

SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN, UTTARAKHAND



COMPUTER GRAPHICS

LABORATORY FILE

(2024-2025)

For
Vth Semester

Submitted To:

Mr. Dinesh
Assistant Professor
[Vth Semester]
School of Computer Science

Submitted By:

Akshat Negi
500106533(SAP ID)
R2142220414(Roll No.)
B.Tech. CSF (Batch-1)

LAB EXPERIMENT – 3

Drawing a Circle and an Ellipse

[Usage of Open GL]

Take the value of radius, major axis and minor axis as input from the user.

a) Draw the circle with the help of polar equations

```
#include <GL/freeglut.h>
#include <iostream>
#include <cmath>
#include <math.h>
# define M_PI          3.14159265358979323846  /* pi */
using namespace std;

int radius = 100;
int centerX = 320;
int centerY = 240;

void init() {
    glClearColor(1.0, 1.0, 1.0, 1.0);
    gluOrtho2D(0, 640, 0, 480);
}

void drawCirclePolarEquation() {
    glBegin(GL_LINE_LOOP);
    for (double angle = 0; angle <= 360; angle += 1) {
        double x = centerX + radius * cos(angle * M_PI / 180);
        double y = centerY + radius * sin(angle * M_PI / 180);
        glVertex2i(x, y);
    }
    glEnd();
}

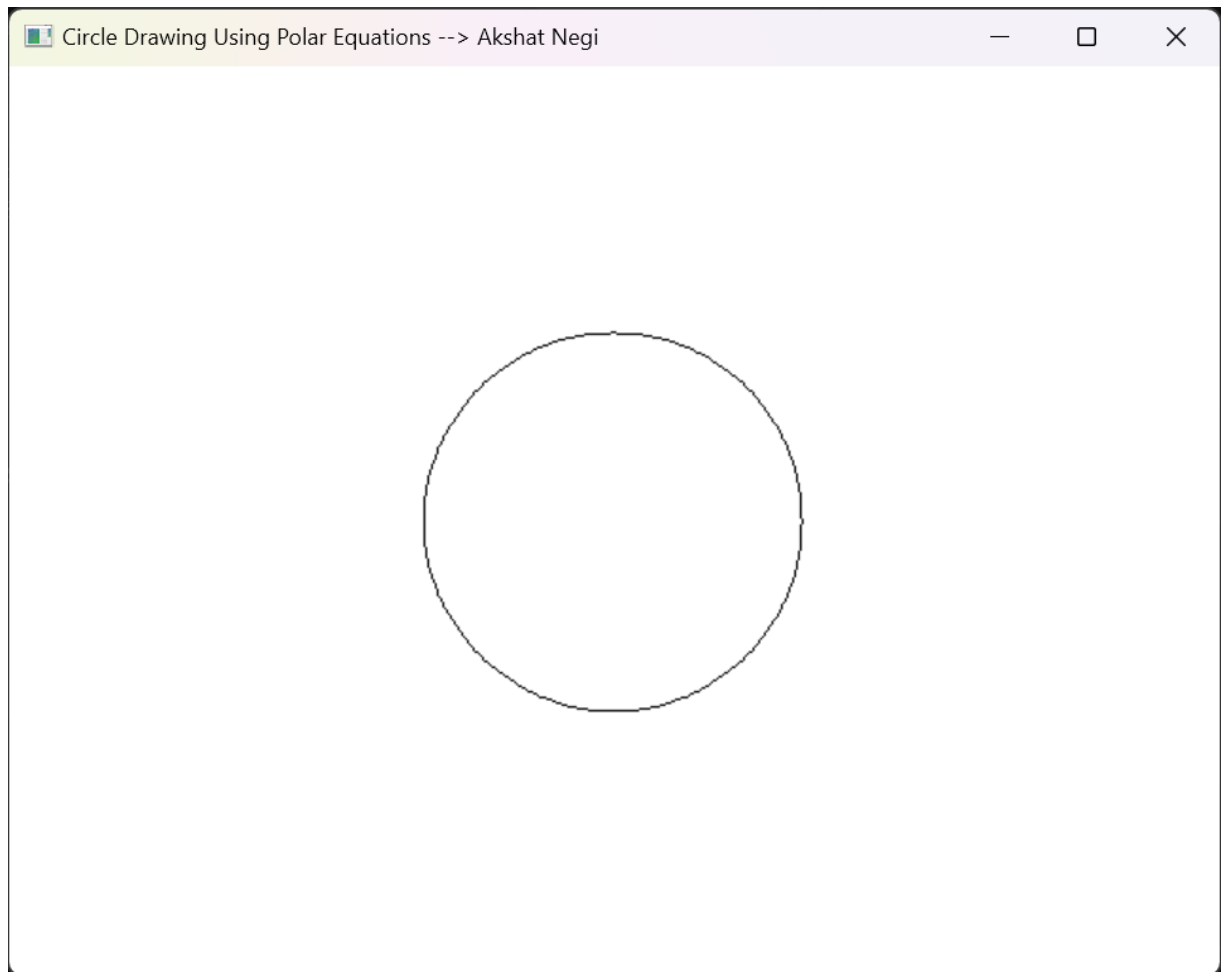
void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.0, 0.0);

    // Draw the circle using polar equations
    drawCirclePolarEquation();

    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Circle Drawing Using Polar Equations --> Akshat Negi");
    init();
    glutDisplayFunc(display);
}
```

```
glutMainLoop();  
return 0;  
}
```



b) Draw the circle with the help of mid-point method.

```
#include <iostream>  
#include<math.h>  
#include<GL/freeglut.h>  
  
using namespace std;  
  
void circle() {  
    glColor3f(0.0, 0.0, 0.0);  
    glPointSize(2.0);  
    float r = 100;  
    float x = 0, y = r;  
    float p = 1 - r;  
    glBegin(GL_POINTS);  
    while (x != y)  
    {  
        x++;  
        y--;  
        p = p - x + y;  
        if (p < 0)  
            p = 2 * y - p;  
        if (p > 0)  
            p = 2 * x - p;  
        glVertex2f(x, y);  
    }  
    glEnd();  
}
```

