#include<GL/glut.h>

#include<stdio.h>

int xc, yc, r;

void drawDot(GLint x, GLint y, GLfloat r, GLfloat g, GLfloat b)

{

glColor3f(r, g, b);

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

}

void symmetricPixels(int x, int y, int xc, int yc, float r, float g, float b)

{

drawDot(xc + x, yc + y, r, g, b);

drawDot(xc - x, yc + y, r, g, b);

drawDot(xc + x, yc - y, r, g, b);

drawDot(xc - x, yc - y, r, g, b);

}

void Ellipse(int a, int b, int xc, int yc, float r, float g, float bl)

{

int aSq, bSq, twoASq, twoBSq, d, dx, dy, x, y;

aSq = a \* a;

bSq = b \* b;

twoASq = 2 \* aSq;

twoBSq = 2 \* bSq;

d = bSq - b \* aSq + aSq / 4;

dx = 0;

dy = twoASq \* b;

x = 0;

y = b;

symmetricPixels(x, y, xc, yc, r, g, bl);

while (dx < dy)

{

x++;

dx += twoBSq;

if (d >= 0)

{

y--;

dy -= twoASq;

}

if (d < 0)

d += bSq + dx;

else

d += bSq + dx - dy;

symmetricPixels(x, y, xc, yc, r, g, bl);

}

d = (int)(bSq \* (x + 0.5) \* (x + 0.5) + aSq \* (y - 1) \* (y - 1) -

aSq \* bSq);

while (y > 0)

{

y--;

dy -= twoASq;

if (d <= 0)

{

x++;

dx += twoBSq;

}

if (d > 0)

d += aSq - dy;

else

d += aSq - dy + dx;

symmetricPixels(x, y, xc, yc, r, g, bl);

}

}

void Set\_Graphics(int width, int height)

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(0.0f, 0.0f, 0.0f);

glViewport(0, 0, width, height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glEnable(GL\_DEPTH\_TEST);

gluPerspective(45, (float)width / height, 1.0, 0);

glMatrixMode(GL\_MODELVIEW);

glTranslatef(0, 0, 5 \* (-height));

}

void Start\_Line()

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

// Line\_Algo( 5, 8, 29,32 );

// Circle (40,0,0,1);

Ellipse(100, 30, 20, 20, 0, 1, 0);

// glFlush();

glutSwapBuffers();

}

void draw\_circle(int xc, int yc, int x, int y)

{

glBegin(GL\_POINTS);

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);

glEnd();

}

void circlebres()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

int x = 0, y = r;

int d = 3 - 2 \* r;

while (x <= y)

{

draw\_circle(xc, yc, x, y);

x++;

if (d < 0)

d = d + 4 \* x + 6;

else

{

y--;

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

}

draw\_circle(xc, yc, x, y);

}

glFlush();

}

void minit()

{

glClearColor(1, 1, 1, 1);

glColor3f(1.0, 0.0, 0.0);

glPointSize(3.0);

gluOrtho2D(10, 500, 10, 500);

}

void main(int argc, char\* argv[])

{

printf\_s("Enter the coordinates of the center:");

scanf\_s("%d%d", &xc, &yc);

printf\_s("Enter the Radius:");

scanf\_s("%d", &r);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(300, 300);

glutInitWindowPosition(5, 5);

glutCreateWindow("Bresenham's Circle");

glutDisplayFunc(circlebres);

minit();

glutInitWindowSize(640, 480);

glutInitWindowPosition(100, 150);

glutInitDisplayMode(GLUT\_RGBA | GLUT\_DOUBLE | GLUT\_DEPTH);

glutCreateWindow(" Elipse Drawing");

glutDisplayFunc(Start\_Line);

Set\_Graphics(100,100);

glutMainLoop();

}