

# Department of Computer Science and Engineering

Shivanirudh S G, 185001146, Semester VII

28 September 2021

---

## UCS1712 - Graphics and Multimedia Lab

---

### Exercise 7: Cohen Sutherland Line clipping in C++ using OpenGL

#### Objective:

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

#### Code:

```
1  #ifndef LOPENGL_H
2  #define LOPENGL_H
3
4  #include <GL/freeglut.h>
5  #include <GL/gl.h>
6  #include <GL/glu.h>
7  #include <math.h>
8  #include <stdio.h>
9  #include<iostream>
10 #include<vector>
11 #include<ctime>
12 #include <unistd.h>
13 using namespace std;
14
15 #endif
```

```

1  #ifndef LUTIL_H
2  #define LUTIL_H
3
4  #include "Headers.h"
5
6  //Screen Constants
7  const int SCREEN_WIDTH = 640;
8  const int SCREEN_HEIGHT = 480;
9  const int SCREEN_FPS = 60;
10 const int POINT_SIZE=2;
11
12 //pairs of the form (min, max)
13 pair<double, double> window_x_dims, window_y_dims;
14
15 pair<double, double> original_point1, new_point1;
16 pair<double, double> original_point2, new_point2;
17
18 int region_code1[4] = {0, 0, 0, 0},
19     region_code2[4] = {0, 0, 0, 0};
20
21 bool initGL();
22
23 void update();
24
25 void render();
26
27 void lineloop(double x1, double y1, double x2, double y2);
28
29 void drawWindow();
30
31 void drawOriginalLine();
32
33 void computeRegionCode1();
34
35 void computeRegionCode2();
36
37 bool trivial_accept();
38
39 bool trivial_reject();
40
41 void drawClippedLine();
42
43 void CohenSutherland();
44
45 #endif
46
47 #include "Signatures.h"
48
49 bool initGL(){
50     //Initialize Projection Matrix
51     glMatrixMode( GL_PROJECTION );
52     glLoadIdentity();
53     gluOrtho2D(0.0,640.0,0.0,480.0);
54
55     //Initialize Modelview Matrix
56     glMatrixMode( GL_MODELVIEW );
57     glLoadIdentity();
58
59

```

```

13     // glTranslatef( SCREEN_WIDTH / 3.f, SCREEN_HEIGHT / 3.f, 0.f )
14     ;
15     //Initialize clear color
16     glClearColor( 0.f, 0.f, 0.f, 1.f );
17
18     glPointSize(POINT_SIZE);
19     glEnable(GL_POINT_SMOOTH);
20
21     //Check for error
22     GLenum error = glGetError();
23     if( error != GL_NO_ERROR )
24     {
25         printf( "Error initializing OpenGL! %s\n", gluErrorString(
26             error ) );
27         return false;
28     }
29     return true;
30 }
31
32 void update(){
33
34 }
35
36 void render(){
37     drawWindow();
38     drawOriginalLine();
39
40     CohenSutherland();
41
42     glFlush();
43 }
44
45 void lineloop(double x1, double y1, double x2, double y2) {
46
47     glBegin(GL_LINE_LOOP);
48
49     glVertex2d(x1,y1);
50     glVertex2d(x2,y1);
51     glVertex2d(x2,y2);
52     glVertex2d(x1,y2);
53
54     glEnd();
55 }
56
57 void drawWindow(){
58     glColor3f(1.0,1.0,1.0);
59     lineloop(window_x_dims.first, window_y_dims.first,
60         window_x_dims.second, window_y_dims.second);
61 }
62
63 void drawOriginalLine(){
64     glColor3f(1.0,0.0,0.0);
65     glBegin(GL_LINES);
66     glVertex2d(original_point1.first, original_point1.second);

