

**NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR**

**Cachar, Assam**

**B.Tech. VI<sup>th</sup> Sem**

**Subject Code:** CS-317

**Subject Name:** Graphics and Multimedia Lab

**Submitted By:**

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

## 1. Implementation of Midpoint ellipse drawing algorithm

- Divide the coordinate axes into four quadrants and then draw only a portion of an ellipse with the major axis as X-axis.
- Draw another portion of the ellipse with the major axis as Y-axis to the quadrant diagonally opposite to each other.

Assign different colours to them respectively.

➔ CODE:

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

GLint rx, ry; //semi-Major axis & semi Minor Axis (OFFSET)
GLint xCenter, yCenter; //center of ellipse
GLint showQuadI, showQuadII, showQuadIII, showQuadIV;
GLint xxStore [4][1000], yyStore [4][1000], size [4];

void myinit (void) {
    glClear (GL_COLOR_BUFFER_BIT);
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (-500, 500, -500, 500);
}

void setPixel (GLint plotX, GLint plotY) {
    if (showQuadI && (plotX >= 0 && plotY >= 0)) {
        xxStore [0][(size[0])] = plotX; yyStore [0][(size[0])] = plotY;
        ++size[0];
    }
    if (showQuadII && (plotX <= 0 && plotY >= 0)) {
        xxStore [1][(size[1])] = plotX; yyStore [1][(size[1])] = plotY;
        ++size[1];
    }
    if (showQuadIII && (plotX <= 0 && plotY <= 0)) {
        xxStore [2][(size[2])] = plotX; yyStore [2][(size[2])] = plotY;
        ++size[2];
    }
    if (showQuadIV && (plotX >= 0 && plotY <= 0)) {
        xxStore [3][(size[3])] = plotX; yyStore [3][(size[3])] = plotY;
        ++size[3];
    }
}

void drawPixel (GLint quadNum, GLint xaxis, GLint yaxis) {
```

```

glBegin (GL_POLYGON);
    for (GLint ii = 0; ii < size[quadNum]; ++ii)
        glVertex2i (xxStore[quadNum][ii], yyStore[quadNum][ii]);
    glVertex2i (xaxis, 0);
    glVertex2i (0, 0);
    glVertex2i (0, yaxis);
glEnd ();
glFlush();
}

void ellipseMidPoint (GLint QI, GLint QII, GLint QIII, GLint QIV) {
    showQuadI = QI;
    showQuadII = QII;
    showQuadIII = QIII;
    showQuadIV = QIV;

    size [0] = 0; size [1] = 0; size [2] = 0; size [3] = 0;

    float xx = 0;
    float yy = ry;
    float p1 = ry * ry - (rx * rx) * ry + (rx * rx) * (0.25);
    //slope
    float dx = 2 * (ry * ry) * xx;
    float dy = 2 * (rx * rx) * yy;
    float p2 = (ry * ry) * (xx + 0.5) * (xx + 0.5) + (rx * rx) * (yy -
1) * (yy - 1) - (rx * rx) * (ry * ry);

    //plotting for 1st region of 1st quadrant and the slope will be < -
1
    while (dx < dy) {
        //plot (x,y)
        setPixel (xCenter + xx, yCenter + yy);
        setPixel (xCenter - xx, yCenter + yy);
        setPixel (xCenter + xx, yCenter - yy);
        setPixel (xCenter - xx, yCenter - yy);
        if (p1 < 0) {
            xx = xx + 1;
            dx = 2 * (ry * ry) * xx;
            p1 = p1 + dx + (ry * ry);
        }
        else {
            xx = xx + 1;
            yy = yy - 1;
            dx = 2 * (ry * ry) * xx;
            dy = 2 * (rx * rx) * yy;
            p1 = p1 + dx - dy + (ry * ry);
        }
    }
}

```

```

// Plotting for 2nd region of 1st quadrant and the slope will be > -
1
while (yy > 0) {
    //plot (x,y)
    setPixel (xCenter + xx, yCenter + yy);
    setPixel (xCenter - xx, yCenter + yy);
    setPixel (xCenter + xx, yCenter - yy);
    setPixel (xCenter - xx, yCenter - yy);
    if (p2 > 0) {
        xx = xx;
        yy = yy - 1;
        dy = 2 * (rx * rx) * yy;
        p2 = p2 - dy + (rx * rx);
    }
    else {
        xx = xx + 1;
        yy = yy - 1;
        dy = dy - 2 * (rx * rx) ;
        dx = dx + 2 * (ry * ry) ;
        p2 = p2 + dx -
            dy + (rx * rx);
    }
}
if (showQuadI) drawPixel (0, rx ,ry);
if (showQuadII) drawPixel (1, -rx, ry);
if (showQuadIII) drawPixel (2, -rx, -ry);
if (showQuadIV) drawPixel (3, rx, -ry);
}

void drawQuadrants () {
    glPointSize (3.0);
    glColor3f (0.0f, 0.5f, 0.5f);

    glBegin (GL_LINE_LOOP);
        glVertex3f (-500.0, 0.0, 0.0);
        glVertex3f (500.0, 0.0, 0.0);
    glEnd ();

    glBegin (GL_LINE_LOOP);
        glVertex3f (0.0, -500.0, 0.0);
        glVertex3f (0.0, 500.0, 0.0);
    glEnd ();
}

void display () {
    // A. PORTION OF AN ELLIPSE WITH THE MAJOR AXIS AS X-AXIS
    xCenter = 0; yCenter = 0; rx = 400; ry = 100;
    glColor3f (1.0, 0.0, 0.0);

```

```

    ellipseMidPoint (1, 0, 0, 0); // Displays only Quadrant I

    // B. PORTION OF AN ELLIPSE WITH THE MAJOR AXIS AS Y-AXIS
    xCenter = 0; yCenter = 0; rx = 100; ry = 400;
    glColor3f (0.0, 1.0, 0.0);
    ellipseMidPoint (0, 0, 1, 0); // Displays only Quadrant III

    drawQuadrants ();
    glFlush ();
}

int main (int argc, char **argv) {
    glutInit (&argc, argv);
    glutInitWindowSize (1000, 1000);
    glutInitWindowPosition (500, 0);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow ("Midpoint Ellipse Drawing Algorithm");
    myinit ();
    glutDisplayFunc (display);
    glutMainLoop ();
    return 0;
}

```

**OUTPUT:**

