Lab Report: 02
Title: Scan Converting of a Circle and an Ellipse

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
Dr. Morium Akter

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka-1342

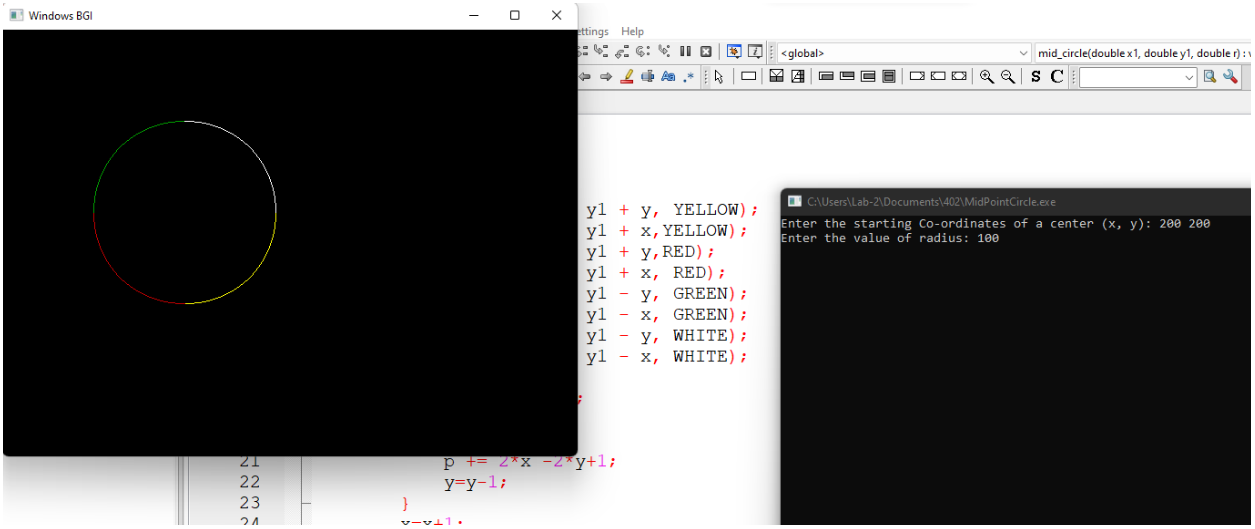
Name	Class Roll	Exam Roll
Jubaer Ahmad Khan	402	202214

Report Title: Midpoint Circle Algorithm

Source Code:

```
#include <graphics.h>
#include <iostream>
using namespace std;
void mid_circle(double x1, double y1, double r) {
    int x = 0;
    int y = r;
    int p = 1-r;
    while(y > x) {
        putpixel(x1 + x, y1 + y, YELLOW);
        putpixel(x1 + y, y1 + x,YELLOW);
        putpixel(x1 - x, y1 + y,RED);
        putpixel(x1 - y, y1 + x, RED);
        putpixel(x1 - x, y1 - y, GREEN);
        putpixel(x1 - y, y1 - x, GREEN);
        putpixel(x1 + x, y1 - y, WHITE);
        putpixel(x1 + y, y1 - x, WHITE);
        if (p < 0) {
            p += 2*x + 1;
        }
        else {
            p += 2*x -2*y+1;
            y=y-1;
        }
        x=x+1;
        delay(1000);
    }
}
int main(){
    double x1, y1, r;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, (char*)"");
    cout<<"Enter the starting Co-ordinates of a center (x, y): ";
    cin>>x1>>y1;
    cout<<"Enter the value of radius: ";
    cin>>r;
    mid_circle(x1, y1, r);
    getch();
    closegraph();
    return 0;
}
```

GUI Output:



```
y1 + y, YELLOW);
y1 + x,YELLOW);
y1 + y,RED);
y1 + x, RED);
y1 - y, GREEN);
y1 - x, GREEN);
y1 - y, WHITE);
y1 - x, WHITE);

        p += 2*x -2*y+1;
        y=y-1;
    }
```

