President University

Report of Programming Assignment 12

Weiler-Atherton Algorithm for Polygon Clipping

Computer Graphics and Animations

Hilman Revanda (001201500038)

Nikita Chrissandha M (001201500031)

Predrika Br Ginting (001201500032)

CIT 2 2015

# Introduction

The program is about drawing and clipping polygon/polygons. The Weiler – Atherton Algorithm for polygon clipping is fully implemented in this program. Other than those, user can also refresh the screen, delete, and save the polygon/polygons.

This program was created using Visual Basic programming language. The report includes basic theory, how to use the application, design of the application, evaluation of the main features, work log, and conclusion and remarks.

# Basic Theory

## Explain what clipping is about.

## Explain how to clip a point using the Cyrus – Beck algorithm.

### Explain how to determine whether a point is inside or outside an edge.

### Explain how to determine whether a line crosses an edge.

### Explain how to find the intersection between a line and an edge.

## Explain what polygon clipping is.

## Explain how to clip a polygon using the Weiler – Atherton algorithm.

### Explain the overall steps.

### Explain the four cases in processing each edge of the polygon.

### Write down the pseudocode (**not** code) for the Weiler – Atherton clipping algorithm. Include explanations; explain in detail.

## Explain the main limitations of the Sutherland – Hodgman algorithm

# Implementation

## Main interface



The application is named Hil-Nik-Cha Drawing App which was named after the developers. With color palette of silver, brown, dark turquoise, black and white, the user interface looks flat and modern. The features of the app include draw single and multiple polygon, Clipping the polygon with rectangle and polygon, delete polygon, refresh the screen, save polygon, and showing the list of polygon and clipping.

## Canvas/Picture Image

The picture box is used as canvas to draw polygons. After selecting the draw button, the user could click anywhere on the canvas to create polygon. Right click is used to finish the polygon.

## List Box

List box is where the points of all polygons and clipping showed.

## Draw

Single Polygon and Multiple Polygon are both in the Draw group box. Single Polygon is used to draw a single polygon. When the user use this button, the previous polygon will be deleted.

## Clipping

In the Clipping group box, there are Rectangular and Polygon Clipping. The rectangular clipping is used to clip with a rectangle. On the other hand, Polygon button is used to clip with a free and multiple angles(polygon).

## Properties

In the properties, there are Delete, Refresh, and Save button. Delete button is used for deleting line or polygon. Refresh button is used for clearing the screen and the list box. And Save button is used for saving the canvas into bitmap.

## Exit

Exit button is used to exit the app.

# Design

## Explain the main data structures (if any) used in the program.

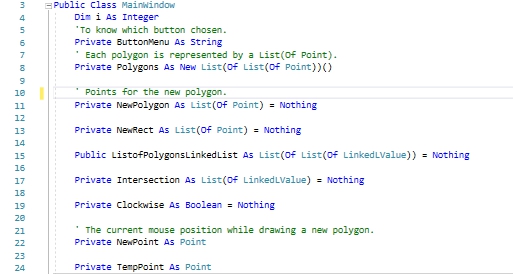
The point is represented as list and the polygon is represented as list of lists of point.

### How are the points and polygons represented in the program?

### How is the clipping window represented in the program?

### How are the CLP and CLW represented in the program?

## Explain the main/global variables used in the program.

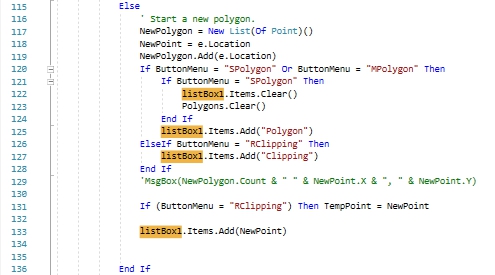


The variables:

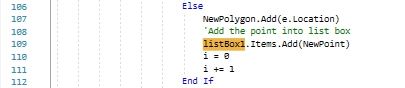
* i is index and declared as integer.
* ButtonMenu is declared as string. It is declared to know which button is chosen.
* Polygons to store the list of lists of point.
* NewPolygon to store the points of the new polygon.
* NewRect is to store the points of the rectangular clipping.
* ListofPolygonLinkedList to store the polygon in linked list.
* Intersection is to store the intersection of the polygon and the clipping in the linked list.
* Clockwise is declared as Boolean to determine whether the clipping is clockwise or not.
* New point is the current position of the mouse while drawing a new polygon.
* TempPoint is to store the temporary point.

## Implementation of Bonus

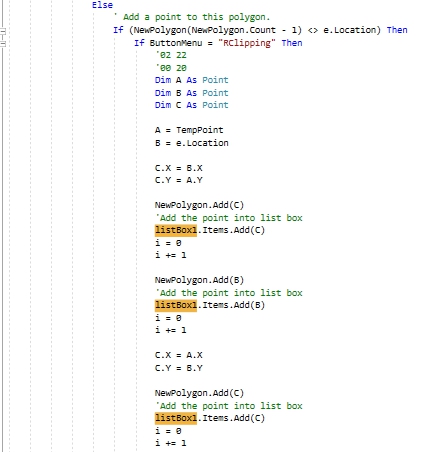
### List Box of Polygon and Clipping



When starting a new polygon, the program will check whether it’s polygon or clipping. If it is polygon, then “Polygon” will showed in list box, then the next line will be followed by the first point of the polygon. If the polygon chosen is Single Polygon, then the previous list box will be deleted.

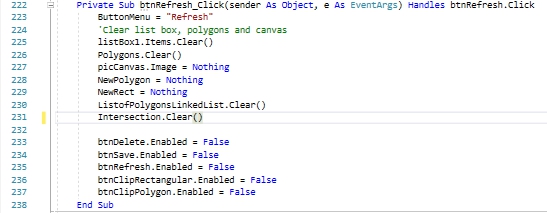


And then, the other points will be added after click from the user.



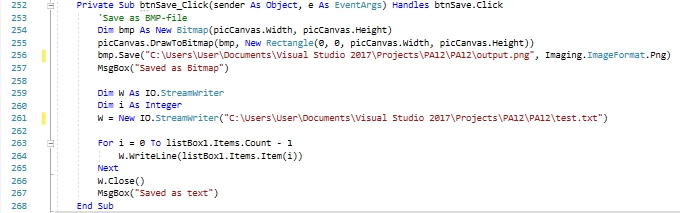
For rectangular clipping, the next point of polygon will be calculated first because the user only needs to click on two points and then the clipping will become rectangular.

### Refresh the Screen



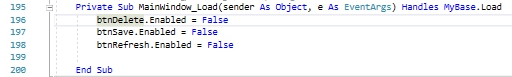
When the button is clicked, polygons, the list box, list of polygons linked list, intersection will be emptied. New polygon, new rectangular clipping, and the canvas will be set to be nothing.

### Save Polygon into Bitmap and Text File



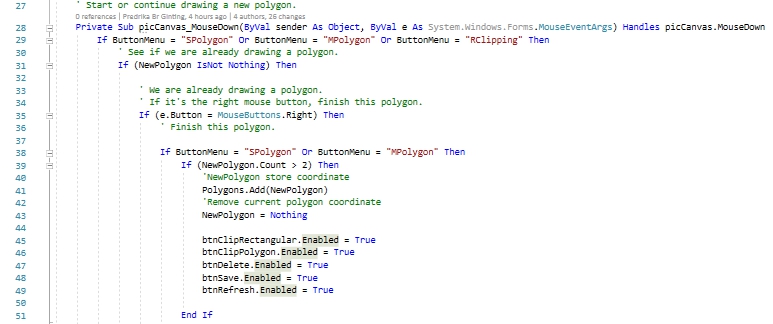
The polygon drawn on the canvas will saved as text and bitmap when the user click the Save button.

### Enable and Disable Button





When there’s still nothing on the canvas, the Delete, Save, and Refresh button will be disabled.

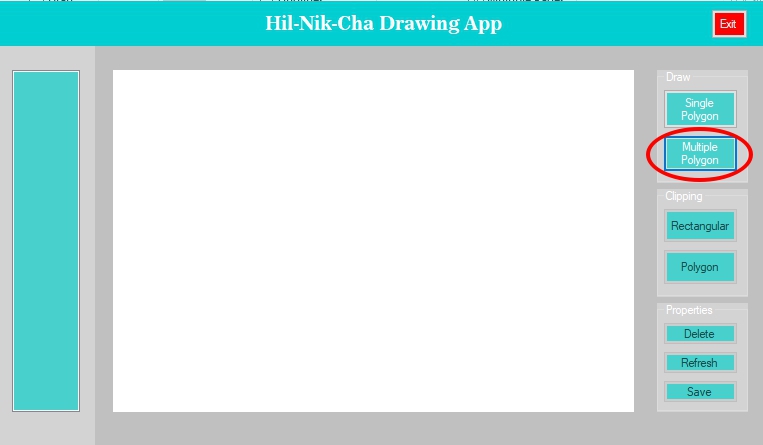


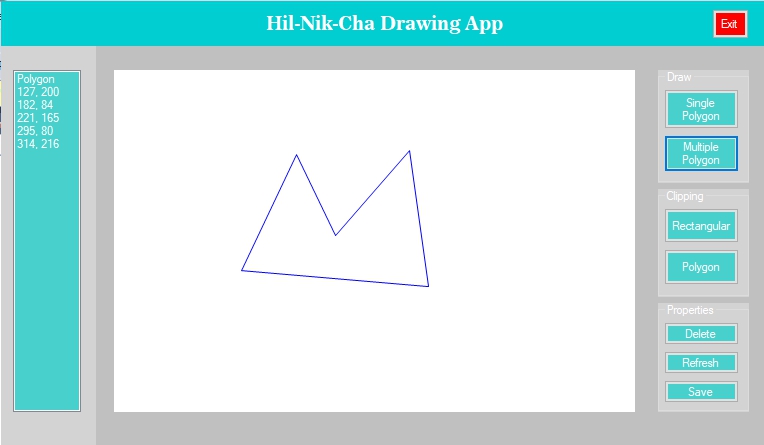
After a polygon is drawn, the Clipping and all the Properties Buttons will be enabled.

### Delete the Line

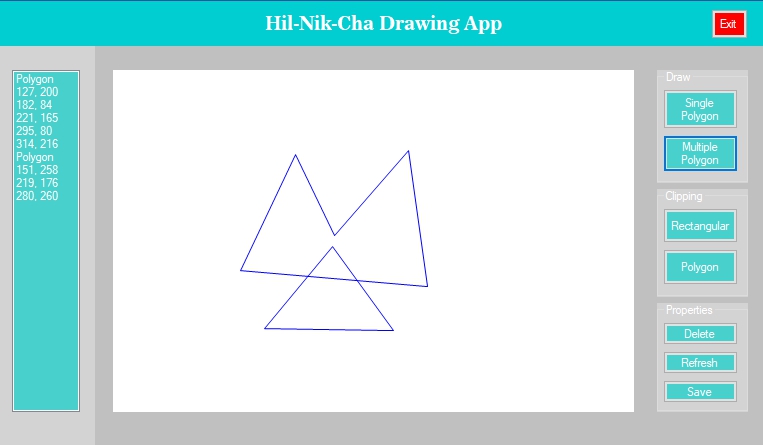
# Evaluation

## Adding a Polygon into the Screen

First, select button from Draw Group Box, which one the user wants to draw. click The Single Polygon button for one polygon only or Multiple Polygon button for multiple polygon.

Then click in the picture box to draw the polygon. Right click to finish the polygon.

## Adding Another Polygon into the Screen

When the user wants to add another polygon, the user just needs to clicks on the picture box to determine the points of the polygon and then right click to finish the polygon. Do the same if the user wants to add more.

## Clipping

#### A polygon partially inside the clipping window resulting in no degenerate lines.

#### A polygon partially inside the clipping window resulting in degenerate lines.

#### A polygon entirely inside the clipping window.

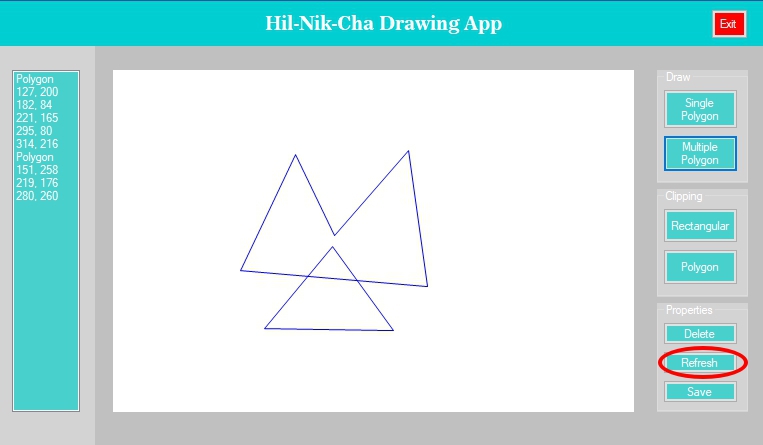
#### A polygon entirely outside the clipping window.

#### A polygon entirely covering the clipping window.

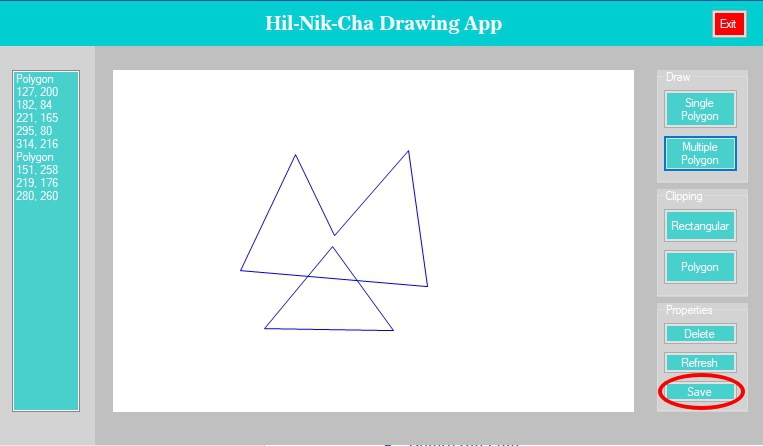
## List Box of Polygon and Clipping

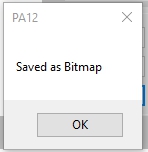
The list box will show the list of point of polygons and clipping.

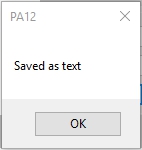
## Refresh the Screen

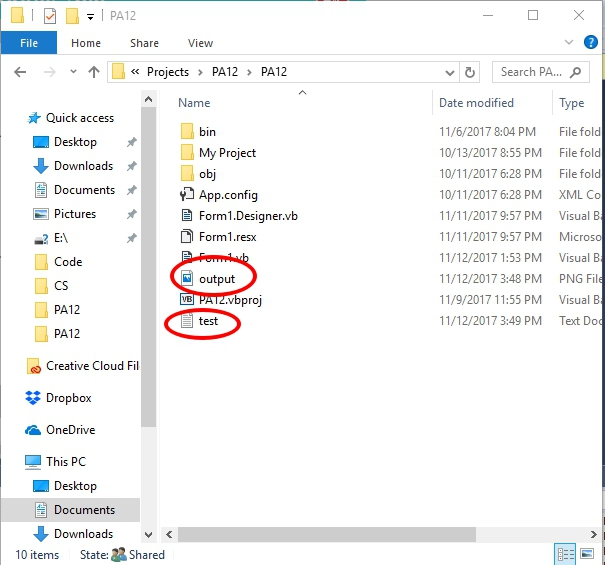
When the user wants to clear the screen, he/she only needs to click the Refresh button.

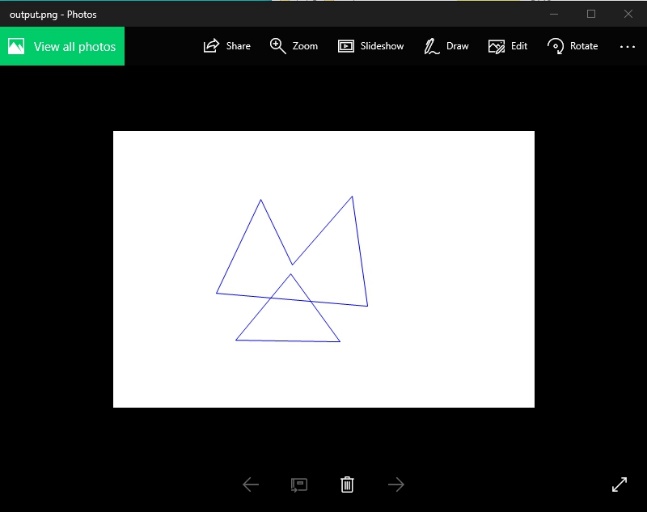
## Save Polygon into Bitmap and Text File

If the user wants to save the polygons which he/she has drawn, he/she only needs to click the Save button, then the app will save it into bitmap and text file.

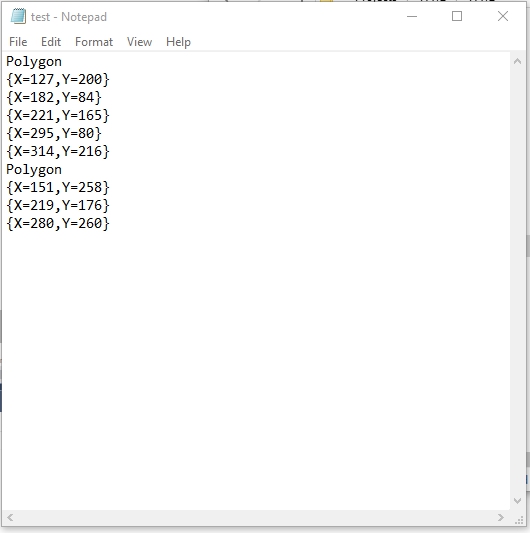
This means, the drawing is already saved as bitmap file.

And this means, the drawing is already saved as text.

If the user checks the folder, the bitmap and text is already there.

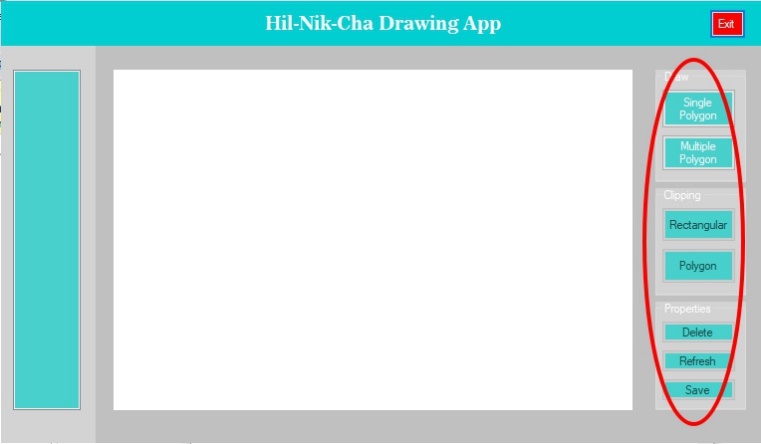


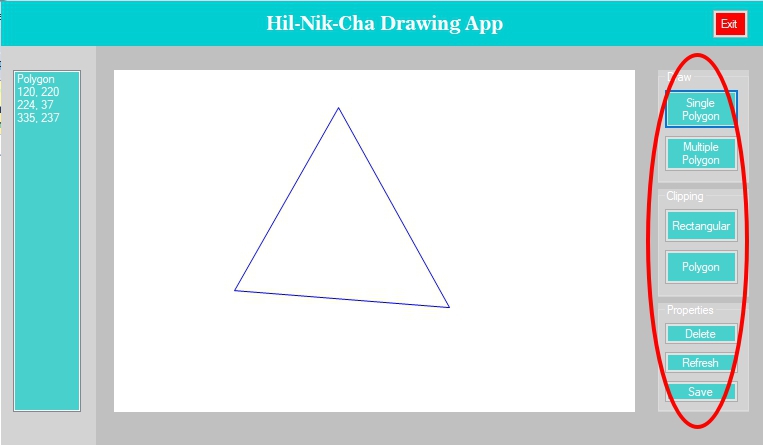
This is the output of bitmap file.

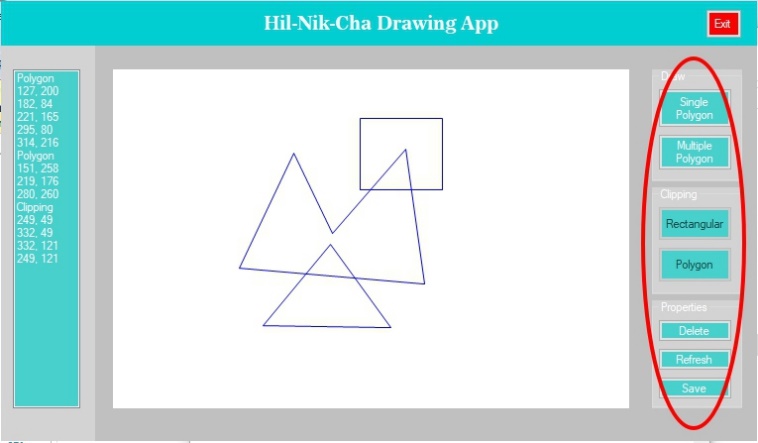


And this is the output of text file.

## Enable and Disable Button

When there’s no polygon drawn yet, the Clipping and Properties buttons will be disabled to prevent user clip or delete something that’s not exist yet.

After the user drawn something on the picture box, All the buttons in Draw, Clipping, and Properties will be enabled.

After there’s a clipping, the Clipping buttons will be disabled.

## Delete the Line

# Work Log

The work log is extracted directly from Visual Studio’s Git Log History, which is also available publicly at <https://github.com/ichapredrika/PA12/commits/master>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Author | Date | Time | Commit Message |
| 1 | Nikita | 10/11/2017 | 20:04:59 | Create Basic Theory in Report |
| 2 | Predrika | 10/13/2017 | 19:16:12 | User interface |
| 3 | Predrika | 10/15/2017 | 18:35:37 | Draw multiple polygon |
| 4 | Hilman | 10/15/2017 | 22:13:00 | Add ButtonMenu variable for SPolygon and MPolygon buttons |
| 5 | Hilman | 16 | 00:06:28 | Differentiate draw single and multiple polygon |
| 6 | Predrika | 16 | 14:53:21 | From Single Polygon button, when we draw the second polygon, the first will disappear |
| 7 | Predrika | 27 | 15:22:09 | Delete item on list box |
|  | Predrika | 27 | 15:43:14 | Adding polygon's point into list box and refresh canvas |
|  | Hilman | 30 | 01:31:01 | Determine whether the clipping is convex or not |
|  | Hilman | 30 | 22:30:27 | If polygon clipping is not convex, don’t draw the polygon |
|  | Predrika | 31 | 11:53:29 | Determine whether the clipping is convex or not (correcting) |
|  | Predrika | 31 | 12:03:45 | Determine whether the clipping is clockwise or anticlockwise |
|  | Predrika | 3 | 17:04:53 | Rectangular Clipping (not finished yet) |
|  | Hilman | 3 | 03:12:29 | Add insidepoint function and clippingpoint funtion |
|  | Hilman | 3 | 23:02:58 | Draw rect with 2 points |
|  | Hilman | 3 | 23:30:47 | Rect bug fixed |
|  | Nikita | 4 | 17:20:21 | Correcting the point clipping in the listbox |
|  | Hilman | 4 | 07:29:42 | Test in and out |
|  | Hilman | 6 | 11:42:33 | Inside - outside (Fixed) |
|  | Hilman | 7 | 10:42:26 | linked list using lis of object(not finished yet) |
|  | Predrika | 8 | 15:53:56 | Better user interface |
|  | Hilman | 8 | 01:34:21 | Initialized all polygon into linked list after rectangular clipping is drawn |
|  | Predrika | 8 | 16:57:06 | Disable properties button when there's no polygon |
|  | Hilman | 9 | 06:15:14 | Linkedlist update |
|  | Predrika | 9 | 21:52:14 | -Disable clicks on the picture box when the button to draw polygon has not been clicked  - Disable clipping and properties after the user click refresh |
|  | Predrika | 10 | 20:43:21 | Implementation and edit introduction in report |
|  | Predrika | 11 | 17:54:33 | -Save to bitmap  -Save into txt |
|  | Hilman | 11 | 05:32:22 | Clipping (still has bug) |
|  | Predrika | 12 | 13:23:57 | Refresh (also delete the linked list) |
|  | Predrika | 12 | 13:46:36 | The second Single Polygon will delete all the previous polygon along with the listbox |
|  | Predrika | 12 | 19:42:20 | -Evaluation  -Design  -Worklog |

# Conclusion and Remarks

## Does the program work as expected?

## If some parts of the program do not work as expected, explain why.

## What are your comments about this assignment