**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**Ex. No.7 Cohen Sutherland Line clipping in C++ using OpenGL**

**Date:** 6/9/21 **Name:** Srinath S

**Class:** CSE-C  **Roll:** 185001205

**Question:**

Apply Cohen Sutherland line clipping on a line (x1,y1) (x2,y2) with respect to a clipping window (XWmin,YWmin) (XWmax,YWmax).

After clipping with respect to an edge, display the line segment with the calculated intermediate intersection points and the vertex list.

Input: The clipping window co-ordinates and the line endpoints

Note: The output should show the clipping window and the line to be clipped in different colors. You can show the intermediate steps using time delay.

**Code:**

#include <bits/stdc++.h>

#include <GL/glut.h>

using namespace std;

using ld = long double;

const int WINDOW\_WIDTH = 850;

const int WINDOW\_HEIGHT = 700;

void myInit();

void myDisplay();

void printCohen();

const ld PADDING = 250;

const ld STEP = 1;

const ld SCALE = 5;

const int xmin = -50;

const int xmax = 50;

const int ymin = -50;

const int ymax = 50;

double x1 = -54.0;

double y11 = 60.0;

double x2 = 40.0;

double y2 = 55.0;

int main(int argc, char \*argv[])

{

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);

printf("COHEN SUTHERLAND CLIPPING ALGORITHM\n Window:xmin,ymin =

(-50,-50) xmax,ymax = (50,50)\n");

printf("Enter the value of x1 coord:");

scanf("%lf",&x1);

printf("Enter the value of y1 coord : ");

scanf("%lf",&y11);

printf("Enter the value x2 coord : ");

scanf("%lf",&x2);

printf("Enter the value y2 coord : ");

scanf("%lf",&y2);

glutCreateWindow("Cohen's Algorithm");

glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

return 1;

}

void myInit()

{

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(15.0);

    glLineWidth(4);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(-100.0, 100, -100, 100);

}

void myDisplay()

{

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glColor3f(0.0f, 0.0f, 0.0f);

    glRectf(xmin, ymin, xmax, ymax);

    glColor3f(0, 0, 1);

    glBegin(GL\_LINES);

    glVertex2f(x1, y11);

    glVertex2f(x2, y2);

    glEnd();

    printCohen();

    glFlush();

}

char sign(int x)

{

    if (x <= 0)

        return '0';

    else

        return '1';

}

string codeGen(double x, double y)

{

    string code = "";

    code += sign(y - ymax);

    code += sign(ymin - y);

    code += sign(x - xmax);

    code += sign(xmin - x);

    return code;

}

pair<double, double> CohenPoint(int x, int y, string code, double m)

{

    pair<double, double> point;

    point.first = x;

    point.second = y;

    double xclip, yclip;

    if (code[0] == '1')

    {

        xclip = x + (ymax - y) / m;

        if (xclip <= xmax && xclip >= xmin)

        {

            point.first = xclip;

            point.second = ymax;

        }

    }

    if (code[1] == '1')

    {

        xclip = x + (ymin - y) / m;

        if (xclip <= xmax && xclip >= xmin)

        {

            point.first = xclip;

            point.second = ymin;

        }

    }

    if (code[2] == '1')

    {

        yclip = y + m \* (xmax - x);

        if (yclip <= ymax && yclip >= ymin)

        {

            point.first = xmax;

            point.second = yclip;

        }

    }

    if (code[3] == '1')

    {

        yclip = y + m \* (xmin - x);

        if (yclip <= ymax && yclip >= ymin)

        {

            point.second = yclip;

            point.first = xmin;

        }

    }

    cout << point.first << " " << point.second << " ";

    return point;

}

void printCohen()

{

    string code1, code2;

    pair<double, double> p1, p2;

    code1 = codeGen(x1, y11);

    code2 = codeGen(x2, y2);

    cout << code1 << " " << code2;

    int code\_int1, code\_int2;

    code\_int1 = stoi(code1, 0, 2);

    code\_int2 = stoi(code2, 0, 2);

    cout << code\_int1 << " " << code\_int2;

    double m;

    m = (y2 - y11) \* 1.0 / (x2 - x1);

    if ((code\_int1 | code\_int2) == 0)

    {

        cout << "Line Inside";

        glColor3f(0, 1, 1);

        glBegin(GL\_LINES);

        glVertex2f(x1, y11);

        glVertex2f(x2, y2);

        glEnd();

    }

    else if ((code\_int1 & code\_int2) != 0)

    {

        cout << "Line Outside";

    }

    else

    {

        cout << "Line to be clipped\n";

        p1 = CohenPoint(x1, y11, code1, m);

        p2 = CohenPoint(x2, y2, code2, m);

        glColor3f(1, 0, 1);

        glBegin(GL\_LINES);

        glVertex2f(p1.first, p1.second);

        glVertex2f(p2.first, p2.second);

        cout << "Line after Clipping: "

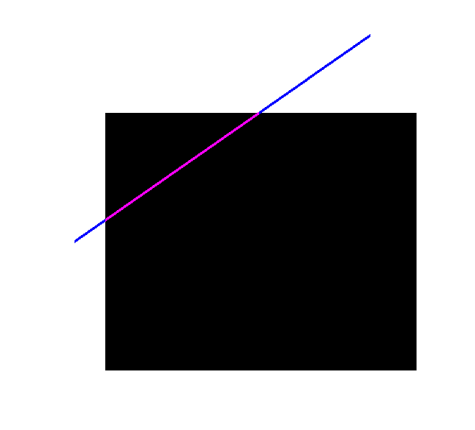
             << "p1 :(" << p1.first << ", " << p1.second << ")p2 : (" << p2.first << ", " << p2.second << ")\n ";

        glEnd();

    }

}

**Output:**

****