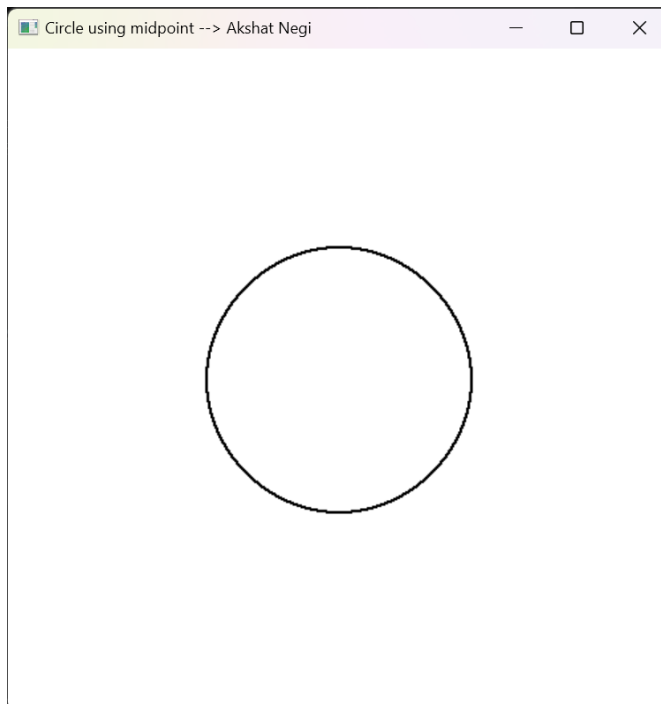
```

        if (p < 0) {
            p += 2 * (x + 1) + 1;
        }
        else {
            y--;
            p += 2 * (x + 1) + 1 - 2 * (y - 1);
        }
        glVertex2i(x, y);
        glVertex2i(-x, y);
        glVertex2i(x, -y);
        glVertex2i(-x, -y);

        glVertex2i(y, x);
        glVertex2i(-y, x);
        glVertex2i(y, -x);
        glVertex2i(-y, -x);
    }
    glEnd();
    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Circle using midpoint --> Akshat Negi");
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT);
    gluOrtho2D(-250, 250, -250, 250);
    glMatrixMode(GL_PROJECTION);
    glViewport(0, 0, 500, 500);
    glutDisplayFunc(circle);
    glutMainLoop();
    return 0;
}

```



c) Draw the Ellipse with the mid-point method.

```
#include <GL/freeglut.h>
#include <iostream>
using namespace std;
int rx, ry;
int xi, yi;

void ellipseMidPoint() {
    int x = 0, y = ry;
    int p1 = (ry * ry) + (rx * rx * 0.25) - (ry * rx * rx);
    int dx = 2 * x * (ry * ry);
    int dy = 2 * y * (rx * rx);
    while (dy > dx) {
        glVertex2i(x + xi, y + yi);
        glVertex2i(x + xi, -y + yi);
        glVertex2i(-x + xi, -y + yi);
        glVertex2i(-x + xi, y + yi);
        if (p1 < 0) {
            x++;
            dx = 2 * x * (ry * ry);
            p1 += dx + (ry * ry);
        }
        else {
            x++;
            y--;
            dx = 2 * x * (ry * ry);
            dy = 2 * y * (rx * rx);
            p1 += dx + (ry * ry) - dy;
        }
    }
    int p2 = (ry * ry * (x + 0.5) * (x + 0.5)) + (rx * rx * (y - 1) * (y - 1))
- (rx * rx * ry * ry);
    while (y > 0) {
        glVertex2i(x + xi, y + yi);
        glVertex2i(x + xi, -y + yi);
        glVertex2i(-x + xi, -y + yi);
        glVertex2i(-x + xi, y + yi);
        if (p2 > 0) {
            y--;
            dy = 2 * y * (rx * rx);
            p2 += (rx * rx) - dy;
        }
        else {
            y--;
            x++;
            dy -= 2 * (rx * rx);
            dx += 2 * (ry * ry);
            p2 += dx + (rx * rx) - dy;
        }
    }
}

void display() {
    //glClear(GL_COLOR_BUFFER_BIT); already mentioned in main program
    glColor3f(0.0, 1.0, 1.0);
    glPointSize(5.0);
    glBegin(GL_POINTS);
    //int rx = 40, ry = 50, xi = 200, yi = 250;
    ellipseMidPoint();
}
```

```

        glEnd();
        glFlush();
    }
    int main(int argc, char** argv)
    {
        cout << "\n\nEnter Center Of Ellipse \n\n";
        cout << "\n x = ";
        cin >> xi;

        cout << "\n y = ";
        cin >> yi;

        cout << " Enter a Semi Major Axix : ";
        cin >> rx;
        cout << " \nEnter a Semi Minor Axis: ";
        cin >> ry;

        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(100, 100);
        glutCreateWindow("Drawing Algorithm --> Akshat Negi");
        glClearColor(0.0, 0.0, 0.0, 1.0);
        glClear(GL_COLOR_BUFFER_BIT);
        gluOrtho2D(0, 500, 0, 500);
        glMatrixMode(GL_PROJECTION);
        glViewport(0, 0, 500, 500);
        glutDisplayFunc(display);
        glutMainLoop();
        return 0;
    }
}

```

