A microproject report on

' Implementing program to clip line using cohen-sutherland algorithm '

Submitted for

DIPOMA IN COMPUTER ENGINEERING

MSBTE , Mumbai

Abhaysinhraje Bhonsale Institute of Technology shahunagar - slendre - satara

(2020 - 21)

Vidyavardhini charitable Trust's
Abhaysinhraje Bhonsle Institute
of Technology (Polytechnic)
Shahunagar - shendre satar 415519

CERTIFICATE

This is certified that micro-project report submitted by following members of (103-I)

SANIKA	PATIL	15
AKSHAT	MARUDA	13
PRANAY	NEJKAR	14

GUIDE	HOD	PRINCIPAL
Mr. SARATE S R	Mrs. Nikam R A	Mr. DHUMAL S.U

TABLE OF CONTENTS.

ACTION PLAN

Sr.no	Details of activity	Start Date	Finish Date	Team member
1.	Selected the topic	15-12-20	20-12-20	Akshat
2.	we organised things required	20-12-20	26-12-20	Pranay
3.	we browsed information on internet and raw data	26-12-20	5-01-21	Sanika
4.	we attented extra lectures	5-01-21	15-01-21	Akshat
5.	we made points /notes	15-01-21	20-01-21	Pranay
6.	we created a word document	20-01-21	25-01-21	Sanika
7.	we made corrections by discussing with teachers	25-01-21	30-01-21	Akshat
8.	we also created a PDF document to make a hard copy	30-01-21	04-02-21	Pranay

INTRODUCTION

Computer graphics deals with generating images with the aid of computers. Today, computer graphics is a core technology in digital photography, film, video games, cell phone and computer displays and many specialised applications. computer graphics is used where a set of images needs to be manipulated or the creation of the image in the form of pixels and is drawn on the computer.

Cohen-Sutherland line clipping algorithm is characterized by its ease and speed in obtaining results and it is one of the most famous and simplest clipping algorithm.

PROJECT OBJECTIVE

The cohen-sutherland algorithm is a computer graphics algorithm used for line clipping The algorithm divides a two-dimensional space into 9 regions and then efficiently determines the lines and portions of lines that are visible in the central region of of interest (the viewport). The algorithm quickly detects and dispenses with two common and trivial cases. To clip a line we need to consider only its endpoints. If both endpoints of a line lie inside the window. the entire line lies inside the window. It is trivially accepted and needs no clipping. On the other hand, if both endpoints of a line lie entirely to one side of the window the line must lie entirely outside the window.

ALGORITHM.

Step 1. Start

Step 2. given line segment with endpoint $P_1(x_1, y_1)$, $P_2 = (x_2, y_2)$

Step 3. computer 4 bit code for each endpoint.

Step 4. If both codes have a line the same bit code position the line lies outside window. It can trivally rejected.

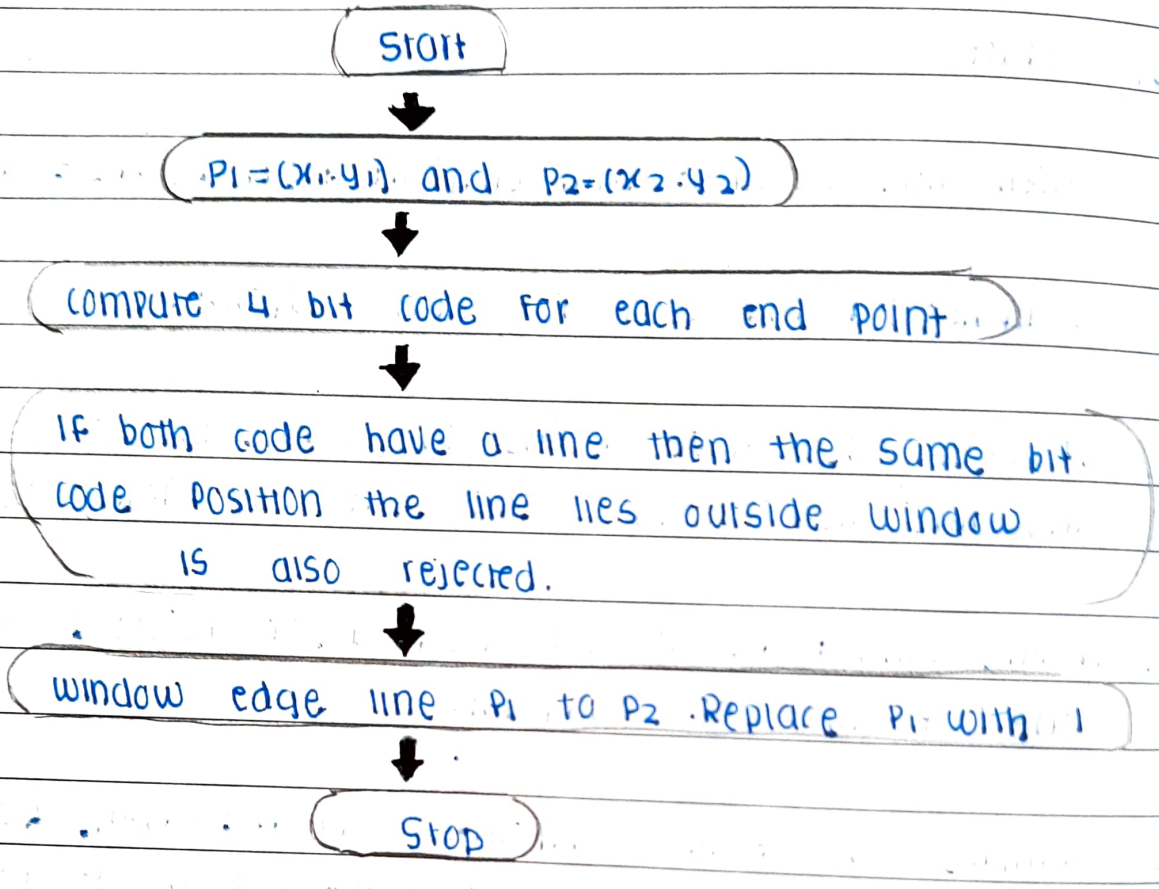
Step 5. If line cannot be trivally accepted or rejected at least one of the two endpoints

Step 6. Examine one of the endpoints say $P_1 = (x_1, y_1)$ Read 4 bit code in order left to right. Bottom to top.

Step 7. When set bit (1) is found compute to intersection of corresponding window edges with line form P_1 to P_2

Step 8. END

FLOW CHART.



SOFTWARE RESOURCES.

Sr.no	Name of resources	specification
1	Computer system with broad specification	computer (13-15) RAM minimum 2GiB
2	Software	Turbo c++

"C" Program code.

```
# include <stdio.h>
# include <stdlib.h>
# include <math.h>
# include <graphics.h>
# include <dos.h>
typedef struct coordinate
{
int x, y;
char code [4];
} PT;
void drawwindow ();
void drawline (PT P1 , PT P2 );
PT setcode (PT p);
int visibility (PT P1 , PT P2);
PT resetendpt (PT P1 , PT P2);
void main ()
{
int gd = DETECT , v, gm;
PT P1 , P2 , P3 , P4 , ptemp ,
printf (" \n Enter x1 and y1 \n");
scanf (" %d %d " , & P1 . x & P1. y );
printf (" \n Enter x2 and y2 \n");
scanf (" %d %d " , & P2 . x & P2 . y);
initgraph (& gd , & gm , "c: \\turboc3 \\bgi");
drawwindow ();
delay (500);
drawline (P1 , P2 ).
```

```
delay (500);
cleardevice ();
delay (500);
P1 = setcode (P1);
P2 = setcode (P2);
V = visibility (P1 , P2);
delay (500);
switch (v)
{
case 0: drawwindow ();
        delay (500);
        drawline (P1 , P2);
        break;
case 1 : drawwindow ();
        delay (500);
        break ;
case 2 : B3 = resetendpt (P2 . P1);
        P4 = resetendpt (P2 . P1);
        drawwindow ();
        delay (500);
        drawline (P3 . P4);
        break ;
}
delay (500);
closegraph ();
}
void drawwindow ()
{
line ( 150 , 100 , 450 , 100);
line ( 450 , 100 , 450 , 350);
line ( 450 , 350 , 150 , 350);
line ( 150 , 350 , 150 , 100);
```

```
}
void drawline (PT p1, PT p2)
{
line (p1.x, p1.y, p2.x, p2.y);
}
PT setcode (PT p)
{
 PT ptemp;
if (p.y<100)
 ptemp.code [0] = '1';
else
ptemp.code [0] = '0';
if (p.y > 350)
ptemp.code [1] = '0';
else
ptemp.code [1] = '0';
if (p.x > 450)
ptemp.code [2] = '1';
else
ptemp.code [2] = '0';
if (p.x<150)
 ptemp.code [3] = '1';
 else
 ptemp.code [3] = '0';
 ptemp.x = p.x;
 ptemp.y = p.y;
 return (ptemp);
 }
int visibility (PT p1, PT p2)
{
int i, flag = 0;
for(i=0; i<4; i++)
```

