#include <stdio.h>

#include <math.h>

#include <GL/glut.h>

// Center of the cicle = (320, 240)

int xc = 320, yc = 240;

// Plot eight points using circle's symmetrical property

void plot\_point(int x, int y)

{

glBegin(GL\_POINTS);

glVertex2i(xc + x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - x, yc - y);

glVertex2i(xc - y, yc - x);

glVertex2i(xc - x, yc + y);

glVertex2i(xc - y, yc + x);

glEnd();

}

// Function to draw a circle using bresenham's

// circle drawing algorithm

void bresenham\_circle(int r)

{

int x = 0, y = r;

float pk = (5.0 / 4.0) - r;

/\* Plot the points \*/

/\* Plot the first point \*/

plot\_point(x, y);

int k;

while (x < y)

{

x = x + 1;

if (pk < 0)

pk = pk + 2 \* x + 1;

else

{

y = y - 1;

pk = pk + 2 \* (x - y) + 1;

}

plot\_point(x, y);

}

glFlush();

}

void draw\_circle(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

int radius = 100;

bresenham\_circle(radius);

}

void Init()

{

glClearColor(0.0, 1.0, 1.0, 0);

glColor3f(0.0, 0.0, 0.0);

gluOrtho2D(0, 640, 0, 480);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(0, 0);

glutInitWindowSize(640, 480);

glutCreateWindow("bresenham\_circle");

Init();

glutDisplayFunc(draw\_circle);

glutMainLoop();

}