Date
Page

Assignment No - A-5

Student Name:- Sameer Manoj Bramhecha.
Roll No :- 21115
Batch:- F-1
Performance Date: 19-10-2021
Submission Date: 12-10-2021.

Title:→ Cohen-Sutherland Line Clipping

Problem Statement:-> Write a C++ program to implement Cohen Sutherland line clipping algorithm.

Learning Objectives:->
1) To understand the Cohen-Sutherland line clipping algorithm
2) To understand the use of mouse interfacing.

Learning Outcomes:-> After completion of this assignments, students will be able to implement Cohen-Sutherland line clipping algorithm.

S/W & H/W requirement:-
1.> 64-bit open source Linux.
2.> Open Source C++ programming tool like G++/GCC.
3.> Open GL.

Theory :→

Cohen Sutherland algorithm is one of the popular line clipping algorithm. To speed up the proceseing, this algorithm performs initial tests that reduce the number of intersections that must be calulated. This algorithm uses a four digit code to indicate which of nine regions contain the end point of line. The four bit codes are called region codes or out codes. These codes identify the location of the point relative to the boundaries of the clipping rectangle. Each bitt position in the region code is used to indicate one of the four relative coordinate positions of the point with respect~~ive~~ to the clipping window: to the left, right, top or bottom. The bits are set to 1 based on the following scheme.

set bit 1 : if the end point~~s~~ is to the left of the window.
set bit 2 : if the end point~~s~~ is to the right of the window.
set bit 3 : if the end point~~s~~ is below the window.
set bit 4 : if the end point~~s~~ is above the window.

1010 region 1	1000 region 2	1001 region 3
0010 region.	0000 Clipping Window	0001 region 4
0110 region 7	0100 region 6	0101 region 5

The equation for line passing through points p1(x_1, y_1) and P2 (x_2, y_2) is:

~~$y = m(x$~~ $y = m(x - x_1) + y_1$

OR

$y = m(x - x_2) + y_2$

where $m = \frac{y_2 - y_1}{x_2 - x_1}$ (slope of a line)

Therefore, the intersections with the clipping boundaries of the window are given as:

Left : xL, $y = m(xL - x_1) + y_1$
Right : xR, $y = m(xR - x_1) + y_1$

Top : yT, $x = x_1 + (1/m)(yT - y_1)$
Bottom yB, $x = x_1 + (1/m)(yB - y_1)$

Algorithm:→

1.) Read two endpoints of the line say p1(x_1, y_1) and P2 (x_2, y_2).

2) Read two corners (left-top & right bottom) of the window, say (wx_1, wy_1, wx_2, wy_2).

3.) Assign the region codes for two endpoints p1 & p2 using following steps:-

Set bit1 - if ($x < wx_1$)
Set bit2 - if ($x > wx_2$)
Set bit3 - if ($y < wy2$)
Set bit4 - if ($y > wy1$)

4.) Check for visibility of line p1p2

a.) if region codes for both end points p1 and p2 are zero then the line is completely visible. Hence draw the line and go to step. 9.

b.) if region codes for endpoints are not zero and the logical Anding of them is also non-zero then the line is completely invisible, so reject the line and go to step 9.

c.) If region codes for two endpoints do not satisfy the conditions in (4a) and (4b), then the line ~~is~~ is partially visible.

5.) Determine the intersecting edge of the clipping window by inspecting the region codes of two endpoints.

a.) if region codes for both the end points are non-zero, find intersection points p1' and p2' with boundary edges of clipping window with respect to point p1 and point p2, respectively.

b.) if region code for any one end point is non-zero then find intersection point p1' or p2' with the boundary edge of the clipping window with respect to it.

6.) Divide the line segments considering intersection points.

7.) Reject the line segment if any one end point of it appears outsides the clipping window.

8.) Draw the remaining line segments.

9.) ~~Stop~~. Stop.

Conclusion:- Thus line can be clipped against a window using Cohen Sutherland line clipping algorithm.