The Apple PAINTBOX

by Al Evans
1206 Karen Ave.
Austin, TX 78757

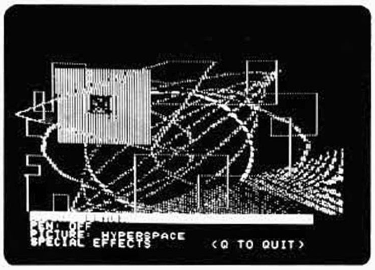
SYSTEM REQUIREMENTS: Apple II Plus or Apple II with firmware Applesoft or Language Card, game paddles, 32K, 1 disk drive

Once there was a high-resolution graphics program for the APPLE II which turned the television into a combination drawing board, Etch-a-Sketch, and Spirograph. Push a button, an orange line would appear; press a key, a star would be drawn.

"Fascinating!" everybody said.

Unfortunately, the program began with one meager screenful of incomprehensible instructions, and the only feedback offered during a run was an occasional low "Boop" to indicate that a command had been carried out. Nobody, including the programmer, could remember what the commands were from one run to the next.

That programmer (me, if you hadn't guessed) eventually learned that complex programs, whether used for business or fun, must have simple, interactive, error-proof control routines. Their instructions should provide enough data to make operation possible without outside documentation.



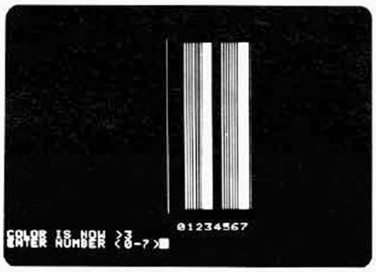
PAINTBOX is a result of this learning experience. We'll discuss the techniques used a little further on. First, the program.

HOW TO USE PAINTBOX

When you run PAINTBOX, you'll see a title page with the query "INSTRUCTIONS?". Answer "Y" to get a short instruction manual. Press <RETURN> after reading each page to get the next. At the end, press "A" to read the instructions again or <RETURN> to proceed.

After the instructions (or if you answered the initial query with "N") you will see a blank Hi-Res screen with a four-line menu at the bottom. The arrow keys step forward and backward through the menu. The <RETURN> key selects the item presently "lit up".

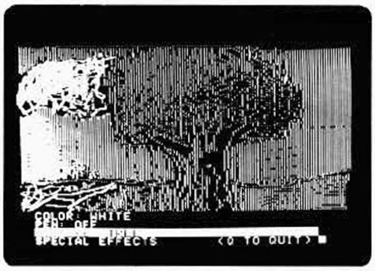
First, select a color (no point in drawing in black on black). Press the arrow keys until the **COLOR:** line on the menu lights up, press <RETURN>, and enter the first letter of one of the colors shown. Note that the menu now shows the color you have selected.



The **<ESC> key** switches back and forth from the menu to the full graphics page. Press it to make the menu disappear, and note the flashing dot somewhere on the screen. The paddle controls move this dot. Place it in an artistically satisfying position and push the button on PDL(0). This should leave a mark on the screen. Now move the flashing "cursor" somewhere else and press the button on PDL(1). The two locations should now be connected by a line in the color you chose.

Move again and push the button on PDL(1). The new starting point is the old ending point. The button on PDL(0) resets the starting point to the present position of the flashing dot; the button on PLD(1) draws a line to the present starting point.

Now push the button on PDL(0) and press <ESC> to return to the menu. Select a new color if you wish. Use the arrow keys to move to the line which says **PEN:OFF** and press <RETURN> to turn the pen on. Press <ESC> and move the paddle controls. You now have a color "Etch-a-Sketch".



The next item on the menu is used to **NAME** or **ERASE** a picture and to load from or save to a disk. When saving a picture, be sure your disk has plenty of room; each picture takes up 34 sectors.

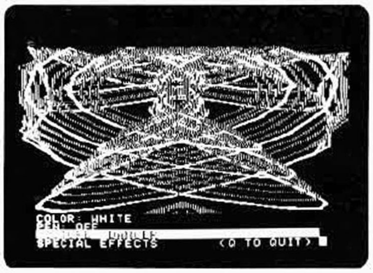
The last line of the menu is **SPECIAL EFFECTS**. Special effects presently in the program are **FILL** and **CURVES**.

FILL is intended for filling spaces with a solid color, and will do so when 4 dots are entered in the order shown in the instructions. It does other things, some of them very pretty, when different entry orders are used.

