**COMPUTER GRAPHICS AND ANIMATION**

**Programming Assignment 5**

**Cyrus-Beck Clipping**

**Report**

Fadel Cahyo M (001201700040)

Faisal Abdul Rafi (001201700035)

Septian Calvin Infa Waruwu (001201700055)

**CHAPTER 1**

**INTRODUCTION**

The program that developed in this project allows the user to draw dots or lines on the screen and then can clip what lines or dots in the screen with convex polygon clipping window. Every lines and dots that clipped by the convex polygon window will be highlighted using Cyrus-Beck method. This program made by using JavaScript programming language.

**CHAPTER 2**

**BASIC THEORY**

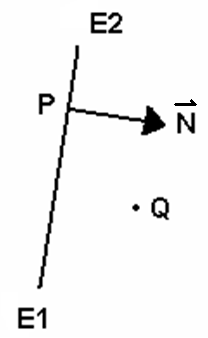
**CLIPPING**

In computer graphics the screen act as 2D coordinate system. User view points, which lie in particular range (0, ) and ( , ). So, clipping is a procedure that identifies those portions of a picture that are either inside or outside of user viewing pane.

**POINT CLIPPING**

Point clipping only show points on user window which are in range of user viewing pane, the others point which are outside the range are discarded.

The statement below will explain how to determine whether a point is inside or outside an edge

 E1E2 = window edge

P = a point on the line

= normal vector of E1E2 pointing inside the window

• = dot product

Q = a point

In this case, user want to know the point Q is inside or outside of the edge. We have to find the normal vector of E1E2. We use this equation :

F(Q) = PQ **.** N

F(Q) = (Q – P) **.** N

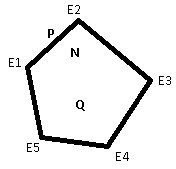
After we know F(Q) and F(Q) determine the angle between PQ and N is greater or less then 90

If F(Q) > 0 Q is “ inside ” the line.

If F(Q) < 0 Q is “outside” the line.

If F(Q) = 0 Q is on the line.

How do we determine whether a points is inside or outside a polygon.

 E1,E2,E3,E4,E5 = window edge/polygon edge

P = a point on the line

= normal vector of E1E2 pointing inside the window

• = dot product

Q = a point

User want to know whether point Q is inside or outside the convex polygon window, to determine the polygon is convex or not we will be explained later on this report, first determine the normal vector, the normal vector always pointing inside the polygon.

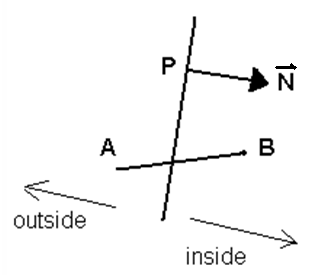
User must define the normal vector and F(Q) for all lines in polygon by using above formula in every line. The dot product of the normal vector and F(Q) must be greater than 0, so Q point is inside the polygon. If one of the dot product(F(Q)) value < 0, it means Q point is outside the polygon.

**LINE CLIPPING**

In computer graphics, line clipping is the process of removing lines or portions of lines outside an area of interest. Typically, any line or part thereof which is outside of the viewing area is removed

The **Cyrus–Beck algorithm** is a generalized line-clipping algorithm. It was designed to be more efficient than the Sutherland–Cohen algorithm, which uses repetitive clipping. Cyrus–Beck is a general algorithm and can be used with a convex polygon clipping window, unlike Sutherland–Cohen, which can be used only on a rectangular clipping area.

User can determine whether a line is accepted, rejected, or trivially accepted by checking both points of the line.



Window edge:

Line segment :

Trivial reject :

and for at least one edge of the clipping region.

Trivial accept :

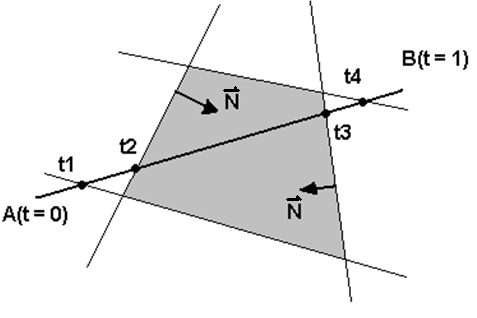
and for all edges of the clipping region.

From the equation above, we can find the point where the line and the window intersected.

Before we use the equation above, first of all we must to find the value of t with this formula :

=

Example in this case

First we have to find N

After this we check the value of D. which is :

If D is already know we check

D > 0 = D < 0 =

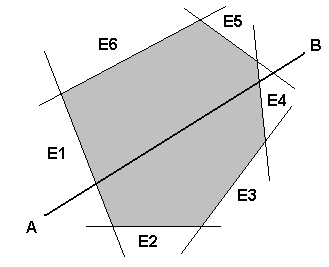
**POLYGON CONVEXITY**

A convex polygon is a [simple polygon](https://en.wikipedia.org/wiki/Simple_polygon) (not [self-intersecting](https://en.wikipedia.org/wiki/Self-intersecting_polygon)) in which no line segment between two points on the boundary ever goes outside the polygon. Equivalently, it is a simple polygon whose [interior](https://en.wikipedia.org/wiki/Interior_(topology)) is a [convex set](https://en.wikipedia.org/wiki/Convex_set). In a convex polygon, all interior angles are less than or equal to 180 degrees, while in a strictly convex polygon all interior angles are strictly less than 180 degrees. Non-convex is also referred to as concave.

Convex and non-convex both define the types of curvature. Convex defines the curvature that extends outwards or bulges out. non-convex defines a curvature that extends or bends inward.

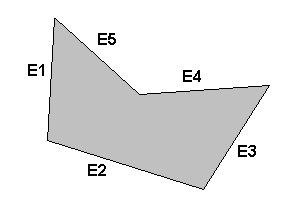
To determine whether a polygon is convex or not is by the sign of all cross product of adjacent edges is the same.

Picture below shown a convex polygon.



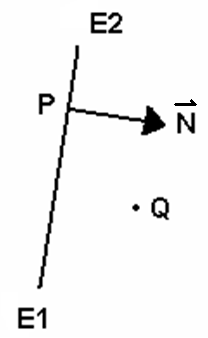
This polygon is convex

Picture below shown a concave or not convex polygon.



**EDGE NORMALS**

Edge normal is determine by normal vector of each edge, and edge normal always pointing inside the window.

Window edge = E1E2

N = normal vector pointing inside window

In this case N is determine by normal vector of E1E2 pointing inside the window and E1E2 is clockwise. So, the normal vector in this case is

(-dy dx)T.

To determine whether a polygon is clockwise or anti-clockwise is by using the cross product of each edge. In convex polygon above is the way how user determine whether a polygon is clockwise or not. If the cross product of the edges is all positive it means that the polygon is clockwise and vice versa(the polygon is anti-clockwise).

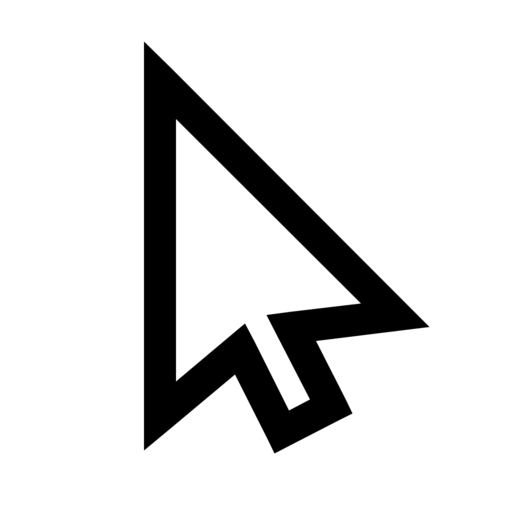
**CHAPTER 3**

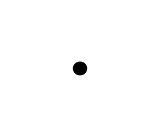
**IMPLEMEMTATION**

Main interface

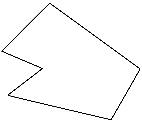


Feature :

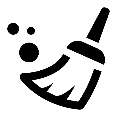
 = make a mouse a usual pointer

 = to draw dots

 = to draw lines

 = to draw polygon window

 = to delete a selected dot or line

 = to clear the canvas(redraw)

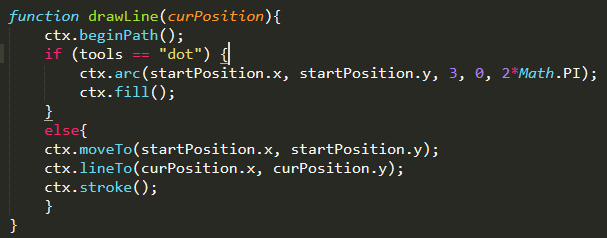
 = to clear only the polygon

 = to save as picture

**CHAPTER 4**

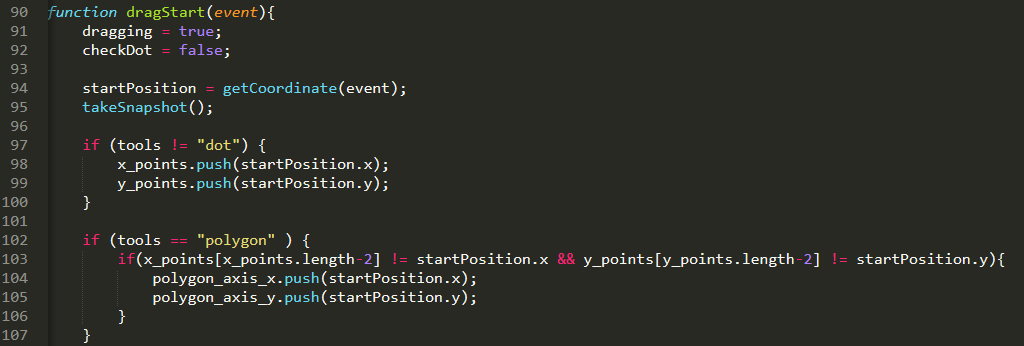
**DESIGN**

Function to draw line and dot.



To draw polygon clipping window :

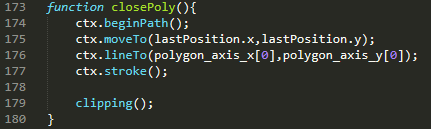
1. in dragStart function, push the coordinate for starting polygon coordinate to array.



1. in dragStop function, push the next polygon coordinate to array



1. If user double click on the screen, the polygon will be closed, so the last coordinate will automatically go to the first polygon coordinate.



**CHAPTER 5**

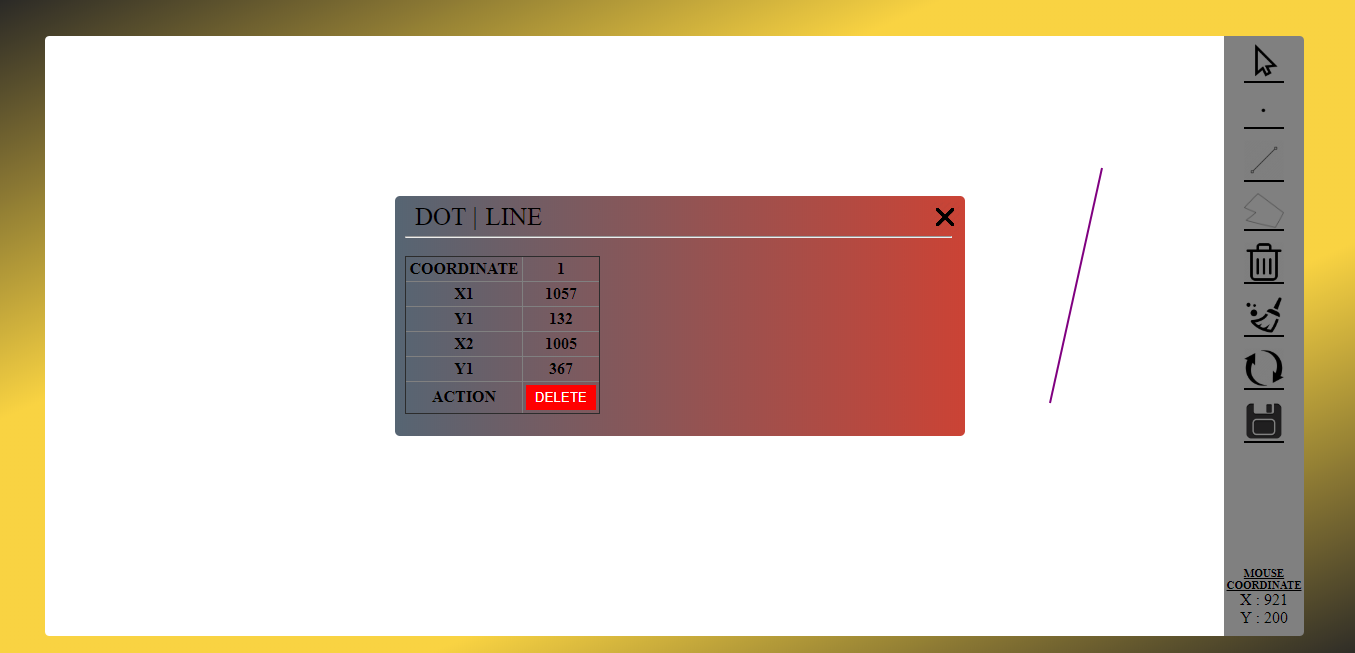
**EVALUATION**

Test case :

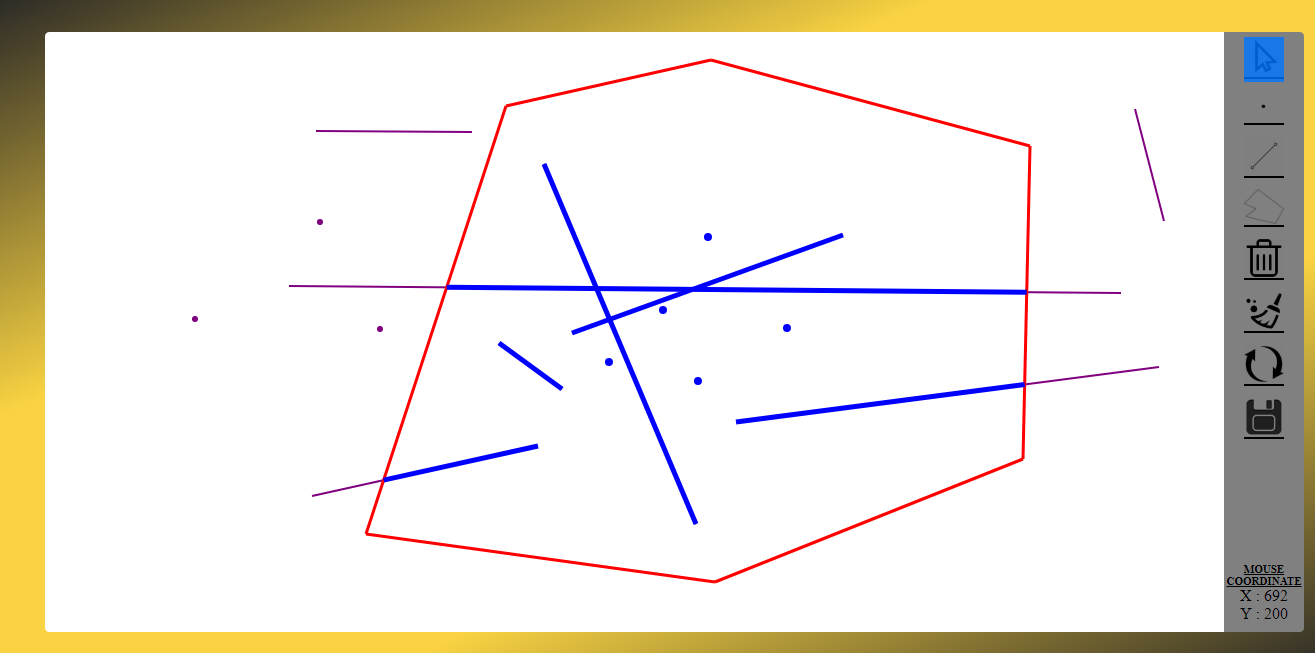
Adding a point/line on the screen



Deleting point/line from screen

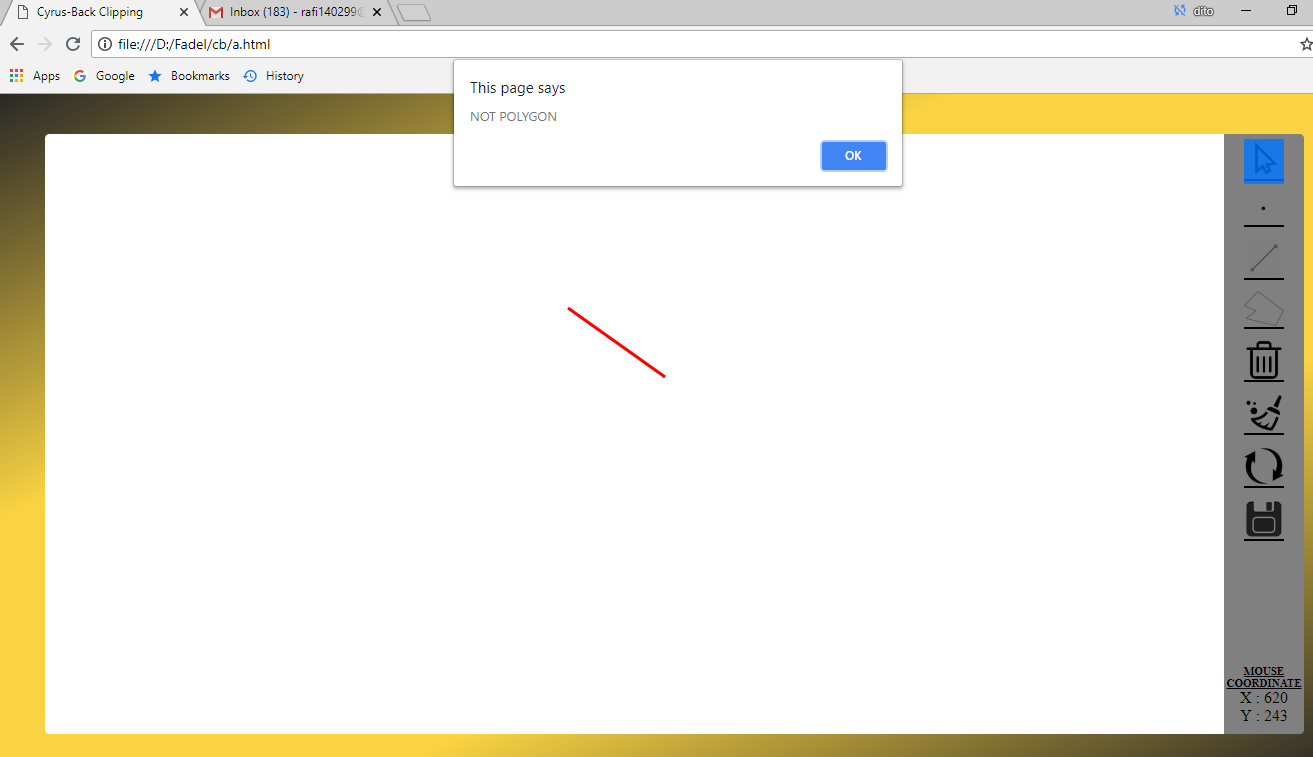


Clipping

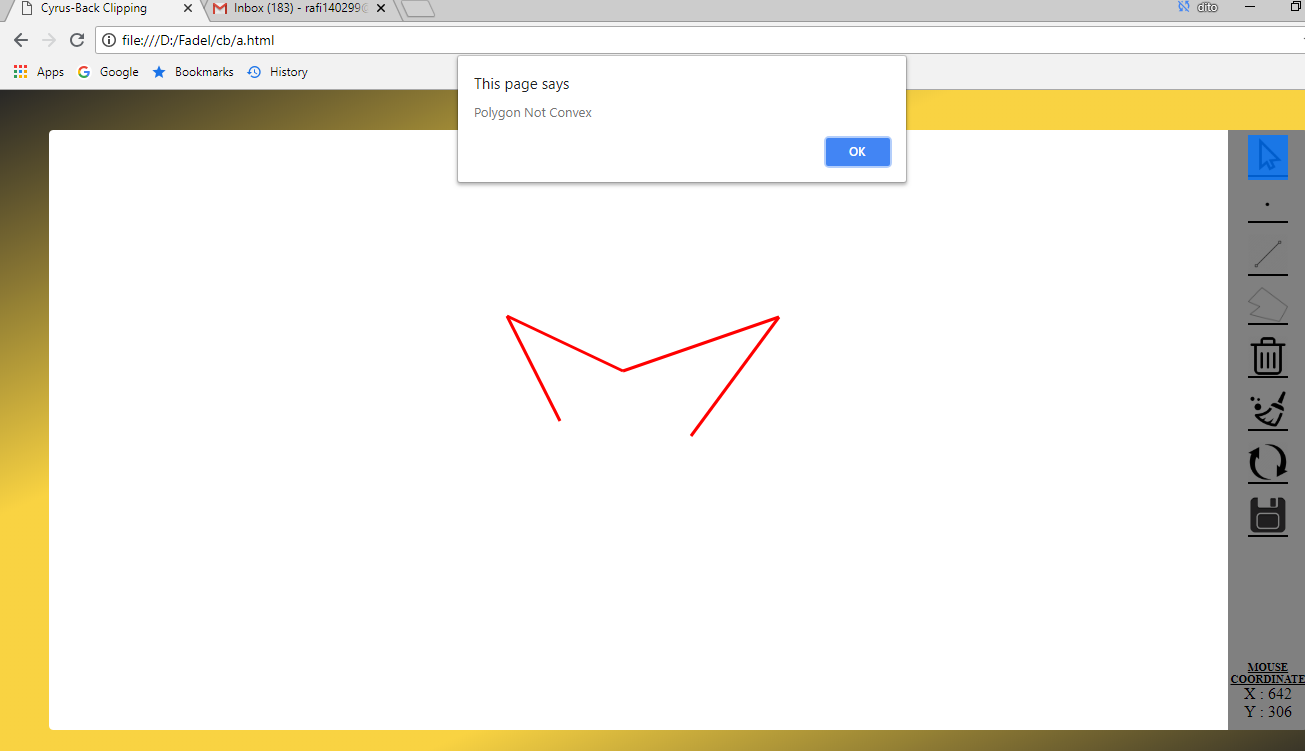


Validating Clipping window

Window with less than 3 vertices



Non-Convex clipping window



**CHAPTER 6**

**WORKLOG**

**Summary**

On the first week, we can draw a dot/line on the screen and save data. Also, on the first week, we can draw a polygon clipping on the screen, but we haven’t make it clip successfully yet because there’s a problem when we try to draw the polygon. The same coordinate of polygon can be inserted more than one into the array and it makes the array of polygon becomes chaotic, in other word, the array isn’t arranged sequentially. So we try to fix this problem to avoid inserting the same coordinate by adding additional codes. After all the problems in the first week are fixed, finally on the second week we can make a clipping successfully. On the second week, we complete all the requirements included how to delete dots and lines, removing the clipping and clear the screen. Overall, we fulfill all the requirements and the program runs successfully without error.

|  |  |  |
| --- | --- | --- |
| Date | Record Job Description | In-Charge |
| 7 February 2018 | Programming Assignment given by the lecturer | Fadel |
| 8 February 2018 | Discussed about what kind programming language will be used in this programming assignment | Fadel, Septian, Faisal |
| 9 February 2018 | Decided use JavaScript as the programming language in this programming assignment | Fadel, Septian, Faisal |
| 10 February 2018 | Make the user interface design | Septian, Faisal |
| 11 February 2018 | Draw dots and lines on the screen | Fadel, Faisal |
| 12 February 2018 | Draw polygon | Septian |
| 12 February 2018 | Remove multiple coordinate | Septian |
| 13 February 2018 | Check the polygon convex or not convex | Septian, Fadel |
| 14 February 2018 | Place clipping window on the screen just 1 line | Septian |
| 14 February 2018 | Make progress report(introduction, basic theory) | Faisal |
| 16 February 2018 | Clipping the dots and lines using cyrus-beck method | Septian |
| 20 February 2018 | Design the new user interface | Fadel, Septian |
| 24 February 2018 | Delete dots and lines, Clear screen | Septian, Fadel |
| 26 February 2018 | Remove polygon clipping window | Fadel, Septian |
| 1 March 2018 | Revised Report | Faisal, Fadel |

**CHAPTER 7**

**CONCLUSION and REMARKS**

The program is already completed and works fine. The program can draws some dots and lines, and also can clipping dots and lines. This programming assignment use cyrus-beck method in clipping polygon is already successfully.

This programming assignment we learn about the materials more depth especially about the cyrus-beck method. This programming assignmet also teach us about implemented the algorithm and how implement the cyrus beck method in a program. We have to know more details to make the report and how to explained the materials.