#include<stdlib.h>

#include<stdio.h>

#include <GL/glut.h>

#include <GL/gl.h>

void init() {

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow("My shape");

glClearColor(0.0, 0.0, 0.0, 0.0);

gluOrtho2D(0, 100, 0, 100);

}

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(35.0, 40.0);

glVertex2f(65.0, 40.0);

glEnd();

}

void draw\_triangle() {

glBegin(GL\_TRIANGLES);

glVertex3f(50.0, 55.0, 0.0);

glVertex3f(47.0, 50.0, 0.0);

glVertex3f(53.0, 50.0, 0.0);

glEnd();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

bresenhamCircle(50, 50, 30);

draw\_point();

draw\_line();

draw\_triangle();

glFlush();

}

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

{

glutInit(&argc, argv);

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}