CURVES was designed as a sort of hyper-Spirograph, for those of you familiar with classical toys. Two dots entered as shown in the instructions control the location and size of the figure drawn. Four numbers are entered to determine its shape. Precisely what happens is very difficult to describe in words. Experiment!

If you quit accidentally or hit RESET, enter 3D0G <RETURN> if required to return to Applesoft, then enter GOTO 1000<RETURN> to re-enter the program without erasing your picture.

Finally, press "Q" to quit, and we'll get back to the question of program operation.



CONTROL ROUTINES AND INSTRUCTIONS

Although much simpler in operation than the drawing functions, the control routines proved much trickier to design.

Complicating the problem in this particular case is the fact that the Hi-Res graphics display allows only 4 lines of text. The original program used 10 different commands and the present version uses 13. In addition, Applesoft storage space when Hi-Res page 1 is used extends from location 2049 to location 8192 — a total of 6144 bytes for both program and workspace. Here's how PAINTBOX makes maximum use of the memory and display space available.

Lines 1000–1085 of the program operate the menu. The variable F is set by the right- and left-arrow keys to 1, 2, 3, or 4 (lines 1060–1065). This variable is used to print the menu line currently selected in black-on-white (lines 1015, 1025, 1035, 1045). When a carriage return is entered (line 1070), F points to the routine to be executed.

This menu has several advantages. First, it can be completely controlled by 3 keys, which makes it extremely easy to use. Second, the approach of "lighting up" the selections greatly reduces the probability of an incorrect entry. Even though the absolute number of keystrokes required may be greater (for example, if there are 10 or 12 possible selections), most people find this type of menu faster to use than the usual "pick a number" kind. The number of steps required is reduced by placing the items most often selected near the top or bottom of the menu. Finally, there is no possible way, short of pushing <RESET>, to input something the menu doesn't understand.

This particular menu allows two other entries: <ESC> (line 1075) transfers control to the main drawing loop of the program (lines 50–105). "Q" (line 1080) exits the program after restoring the computer to its initial state (line 15 has saved the original HIMEM address and set HIMEM:8192 to protect the graphics; line 1010 has set the text window to the bottom four lines only).

The menu also uses COL$, SW$, and PIC$, which are changed by the related routines (lines 200–250, 300–320, and 400–480), to provide a constant display of the state of the system. The final result is a simple method of controlling a versatile program.

TEXT FILE INSTRUCTIONS

The next problem was where to put a rather long set of instructions for what had to be a rather short program. My solution was to leave the instructions out of the program entirely. Instead, I used lines 2100–2225 to read a standard Apple sequential text file from disk. (Note: these can be deleted if you use the instructions reprinted here. Otherwise use the Listing #2 to create the Text file.)

This type of file can be written by many text editors and word processors, or with the short program provided (listing 3). (I used Gary Shannon and Bill Depew's "Magic Window".)

This technique has several advantages over the usual method of including the instructions in the program itself. First, it conserves memory. This is particularly important in the present case, where the instructions are long and the workspace short. Second, it is much easier to format screens of information with a text editor than by using BASIC print statements. Changes and corrections are simpler, too. Third, the same routine can be used to read and display any set of instructions (or any other sequential text file) by simply changing the name of the file.

Line 2105 initializes the line count (LC) and the string variable (I$) used to hold each instruction line. Line 2110 causes the program to jump to line 2200 when the end of data is reached. Lines 2115 and 2120 start the file-reading process. Note that an extra PRINT is used before each disk operation. This is required by Apple DOS to cancel out preceding GET's.

Line 2125 reads characters from the file until it finds a carriage return [CHR$(13)] indicating the end of a line. Line 2130 prints each line and increments the line counter until the screen is full (LC = 22). Lines 2135-2145 cancel the read operation, wait for a <RETURN>, reset the line counter, and restart the process.

When an OUT OF DATA condition is reached (i.e. at the end of the file), the ONERR GOTO in line 2110 causes a jump to line 2200. Lines 2200-2225 finish up the presentation, close the file, clear the ONERR GOTO condition, and wait for the user to choose the next operation.