```

```

67     glVertex2d(original_point2.first, original_point2.second);
68
69     glEnd();
70     glFlush();
71 }
72
73 void computeRegionCode1(){
74     if(new_point1.second > window_y_dims.second){
75         region_code1[0] = 1;
76     }
77     if(new_point1.second < window_y_dims.first){
78         region_code1[1] = 1;
79     }
80     if(new_point1.first > window_x_dims.second){
81         region_code1[2] = 1;
82     }
83     if(new_point1.first < window_x_dims.first){
84         region_code1[3] = 1;
85     }
86 }
87
88 void computeRegionCode2(){
89     if(new_point2.second > window_y_dims.second){
90         region_code2[0] = 1;
91     }
92     if(new_point2.second < window_y_dims.first){
93         region_code2[1] = 1;
94     }
95     if(new_point2.first > window_x_dims.second){
96         region_code2[2] = 1;
97     }
98     if(new_point2.first < window_x_dims.first){
99         region_code2[3] = 1;
100    }
101 }
102
103 bool trivial_accept(){
104     computeRegionCode1();computeRegionCode2();
105
106     int sum=0;
107     for(int i=0;i<4;i++){
108         sum += (region_code1[i]+region_code2[i]);
109     }
110
111     return (sum == 0)?true:false;
112 }
113
114 bool trivial_reject(){
115     computeRegionCode1();computeRegionCode2();
116
117     int sum=0;
118     for(int i=0;i<4;i++){
119         sum += (region_code1[i]*region_code2[i]);
120     }
121
122     return (sum >= 1)?true:false;
123 }

```

```

124
125 void drawClippedLine(){
126     glColor3f(0.0,1.0,0.0);
127     glBegin(GL_LINES);
128
129     glVertex2d(new_point1.first, new_point1.second);
130     glVertex2d(new_point2.first, new_point2.second);
131
132     glEnd();
133     glFlush();
134 }
135
136 void CohenSutherland(){
137     new_point1 = original_point1;
138     new_point2 = original_point2;
139     for(int iteration=0; iteration<5; iteration++){
140         if(trivial_accept()){
141             cout<<"Trivially accepted"<<endl;
142             cout<<"New points are : ("
143                 <<new_point1.first<<", " <<new_point1.second
144                 <<") and ("
145                 <<new_point2.first<<", " <<new_point2.second
146                 <<")"<<endl;
147
148             sleep(2);
149             drawClippedLine();
150             sleep(3);
151             break;
152         }
153         if(trivial_reject()){
154             cout<<"Trivially rejected"<<endl;
155             sleep(3);
156             break;
157         }
158
159         double slope = (new_point2.second - new_point1.second)/(
new_point2.first - new_point1.first);
160         //Point 1
161         if(region_code1[0] == 1){
162             new_point1.second = window_y_dims.second;
163             new_point1.first = new_point2.first + (new_point1.
second - new_point2.second)/slope;
164         }
165         if(region_code1[1] == 1){
166             new_point1.second = window_y_dims.first;
167             new_point1.first = new_point2.first + (new_point1.
second - new_point2.second)/slope;
168         }
169         if(region_code1[2] == 1){
170             new_point1.first = window_x_dims.second;
171             new_point1.second = new_point2.second + (new_point1.
first - new_point2.first)*slope;
172         }
173         if(region_code1[3] == 1){
174             new_point1.first = window_x_dims.first;
175             new_point1.second = new_point2.second + (new_point1.
first - new_point2.first)*slope;

```

```

176     }
177
178
179     //Point 2
180     if(region_code2[0] == 1){
181         new_point2.second = window_y_dims.second;
182         new_point2.first = new_point1.first + (new_point2.
second - new_point1.second)/slope;
183     }
184     if(region_code2[1] == 1){
185         new_point2.second = window_y_dims.first;
186         new_point2.first = new_point1.first + (new_point2.
second - new_point1.second)/slope;
187     }
188     if(region_code2[2] == 1){
189         new_point2.first = window_x_dims.second;
190         new_point2.second = new_point1.second + (new_point2.
first - new_point1.first)*slope;
191     }
192     if(region_code2[3] == 1){
193         new_point2.first = window_x_dims.first;
194         new_point2.second = new_point1.second + (new_point2.
first - new_point1.first)*slope;
195     }
196
197     drawClippedLine();
198     cout<<"New points are : ("
199         <<new_point1.first<<", "<<new_point1.second
200         <<") and ("
201         <<new_point2.first<<", "<<new_point2.second
202         <<")"<<endl;
203     sleep(1);
204 }
205 }

1 #include "Helpers.h"
2
3 void runMainLoop(int val);
4
5 int main( int argc, char* args[] ){
6
7     glutInit( &argc, args );
8
9     glutInitContextVersion( 2, 1 );
10
11     glutInitDisplayMode( GLUT_SINGLE|GLUT_RGB );
12     glutInitWindowSize( SCREEN_WIDTH, SCREEN_HEIGHT );
13     glutCreateWindow( "OpenGL" );
14
15     cout<<"Enter window dimensions: "<<endl;
16     cout<<"Enter minimum X value: "; cin>>window_x_dims.first;
17     cout<<"Enter maximum X value: "; cin>>window_x_dims.second;
18     cout<<"Enter minimum Y value: "; cin>>window_y_dims.first;
19     cout<<"Enter maximum Y value: "; cin>>window_y_dims.second;
20
21     cout<<"Enter vertices: "<<endl;
22     cout<<"Vertex " <<1<<" (x y): ";
23     cin>>original_point1.first>>original_point1.second;

