

## LAB Assignment-02

*Course title: Computer Graphics Laboratory*

*Course code: CSE-304*

*3<sup>rd</sup> Year 1<sup>st</sup> Semester Examination 2022*

**Date of Submission:** 04 June,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	386	202198	Sunirmol Mollik

## **Labwork:01** Mid-Point Circle Algorithm

### **Source Code:**

```
#include <iostream>
#include <graphics.h>
void drawCircle(int xc, int yc, int radius)
{
    int x = 0;
    int y = radius;
    int decision = 1 - radius;
    while (y >= 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);
        putpixel(xc + y, yc + x, WHITE);
        putpixel(xc - y, yc + x, WHITE);
        putpixel(xc + y, yc - x, WHITE);
        putpixel(xc - y, yc - x, WHITE);

        if (decision <= 0)
        {
            x++;
            decision += 2 * x + 1;
        }
        else
        {
            y--;
            x++;
            decision += 2 * (x - y) + 1;
        }
    }
}
```

```
int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int xc, yc, radius;
    std::cout << "Enter the center coordinates (xc, yc): ";
    std::cin >> xc >> yc;
    std::cout << "Enter the radius: ";
    std::cin >> radius;

    drawCircle(xc, yc, radius);

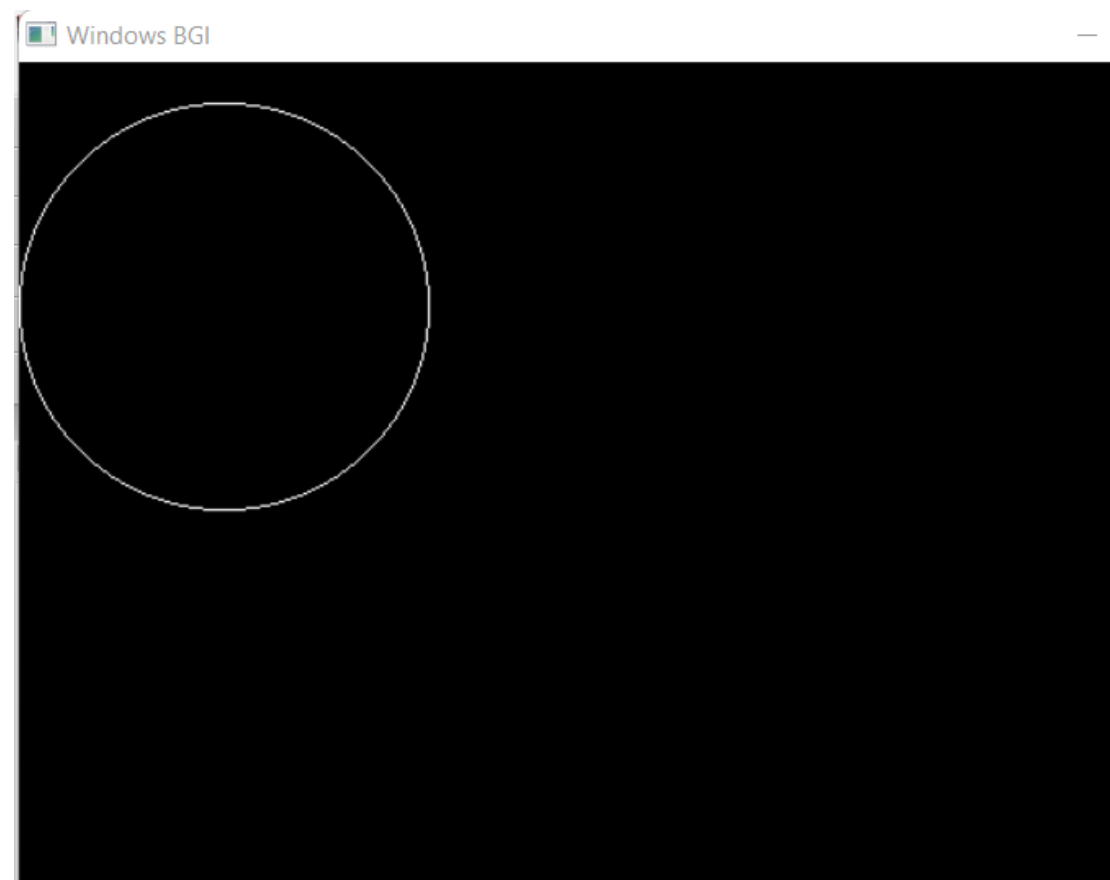
    delay(5000);
    closegraph();

    return 0;
}
```

## Output:

```
F:\midpoint_circle.exe
Enter the center coordinates (xc, yc): 100 120
Enter the radius: 100

Process returned 0 (0x0)   execution time : 12.354 s
Press any key to continue.
```



## Labwork:02 Scan Conversion of Ellipse

### Source Code:

```
#include <iostream>
#include <graphics.h>
void drawEllipse(int xc, int yc, int rx,
int ry)
{
    int x = 0;
    int y = ry;
    // Decision parameter for region 1
    int d1 = (ry * ry) - (rx * rx * ry) +
(0.25 * rx * rx);

    int dx = 2 * ry * ry * x;
    int dy = 2 * rx * rx * y;

    while (dx < dy)
    {
        putpixel(xc + x, yc + y, WHITE);
        putpixel(xc - x, yc + y, WHITE);
        putpixel(xc + x, yc - y, WHITE);
        putpixel(xc - x, yc - y, WHITE);

        if (d1 < 0)
        {
            x++;
            dx = dx + (2 * ry * ry);
            d1 = d1 + dx + (ry * ry);
        }
        else
        {
            x++;
            y--;
            dx = dx + (2 * ry * ry);
            dy = dy - (2 * rx * rx);
            d1 = d1 + dx - dy + (ry * ry);
        }
    }
    int d2 = ((ry * ry) * ((x + 0.5) * (x +
0.5))) + ((rx * rx) * ((y - 1) * (y - 1)))
- (rx * rx * ry * ry);

    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);

        if (d2 > 0)
        {
            y--;
            dy = dy - (2 * rx * rx);
            d2 = d2 + (rx * rx) - dy;
        }
        else
        {
            y--;
            x++;
            dx = dx + (2 * ry * ry);
            dy = dy - (2 * rx * rx);
            d2 = d2 + dx - dy + (rx *
rx);
        }
    }
}

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int xc, yc, rx, ry;
    std::cout << "Enter the center
coordinates (xc, yc): ";
    std::cin >> xc >> yc;
    std::cout << "Enter the x-radius
(rx): ";
    std::cin >> rx;
    std::cout << "Enter the y-radius
(ry): ";
    std::cin >> ry;
}
```

```
drawEllipse(xc, yc, rx, ry);  
    delay(5000);  
    closegraph();  
    return 0;  
}
```

## Output:

```
F:\scan_conversion_ellipse.exe  
Enter the center coordinates (xc, yc): 150 200  
Enter the x-radius (rx): 100  
Enter the y-radius (ry): 70  
  
Process returned 0 (0x0)   execution time : 23.952 s  
Press any key to continue.
```