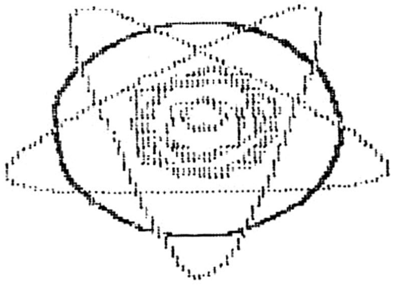
PAINTBOX INSTRUCTIONS

Listing 3 is a printout of the PAINTBOX INSTRUCTIONS file. In this listing, I have substituted periods for all blank spaces used in formatting. This was done simply to make counting easy if you are using TEXTFILE WRITER (listing 2) to make up the PAINTBOX INSTRUCTIONS file. Replace the periods with spaces or carriage returns, as required, in your actual text file.

TEXTFILE WRITER can be used to write any sequential text file. Enter each line (including any spaces), followed by a <RETURN>. You can backspace and make corrections before pressing <RETURN>. <RETURN> alone will write a blank line. After input is complete, enter the word "<FINISHED>" exactly as shown and press <RETURN> to complete the operation.

DRAWING ROUTINES

The basic draw loop (lines 50-105) shares control of the program with the menu. Control is transferred to the menu by line 55 when <ESC> is pressed. Lines 60 and 65 save the last X and Y values and read new ones from the paddles. If the "PEN:" switch (SW) is on (lines 300-320) line 70 short-circuits the loop and goes directly to the subroutine which draws a line and updates the array H(4),V(4). This array is used to store the last four endpoints entered (lines 150-190).

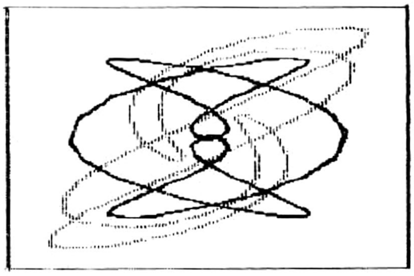


Rose in Atom

If the **PEN** is off, line 75 checks the button on PDL(0). If this button is not pressed [PEEK(B1)<127], lines 90 and 95 plot a white dot in the present position and erase the previous dot. If the button has been pressed, line 80 places a white point 1 dot away from the current position (so it won't be erased) and updates the array of "significant" points. Line 85 simply provides time to release the button.

Line 100 tests whether the button on PDL(1) has been pressed. If so, a line is drawn from the most recent endpoint to the present position.

The **FILL** routine (lines 700-810) uses the last four endpoints entered in the H and V arrays to fill in a space which is assumed to be bounded by four points entered in a certain order. However, complete range-checking (lines 740-745, 755-760, etc.) allows the points to be entered in any order. This makes it possible to obtain a variety of other interesting effects from the same routine.



It would be nearly impossible to describe the **CURVES** routine (lines 535-650) in words alone. Figure 1 is a diagram of approximately what this routine would "look like" if it were a mechanical device.

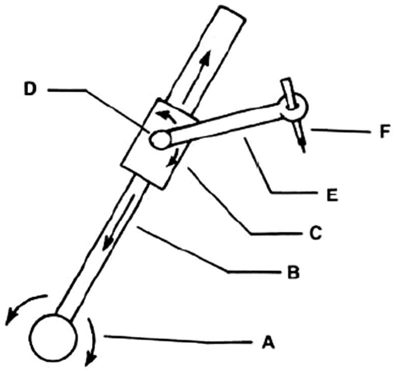


Figure 1

In this Figure, main arm B rotates about main axis A. Carriage C slides back and forth on arm B during rotation. Meanwhile, secondary arm E (with a length determined by "% AMPLITUDE") rotates about secondary axis D at a rate determined by "FREQUENCY". "Pen" F does the actual drawing.

Lines 570-645 implement this action as a polar plotting routine. Again, complete range checking is provided. In this case, lines 605-620 cause the "pen" to be "reflected" off the top, bottom, and sides of the screen. Pushing the button on PDL(1) terminates execution of the routine.

CONCLUSION

As shown, PAINTBOX uses approximately 4700 bytes of memory, leaving about 1300 bytes free for additions and extensions. One improvement would be to extend the LOAD and SAVE error-handling routines (lines 484-496), which presently assume FILE NOT FOUND and DISK FULL errors, respectively. A fairly simple addition could be made to display a complete catalog when one of these errors is encountered. Refer to the Applesoft and DOS manuals for the techniques and error codes involved.

But enough of technicalities! Once you've typed in the program, you're an electronic artist. Turn those knobs, push those buttons, and create your first computer video masterpiece!

