Lab Report. 02

Course title: Computer Graphics Lab

Course code: CSE-304

3rd Year 1st Semester 2022

Date of Submission: 04-06-2023

Submitted to-

Dr. Mohammad Shorif Uddin

Professor
Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342

And

Dr. Morium Akter

Associate Professor
Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342

Sl	Class Roll	Exam Roll	Name
01	364	202176	Suraiya Mahmuda

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

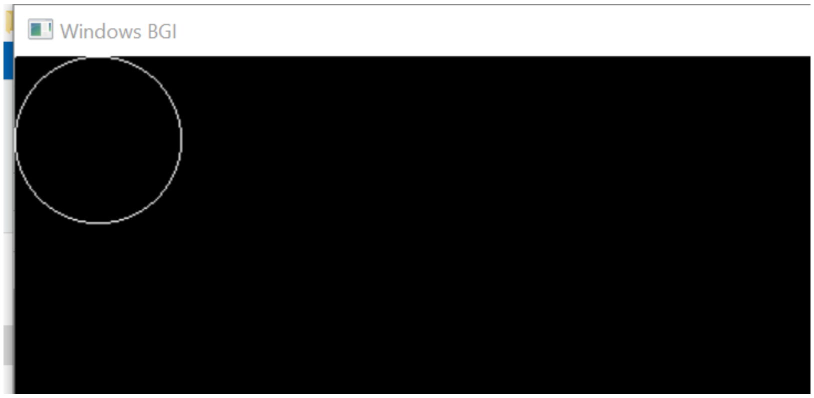
1.Midpoint Circle Algorithm

Sourcecode:

```
#include<bits/stdc++.h>
#include<graphics.h>
using namespace std;
void drawcircle(double a, double b, double r)
{
    double x=0, y=r, p;
    putpixel (a, b+r, WHITE);
    putpixel (a, b-r, WHITE);
    putpixel (a-r, b, WHITE);
    putpixel (a+r, b, WHITE);
    p=(5/4)-r;
    while (x<=y)
    {
        if(p<0)
            p+= (2*x)+3;
        else
        {
            p+=(2*(x-y))+5;
            y--;
        }
        x++;
```

```
int main(){
    float x, y,a, b, r, p;
    int gd = DETECT, gm;
                initgraph(&gd, &gm, "");
     cout<< "Enter center of the circle(a, b): ";
    cin>>a>>b;
    cout<<"Enter radius of the circle: ";
    cin>>r;
   drawcircle(a, b, r);
    getch();
                closegraph();
}
        putpixel (a+x, b+y, WHITE);
        putpixel (a-x, b+y, WHITE);
        putpixel (a+x, b-y, WHITE);
        putpixel (a-x, b-y, WHITE);
        putpixel (a+y, b+x, WHITE);
        putpixel (a+y, b-x, WHITE);
        putpixel (a-y, b+x, WHITE);
        putpixel (a-y, b-x, WHITE);
        delay(100);
    }
}
```

Output:



2.Scan conversion of an Ellipse

Sourcecode:

```
#include<bits/stdc++.h>
#include<graphics.h>
using namespace std;
void drawEllipse(int centerX,int centerY,int radiusX,int radiusY)
 {
    int x,y;
    float d1,d2;
    initwindow(800,600,"Scan Conversion of Ellipse");
    setcolor(WHITE);
        if(d1<0)
            {
            d1+=(2*radiusY*radiusY*(x+1))+(radiusY*radiusY);
        }
```

```
    x=0,y=radiusY;
    d1=(radiusY*radiusY)-
(radiusX*radiusX*radiusY)+(0.25*radiusX*radiusX);
    putpixel(centerX+x,centerY+y,WHITE);
    putpixel(centerX+x,centerY-y,WHITE);
    putpixel(centerX+x,centerY+y,WHITE);
    putpixel(centerX-x,centerY-y,WHITE);
while((radiusX * radiusX * (y - 0.5)) > (radiusY * radiusY * (x +
1))) {
           d2+=(2 * radiusY * radiusY * (x + 1))-(2 * radiusX *
radiusX * (y - 1))+(radiusX * radiusX);
            x++;
        }
        else
            {
            d1+=(2*radiusY*radiusY*(x+1))+(radiusY*radiusY)-
(2*radiusX*radiusX*(y-1));
            y--;
        }
        x++;
        putpixel(centerX+x,centerY+y,WHITE);
        putpixel(centerX+x,centerY-y,WHITE);
        putpixel(centerX-x,centerY+y,WHITE);
        putpixel(centerX-x,centerY-y,WHITE);
    }
    d2=((radiusY*(x+0.5))*(radiusY*(x+0.5)))+((radiusX* (y -
1))*(radiusX* (y - 1)))-(radiusX * radiusX * radiusY * radiusY);

    while (y > 0)
     {
        if (d2 < 0)
            {
```

```
else
            d2+= -(2*radiusX*radiusX*(y - 1))+(radiusX * radiusX);
        y--;
putpixel(centerX + x, centerY + y, WHITE);
        putpixel(centerX + x, centerY - y, WHITE);
        putpixel(centerX - x, centerY + y, WHITE);
        putpixel(centerX - x, centerY - y, WHITE);
    }
    delay(50000);
    closegraph();
}
int main() {
    int centerX = 400;
    int centerY = 300;
    int radiusX = 200;
    int radiusY = 100;
    drawEllipse(centerX, centerY, radiusX, radiusY);
    return 0;
}
```

Output: