM DISK!": NORMAL : GOSUB 63000: PRINT
    D$"RUNSCANA"
10000  PRINT : INPUT "34 SECTOR SCREEN PICTURE NAME:           ";N$: IF  LEN
    (N$) = 0 THEN 89
10002  PRINT : INPUT "DID YOU GET IT RIGHT? (Y/N):";A$: IF  LEN (A$) = 0 THEN
     10002
10004  IF  ASC (A$) = 78 THEN 10000
```

Listing continued.

Listing continued.

```
10006  FLASH : PRINT "SWITCH TO SHAPE DISK!": NORMAL : GOSUB 63000: PRINT
     D$"BSAVE";N$;",A8192,L8192": GOTO 89
11000  HOME : VTAB 9: TEXT : FLASH : INPUT "SURE YOU WANT TO ERASE SCREEN?
     (Y/N):";A$: IF  LEN (A$) = 0 THEN 11000
11010  NORMAL : IF  ASC (A$) <  > 89 THEN 89
11020  CALL 62450: GOTO 89
12000  HOME : TEXT : VTAB 3: INVERSE : HTAB 3: PRINT "YOUR COLOR #S, IN $6
     -$9 & $DC-$DF ARE:"
12010  VTAB 6: PRINT "(A) $6 = " PEEK (6): PRINT : PRINT "(B) $7 = " PEEK
     (7): PRINT : PRINT "(C) $8 = " PEEK (8): PRINT : PRINT "(D) $9 = " PEEK
     (9): PRINT : PRINT
12020  PRINT "(E) $DC = " PEEK (220): PRINT : PRINT "(F) $DD = " PEEK (221
     ): PRINT : PRINT "(G) $DE = " PEEK (222): PRINT : PRINT "(H) $DF = " PEEK
     (223)
12030  NORMAL : VTAB 23: PRINT "    (HIT ANY KEY TO CONTINUE):": GOSUB 630
     10: GOTO 89
13000  HOME : VTAB 7: HTAB 15: PRINT "MYSTERY COLOR:": INVERSE : VTAB 9: HTAB
     15: PRINT "     ": VTAB 9: HTAB 22: PRINT "     ": VTAB 11: HTAB 15: PRINT
     "     ": VTAB 11: HTAB 22: PRINT "     "
13010  FOR Q = 1 TO 20:Q1 =  INT ( RND (1) * 256): VTAB 9: HTAB 16: PRINT
     Q1;Q2 =  INT ( RND (1) * 256): VTAB 9: HTAB 23: PRINT Q2;Q3 =  INT ( RND
     (1) * 256): VTAB 11: HTAB 16: PRINT Q3;Q4 =  INT ( RND (1) * 256): VTAB
     11: HTAB 23: PRINT Q4
13015  IF Q = 20 THEN 13021
13020  VTAB 9: HTAB 16: PRINT "    ": VTAB 9: HTAB 23: PRINT "    ": VTAB 11
     : HTAB 16: PRINT "    ": VTAB 11: HTAB 23: PRINT "    ":PK =  PEEK ( -
     16336):PK =  PEEK ( - 16336): NEXT
13021  INVERSE : VTAB 14: HTAB 15: PRINT "     ": VTAB 14: HTAB 22: PRINT
     "     ": VTAB 16: HTAB 15: PRINT "     ": VTAB 16: HTAB 22: PRINT "
     "
13022 Q5 = Q1:Q6 = Q2:Q7 = Q3:Q8 = Q4: VTAB 14: HTAB 16: PRINT Q5; VTAB 14
     : HTAB 23: PRINT Q6: VTAB 16: HTAB 16: PRINT Q7: VTAB 16: HTAB 23: PRINT
     Q8
13023  VTAB 10: PRINT "$6-$9:": VTAB 15: PRINT "$DC-$DF:"
13025  POKE 220,Q5: POKE 221,Q6: POKE 222,Q7: POKE 223,Q8
13026  POKE 6,Q1: POKE 7,Q2: POKE 8,Q3: POKE 9,Q4
13030  FOR Q = 1 TO 80:PK =  PEEK ( - 16336): NEXT : NORMAL : PRINT "": VTAB
     21: GOSUB 63000: GOTO 89
15000  IF P9 = 193 OR P9 = 211 THEN V1 = 1: RETURN
15001 Z6 = 0
15010  HOME : VTAB 9: POKE  - 16303,0: POKE  - 16298,0: PRINT "PAINTBRUSH
     HCOLOR? (0-7):";: GET C$: PRINT C$: PRINT  CHR$ (13): IF  LEN (C$) =
     0 THEN 15010
15012  IF Z6 = 1 THEN  PRINT : PRINT "GIVE PAINTBRUSH HEIGHT:";: GET H$: PRINT
     H$: PRINT  CHR$ (13)
15015  IF V5 = 0 THEN V5 = 1: PRINT : PRINT "PDLS TO PAINT, PDL #1 BUTTON
     TO EXIT.": PRINT
15020  IF  VAL (C$) > 7 OR  VAL (C$) < 0 THEN 15010
15030 HT = P9 - 176:HC =  VAL (C$): HCOLOR= HC: IF Z6 = 1 THEN HT =  VAL (
     H$)
15031 Z6 = 1
15035  POKE  - 16304,0: POKE  - 16297,0
15039  IF  PDL (1) * .75 < 191 - (HT - 1) THEN P0 =  PDL (0) * 1.0942:P1 =
     PDL (1) * .75:P4 = P0:P6 = P0
15040  IF  PDL (1) * .75 < 191 - (HT - 1) THEN P0 =  PDL (0) * 1.0942:P1 =
     PDL (1) * .75
15041 P8 = P4:P4 = P0
15042 Z3 = 1: IF P6 > P0 THEN Z3 =  - 1
15043 PK =  PEEK ( - 16286): IF PK > 127 THEN P0 =  PDL (0) * 1.0942:P1 =
     PDL (1) * .75: XDRAW 1 AT P0,P1: FOR Q = 1 TO 100: NEXT : XDRAW 1 AT
     P0,P1: GOTO 15039
15045  IF  PDL (1) * .75 < 191 - (HT - 1) THEN P2 =  PDL (0) * 1.0942:P3 =
     PDL (1) * .75: FOR P = P6 TO P2 STEP Z3: HPLOT P,P3 TO P,(P3 + (HT -
     1)) * (P3 = (HT - 1) > 0) * (P3 + (HT - 1) < 192): NEXT
15050 PP =  PEEK ( - 16384): IF PP > 127 THEN  POKE  - 16368,0: GOTO 15070
15060  GOTO 15040
15070  POKE  - 16303,0: POKE  - 16298,0: HOME : VTAB 9: INPUT "WANT TO PAI
     NT SOME MORE OR FILL OR EXIT?(TYPE P OR F OR E):";A$: IF  LEN (A$) =
     0 THEN 15070
15080  IF  ASC (A$) = 69 THEN  POP : GOTO 89
15082  IF  ASC (A$) = 70 THEN  POKE  - 16304,0: POKE  - 16297,0: RETURN
15084  IF  ASC (A$) = 80 THEN 15010
15086  GOTO 15070
16000  POKE  - 16303,0: POKE  - 16298,0: VTAB 7: PRINT "COORDINATES:": PRINT
     : PRINT "X = " INT (P0): PRINT "Y = " INT (P1): PRINT : PRINT "COLOR
     BYTES:": PRINT
16010  PRINT  PEEK (6)"  " PEEK (7)"  " PEEK (8)"  " PEEK (9): PRINT  PEEK
     (220)"  " PEEK (221)"  " PEEK (222)"  " PEEK (223): GOSUB 63000: POKE
     - 16304,0: POKE  - 16297,0: RETURN
60000  FOR QW = 1 TO 10:PM =  PEEK ( - 16336): NEXT : RETURN
62000  PRINT : PRINT "HIT A TO STOP CLICKS.": PRINT : PRINT "HIT 1-9 TO GE
     T HOR. PAINTBRUSH OF THAT  HEIGHT & HIT 0-7 FOR HI-RES COLORS 0-7.": PRINT
     : PRINT "HIT SPACE BAR TO EXIT PAINTING."
62005  PRINT : PRINT "(HITTING SPACE BAR WHEN IN FILL MODE    WILL GET YOU
     PAINTBRUSH HEIGHT OF 18----THEN YOU MERELY CHOOSE HI-RES COLOR.)"
62010  PRINT : PRINT "TO MOVE PDLS WITHOUT PAINTING, HOLD DOWNPDL BUTTON #
     1."
62020  PRINT : PRINT "HIT C FOR COORD.S  & COLOR BYTES."
62030  GOTO 63000
62090  TEXT : HOME : PRINT "TURN PADDLE #0 COUNTERCLOCKWISE!!": FOR Q0 = 1
     TO 18:QC =  PEEK ( - 16336): NEXT : POKE  - 16304,0: POKE  - 16297,0
     : POKE  - 16299,0: RETURN
63000  PRINT : PRINT "(HIT ANY KEY TO CONTINUE):": PRINT
63010 PP =  PEEK ( - 16384): IF PP > 127 THEN  POKE  - 16368,0: RETURN
63020  GOTO 63010
63990  POKE 216,0:KP =  PEEK (222)
63991  ONERR  GOTO 63990
63995  IF KP = 254 THEN  RESUME
63997  PRINT "";"";""
63998  CALL 54915
63999  GOTO 89
```

byte's address ($2000–$3FFF), and "CBADWN" means the routine that checks downward for the next screen byte (cba) inspection.

"A or B" refers to the fact that in FILL1 there are two different horizontal color bytes (FILL4 has four) and I have named one A and one B. If you put A where B belongs, it changes color, so take care.

"Y = 39" is a place where you hit the right edge of the screen while moving to the right (0–39 are the only possible X-byte column coordinates), and "Y = 0" refers to the left edge of the screen's byte column.

"For a hand-like figure with fingers pointing upward, five fills would be needed."

Seed address is the starting coordinate's screen byte and the "go-back-to" parameter in both X and Y coordinate questions. ">$3FFF" refers to the bottom edge of the screen and "<$2000" refers to the top edge of the hi-res page 1 screen.

The entire block that contains "address>$1FFF" and "address<$2000," etc., is called CBAUP since you are inspecting upward at this point. The entire block below the CBAUP block is the CBADWN block and contains "address>$3FFF" and other goodies. The block above CBAUP is the CBALFT block and the one above that is the CBARGT block (left and right respectively). Since right, left, up and then down is the sequence of inspection, you can see that this flow chart is to be read from top to bottom.

Next month's column is going to be a surprise—I'm not telling! ■