Listing 2 — TEXTFILE WRITER

```
10  REM  TEXTFILE WRITER
15  REM  BY AL EVANS, 1980
20 D$ =  CHR$ (4):I$ = ""
30 F$ = "PAINTBOX INSTRUCTIONS": REM  NAME OF FILE TO BE WRITTEN
50  PRINT D$;"OPEN";F$: PRINT D$;"DELETE";F$: PRINT D$;"OPEN";F$
60  GET CH$: PRINT CH$;
65  IF CH$ =  CHR$ (21) AND I$ <  > "" THEN I$ =  LEFT$ (I$, LEN (I$) - 1)
   : POKE 36, POS (0) - 1
70  IF CH$ <  >  CHR$ (13) THEN I$ = I$ + CH$: GOTO 60
75  IF I$ = "<FINISHED>" THEN 100
80  PRINT D$: PRINT D$;"APPEND";F$: PRINT D$;"WRITE";F$: PRINT I$:I$ = "":
    GOTO 60
100  PRINT : PRINT D$;"CLOSE";F$
105  PRINT "FINISHED": END
```

Listing 3 — PAINTBOX INSTRUCTIONS File (Blanks replaced with periods for easy unformatted entry)

```
........................................
1. PRESS <ESC> TO GET TO AND FROM MENU.
...SELECT A MENU ITEM WITH THE ARROW
...KEYS. PRESS <RETURN> TO CARRY OUT
...THE ACTION SELECTED.
........................................
2. SELECT THE ITEM "COLOR:" AND PRESS
...<RETURN>. CHOOSE A COLOR.
........................................
3. DRAWING WITH PEN:OFF
........................................
...PRESS <ESC> TO LEAVE THE MENU. THE
...PADDLE CONTROLS [PDL(0) AND PDL(1)]
...MOVE A SPOT ON THE SCREEN. BUTTON
...0 MAKES A MARK; BUTTON 1 DRAWS A
...LINE TO THAT MARK IN THE CHOSEN
...COLOR. THE OLD ENDING POINT BECOMES
...THE NEW STARTING POINT.
........................................
........................................
........................................
4. DRAWING WITH PEN:ON
........................................
...PUSH BUTTON 0 TO MAKE A MARK ON THE
...SCREEN, THEN PRESS <ESC> TO GET TO
...THE MENU. SELECT THE ITEM "PEN:"
...WITH THE ARROW KEYS AND PRESS
...<RETURN>. THE PADDLE CONTROLS NOW
...WORK LIKE THE KNOBS ON AN
..."ETCH- A-SKETCH".
........................................
5. SPECIAL EFFECTS: F(ILL
........................................
...FILL IS USED TO FILL AREAS WITH A
...SOLID COLOR. USE BUTTON 0 TO SET 4
...POINTS IN THE FOLLOWING ORDER:
.............1.........2
.............*.........*
........................................
........................................
.........................................
...........*...........*
...........3.............4
...THEN PRESS <ESC> TO GET THE MENU,
...SELECT "SPECIAL EFFECTS", AND
...PRESS "F" FOR FILL. THE AREA INSIDE
...THE 4 POINTS WILL BE FILLED WITH THE
...SELECTED COLOR. OF COURSE, THE
...POINTS CAN BE ENTERED IN OTHER
...ORDERS, WHEREUPON OTHER THINGS WILL
...HAPPEN. TRY IT.
........................................
6. SPECIAL EFFECTS: C(URVES
........................................
...THIS SPECIAL EFFECT WORKS SOMEWHAT
...LIKE A SPIROGRAPH. MARK 2 POINTS
...WITH BUTTON 0 IN THIS FORMAT:
........................................
................. 1
..................*
........................................
........................................
