PROGRAMMING AN ART PACKAGE

Ever fancied writing your own art package? Well in this tutorial I am going to show you how to program a few options that you would normally find in an art package. I have written a full art package called Grafix Art and the routines printed here are taken straight from it.

First of all, this is the routine for the normal freehand draw tool.

4410 rem DRAW

4420 hide on : off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : show on : K=1

4430 set mark 1,1

4440 wait 10 : repeat

4450 X=x mouse : Y=y mouse

4460 if hardkey=97 then hide : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : fastcopy logic,back : show

4470 until mouse key=1 or mouse key=2

4480 hide on : SCR$(SC)=screen$(logic,0,0 to 330,200)

4490 if mouse key=2 then hide on : wait 10 : goto 230

4500 if mouse key=1 then polymark X,Y

4520 repeat

4540 X=x mouse : Y=y mouse

4550 if mouse key=1 then polyline to X,Y

4560 until mouse key=0

4570 fastcopy logic,back

4580 show on : goto 4440

In other STOS art packages, people have tried using the PLOT command for this routine. However, this causes the line to draw broken. What happens here is we use polymarks to make the lines. The opaque command is from the EXTRA extension and what it does is places an image in a string put there using SCREEN$ onto the screen as a normal screen copy and not merged. In an art package, we have workscreens to

place our pictures in so we can design so many different pictures at a time and flick through them. The routine starts by waiting for the user to click the left mouse button before it will draw a polymark. The right mouse button allows you to abort the operation.

This next routine will draw a continuous line.

4590 rem LINE

4600 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : K=1 : set line val(LTYPE$),L_SIZE,0,0 : show on

4610 clear key : wait 10 : repeat

4620 X=x mouse : Y=y mouse : M=mouse key : K$=inkey$

4630 if asc(K$)=0 and scancode=97 then hide : opaque screen logic,0,0,SCR$(SC) : fastcopy logic,back : show

4640 wait vbl : until M

4650 if M=2 then hide on : wait 10 : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : cls : goto 200

4660 Y1=Y : X1=X : hide on : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200)

4670 logic=back : wait 10 : repeat

4680 opaque screen back,0,0,SCR$(SC) : polyline X,Y to X1,Y1

4690 X1=x mouse : Y1=y mouse

4700 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

4710 screen swap : wait vbl

4720 until mouse key=1

4730 logic=physic

4740 show on : goto 4610

You'll notice in these routines that we have implanted an UNDO feature. This means that if you are not happy with the last thing you have done you can erase it from the screen before you carry on. Note that the routine will put the current picture into a screen with SCREEN$. This is your present work screen.

Drawing a box is easy as well, here is the routine.

4750 rem BOX

4760 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : K=1 : set line val(LTYPE$),L_SIZE,0,0 : show on

4770 wait 10 : repeat

4780 X=x mouse : Y=y mouse : wait vbl

4790 if hardkey=97 then hide : opaque screen logic,0,0,SCR$(SC) : fastcopy logic,back : show

4800 M=mouse key : until M

4810 Y1=Y : X1=X : hide on : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200)

4820 if M=2 then hide on : wait 10 : cls : goto 200

4830 logic=back : repeat

4840 opaque screen back,0,0,SCR$(SC) : box X,Y to X1,Y1

4850 X1=x mouse : Y1=y mouse

4860 screen swap : wait vbl

4870 until mouse key

4880 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

4890 logic=physic : fastcopy logic,back

4900 show on : goto 4770

You can also alter the routine to draw a rounded box just by changing the command BOX to RBOX.

Something a little more interesting……a triangle.

4910 rem TRIANGLE SHAPE

4920 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : K=1 : set line val(LTYPE$),L_SIZE,0,0 : show on

4930 repeat

4940 X=x mouse : Y=y mouse

4950 if hardkey=97 then hide : opaque screen logic,0,0,SCR$(SC) : fastcopy logic,back : show

4960 wait vbl : M=mouse key : until mouse key : hide on

4970 X1=X : Y1=Y : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : logic=back : wait 10

4980 if M=2 then logic=physic : wait 10 : cls : goto 200

4990 repeat

5000 opaque screen logic,0,0,SCR$(SC) : X1=x mouse : Y1=y mouse

5010 polyline X,Y to X1,Y1 to X,Y to X,Y

5020 screen swap : wait vbl

5030 until mouse key

5040 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

5050 wait 10 : repeat

5060 opaque screen logic,0,0,SCR$(SC) : X2=x mouse : Y2=y mouse

5070 polyline X,Y to X1,Y1 to X2,Y2 to X,Y

5080 screen swap : wait vbl

5090 until mouse key

5100 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

5110 wait 10 : logic=physic : fastcopy logic,back : show on : goto 4930

The first part of the routine draws the first angle of the triangle and the second part draws the rest of it.

And now, an eclipse type circle.

