Graphics Assignment No: 7

Shikhar Jaiswal
1601CS44

April 4, 2019

1 Assignment Description

Implement Cohen-Sutherland line clipping algorithm and draw and clip a line segment using its end points.

2 Procedure

Installation

Install the following packages from the Ubuntu repository:
- freeglut3-dev
- mesa-common-dev
- libglm-dev

```
sudo apt-get install freeglut3 freeglut3-dev mesa-common-dev
libglm-dev
```

Check your */usr/include/GL* folder to verify the installation of the OpenGL headers that you intend to use.

Compiling and Linking

We will have to use the $-lglut$ linker option with $gcc/g{+}{+}$ to compile a program with *glut* library.

For example, to compile the program, use the following to get the binary executable code:

```
g++ cohen_sutherland.cpp -lGL -lGLU -lglut -o cohen_sutherland
```

3 Discussion

The primary objective of the assignment is to implement an algorithm capable of determining the end points of a line segment with respect to a clipping window. For this purpose, Cohen-Sutherland line clipping algorithm is implemented. The algorithm works by tracing the end points of the line segment, and assigning them to one of the eight zones (with respect to the clipping window). Then, the segments are clipped according to the zone of interest. For a detailed explanation, please refer the code.

OpenGL Code

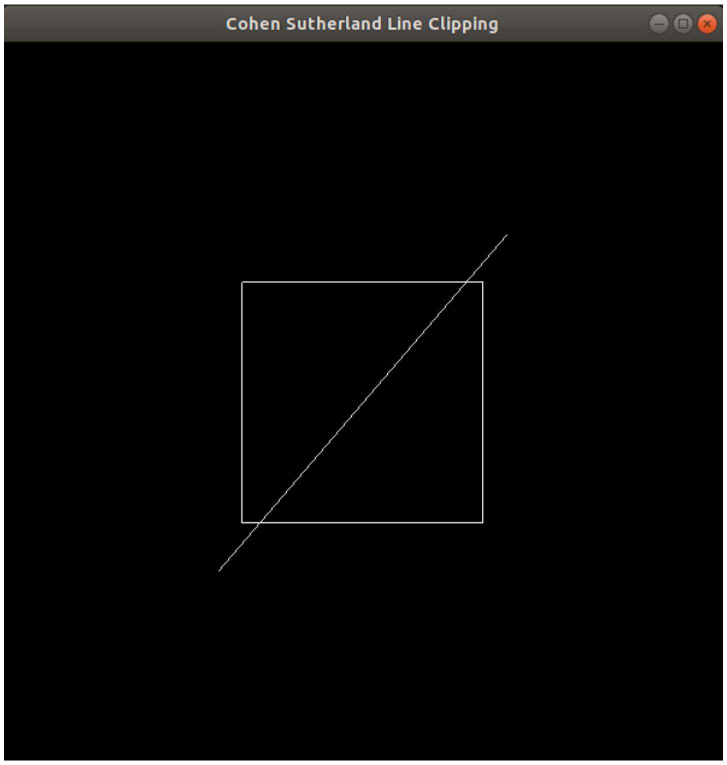
```
#include <iostream>
#include <GL/glut.h>

using namespace std;

// Parameters for setting the window dimensions
// and line segment points.
float xmin = -100;
float ymin = -100;
float xmax = 100;
float ymax = 100;
float xd1, yd1, xd2, yd2;

// Function for assigning the section codes to end points.
int code(float x, float y)
{
    int c = 0;
    if (y > ymax)
    {
        c = 8;
    }
    if (y < ymin)
    {
        c = 4;
    }
    if (x > xmax)
    {
        c = c | 2;
    }
    if (x < xmin)
    {
        c = c | 1;
    }
    return c;
}

// Function to display the drawn and clipped line.
void display()
{
    // Set the display area colour set using glClearColor().
    glClear(GL_COLOR_BUFFER_BIT);

    // Set the colour of the clipping window.
    glColor3ub(255, 255, 255);
    // Draw the clipping window.
    glBegin(GL_LINE_LOOP);
        glVertex2i(xmin, ymin);
        glVertex2i(xmin, ymax);
        glVertex2i(xmax, ymax);
        glVertex2i(xmax, ymin);
    glEnd();

    // Set the colour of the line segment.
    glColor3ub(230, 230, 230);
    // Draw the line segment.
    glBegin(GL_LINES);
        glVertex2i(xd1, yd1);
        glVertex2i(xd2, yd2);
    glEnd();

    glFlush();
}

// Function to compute the clipped points of the line.
void cohen_line(float x1, float y1, float x2, float y2)
{
    int c1 = code(x1, y1);
    int c2 = code(x2, y2);
    float m = (y2 - y1) / (x2 - x1);
    while ((c1 | c2) > 0)
    {
        if ((c1 & c2) > 0)
        {
           exit(0);
        }

        float xi = x1;
        float yi = y1;
        int c = c1;
        if (c == 0)
        {
            c = c2;
            xi = x2;
            yi = y2;
        }

        float x, y;
        if ((c & 8) > 0)
        {
            y = ymax;
            x = xi + 1.0 / m * (ymax - yi);
        } else if ((c & 4) > 0)
        {
            y = ymin;
            x = xi + 1.0 / m * (ymin - yi);
        } else if ((c & 2) > 0)
        {
            x = xmax;
            y = yi + m * (xmax - xi);
        } else if ((c & 1) > 0)
        {
            x = xmin;
            y = yi + m * (xmin - xi);
        }

        if (c == c1)
        {
            xd1 = x;
            yd1 = y;
            c1 = code(xd1, yd1);
        }

        if (c == c2)
        {
           xd2 = x;
           yd2 = y;
           c2 = code(xd2, yd2);
        }
    }

    display();
}

// Function to trigger the line clipping.
void mykey(unsigned char key, int x, int y)
```

```
{
    // On pressing the key `a`, the clipper gets triggered.
    if (key == 'a')
    {
        cohen_line(xd1, yd1, xd2, yd2);
        glFlush();
    }
}

int main(int argc, char **argv)
{
    cout << "Welcome To Cohen Sutherland Line Clipper" << endl;
    cout << "Enter Line Co-Ordinates:" << endl;
    cin >> xd1 >> yd1 >> xd2 >> yd2;

    // Initialize to the command-line arguments.
    glutInit(&argc, argv);
    // Setup the colour depth of the window buffers.
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

    // Assign the position, size and name to the window.
    glutInitWindowSize(600, 600);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Cohen Sutherland Line Clipping");

    // Setup a black background.
    glClearColor(0.0, 0.0, 0.0, 0.0);
    // Setup viewing projection.
    glMatrixMode(GL_PROJECTION);
    // Setup a viewport.
    gluOrtho2D(-300.0, 300.0, -300.0, 300.0);

    // Pass the display function to generate the display.
    glutDisplayFunc(display);
    // Pass the keyboard function to trigger the clipping.
    glutKeyboardFunc(mykey);
    // Hand over the execution to the glut library.
    glutMainLoop();

    return 0;
}
```