..................*
..................2
........................................
...THESE POINTS DETERMINE THE SIZE AND
...LOCATION OF THE CURVE. PRESS <ESC>
...TO GET THE MENU, CHOOSE "SPECIAL
...EFFECTS", AND PRESS "C" FOR CURVES.
...ENTER VALUES FOR A AND B (THESE
...DETERMINE THE SHAPE OF THE CURVE).
...NEGATIVE NUMBERS INVERT THE CURVE.
...THE CURVE "BOUNCES OFF" THE TOP,
...BOTTOM, & SIDES OF SCREEN. NEXT
...ANSWER "MODULATE THE CURVE?" WITH Y
...OR N. MODULATION ALTERS THE BASIC
...SHAPE. ENTER NUMBERS FOR "% AMPLI-
...TUDE" AND "FREQUENCY". USING 3,3 FOR
...A AND B (WHICH, INCIDENTALLY,
...PRODUCES A CIRCLE), TRY:
........................................
.......AMPLITUDE=20,FREQUENCY=9
.......AMPLITUDE=30,FREQUENCY=9
.......AMPLITUDE=30,FREQUENCY=-9
........................................
........................................
........................................
........................................
........................................
...OTHER INTERESTING SHAPES ARE
...PRODUCED BY A=4,  B=4;
................A=10, B=23;
................A=33, B=33;
...........AND A=12, B=6.
........................................
...FINALLY, TRY A=-3,B=-3,AMPLITUDE=30,
...FREQUENCY=3. SURPRISE!!
........................................
................**NOTE**
........................................
.BUTTON 1 CAN BE USED TO BREAK OUT OF
......EITHER EFFECT AT ANY TIME.
.IN CASE OF ACCIDENTAL EXIT, RETURN TO
..BASIC BY ENTERING 3D0G <RETURN> IF
REQUIRED, THEN TYPE GOTO 1000 <RETURN>.
........................................
........................................
```

```
10  GOTO 2000
15 SW$ = "OFF":ML =  PEEK (115):MH =  PEEK (116): HIMEM:
     8192
20 COL = 0:COL$ = "BLACK      "
25 PIC$ = "PICTURE NOT NAMED        "
30 KBD =  - 16384:SB =  - 16368:B1 =  - 16287:B2 =  - 16
     286:T = 3:K = 1.18
35 [illegible]28318531:S = .03498066585: DIM H(4),V(4):D$ =

4[illegible]5:COL = 0:COL$ = "BLACK      ":PIC$ = "PIC
     [illegible]MED        "
45  HGR : GOTO 1000
50  REM  BASIC DRAW LOOP
55  IF  PEEK (KBD) = 155 THEN  POKE SB,0: GOTO 1000
60 X2 = X1:Y2 = Y1
65 X1 =  INT ( PDL (0) * 277 / 255 + 1):Y1 =  INT ( PDL
     (1) * 189 / 255 + 1)
70  IF SW THEN  GOSUB 150: GOTO 50
75  IF  PEEK (B1) < = 127 THEN 90
80 XO = X1 + 1:YO = Y1 + 1: HCOLOR= 3: HPLOT XO,YO: GOSUB
     175
85  FOR D = 1 TO 500: NEXT
90  HCOLOR= 3: HPLOT X1,Y1
95  HCOLOR= 0: HPLOT X2,Y2
100  IF  PEEK (B2) > 127 THEN  GOSUB 150
105  GOTO 50
150  REM  PLOT AND UPDATE
155 X3 = X1:Y3 = Y1
160  HCOLOR= COL
165  HPLOT XO,YO TO X3,Y3: HPLOT XO - 1,YO TO X3 - 1,Y3
170 XO = X3:YO = Y3
175  REM  ENTER HERE TO UPDATE ONLY
180 H(0) = XO:V(0) = YO
185  FOR I = 4 TO 1 STEP  - 1:H(I) = H(I - 1):V(I) = V(I
      - 1): NEXT
190  RETURN
200  REM  SELECT COLOR
205  HOME
210  PRINT "COLOR: B(LACK, L(IGHT BLUE, W(HITE,": PRINT
