#include<stdlib.h>

#include<stdio.h>

#include <GL/glut.h>

#include <GL/gl.h>

void init() {

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(600, 600);

//glutInitWindowPosition(100, 100);

glutCreateWindow("My shape");

glClearColor(0.0, 0.0, 0.0, 0.0);

gluOrtho2D(0, 400, 0, 400);

}

void circlePoints(int xc, int yc, int x, int y)

{

glVertex2i(xc + x, yc + y);

glVertex2i(xc - x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc - x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc - y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - y, yc - x);

}

void bresenhamCircle(int xc, int yc, int r) {

int x = 0, y = r;

int d = 3 - 2 \* r;

glBegin(GL\_POINTS);

while (y >= x) {

circlePoints(xc, yc, x, y);

x++;

if (d > 0)

{

y--;

d = d + 4 \* (x - y) + 10;

}

else

{

d = d + 4 \* x + 6;

}

}

glEnd();

}

void draw\_point() {

glPointSize(10);

glBegin(GL\_POINTS);

glVertex2f(65.0, 70.0);

glVertex2f(35.0, 70.0);

glEnd();

}

void draw\_line() {

glBegin(GL\_LINES);

glVertex2f(250.0, 350.0);

glVertex2f(200.0, 150.0);

glEnd();

}

void draw\_line1() {

glBegin(GL\_LINES);

glVertex2f(50.0, 350.0);

glVertex2f(100.0, 150.0);

glEnd();

}

void draw\_triangle() {

glBegin(GL\_TRIANGLES);

glVertex3f(90.0, 200.0, 0.0);

glVertex3f(130.0, 200.0, 0.0);

glVertex3f(110.0, 230.0, 0.0);

glEnd();

}

void draw\_triangle1() {

glBegin(GL\_TRIANGLES);

glVertex3f(130.0, 200.0, 0.0);

glVertex3f(150, 230.0, 0.0);

glVertex3f(170.0, 200.0, 0.0);

glEnd();

}

void draw\_triangle2() {

glBegin(GL\_TRIANGLES);

glVertex3f(170.0, 200.0, 0.0);

glVertex3f(210.0, 200.0, 0.0);

glVertex3f(190.0, 230.0, 0.0);

glEnd();

}

void drawEllipse(int xCenter, int yCenter, int rx, int ry)

{

int x = 0;

int y = ry;

int p1 = ry \* ry - rx \* rx \* ry + rx \* rx / 4;

int dx = 2 \* ry \* ry \* x;

int dy = 2 \* rx \* rx \* y;

glBegin(GL\_POINTS);

while (dx < dy)

{

glVertex2i(x + xCenter, y + yCenter);

glVertex2i(-x + xCenter, y + yCenter);

glVertex2i(x + xCenter, -y + yCenter);

glVertex2i(-x + xCenter, -y + yCenter);

if (p1 < 0)

{

x++;

dx += 2 \* ry \* ry;

p1 += dx + ry \* ry;

}

else

{

x++;

y--;

dx += 2 \* ry \* ry;

dy -= 2 \* rx \* rx;

p1 += dx - dy + ry \* ry;

}

}

int p2 = ry \* ry \* (x + 0.5) \* (x + 0.5) + rx \* rx \* (y - 1) \* (y - 1) - rx \* rx \* ry \* ry;

while (y >= 0)

{

glVertex2i(x + xCenter, y + yCenter);

glVertex2i(-x + xCenter, y + yCenter);

glVertex2i(x + xCenter, -y + yCenter);

glVertex2i(-x + xCenter, -y + yCenter);

if (p2 > 0)

{

y--;

dy -= 2 \* rx \* rx;

p2 += rx \* rx - dy;

}

else

{

y--;

x++;

dx += 2 \* ry \* ry;

dy -= 2 \* rx \* rx;

p2 += dx - dy + rx \* rx;

}

}

glEnd();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

// bresenhamCircle(50, 50, 30);

drawEllipse(150, 350, 100, 50);

drawEllipse(150, 150, 50, 25);

//draw\_point();

draw\_line();

draw\_line1();

draw\_triangle();

draw\_triangle1();

draw\_triangle2();

glFlush();

}

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

{

glutInit(&argc, argv);

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}