5120 rem CIRCLE

5130 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : show on : K=1

5140 repeat

5150 X=x mouse : Y=y mouse

5160 if hardkey=97 then hide : opaque screen logic,0,0,SCR$(SC) : fastcopy logic,back : show

5170 wait vbl : M=mouse key : until M

5180 if M=2 then hide on : wait 10 : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : cls : goto 200

5190 hide on : logic=back : wait 10

5200 SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200)

5210 repeat

5220 opaque screen logic,0,0,SCR$(SC) : X1=x mouse : Y1=y mouse

5230 earc X,Y,abs(X1-X),abs(Y1-Y),0,3600

5240 screen swap : wait vbl

5250 until mouse key

5260 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

5270 logic=physic : fastcopy logic,back : show on : wait 10 : goto 5140

The ABS command stops the circle from drawing inwards.

No picture is complete without a bit of colour. This routine will fill an area of the screen with the current ink and pattern number, which can be altered with the SET PAINT and INK commands.

5280 rem FILL SHAPE

5290 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : show on : set paint 2,PSTYLE,1 : K=1

5300 repeat

5310 if mouse key=1 then hide : paint x mouse,y mouse : show

5320 if hardkey=97 then hide : opaque screen logic,0,0,SCR$(SC) : fastcopy logic,back : show

5330 wait vbl : until mouse key=2

5340 hide on : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : wait 10 : cls : goto 200

Another interesting tool is the RAYS effect.

5350 rem RAYS

5360 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : K=1 : set line val(LTYPE$),L_SIZE,0,0 : show on : wait 10

5370 repeat

5380 X=x mouse : Y=y mouse : wait vbl

5390 M=mouse key : until M

5400 Y1=Y : X1=X : hide on : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200)

5410 if M=2 then hide on : wait 10 : cls : goto 200

5420 logic=back : repeat

5430 X1=x mouse : Y1=y mouse

5440 opaque screen back,0,0,SCR$(SC) : polyline X,Y to X1,Y1

5450 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

5460 screen swap : wait vbl

5470 until mouse key=1

5480 logic=physic : polyline X,Y to X1,Y1 : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : wait 2 : goto 5420

A K line is like a normal line except that it does not stop drawing itself when you reclick the mouse button. Here is the routine……

5490 rem K-LINE

5500 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : K=1 : set line val(LTYPE$),L_SIZE,0,0 : show on

5510 repeat

5520 X=x mouse : Y=y mouse : wait vbl

5530 M=mouse key : until M : wait 10

5540 Y1=Y : X1=X : hide on : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200)

5550 if M=2 then goto 200

5560 logic=back : repeat

5570 opaque screen back,0,0,SCR$(SC) : polyline X,Y to X1,Y1

5580 X1=x mouse : Y1=y mouse

5590 if mouse key=2 then logic=physic : wait 10 : cls : goto 200

5600 screen swap : wait vbl

5610 until mouse key=1

5620 logic=physic : polyline X,Y to X1,Y1 : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : wait 2 : X=X1 : Y=Y1 : goto 5560

Two more interesting tools are the SPRAY and BRUSH.

6010 rem SPRAY

6020 off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : show on : K=1 : wait 10

6030 repeat

6040 K$=inkey$: X=x mouse+4 : Y=y mouse+4 : wait vbl

6050 if hardkey=97 then hide : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : show

6060 if mouse key=0 then show on

6070 M=mouse key : until M : hide on : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : wait vbl

6080 if M=1 and X>=SIZE and X<=319-SIZE and Y>=SIZE and Y<=199-SIZE then plot X+rnd(SIZE),Y+rnd(SIZE) : plot X-rnd(SIZE),Y-rnd(SIZE) : plot X+rnd(SIZE),Y-rnd(S IZE) : plot X-rnd(SIZE),Y+rnd(SIZE) 6090 if SPEED>0 and M=1 then wait SPEED

6100 if M=2 then SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : wait 10 : cls : goto 200

6110 if mouse key=0 then show on

6120 goto 6030

6180 rem BRUSH

6190 hide on : off : cls : ink PE-1 : opaque screen back,0,0,SCR$(SC) : fastcopy back,logic : wait vbl : wait 10 : K=1 : show on

6200 repeat

6210 XM=x mouse : YM=y mouse

6220 if hardkey=97 then off : opaque screen logic,0,0,SCR$(SC) : fastcopy logic,back

6230 if M=1 and XM+BRUSH<320 and YM+BRUSH<200 then hide on : bar XM,YM to XM+BRUSH,YM+BRUSH else show on

6240 M=mouse key : until M=2

6250 if M=2 then hide on : wait 10 : SCR$(SC)="" : SCR$(SC)=screen$(logic,0,0 to 330,200) : cls : goto 200

6260 goto 6200

These are some of the more common tools used in an art package. There are more interesting features like MAGNIFY and RESIZE and all kinds of effects. But with these simple routines and a bit of imagination, there is no reason why you can't write your very own art package in STOS.