     "       O(RANGE, P(URPLE, G(REEN"
215  GET CH$
220  IF CH$ = "B" THEN COL$ = "BLACK      ":COL = 0: GOTO
     1000
225  IF CH$ = "L" THEN COL$ = "LIGHT BLUE":COL = 6: GOTO
     1000
230  IF CH$ = "W" THEN COL$ = "WHITE      ":COL = 3: GOTO
     1000
235  IF CH$ = "O" THEN COL$ = "ORANGE     ":COL = 5: GOTO
     1000
240  IF CH$ = "P" THEN COL$ = "PURPLE     ":COL = 2: GOTO
     1000
245  IF CH$ = "G" THEN COL$ = "GREEN      ":COL = 1: GOTO
     1000
250  GOTO 205
300  REM  TURN PEN ON AND OFF
305 SW =  NOT SW
310  IF  NOT SW THEN SW$ = "OFF"
315  IF SW THEN SW$ = " ON"
320  GOTO 1000
400  REM  CHANGE PICTURES
404  HOME
408  PRINT "PICTURES: N(AME, L(OAD, S(AVE,": PRINT "
          E(RASE, (ESC)"
412  ONERR  GOTO 484
416  GET CH$
420  IF CH$ =  CHR$ (27) THEN  POKE 216,0: GOTO 1000
424  IF CH$ = "E" THEN  HOME : PRINT "ARE YOU SURE? ";: GET
     CH$: IF CH$ = "Y" THEN PIC$ = "PICTURE NOT NAMED
        ": HGR : POKE 216,0: GOTO 1000
428  IF CH$ = "N" THEN 460
432  IF CH$ = "S" THEN 476
436  IF CH$ < > "L" THEN  POKE 216,0: GOTO 404
440  HOME : PRINT "NAME OF PICTURE TO LOAD:"
444  INPUT "";PIC$: IF  LEN (PIC$) > 25 THEN PIC$ =  LEFT$
     (PIC$,25)
448  IF  LEN (PIC$) < 25 THEN PIC$ = PIC$ + " ": GOTO 44
     8
452  PRINT D$;"BLOAD PBX.";PIC$;",A$2000"
456  POKE 216,0: GOTO 1000
460  HOME : PRINT "NAME THIS PICTURE:"
464  INPUT "";PIC$: IF  LEN (PIC$) > 25 THEN PIC$ =  LEFT$
     (PIC$,25)
468  IF  LEN (PIC$) < 25 THEN PIC$ = PIC$ + " ": GOTO 46
     8
472  POKE 216,0: GOTO 404
476  PRINT : PRINT D$;"BSAVE PBX.";PIC$;",A$2000,L$1FFF"
480  POKE 216,0: GOTO 1000
484  REM  LOAD/SAVE ERRORS
488 EC =  PEEK (222): IF EC = 6 THEN  PRINT "NOT ON THIS
      DISK": FOR D = 1 TO 1500: NEXT : POKE 216,0: GOTO
     404
492  IF EC = 9 THEN  PRINT D$;"DELETE PBX.";PIC$: PRINT
     "THIS DISK IS FULL": FOR D = 1 TO 1500: NEXT : POKE
     216,0: GOTO 404
496  PRINT "ERROR ENCOUNTERED. CHECK DISK, DRIVE,   SYNT
     AX AND TRY AGAIN.": FOR D = 1 TO 2500: NEXT : POKE
     216,0: GOTO 404
500  REM  SPECIAL EFFECTS
505  HOME
510  PRINT "SPECIAL EFFECTS: F(ILL, C(URVES, (ESC)"
515  GET CH$
520  IF CH$ =  CHR$ (27) THEN 1000
525  IF CH$ = "F" THEN 700
530  IF CH$ < > "C" THEN 505
535 Z =  ABS (V(2) - V(1))
540  HOME : INPUT "ENTER NUMBER FOR 'A' AND (RETURN): ";
     A
545  INPUT "ENTER NUMBER FOR 'B' AND (RETURN): ";B
550  PRINT "MODULATE THIS CURVE? ";: GET CH$: PRINT : IF
     CH$ < > "Y" THEN M = 0:N = 0: GOTO 565
555  INPUT "ENTER % AMPLITUDE AND (RETURN): ";M:M = M /
     100 * Z
560  INPUT "ENTER FREQUENCY AND (RETURN): ";N
565  HCOLOR= COL
570  FOR TH = 0 TO P2 STEP S
575 R = Z *  SIN (TH * T)
580 X2 = K * R *  COS (A * TH) + H(2)
585 X2 =  ABS (X2 + (K * M *  COS (N * TH * T)))
590 Y2 = R *  SIN (B * TH) + V(2)
595 Y2 =  ABS (Y2 + (M *  SIN (N * TH * T)))
600  IF TH = 0 THEN X1 = X2:Y1 = Y2
605  IF X1 > 278 THEN X1 = 278 - (X1 - 278)
