**Lab Report-02**

**Title: Scan Conversion of circle using mid-point algorithm and**

**scan converting an ellipse**

*Course Title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

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

****

**Submitted to-**

***Dr. Mohammad Shorif Uddin***

Professor

***Dr. Morium Akter***

Associate Professor

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 358 | 202170 | Jannati Tajrimin Mitu |

**Mid-Point Circle Algorithm:**

#include<bits/stdc++.h>

#include<graphics.h>

using namespace std;

void drawcircle(int x0, int y0, int radius)

{

int x = radius;

int y = 0;

int err = 0;

while (x >= y)

{

putpixel(x0 + x, y0 + y, 7);

putpixel(x0 + y, y0 + x, 7);

putpixel(x0 - y, y0 + x, 7);

putpixel(x0 - x, y0 + y, 7);

putpixel(x0 - x, y0 - y, 7);

putpixel(x0 - y, y0 - x, 7);

putpixel(x0 + y, y0 - x, 7);

putpixel(x0 + x, y0 - y, 7);

if (err <= 0)

{

y += 1;

err += 2\*y + 1;

}

if (err > 0)

{

x -= 1;

err -= 2\*x + 1;

}

delay(100);

}

}

int main()

{

int gdriver=DETECT, gmode, error, x, y, r;

initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");

cout<<"Enter radius of circle: ";

cin>>r;

cout<<"Enter co-ordinates of center(x and y): ";

cin>>x>>y;

drawcircle(x, y, r);

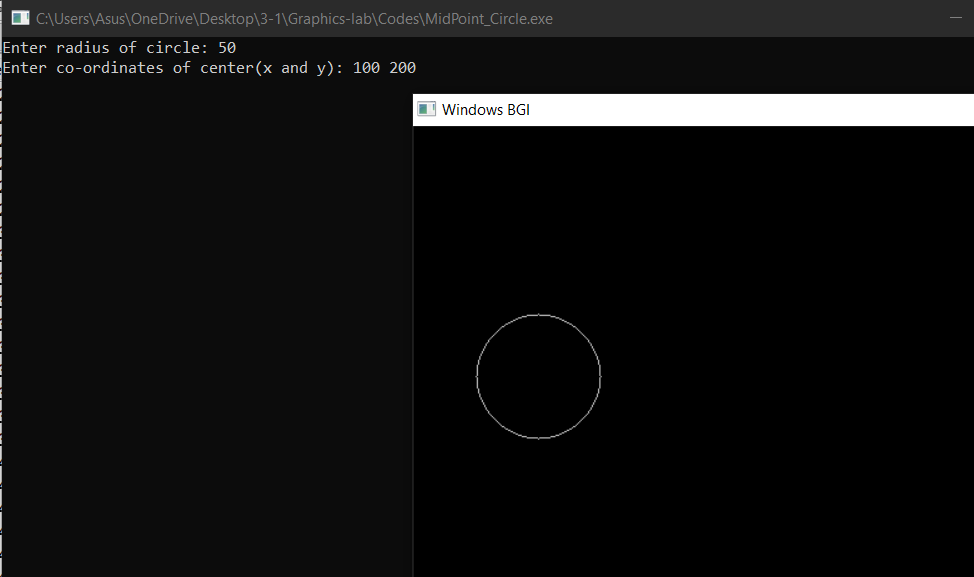
getch();

closegraph();

return 0;

}

**Output:**



**Scan Converting an Ellipse:**

#include <iostream>

#include <graphics.h>

void drawEllipsePoints(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);

}

void midpointEllipse(int xc, int yc, int rx, int ry)

{

    int x = 0;

    int y = ry;

    int rx\_square = rx \* rx;

    int ry\_square = ry \* ry;

    int two\_rx\_square = 2 \* rx\_square;

    int two\_ry\_square = 2 \* ry\_square;

    int p;

    int px = 0;

    int py = two\_rx\_square \* y;

    drawEllipsePoints(xc, yc, x, y);

    // Region 1

    p = ry\_square - (rx\_square \* ry) + (0.25 \* rx\_square);

    while (px < py)

    {

        x++;

        px += two\_ry\_square;

        if (p < 0)

        {

            p += ry\_square + px;

        }

        else

        {

            y--;

            py -= two\_rx\_square;

            p += ry\_square + px - py;

        }

        drawEllipsePoints(xc, yc, x, y);

    }

    // Region 2

    p = ry\_square \* (x + 0.5) \* (x + 0.5) + rx\_square \* (y - 1) \* (y - 1) - rx\_square \* ry\_square;

    while (y > 0)

    {

        y--;

        py -= two\_rx\_square;

        if (p > 0)

        {

            p += rx\_square - py;

        }

        else

        {

            x++;

            px += two\_ry\_square;

            p += rx\_square - py + px;

        }

        drawEllipsePoints(xc, yc, x, y);

    }

}

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 = 200;   // X-axis radius

    int ry = 150;   // Y-axis radius

    midpointEllipse(xc, yc, rx, ry);

    getch();

    closegraph();

    return 0;

}

**Output:**

