

UCS 1712 – GRAPHICS AND MULTIMEDIA LAB

ASSIGNMENT – 4

VISHAL N

185001198

08.08.2021

CSEC

1. MIDPOINT CIRCLE DRAWING ALGORITHM:

```
#include <GL/glut.h>
#include <stdio.h>
#include <iostream>
using namespace std;

int x_centre, y_centre, r;

void myInit() {
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPointSize(0.05);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-600.0, 600.0, -600.0, 600.0);
}

void draw_pixel(int x, int y) {
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

void midPointCircleDraw(int x_centre, int y_centre, int r)
{
    int x = r, y = 0;

    if (r > 0)
    {
        draw_pixel(x + x_centre, -y + y_centre);
        draw_pixel(y + x_centre, x + y_centre);
        draw_pixel(-y + x_centre, x + y_centre);
    }

    int P = 1 - r;
```

```

while (x > y)
{
    y++;

    if (P <= 0)
        P = P + 2*y + 1;

    else
    {
        x--;
        P = P + 2*y - 2*x + 1;
    }

    if (x < y)
        break;

    draw_pixel(x + x_centre, y + y_centre);
    draw_pixel(-x + x_centre, y + y_centre);
    draw_pixel(x + x_centre, -y + y_centre);
    draw_pixel(-x + x_centre, -y + y_centre);

    if (x != y)
    {
        draw_pixel(y + x_centre, x + y_centre);
        draw_pixel(-y + x_centre, x + y_centre);
        draw_pixel(y + x_centre, -x + y_centre);
        draw_pixel(-y + x_centre, -x + y_centre);
    }
}

}

void myDisplayMidpointCircleDrawing(){
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINES);
    glVertex2d(-600, 0);
    glVertex2d(600, 0);
    glEnd();
    glBegin(GL_LINES);
    glVertex2d(0, -600);
    glVertex2d(0, 600);
    glEnd();
    midPointCircleDraw(x_centre, y_centre, r);
    glFlush();
}

int main(int argc, char **argv) {

```

```

cout << "Enter (x_centre, y_centre, r)" << endl;
cin >> x_centre >> y_centre >> r;
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(600, 600);
glutInitWindowPosition(0, 0);
glutCreateWindow("Midpoint Circle Drawing Algorithm");
myInit();
glutDisplayFunc(myDisplayMidpointCircleDrawing);
glutMainLoop();
return 0;
}

```

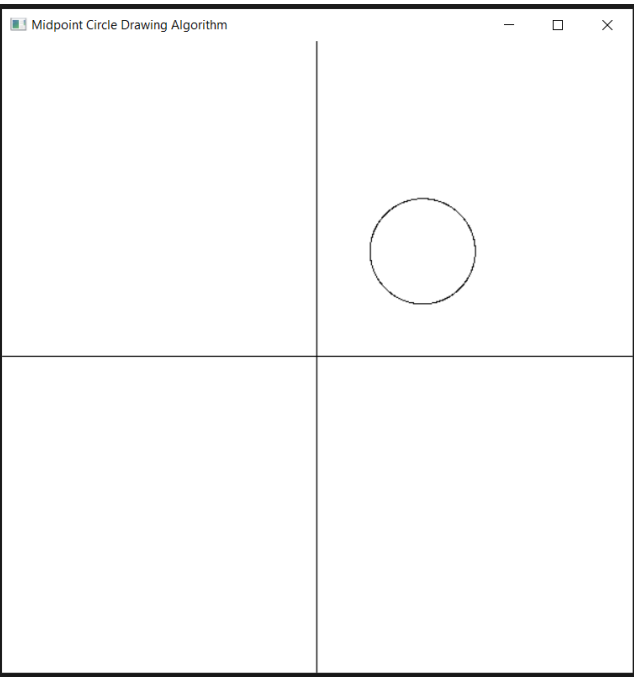
OUTPUT:

1)



2)

```
PS C:\SSN 7th sem materials\Graphics-and-Multimedia-Lab\198-A4> ./circle
Enter (x_centre, y_centre, r)
0 0 100
PS C:\SSN 7th sem materials\Graphics-and-Multimedia-Lab\198-A4> ./circle
Enter (x_centre, y_centre, r)
200 200 100
█
```

A screenshot of a Windows application window titled "Midpoint Circle Drawing Algorithm". The window contains a 2D coordinate system with a horizontal and vertical axis intersecting at the center. A circle is drawn in the first quadrant, centered at (200, 200) with a radius of 100. The circle is drawn using a midpoint circle drawing algorithm, resulting in a smooth, pixelated appearance. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.