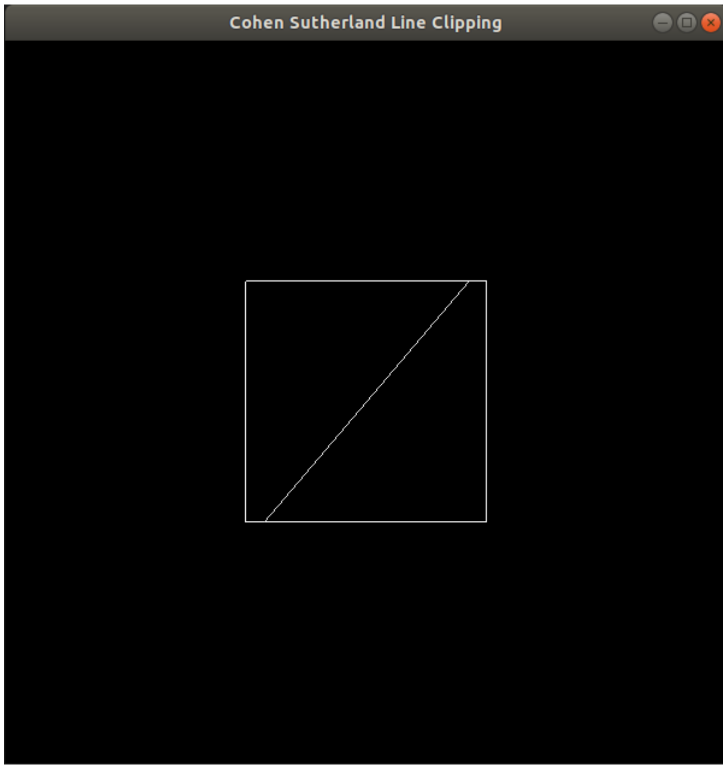
4 Result

OpenGL

Original



Clipped



5 References

[1] How to install OpenGL/GLUT libraries

[2] An Introduction to OpenGL Programming

[3] Turtle Graphics