CGPractcal3

#include <GL/glut.h>

#include <iostream>

using namespace std;

void setPixel(int x, int y) {

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

glFlush();

}

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

int x = 0, y = r;

int d = 3 - 2 \* r;

while (x <= y) {

// Drawing all 8 symmetric points

setPixel(xc + x, yc + y);

setPixel(xc - x, yc + y);

setPixel(xc + x, yc - y);

setPixel(xc - x, yc - y);

setPixel(xc + y, yc + x);

setPixel(xc - y, yc + x);

setPixel(xc + y, yc - x);

setPixel(xc - y, yc - x);

if (d < 0)

d += 4 \* x + 6;

else {

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

y--;

}

x++;

}

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// Draw X and Y axes

glColor3f(0.0, 1.0, 0.0); // Green axes

glBegin(GL\_LINES);

glVertex2i(-250, 0); glVertex2i(250, 0); // X-axis

glVertex2i(0, -250); glVertex2i(0, 250); // Y-axis

glEnd();

// Draw Circle

glColor3f(1.0, 0.0, 0.0); // Red circle

BresenhamCircle(0, 0, 100);

glFlush();

}

void init() {

glClearColor(0.0, 0.0, 0.0, 1.0); // Black background

glColor3f(1.0, 1.0, 1.0); // Default drawing color

gluOrtho2D(-250, 250, -250, 250); // Coordinate system

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow("Bresenham Circle Drawing");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}