

EXP3

AIM: Drawing a circle using circle generation algorithm

CODE:

```
#include <stdio.h>

#include <iostream>

#include <GL/glut.h>

using namespace std;

int pntX1, pntY1, r;

void plot(int x, int y)
{
    glBegin(GL_POINTS);
    glVertex2i(x+pntX1, y+pntY1);
    glEnd();
}

void myInit (void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPointSize(4.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}

void midPointCircleAlgo()
```

```

{
    int x = 0;
    int y = r;
    float decision = 5/4 - r;
    plot(x, y);

    while (y > x)
    {
        if (decision < 0)
        {
            x++;
            decision += 2*x+1;
        }
        else
        {
            y--;
            x++;
            decision += 2*(x-y)+1;
        }
        plot(x, y);
        plot(x, -y);
        plot(-x, y);
        plot(-x, -y);
        plot(y, x);
        plot(-y, x);
        plot(y, -x);
        plot(-y, -x);
    }
}

```

```

void myDisplay(void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (0.0, 0.0, 0.0);
    glPointSize(1.0);

    midPointCircleAlgo();

    glFlush ();
}

int main(int argc, char** argv)
{
    cout << "Enter the coordinates of the center:\n\n" << endl;

    cout << "X-coordinate  : "; cin >> pntX1;
    cout << "\nY-coordinate : "; cin >> pntY1;
    cout << "\nEnter radius : "; cin >> r;

    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (640, 480);
    glutInitWindowPosition (100, 150);
    glutCreateWindow ("Circle Mid Point Algorithm");
    glutDisplayFunc(myDisplay);
    myInit ();
    glutMainLoop();
}

```

OUTPUT:

