#include<stdio.h>

#include<stdlib.h>

#include<GL/gl.h> // Header File For The GLUT Library

#include<GL/glu.h>

#include<GL/glut.h>

typedef struct pixel

{

GLubyte red,green,blue;

}pixel;

pixel c,d,boundary,fill; // objects of pixel

int count=0,ch=1,X,Y;

struct vertex

{

int x,y;

}v[10];

void Init()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1,1,1,0);

gluOrtho2D(0 , 800 , 0 , 700);

}

int sign(int a)

{

if(a>=0)

return 1;

else

return -1;

}

void dda(int X1,int Y1,int X2,int Y2)

{

float x,y,dx,dy,length;

int i;

dx=abs(X2-X1);

dy=abs(Y2-Y1);

if(dx>=dy)

length=dx;

else

length=dy;

dx=(X2-X1)/length;

dy=(Y2-Y1)/length;

x=X1 + 0.5\*sign(X1);

y=Y1 + 0.5\*sign(Y1);

i=1;

while(i<=length)

{

// glColor3f(1.0,0.0,0.0);

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

glFlush();

x=x+dx;

y=y+dy;

i=i+1;

}

glFlush();

}

void boundary\_fill(int x,int y,pixel b,pixel f)

{

glReadPixels(x,y,1,1,GL\_RGB,GL\_UNSIGNED\_BYTE,&c);

if(c.red!=b.red && c.green!=b.green && c.blue!=b.blue && c.red!=f.red && c.green!=f.green && c.blue!=f.blue)

{

glColor3ub(f.red,f.green,f.blue);

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

glFlush();

boundary\_fill(x+1,y,b,f);

boundary\_fill(x-1,y,b,f);

boundary\_fill(x,y+1,b,f);

boundary\_fill(x,y-1,b,f);

}

}

void polygon()

{

int i;

glColor3ub(10,10,10);

for(i=0;i<count-1;i++)

{

dda(v[i].x,v[i].y,v[i+1].x,v[i+1].y);

}

dda(v[i].x,v[i].y,v[0].x,v[0].y);

}

void mouse(int btn,int state,int x,int y)

{

int ymax=glutGet(GLUT\_WINDOW\_HEIGHT);

if(btn==GLUT\_LEFT\_BUTTON&&state==GLUT\_DOWN)

{

v[count].x=x;

v[count].y=ymax-y;

count++;

}

if(btn==GLUT\_RIGHT\_BUTTON && state==GLUT\_DOWN)

{

switch(ch)

{

case 1:

X=x;

Y=ymax-y;

glReadPixels(X,Y,1,1,GL\_RGB,GL\_UNSIGNED\_BYTE,&d);

fill.red = d.red;

fill.green = d.green;

fill.blue = d.blue;

polygon();

ch=2;

break;

case 2:

X=x;

Y=ymax-y;

boundary\_fill(X,Y,boundary,fill);

break;

}

}

}

void draw()

{

glClear(GL\_COLOR\_BUFFER\_BIT );

glBegin(GL\_QUADS);

glColor3ub(100,150,250);glVertex2i(100,100);glVertex2i(150,100);glVertex2i(150,150);glVertex2i(100,150);

glColor3ub(0,0,100);glVertex2i(151,100);glVertex2i(200,100);glVertex2i(200,150);glVertex2i(151,150);

glColor3ub(0,100,0);glVertex2i(201,100);glVertex2i(250,100);glVertex2i(250,150);glVertex2i(201,150);

glColor3ub(0,100,100);glVertex2i(251,100);glVertex2i(300,100);glVertex2i(300,150);glVertex2i(251,150);

glColor3ub(100,0,0);glVertex2i(301,100);glVertex2i(350,100);glVertex2i(350,150);glVertex2i(301,150);

glColor3ub(100,0,100);glVertex2i(351,100);glVertex2i(400,100);glVertex2i(400,150);glVertex2i(351,150);

glColor3ub(100,100,0);glVertex2i(401,100);glVertex2i(450,100);glVertex2i(450,150);glVertex2i(401,150);

glColor3ub(175,175,100);glVertex2i(451,100);glVertex2i(500,100);glVertex2i(500,150);glVertex2i(451,150);

glColor3ub(175,0,0);glVertex2i(501,100);glVertex2i(550,100);glVertex2i(550,150);glVertex2i(501,150);

glColor3ub(0,175,0);glVertex2i(551,100);glVertex2i(600,100);glVertex2i(600,150);glVertex2i(551,150);

glColor3ub(0,0,175);glVertex2i(601,100);glVertex2i(650,100);glVertex2i(650,150);glVertex2i(601,150);

glColor3ub(175,0,175);glVertex2i(651,100);glVertex2i(700,100);glVertex2i(700,150);glVertex2i(651,150);

glEnd();

glFlush();

}

int main(int argc,char \*\* argv)

{

boundary.red = 10;

boundary.green = 10;

boundary.blue = 10;

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

/\* Set the initial window position and size \*/

glutInitWindowPosition(0,0);

glutInitWindowSize(800,700);

glutCreateWindow("mouse click");

Init();

glutMouseFunc(mouse);

glutDisplayFunc(draw);

glutMainLoop();

return 0;

}