

Course title: Computer Graphics Lab
Course code: CSE-304

Date of Submission: 04-06-23



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	376	202188	Md. Shakhwat Hosen

Scan conversion of a circle using mid point rule:

Code:

```
#include <iostream>
#include <graphics.h>
using namespace std;

void drawCircle(int xc, int yc, int radius) {
    int x = 0;
    int y = radius;
    int p = 1 - radius;

    while (x <= y) {
        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);

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

int main() {

    int xc, yc, radius;
    cout << "Enter center coordinates (xc and yc): ";
    cin >> xc >> yc;
    cout << "Enter radius: ";
    cin >> radius;

    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    drawCircle(xc, yc, radius);
```

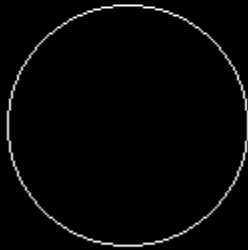
```
    getch();  
    closegraph();  
    return 0;  
}
```

Output:

F:\I0\circleMid.exe

Enter center coordinates (xc and yc): 200 200
Enter radius: 60

Windows BGI



Scan conversion of an ellipse:

Code:

```
#include <iostream>
#include <graphics.h>

// Function to plot points in all quadrants
void plotEllipsePoints(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);
}

// Function to draw ellipse using midpoint algorithm
void drawEllipse(int xc, int yc, int rx, int ry)
{
    int x = 0;
    int y = ry;

    // Decision parameters
    int rxSq = rx * rx;
    int rySq = ry * ry;
    int twoRxSq = 2 * rxSq;
    int twoRySq = 2 * rySq;
    int p;
    int px = 0;
    int py = twoRxSq * y;

    // Plot initial point in all quadrants
    plotEllipsePoints(xc, yc, x, y);

    // Region 1
    p = rySq - (rxSq * ry) + (0.25 * rxSq);
    while (px < py)
    {
        x++;
        px += twoRySq;

        if (p < 0)
        {
            p += rySq + px;
        }
        else
```

```

    {
        y--;
        py -= twoRxSq;
        p += rySq + px - py;
    }

    plotEllipsePoints(xc, yc, x, y);
}

// Region 2
p = rySq * (x + 0.5) * (x + 0.5) + rxSq * (y - 1) * (y - 1) - rxSq * rySq;
while (y > 0)
{
    y--;
    py -= twoRxSq;

    if (p > 0)
    {
        p += rxSq - py;
    }
    else
    {
        x++;
        px += twoRySq;
        p += rxSq - py + px;
    }

    plotEllipsePoints(xc, yc, x, y);
}
}

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    int xc = 250; // X-coordinate of the center of the ellipse
    int yc = 250; // Y-coordinate of the center of the ellipse
    int rx = 150; // X-radius of the ellipse
    int ry = 100; // Y-radius of the ellipse

    drawEllipse(xc, yc, rx, ry);

    // delay(5000);

```

```
    getch();  
    closegraph();  
  
    return 0;  
}
```

Output:

