Expt No. B3

Assignment No 6.

Title - ~~Polygon~~ Line clipping.

Problem statement :- Write C++/Java program to implement line clipping algorithm foe given window. Draw a line using the mouse inteefacing to draw polygon.

Objective :- To leaen & implement the cohen Sutheeland line clipping algorithm

Software used :- Qt creatoe, cpp

Haredware used :- Linux based OS.

Outcome :- • We should able to implement the cohen Sutheeland line clipping algorithm.
• we should able to understand the line drawing & concept of line clipping.

Theoey :-

Digital diffrential analyzee (DDA) :-

In any 2-D plane if we connect two points (x_0, y_0) & (x_1, y_1) we get line segment. But in ca of computee graphics we can directly join two coordinate point foe that we need to calculate intermediate co-ordinate.

DDA is simple line generation algorithm

algorithm :-

Integee : integee function
Sign : returns 1, 0, -1 for argument.

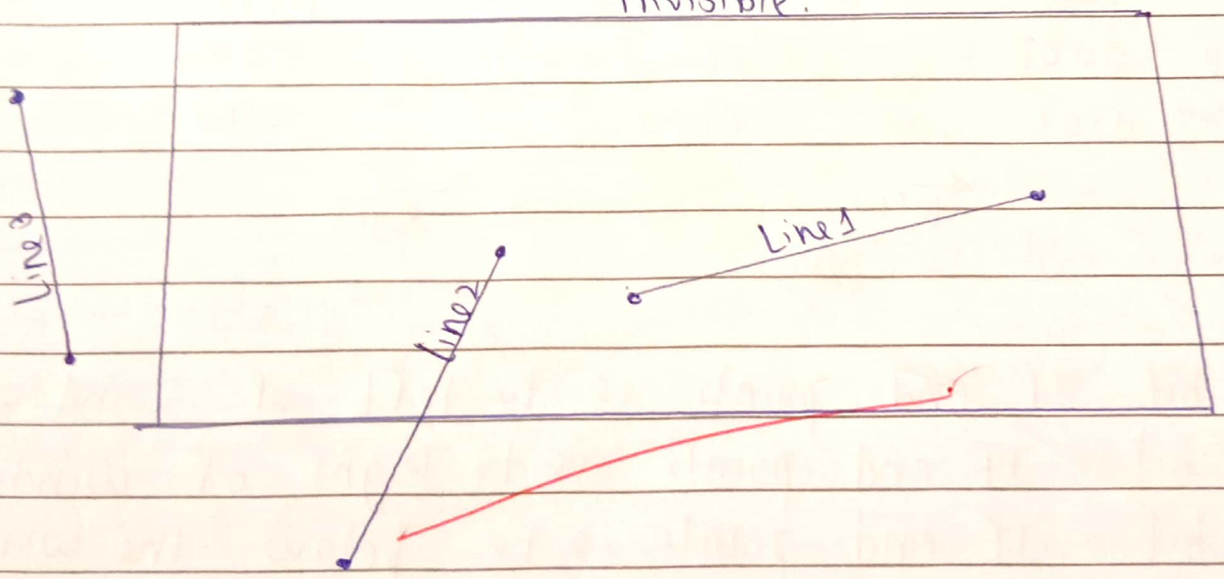
```
step 1: Read end points (x1, y1) (x2, y2)
step 2: approximate the length of line
        if (abs(x2-x1)) >= (abs(y2-y1))
            length = abs(x2-x1)
        else
            length = abs(y2-y1)

step 3: select raptee unit
        Δx = (x2-x1) / length
        Δy = (y2-y1) / length

step 4: Round the values
        x = x1 + 0.5 + sin(Δx)
        y = y1 + 0.5 + sin(Δy)

step 5: plot the pixel  i = 1
        while (i <= length)
        {
            setpixel (Integee(x), Integee(y))
                x = x + Δx ,  y = y + Δy
                i = i+1
        }
```

- Visibility of line :-

i) Completely visible :- When two endpoints of line are lies completely inside or at point of window intersection then line called as completely visible

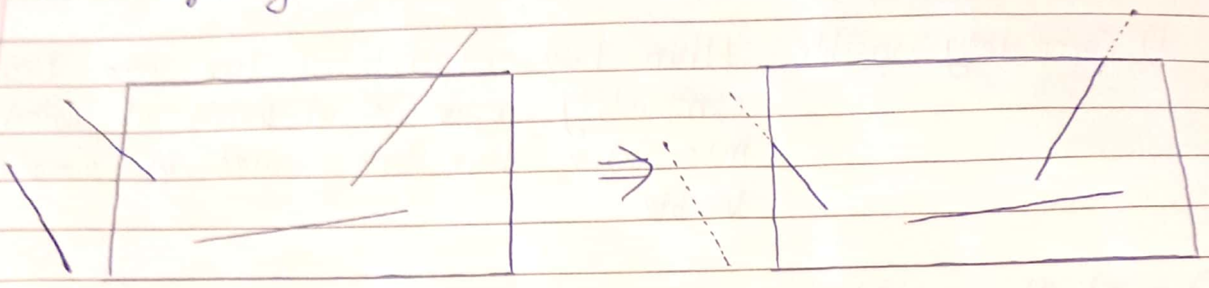
ii) partially visible :- when one of the endpoint of line lies inside the window & other lies in outside the window then line is partially visible.

iii) Completely invisible :- when both endpoints lies outside the window then line is completely invisible.



Line 1 is completely visible
Line 2 is partially visible
Line 3 is completely invisible.

• Visibility algorithms :-



Before applying visibility algorithm

after applying visibility algorithm.

• End point codes :-
Cohen & sutherland technique uses 4-bit (digit) code to indicate nine regions which contain end point

↑	1001	1000	1010
top	0001	window (0000)	0010
↓ bottom	0101	0100	0110
	← left		→ Right

Set
1st bit: If end points is to left of window.
2nd bit :- If end points is to Right of window.
3rd bit :- If end points is to below the window.
4th bit :- If end points is to above the window.

- Cohen-Sutherland algorithm end point code to accept or reject line segment.
- If the line segment is not trivially accepted or rejected then search for end points which is outside the window & segment from this point to the intersection point can always be rejected.

Cohen-Sutherland line clipping algorithm -

1.) Read two end points of the line P1, P2.
2.) Read left top & Right bottom of window.
3.) Assign the region code to P1 & P2
    initialize code with bits 0000
    Set Bit 1 - if $(x < WL)$
    set Bit 2 - if $(x > WR)$
    set Bit 3 - if $(y < WB)$
    Set Bit 4 - if $(y > WT)$
4.) check the visibility of P1 & P2.
5.) If region code for both end points P1 & P2 are
    then line is completely visible, Hence draw t
    line & stop.
6) If region code for end points are not zero &
    AND of them is also nonzero then the lin
    completely invisible so reject the line & stop.
7) If region code for endpoints don't satisfy
    one & second condn then line is partially

8) Determine the intersection edge of the clipping window by inspecting code of endpoints.
9) If region code for both end points are non-zero find intersection point P1 & P2 with boundary edge of clippin
window.
10) Divide the line segment at intersection points appears outside the clipping window.
11) Reject the line segment of any one end point appears outside the clipping window.
12) Draw the remaining line segment.
13) Stop.

Conclusion:-

We have learn & implement the Cohen-Sutherla
line-clipping algorithm

C

[illegible]