```

```

24
25     cout<<"Vertex "<<2<<" (x y): ";
26     cin>>original_point2.first>>original_point2.second;
27
28     drawOriginalLine();
29
30
31     if( !initGL() )
32     {
33         printf( "Unable to initialize graphics library!\n" );
34         return 1;
35     }
36
37     glutDisplayFunc( render );
38
39     glutTimerFunc( 1000 / SCREEN_FPS, runMainLoop, 0 );
40
41     glutMainLoop();
42
43     return 0;
44 }
45
46 void runMainLoop( int val ){
47     update();
48     render();
49
50     glutTimerFunc( 1000 / SCREEN_FPS, runMainLoop, val );
51 }

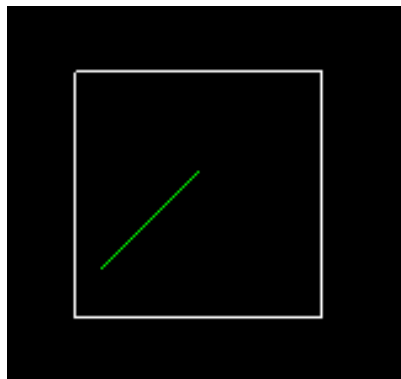
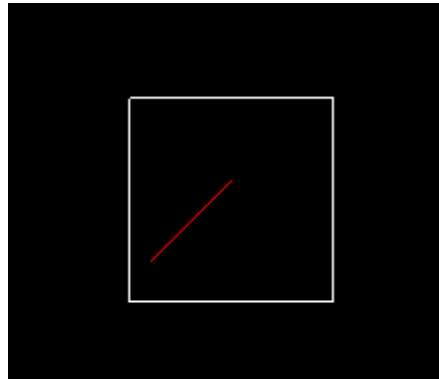
```

## Output:

### Trivial Accept:

Enter window dimensions:  
Enter minimum X value: 200  
Enter maximum X value: 300  
Enter minimum Y value: 200  
Enter maximum Y value: 300

Enter vertices:  
Vertex 1 (x y): 210 220  
Vertex 2 (x y): 250 260  
Trivially accepted  
New points are : (210, 220) and (250, 260)





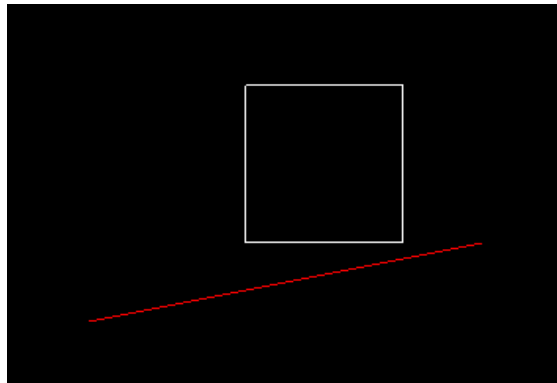
**Trivial Reject:**

Enter vertices:

Vertex 1 (x y): 100 150

Vertex 2 (x y): 350 200

Trivially rejected



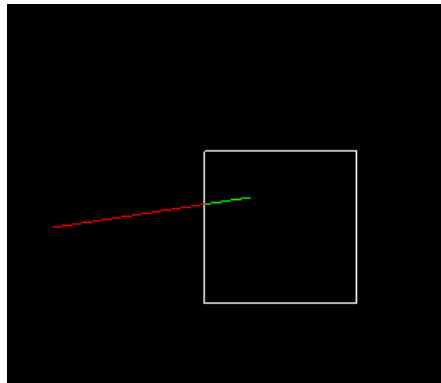
### One vertex outside:

Enter vertices:

Vertex 1 (x y): 100 250

Vertex 2 (x y): 230 270

New points are : (200, 265.385) and (230, 270)



### Both vertices outside:

Enter vertices:

Vertex 1 (x y): 170 320

Vertex 2 (x y): 320 190

New points are : (200, 265.385) and (230, 270)