610  IF Y1 > 190 THEN Y1 = 190 - (Y1 - 190)
615  IF X2 > 278 THEN X2 = 278 - (X2 - 278)
620  IF Y2 > 190 THEN Y2 = 190 - (Y2 - 190)
625  HPLOT X1,Y1 TO X2,Y2
630  HPLOT X1 + 1,Y1 TO X2 + 1,Y2
635 X1 = X2:Y1 = Y2
640  IF  PEEK (B2) > 127 THEN 1000
645  NEXT TH
650  GOTO 1000
700  REM  FILL
705 A = V(2) - V(4): IF A = 0 THEN A = 1E - 6
710 B = V(1) - V(3): IF B = 0 THEN B = 1E - 6
715 C = H(2) - H(4): IF C = 0 THEN C = 1E - 6
720 D = H(1) - H(3): IF D = 0 THEN D = 1E - 6
725  HCOLOR= COL
730  FOR N = 0 TO A
735 X9 = H(4) + N * C / A
740  IF X9 < 0 THEN X9 = 0
745  IF X9 > 279 THEN X9 = 279
750 Y9 = V(4) + N
755  IF Y9 < 0 THEN Y9 = 0
760  IF Y9 > 191 THEN Y9 = 191
765 X8 = H(3) + N * D / B
770  IF X8 < 0 THEN X8 = 0
775  IF X8 > 279 THEN X8 = 279
780 Y8 = V(3) + N * B / A
785  IF Y8 < 0 THEN Y8 = 0
790  IF Y8 > 191 THEN Y8 = 191
795  HPLOT X9,Y9 TO X8,Y8
800  IF  PEEK (B2) > 127 THEN 1000
805  NEXT N
810  GOTO 1000
1000  REM  NEW PAINTBOX MENU
1005 F = 1
1010  POKE  - 16304,0: POKE  - 16297,0: POKE  - 16301,0:
      POKE 34,20: HOME
1015  IF F = 1 THEN  INVERSE
1020  HTAB 3: PRINT "COLOR: ";COL$;"                    ": NORMAL
1025  IF F = 2 THEN  INVERSE
1030  HTAB 3: PRINT "PEN: ";SW$;"
        ": NORMAL
1035  IF F = 3 THEN  INVERSE
1040  HTAB 3: PRINT "PICTURE: ";PIC$: NORMAL
1045  IF F = 4 THEN  INVERSE
1050  HTAB 3: PRINT "SPECIAL EFFECTS       (Q TO QUIT) "
      ;: NORMAL
1055  GET CH$
1060  IF CH$ =  CHR$ (21) THEN F = F + 1: IF F > 4 THEN
      F = F - 4
1065  IF CH$ =  CHR$ (8) THEN F = F - 1: IF F < 1 THEN F
      = F + 4
1070  IF CH$ =  CHR$ (13) THEN  ON F GOTO 200,300,400,50
     0
1075  IF CH$ =  CHR$ (27) THEN  POKE  - 16302,0: GOTO 50

1080  IF CH$ = "Q" THEN  POKE 115,ML: POKE 116,MH: POKE
      34,0: TEXT : POKE  - 16298,0: HOME : END
1085  GOTO 1010
2000  REM  TITLE & INSTRUCTIONS
2005  HOME
2010  VTAB 11: HTAB 13: INVERSE : PRINT "---PAINTBOX---"
      : NORMAL
2015  PRINT : HTAB 8: PRINT "COPYRIGHT 1980 BY AL EVANS"

2025  VTAB 24: HTAB 18: PRINT "INSTRUCTIONS?";: GET CH$:
      IF CH$ = "Y" THEN 2100
2030  CLEAR : GOTO 15
2100  REM  GET INSTRUCTIONS FROM DISK
2105 LC = 0:I$ = "":D$ =  CHR$ (4): HOME
2110  ONERR  GOTO 2200
2115  PRINT : PRINT D$;"OPEN PAINTBOX INSTRUCTIONS"
2120  PRINT : PRINT D$;"READ PAINTBOX INSTRUCTIONS"
2125  GET CH$: IF CH$ < >  CHR$ (13) THEN I$ = I$ + CH$
      : GOTO 2125
2130  PRINT : PRINT I$:I$ = "":LC = LC + 1: IF LC < 22 THEN
      2125
2135  PRINT : PRINT D$: HTAB 11: INVERSE : PRINT "(RETUR
     N) FOR MORE";: NORMAL
2140  GET CH$: IF CH$ < >  CHR$ (13) THEN 2140
2145 LC = 0: GOTO 2120
2200  PRINT : VTAB 24: HTAB 15: PRINT "**THE END**"
2205  PRINT "PRESS 'A' TO SEE INSTRUCTIONS AGAIN"
2210  PRINT "PRESS (RETURN) TO START PAINTING "
2215  PRINT D$;"CLOSE PAINTBOX INSTRUCTIONS"
2220  GET CH$: IF CH$ = "A" THEN 2100
2225  POKE 216,0: CLEAR : GOTO 15
```