**Lab Report**

*Course title: Computer Graphics*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

**Date of Submission**: 04 June 2023

****

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 399 | 202211 | Md Hasan Al Mamun |

**Scan Conversin of Circle Using Mid-Point Algorithm**

**Code:**

*#include <graphics.h>*

**void** drawCircleMidpoint(**int** xc, **int** yc, **int** radius) {

**int** x = 0; *// Initial x-coordinate of the circle's point*

**int** y = radius; *// Initial y-coordinate of the circle's point*

**int** decisionParameter = 1 - radius; *// Initial decision parameter*

*// Loop until x-coordinate becomes greater than y-coordinate*

**while** (x <= y) {

*// Plot the circle's points using symmetry*

putpixel(xc + x, yc + y, WHITE);

putpixel(xc + y, yc + x, WHITE);

putpixel(xc - x, yc + y, WHITE);

putpixel(xc - y, yc + x, WHITE);

putpixel(xc + x, yc - y, WHITE);

putpixel(xc + y, yc - x, WHITE);

putpixel(xc - x, yc - y, WHITE);

putpixel(xc - y, yc - x, WHITE);

*// Update x and y based on the decision parameter*

**if** (decisionParameter < 0) {

decisionParameter += 2 \* x + 3;

} **else** {

decisionParameter += 2 \* (x - y) + 5;

y--;

}

x++;

}

}

**int** main() {

**int** gd = DETECT, gm;

initgraph(&gd, &gm, *""*);

**int** xc = 250; *// X-coordinate of the circle's center*

**int** yc = 250; *// Y-coordinate of the circle's center*

**int** radius = 100; *// Radius of the circle*

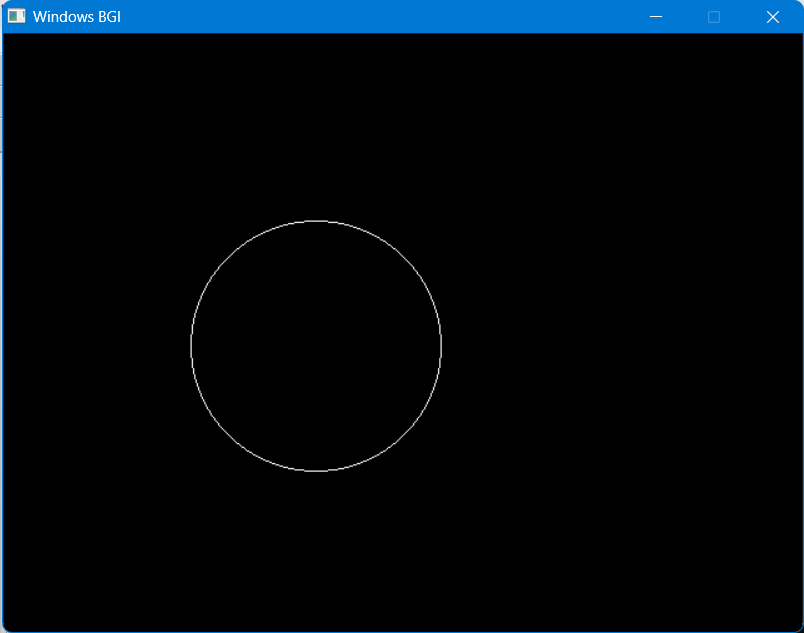
drawCircleMidpoint(xc, yc, radius);

getch();

closegraph();

**return** 0;

}



*Figure 1: Circle*

**Scan Conversin of Ellipse Using Mid-Point Algorithm**

**Code:**

#include <graphics.h>

void drawEllipseMidpoint(int xc, int yc, int rx, int ry) {

int x = 0; // Initial x-coordinate of the ellipse's point

int y = ry; // Initial y-coordinate of the ellipse's point

// Initialize decision parameters

int decisionParameter1 = ry \* ry - rx \* rx \* ry + rx \* rx / 4;

int decisionParameter2;

// Region 1

while (2 \* ry \* ry \* x < 2 \* rx \* rx \* y) {

// Plot points in Region 1

putpixel(xc + x, yc - y, WHITE);

putpixel(xc - x, yc + y, WHITE);

putpixel(xc + x, yc + y, WHITE);

putpixel(xc - x, yc - y, WHITE);

// Update x and decision parameters

x++;

decisionParameter1 += 2 \* ry \* ry \* x + ry \* ry;

if (decisionParameter1 >= 0) {

y--;

decisionParameter1 -= 2 \* rx \* rx \* y;

}

}

// Region 2

decisionParameter2 = ry \* ry \* (x + 0.5) \* (x + 0.5) +

rx \* rx \* (y - 1) \* (y - 1) - rx \* rx \* ry \* ry;

while (y >= 0) {

// Plot points in Region 2

putpixel(xc + x, yc - y, WHITE);

putpixel(xc - x, yc + y, WHITE);

putpixel(xc + x, yc + y, WHITE);

putpixel(xc - x, yc - y, WHITE);

// Update y and decision parameters

y--;

decisionParameter2 -= 2 \* rx \* rx \* y - rx \* rx;

if (decisionParameter2 <= 0) {

x++;

decisionParameter2 += 2 \* ry \* ry \* x;

}

}

}

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int xc = 250; // X-coordinate of the ellipse's center

int yc = 250; // Y-coordinate of the ellipse's center

int rx = 150; // X-axis radius of the ellipse

int ry = 100; // Y-axis radius of the ellipse

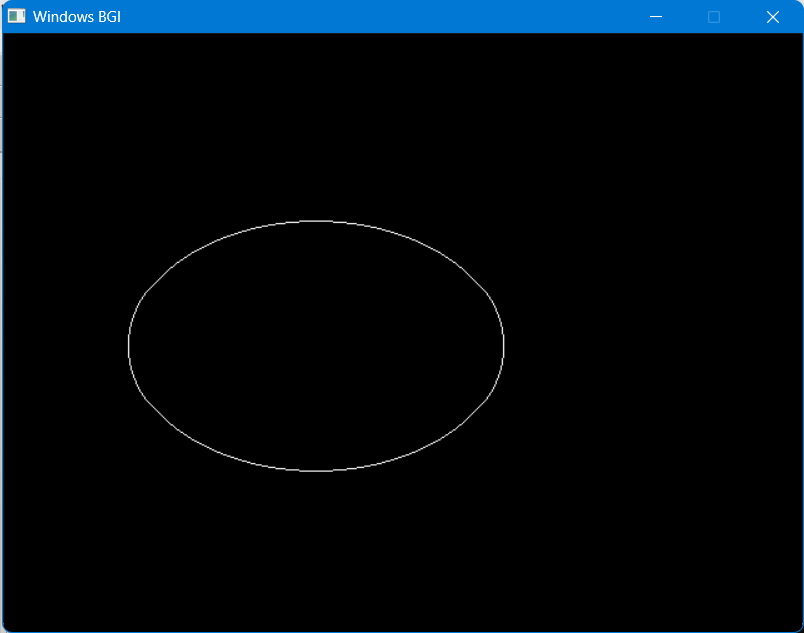
drawEllipseMidpoint(xc, yc, rx, ry);

getch();

closegraph();

return 0;

}



*Figure 2: Ellipse*