// Name – Dahiwal Satyam Santoshkumar

// Roll Number – 207019

#include <GL/glut.h>

#include<math.h>

#include<stdlib.h

#include<iostream>

#include<stdio.h>

#define PI 3.14159265/180

using namespace std;

float rtri=0.0f;

int ww=660,wh=660;

int xi,yi,xf,yf,xx,yy;

float fillcol[3]={0.9,0.0,0.2};

float bordercol[3]={0.0,0.0,0.0};

void setpixel(int pointx,int pointy,float f[3])

{

glBegin(GL\_POINTS);

glColor3f(f[0],f[1],f[2]);

glVertex2d(pointx,pointy);

glEnd();

glFlush();

}

void getpixel(int x,int y,float pixels[3])

{

glReadPixels(x,y,1.0,1.0,GL\_RGB,GL\_FLOAT,pixels);

}

int sign(float arg)

{

if(arg<0)

return -1;

else if(arg==0)

return 0;

else

return 1;

}

double calx(double xold,double yold,double angle)

{

double xnew;

xnew=xold\*cos(PI\*angle)-yold\*sin(PI\*angle);

return xnew;

}

double caly(double xold,double yold,double angle)

{

double ynew;

ynew=xold\*sin(PI\*angle)+yold\*cos(PI\*angle) ;

return ynew;

}

void Bre(int X1,int Y1,int X2,int Y2,double angle)

{

double xold=0,yold=0;

double xnew=0,ynew=0;

xold=X1;

yold=Y1;

xnew=calx(xold,yold,angle);

ynew=caly(xold,yold,angle);

X1=(int)xnew;

Y1=(int)ynew;

xold=X2;

yold=Y2;

xnew=xold\*cos(PI\*angle)-yold\*sin(PI\*angle) ;

ynew=xold\*sin(PI\*angle)+yold\*cos(PI\*angle) ;

X2=(int)xnew;

Y2=(int)ynew;

float dx=abs(X2-X1);

float dy=abs(Y2-Y1);

int s1,s2,exc,y,x,i;

float g,temp;

x=X1;

y=Y1;

s1=sign(X2-X1);

s2=sign(Y2-Y1);

glBegin(GL\_POINTS); // Plot the points

//rotate 45 on x axis

if(dy>dx)

{

temp=dx;

dx=dy;

dy=temp;

exc=1;

}

else

{

exc=0;

}

g=2\*dy-dx;

i=1;

while(i<=dx)

{

glVertex2d(x,y);

while(g>=0)

{

if(exc==1)

x=x+s1;

else

y=y+s2;

g=g-2\*dx;

}

if(exc==1)

y=y+s2;

else

x=x+s1;

g=g+2\*dy;

i++;

}

glEnd();

glFlush();

}

void display()

{

double angle=45;

glClearColor(1.0,1.0,1.0,0); // Set Background color to white

glColor3f(0.0,0.0,0.0);

glViewport(0,0,1360,760);

glLoadIdentity();

gluOrtho2D(0 , 1360 , 0 , 760);

glClear(GL\_COLOR\_BUFFER\_BIT);

Bre(300,00,500,00,angle);

Bre(300,00,300,200,angle);

Bre(500,200,300,200,angle);

Bre(500,200,500,00,angle);//outer

Bre(300,50,500,50,angle);

Bre(300,100,500,100,angle);

Bre(300,150,500,150,angle);//outer

Bre(350,00,350,200,angle);

Bre(400,00,400,200,angle);

Bre(450,00,450,200,angle);//outer

angle=0;

Bre(300,00,500,00, angle);

Bre(300,00,300,200, angle);

Bre(500,200,300,200, angle);

Bre(500,200,500,00, angle);

Bre(300,50,500,50, angle);

Bre(300,100,500,100,angle);

Bre(300,150,500,150,angle);//outer

Bre(350,00,350,200, angle);

Bre(400,00,400,200, angle);

Bre(450,00,450,200, angle);//outer

}

void bfill(int x,int y,float fillcolor[3],float bordercolor[3])

{

float inter\_color[3];

getpixel(x,y,inter\_color);

if((inter\_color[0]!=bordercolor[0]&&inter\_color[1]!=bordercolor[1]&&inter\_color[2]!=bordercolor[2])&&(inter\_color[0]!=fillcolor[0]&&inter\_color[1]!=fillcolor[1]&&inter\_color[2]!=fillcolor[2]))

{

setpixel(x,y,fillcolor);

bfill(x+1,y,fillcolor,bordercolor);

bfill(x-1,y,fillcolor,bordercolor);

bfill(x,y+1,fillcolor,bordercolor);

bfill(x,y-1,fillcolor,bordercolor);

}

}

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

{

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

{

xi=x;

yi=wh-y;

printf("\nPoints collected are:%d and %d",xi,yi);

bfill(xi,yi,fillcol,bordercol);

}

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(ww,wh);

glutInitWindowPosition(0,0);

glutCreateWindow("CheckBox(Click on any box to fill color)");

glutDisplayFunc(display);

glutMouseFunc(mouse);

glutMainLoop();

return 0;

}

Output

satyam@ubuntu:~$ g++ five.cpp -lglut -lGL -lGLEW -lGLU -o five

satyam@ubuntu:~$ ./five