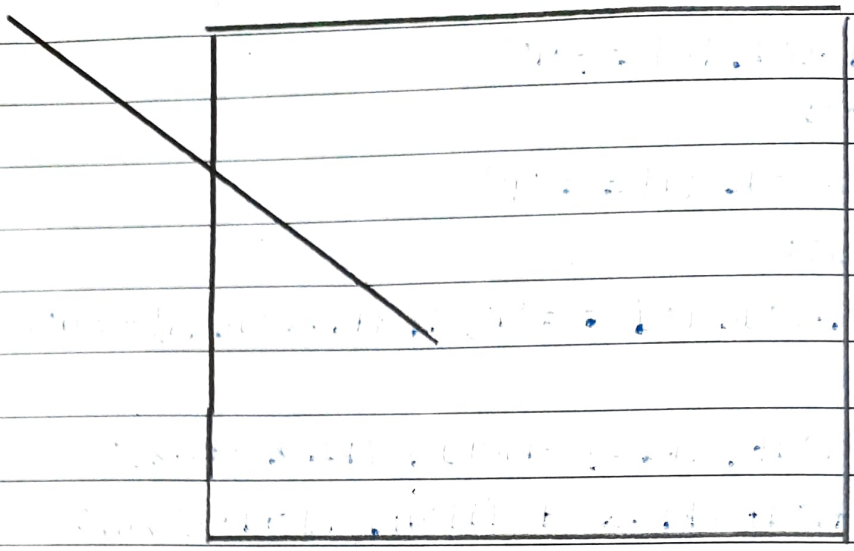
```
{
if ((P1.code [i] != '0') || (P2.code [i] != '0'))
flag = 1;
}
if (flag == 0)
return (0);
for (i=0 ; i<4 ; i++)
{
if ((P1.code [i] == P2.code [i]) && (P1.code [i] == '1'))
flag = '0';
}
if (flag == 0)
return (1);
return (2);
}
PT resetendpt (PT P1, PT P2)
{
pt temp;
int x, y, i;
float m, k;
if (P1.code [3] == '1')
x = 150;
if (P1.code [2] == '1')
x = 450;
if ((P1.code [3] == '1') || (P1.code [2] == '1'))
{
m = (float) (P2.y - P1.y) / (P2.x - P1.x);
k = (P1.y + (m * (x - P1.x)));
temp = k;
temp.x = x;
for (i=0 ; i<4 ; i++)
temp.code [i] = P1.code [i];
```

```
if (temp.y<=350 && temp.y>=100)
return (temp);
}
if (P1.code[0]=='1')
 y=100;
if (P1.code[1]=='1')
 y=350;
if ((P1.code[0]=='1') || (P1.code[1]=='1'))
{
m=(float)(P2.y-P1.y)/(P2.x-P1.x);
k=(float)P1.x+(float)(y-P1.y)/m;
temp.x=k;
temp.y=y;
for(i=0;i<4;i++)
 temp.code[i]=P1.code[i];
 return (temp);
}
else
return (P1);
}
```

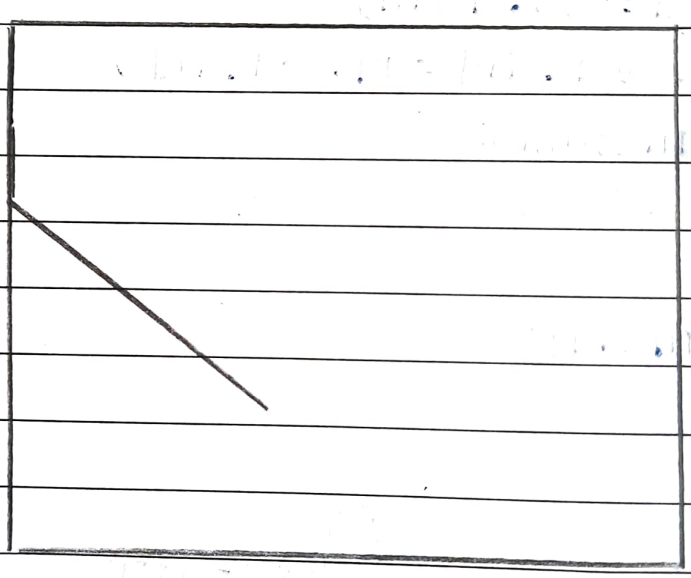
OUTPUT

```
Enter x1 and y1
100
100
Enter x2 and y2
200
200
```

Before clipping :



After clipping



IMPLEMENTATION.

Cohen Sutherland algorithm is considered the primary clipping line algorithm. characterized by easy and simple to apply.

1001	0001	0101
1000	0000	0100
1010	0010	0110

CONCLUSION.

Here we conclude that by using algorithm steps with the help of C program code we can clip a line using cohen-sutherland algorithm.

REFERENES.

Website - www.geeksforgeeks.org
www.thecrazyprogrammer.com

Books - Transformations and projections in computer graphics.
Sketch Based search and composition of 3D models