**Lab Report: 02**

**Title: Scan Conversion of ellipse and mid point of a circle**

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

**Date of Submission**: 04/06/2023

****

**Submitted to-**

*Dr. Mohammad Shorif Uddin*

*Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*&*

*Dr. Morium Akter*

*Associate Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |
| --- | --- | --- |
| Class Roll | Exam Roll | Name |
| 366 |  | Lamia Binta Latif |

**Experiment no-01: Scan conversion of ellipse**

**Source code:**

#include <iostream>

#include <graphics.h>

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

int x, y, p;

int rx2, ry2, twoRx2, twoRy2;

rx2 = rx \* rx;

ry2 = ry \* ry;

twoRx2 = 2 \* rx2;

twoRy2 = 2 \* ry2;

x = 0;

y = ry;

// Region 1

p = ry2 - (rx2 \* ry) + (0.25 \* rx2);

while (twoRx2 \* y >= twoRy2 \* x) {

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

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

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

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

x++;

if (p < 0)

p += ry2 \* (2 \* x + 1);

else {

y--;

p += ry2 \* (2 \* x + 1) - twoRx2 \* y;

}

}

// Region 2

p = ry2 \* (x + 0.5) \* (x + 0.5) + rx2 \* (y - 1) \* (y - 1) - rx2 \* ry2;

while (y >= 0) {

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

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

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

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

y--;

if (p > 0)

p += rx2 \* (1 - 2 \* y);

else {

x++;

p += rx2 \* (1 - 2 \* y) + twoRy2 \* x;

}

}

}

int main() {

int gd = DETECT, gm;

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

int xc = 320; // X coordinate of the center

int yc = 240; // Y coordinate of the center

int rx = 150; // Major radius

int ry = 100; // Minor radius

scanConvertEllipse(xc, yc, rx, ry);

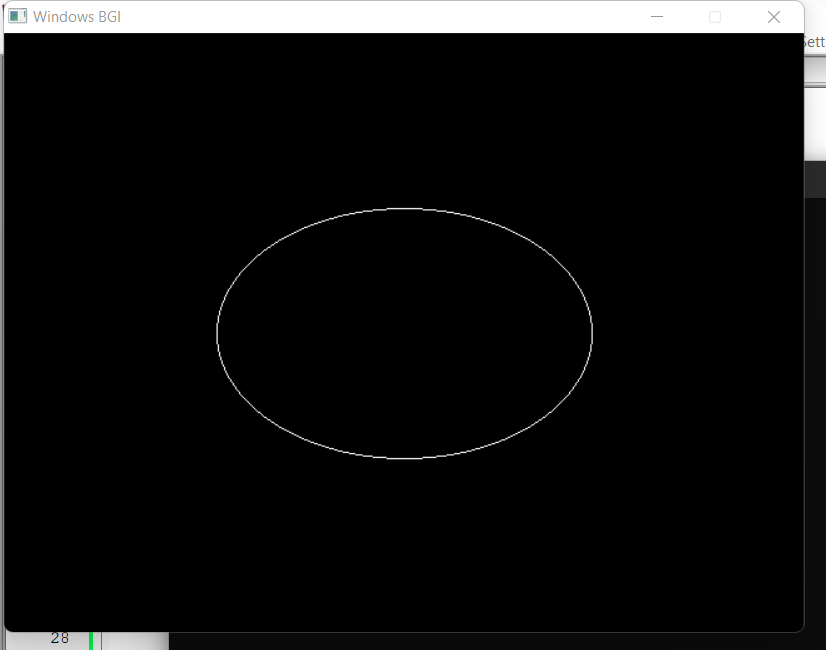
delay(5000); // Delay for 5 seconds

closegraph();

return 0;

}

**Output:**

****

**Experiment no-02: Mid-Point of a circle**

**Source code:**

#include <iostream>

#include <graphics.h>

void drawCircle(int xc, int yc, int x, int y)

{

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

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

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

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

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

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

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

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

}

void circleMidPoint(int xc, int yc, int radius)

{

int x = 0;

int y = radius;

int p = 1 - radius;

drawCircle(xc, yc, x, y);

while (x < y)

{

x++;

if (p < 0)

p += 2 \* x + 1;

else

{

y--;

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

}

drawCircle(xc, yc, x, y);

}

}

int main()

{

int gd = DETECT, gm;

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

int xc = 200;

int yc = 200;

int radius = 100;

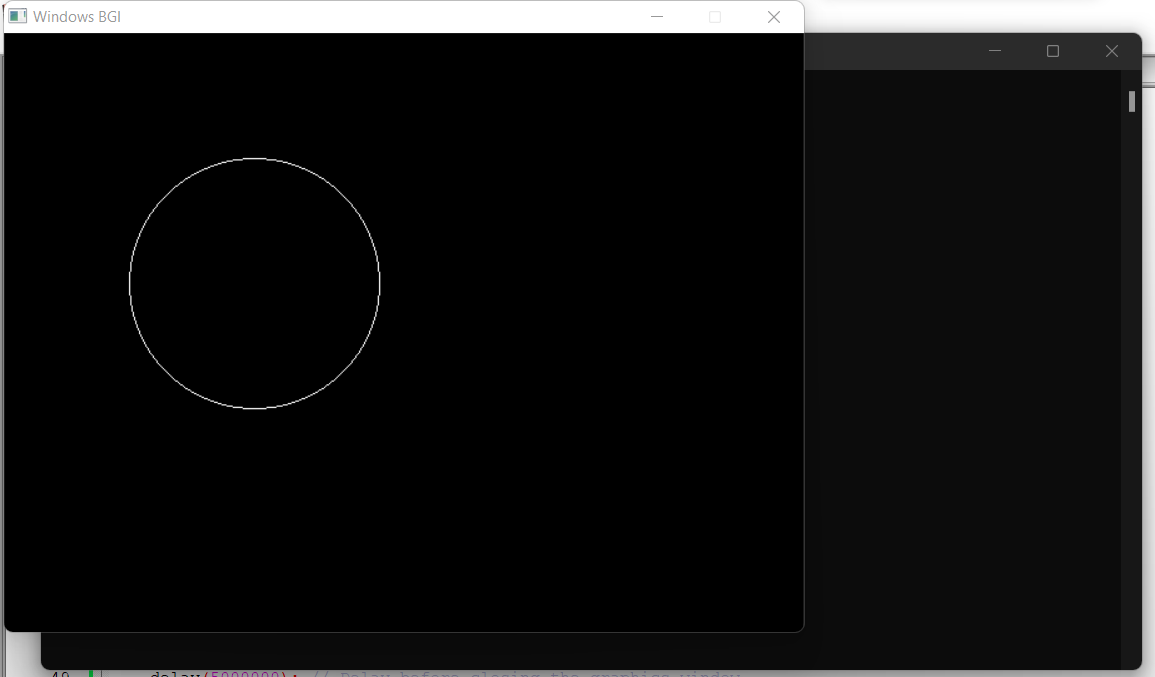
circleMidPoint(xc, yc, radius);

delay(5000000); // Delay before closing the graphics window

closegraph();

return 0;

}

**Output:**