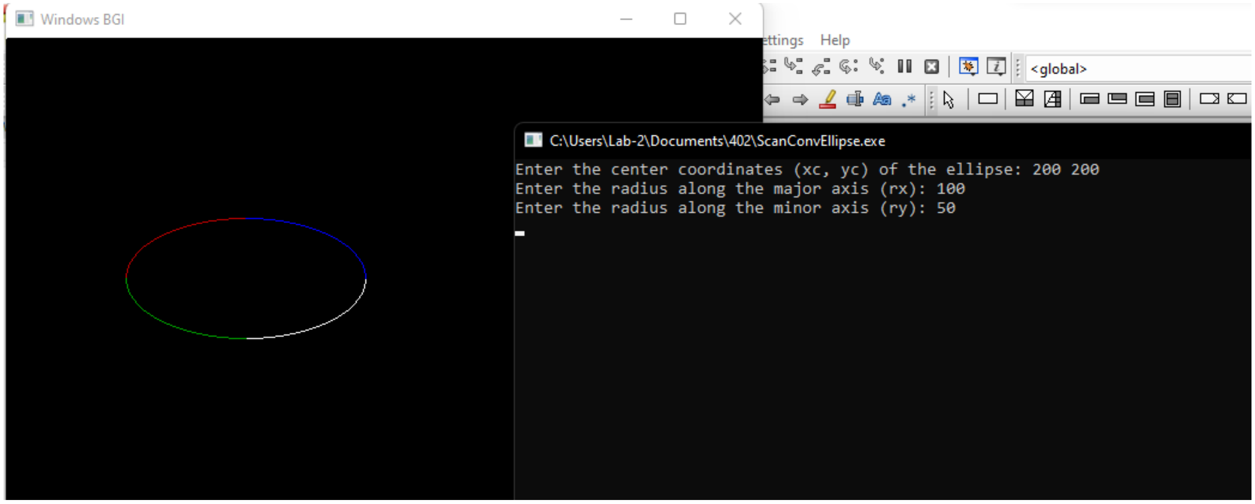
```
C:\Users\Lab-2\Documents\402\MidPointCircle.exe
Enter the starting Co-ordinates of a center (x, y): 200 200
Enter the value of radius: 100
```

Report Title: Midpoint Ellipse Algorithm

Source Code:

```
#include <graphics.h>
#include <iostream>
void drawEllipse(int xc, int yc, int rx, int ry) {
    int x = 0, y = ry;
    int p = ry * ry - rx * rx * ry + rx * rx / 4;
    while (2 * x * ry * ry <= 2 * y * rx * rx) {
        putpixel(xc + x, yc + y, WHITE);
        putpixel(xc - x, yc - y, RED);
        putpixel(xc + x, yc - y, BLUE);
        putpixel(xc - x, yc + y, GREEN);
        if (p < 0) {
            x = x + 1;
            p = p + 2 * ry * ry * x + ry * ry;
        } else {
            x = x + 1;
            y = y - 1;
            p = p + 2 * ry * ry * x - 2 * rx * rx * y - ry * ry;
        }
        delay(1000);
    }
    p = 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, RED);
        putpixel(xc + x, yc - y, BLUE);
        putpixel(xc - x, yc + y, GREEN);
        if (p > 0) {
            y = y - 1;
            p = p - 2 * rx * rx * y + rx * rx;
        } else {
            y = y - 1;
            x = x + 1;
            p = p + 2 * ry * ry * x - 2 * rx * rx * y + rx * rx;
        }
        delay(1000);
    }
}

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

    // Accepting user input for ellipse parameters
    int centerX, centerY, radiusX, radiusY;
    std::cout << "Enter the center coordinates (xc, yc) of the ellipse: ";
    std::cin >> centerX >> centerY;
    std::cout << "Enter the radius along the major axis (rx): ";
    std::cin >> radiusX;
    std::cout << "Enter the radius along the minor axis (ry): ";
    std::cin >> radiusY;

    drawEllipse(centerX, centerY, radiusX, radiusY);

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

GUI Output:



```
Enter the center coordinates (xc, yc) of the ellipse: 200 200
Enter the radius along the major axis (rx): 100
Enter the radius along the minor axis (ry): 50
```