1. **Introduction**

This program is an application for drawing Polygon/Polyline which was created with Visual Basic as its programming language. Drawing a Polygon/Polyline is done by edge processing, in which we use a built-in procedure named “DrawLine” to draw each edge of Polygon/Polyline. There is also a class named “ClassPoly.vb” which is specifically created to store the data structures and some procedures that will mainly be used in the program.

Generally, there are some data structures that can be used to represent the list of points of a Polygon/Polyline (e.g., array, linked-list, linked-table). The Polygon/Polyline in this program will only be represented by array of points.

1. **Basic Theory**

A Polygon is a figure that is created from a list of finite number of vertices, in which each vertex is connected by a straight line (edge) to the next vertex consecutively and the last vertex is connected to the first vertex to form a closed circuit. A figure needs to have at least 3 vertices to be considered as a valid polygon.

A Polyline is a figure that is created from a list of finite number of vertices, in which each vertex is connected by an edge to the next vertex consecutively until it reaches the last vertex. A figure needs to have at least 2 vertices to be considered as a valid polyline.

A Polygon/Polyline is created by taking the coordinates of the first vertex (X1 , Y1), inserting it to the array of points and increase the size of that array by 1. Then, draw a straight line to connect the starting vertex to the next vertex and make that next vertex as the new starting vertex. This step is repeated for each subsequent vertex.

These processes will stop when the size of the array is more than or equal the minimum number of vertices for creating a valid Polygon/Polyline and the coordinates of the last vertex (Xlast , Ylast) is inserted to the array. However, there is one additional step needed to finish drawing a Polygon, which is to draw a straight line to connect the last vertex (Xlast , Ylast) to the first vertex (X1 , Y1).

Basically, there are 2 kinds of Polygon/Polyline processing, which are :

1. Point processing

In this processing, the Polygon/Polyline is processed point by point and each edge is drawn by plotting a number of connected points from the starting vertex to the next vertex. One of the algorithms that can be used to draw an edge of a Polygon/Polyline with point processing is Digital Differential Analysis which is provided below.

**Definitions :**

x : integer

m, y : real

**Algorithm :**

m 🡨 (y2–y1)/(x2–x1)

x 🡨 x1

y 🡨 y1

While x ≤ x2 do

Set(x, Round(y))

x 🡨 x + 1

y 🡨 y + m

{x > x2}

1. Edge processing

In this processing, the Polygon/Polyline is processed edge by edge and each edge is drawn by drawing a straight line from the starting vertex to the next vertex. One of the algorithms that can be used to draw an edge of a Polygon/Polyline with edge processing is provided below.

**Definitions :**

x1, x2, y1, y2 : integer

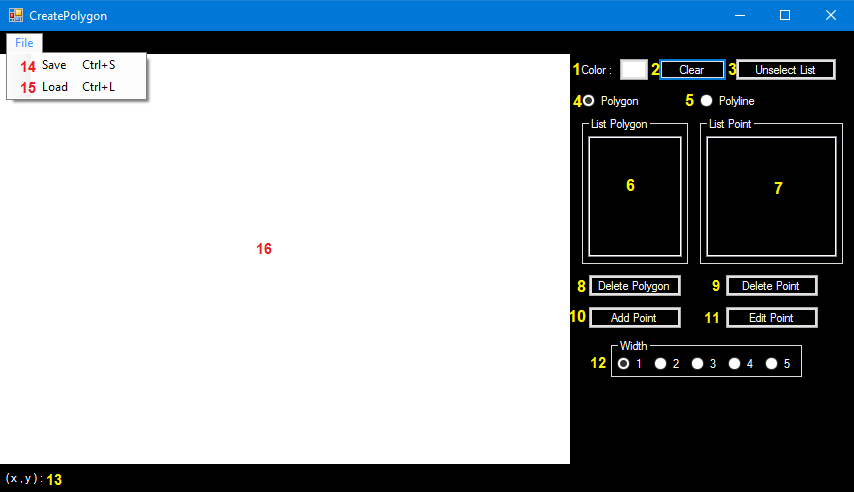
**Algorithm :**

DrawLine(x1, y1, x2, y2)

To handle creating multiple Polygon/Polyline, list (array) of Polygon/Polyline is used to store the data of every Polygon/Polyline that has been created before.

1. **User Manual**

The interface of this application is shown in the picture below.



1. Color Button, used to change/set the color of Polygon/Polyline.
2. Clear Button, used to clear the screen.
3. Unselect List Button, used to unselect Polygon/Polyline on the list.
4. Polygon Radiobutton, used to enable the user to draw or see the list of Polygons.
5. Polyline Radiobutton, used to enable the user to draw or see the list of Polylines.
6. List Polygon/Polyline, used to show the list of Polygons/Polylines.
7. List Point, used to show the vertices of a Polygon/Polyline.
8. Delete Polygon/Polyline Button, used to delete a Polygon/Polyline.
9. Delete Point Button, used to delete a vertex of Polygon/Polyline.
10. Add Point Button, used to add new vertex to Polygon/Polyline.
11. Edit Point Button, used to edit a vertex of Polygon/Polyline.
12. Sets of Width, used to change/set the width of Polygon/Polyline.
13. Coordinates, used to show the coordinates of pointer on the screen.
14. Save File, used to save the data of every Polygon and Polyline on the screen to a text file.
15. Load File, used to load the data of every Polygon and Polyline on the screen from a text file.
16. Screen, used to show the visual/graphical figure of each Polygon/Polyline.

Features

1. Draw a polygon/polyline

In order to draw a polygon/polyline, the user needs to click the starting vertex on the screen then click the next vertex on the screen. To finish drawing a polygon/polyline, user needs to double click the screen at the finishing vertex. The polygon/polyline data will then appear on the list of polygon/polyline. For the polygon, the user needs to have at least 3 vertices on the screen, while for the polyline; the minimum number of vertices is 2.

1. Choose a polygon/polyline

In order to choose a polygon/polyline, the user needs to click a polygon/polyline in the list of polygons/polylines provided in the application

1. Choose a vertex of polygon/polyline and show the coordinates

In order to choose a vertex of polygon/polyline and show the coordinates, the user needs to choose a polygon/polyline, and then the user needs to click the vertex in the list of vertices provided in the application.

1. Unselect a polygon/polyline in the list

In order to unselect a polygon/polyline in the list, the user needs to click the “Unselect List” button provided in the application.

1. Choose to draw or see the list of polygons/polylines

In order to draw or see the list of polygons, the user needs to click the “Polygon” radio button provided in the application.

In order to draw or see the list of polylines, the user needs to click the “Polyline” radio button provided in the application.

1. Delete a polygon/polyline

In order to delete a polygon/polyline, the user needs to choose the polygon/polyline and click the “Delete Polygon” or the “Delete Polyline” button.

1. Delete a vertex of polygon/polyline

In order to delete a vertex of polygon/polyline, the user needs to choose the polygon/polyline then choose the vertex and click the “Delete Point” button.

For polygon, as long as the number of vertices is more than 3 then its vertex can still be deleted.

For polyline, as long as the number of vertices is more than 2 then its vertex can still be deleted.

1. Change or set the color of a polygon/polyline

In order to change the color of a polygon/polyline that has been created before, the user needs to choose the polygon/polyline, then click the “Color” button and choose the color.

In order to set the color of a polygon/polyline that has not been created yet, the user needs to click the “Color” button, then choose the color and draw a polygon/polyline on the screen.

1. Change or set the width of a polygon/polyline

In order to change the width of a polygon/polyline that has been created before, the user needs to choose the polygon/polyline and click the radio button of the width.

In order to set the width of a polygon/polyline that has not been created yet, the user needs to click the radio button of the width and draw a polygon/polyline on the screen.

1. Add new vertex to the polygon/polyline

In order to add new vertex to the polygon/polyline, the user needs to choose the polygon/polyline and click the “Add Point” button. A prompt message will show up and ask the user to input the index of the new vertex. After inputting and clicking OK, the user needs to click the location of the new vertex on the screen.

1. Edit a vertex of polygon/polyline

In order to edit a vertex of polygon/polyline, the user needs to choose the polygon/polyline and choose the vertex. Then, click the “Edit Point” button. Lastly, the user needs to click the location of the edited vertex on the screen.

1. Clear the screen

In order to clear the screen, the user only needs to click the “Clear” button provided in the application.

1. Save all information of the polygon and the polyline on the screen to a text file

In order to save all information of the polygon and the polyline on the screen to a text file, the user needs to click “File” in the menu located at the top of the application, then click “Save”. The user can also save the data of polygon and polyline on the screen by pressing CTRL+S. A window will show up and the user needs to choose the location for the saved text file, then input the name for that file and click “Save”.

1. Load all information of the polygon and the polyline on the screen from a text file

In order to load all information of the polygon and the polyline on the screen from a text file, the user needs to click “File” in the menu located at the top of the application, then click “Load”. The user can also load the data of polygon and polyline on the screen by pressing CTRL+L. A window will show up and the user needs to choose the location for the targeted text file, then choose the file and click “Open”.

1. **How a polygon/polyline is represented**

As it has already been said in the introduction, a Polygon/Polyline in this program is represented by an array of points. This program use class instead of struct for the data structure. There are 3 classes used to represent the Polygon/Polyline, which are TPoint, TPoly and TArrPoly.

1. TPoint class can be defined as follows :

**Class TPoint**

X : integer

Y : integer

Class TPoint is used to store the coordinates of a vertex of the Polygon/Polyline.

Properties : - X, used to store the x-coordinate of a vertex of the Polygon/Polyline.

- Y, used to store the y-coordinate of a vertex of the Polygon/Polyline.

1. TPoly class can be defined as follows :

**Class TPoly**

N : integer

Elmt() : TPoint

Wdth : integer

Clr : color

Class TPoly is used to store the information of a Polygon/Polyline.

Properties : - N, used to store the current size of the Polygon/Polyline.

- Elmt, an array of TPoint, used to store the set of vertices of the Polygon or

Polyline.

- Wdth, used to store the width of the Polygon/Polyline.

- Clr, used to store the color of the Polygon/Polyline.

1. TArrPoly class can be defined as follows :

**Class TArrPoly**

N : integer

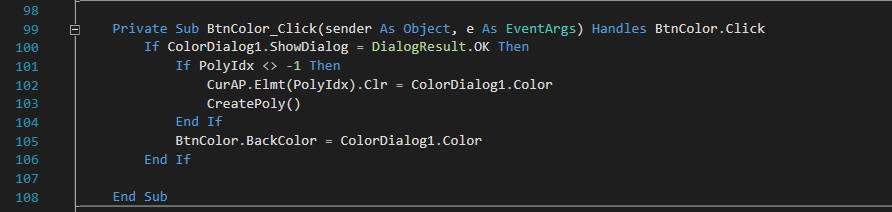
Elmt() : TPoly

Class TArrPoly is used to store all Polygons/Polylines that has been created.

Properties : - N, used to store the current size of the set of Polygons/Polylines.

- Elmt, an array of TPoly, used to store the set of Polygons or Polylines.

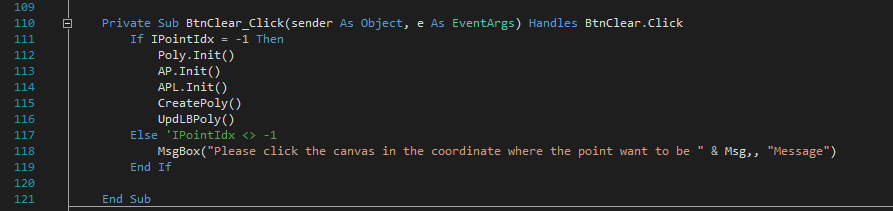
1. **Additional Implementation**
2. Change the color of Polygon/Polyline



In this program, when the user clicks “Color” button, then a color dialog will appear which allow the user to select a color. Variable PolyIdx (integer) used as the index of the currently selected Polygon/Polyline and variable CurAP (TArrPoly) is the current array of Polygon / array of Polyline depending on which one the user is currently selecting.

As long as the user currently selects a Polygon/Polyline ( *PolyIdx <> 1* ), the color of the Polygon/Polyline with index of PolyIdx ( *CurAP.Elmt(PolyIdx).Clr* ) will be changed to the color that is selected by the user. Procedure CreatePoly is used to redraw all Polygons and Polylines on the screen, hence any updates of the color of the Polygon/Polyline will also be updated on the screen. Lastly, the color of the button will also be changed to the selected color.

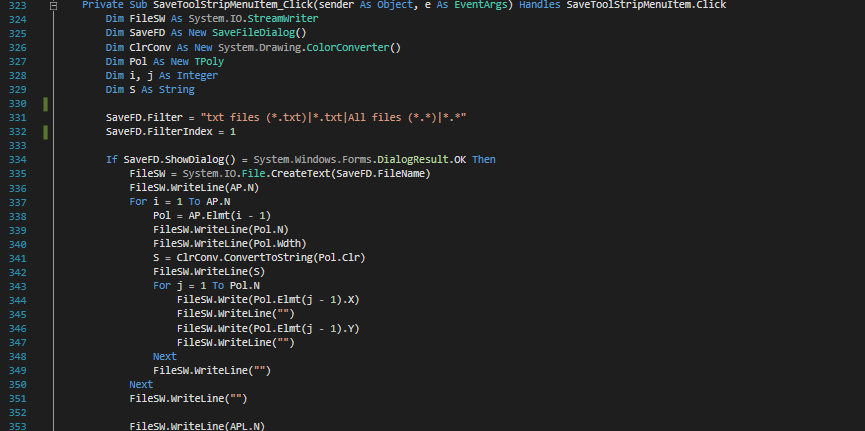
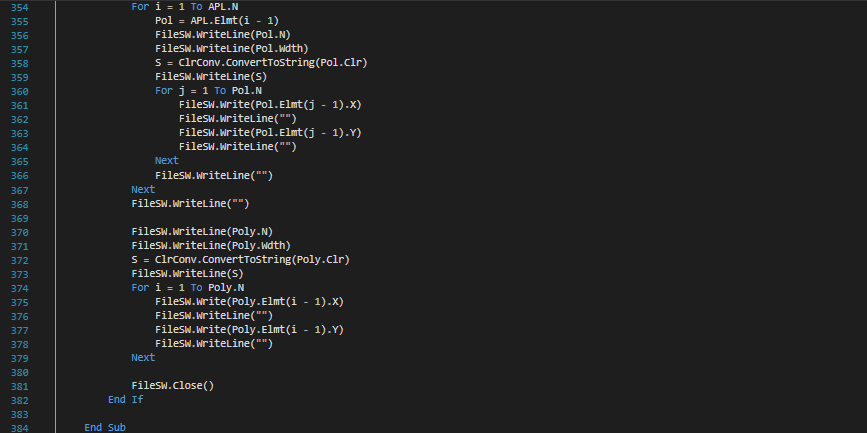
1. Clear the screen



In this program, variable IPointIdx (integer) is used as the index of the added or edited vertex and variable Msg (string) is used to complete the message by adding “added” or “edited” to the message in MessageBox. When user clicks “Clear” button, as long as there is no vertex of a Polygon/Polyline to be added or edited, Poly (TPoly), AP (TArrPoly) and APL (TArrPoly) will be initialized to their default value (the size N is 0 and the array Elmt will be empty).

The program will call procedure CreatePoly to redraw (clear) all the Polygons and Polylines on the screen and the program will also call procedure UpdLBPoly to update (clear) the list of Polygons/Polylines. If currently there is a vertex of a Polygon/Polyline wanted to be added or edited, the message in MessageBox will appear instead.

1. Save data of Polygon/Polyline to a file



The used variables and their declaration can be seen in the above pictures. This program filters the type of saved file(s) to text files “.txt” and all files “.” and make it so that after “Save” in the menu is clicked, when the save window appeared for the first time, the default type for the file is “.txt”. After that, the file is created in the chosen directory with the inputted name and by using StreamWriter, the program writes all information of the Polygon/Polyline on the screen to the file.

The format of the saved text file is as follows :

Number of Polygons

Data of Polygon 1 ( **I** )

Data of Polygon 2

…

Data of Polygon N

Number of Polylines

Data of Polyline 1 ( **II** )

Data of Polyline 2

…

Data of Polyline N

Data of Polygon/Polyline which is drawn by the user but still has not finished yet and is neither considered as a Polygon nor a Polyline. ( **III** )

The first line indicates the number of polygons (N) in AP. The second line indicates the data of first polygon (Elmt(0)), the next line indicates the data of second polygon (Elmt(1)) and continue until the data of Polygon N (Elmt(N-1)).

Then, it is followed by the number of polylines (N) in APL. The next line indicates the data of first polyline (Elmt(0)), the line after that indicates the data of second polyline (Elmt(1)) and continue until the data of Polyline N (Elmt(N-1)).

The last line indicates the data of a Polygon/Polyline which is drawn by the user but still has not finished yet and is neither considered as a Polygon nor a Polyline.

The format of “Data of Polygon/Polyline” is as follows :

Number of vertices in the Polygon/Polyline

Width of Polygon/Polyline

Color of Polygon/Polyline

X1

Y1

X2

Y2

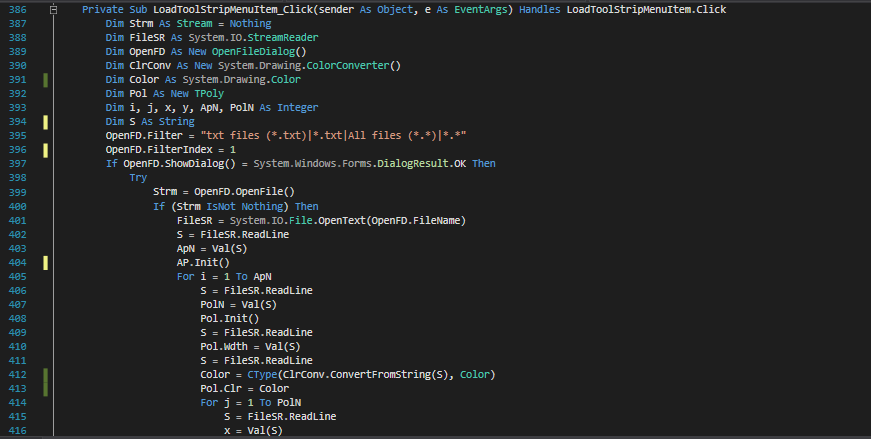
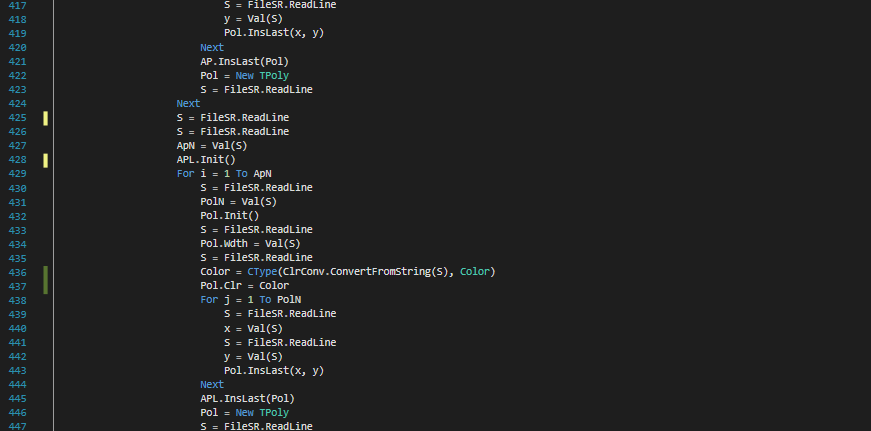
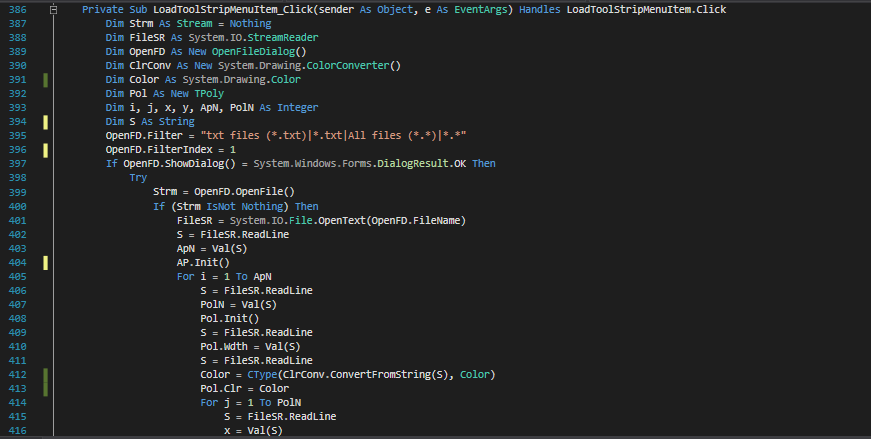
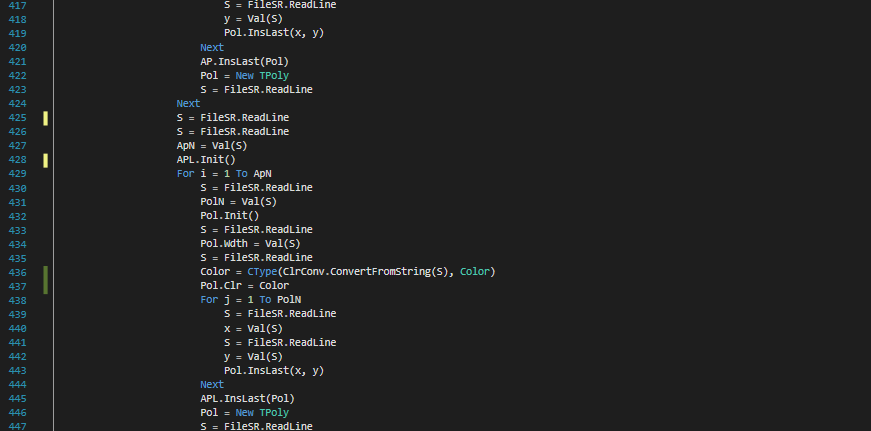
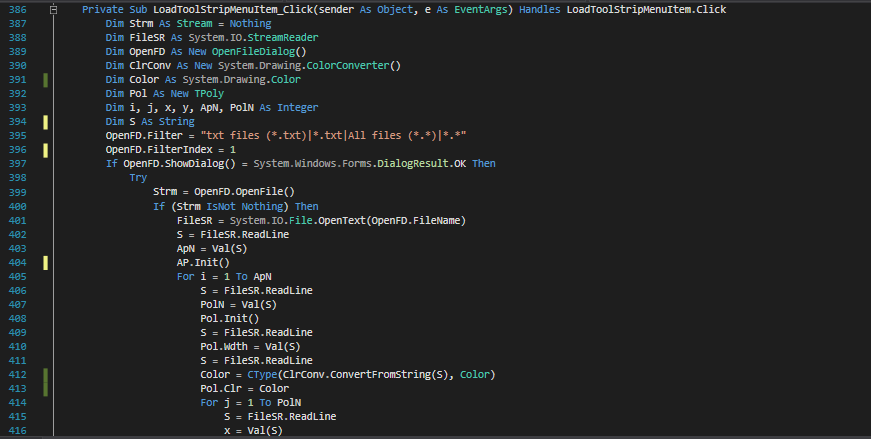
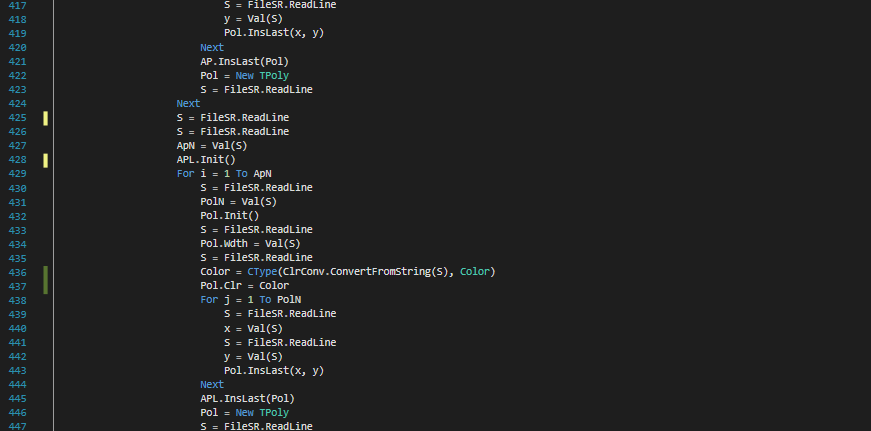
…

Xn

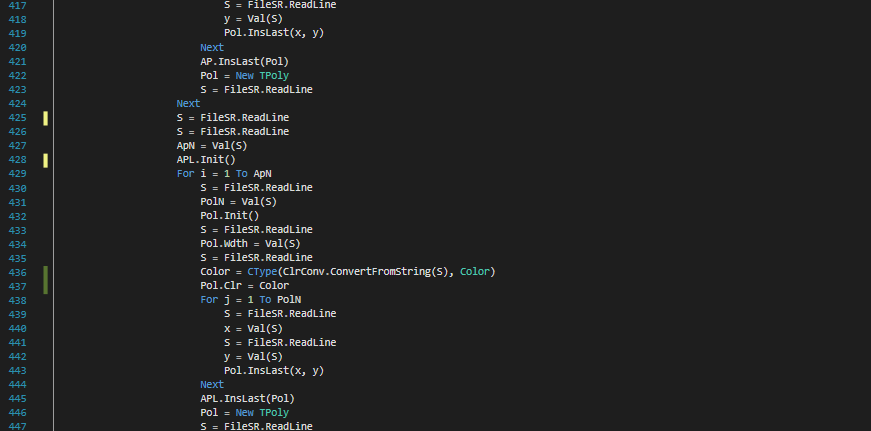
Yn

By taking **I**, **II** and **III** as the example, the first line indicates the number of vertices (N) in the Polygon/Polyline ( AP.Elmt(0), APL.Elmt(0) and Poly ). The second line the indicates width (Wdth) of the Polygon/Polyline. The third line indicates the color (Clr) of the Polygon/Polyline. The fourth line indicates the x-coordinate of the first vertex. The fifth line indicates the y-coordinate of the first vertex. The next line indicates the x-coordinate of the second vertex. The line after that indicates the y-coordinate of the second vertex. This will continue until the x-coordinate of the last vertex and the y-coordinate of the last vertex.

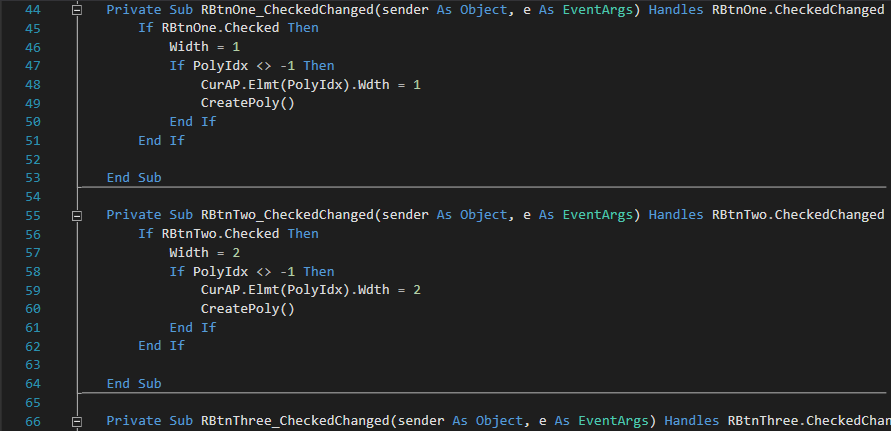
1. Load data of Polygon/Polyline from a file



The used variables and their declaration can be seen in the above pictures. This program also Imports System.IO. This program filters the type of loaded file(s) to text files “.txt” and all files “.” and make it so that after “Load” in the menu is clicked, when the load window appeared for the first time, the default type for the file is “.txt”. After that, the chosen file is loaded from the chosen directory and by using StreamReader, the program reads all information of the Polygon/Polyline from the file and create the Polygon/Polyline based on that information on the screen. If something went wrong in the middle of loading, then the program will show the message of “Cannot find file : “ and completed by the error message.



1. Additional Feature : Change the width of Polygon/Polyline

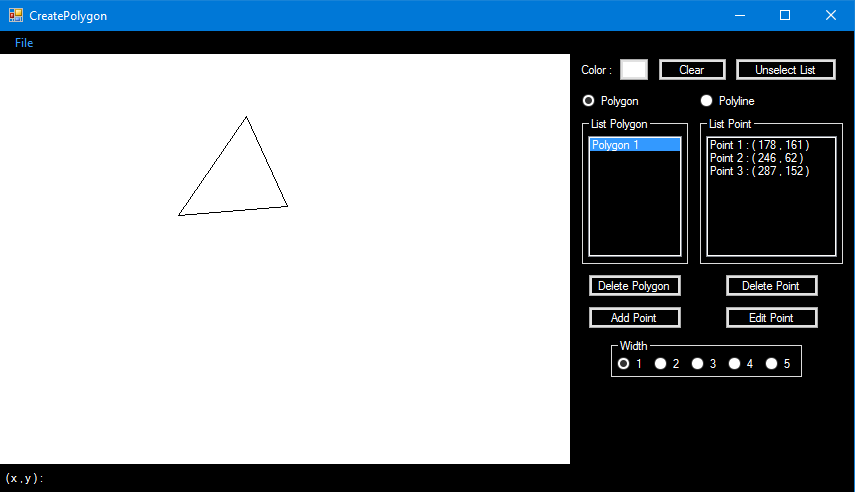


In this program, there are 5 radiobuttons used to change the width of a Polygon/Polyline. However the code is pretty much similar. The bigger the value of the width then the thicker the Polygon/Polyline will be.

If the radiobutton for the desired width has been checked, then variable Width which has previously been declared as integer will have the value of the selected width. This value will be used afterwards as the width of the pen when drawing a new Polygon/Polyline.

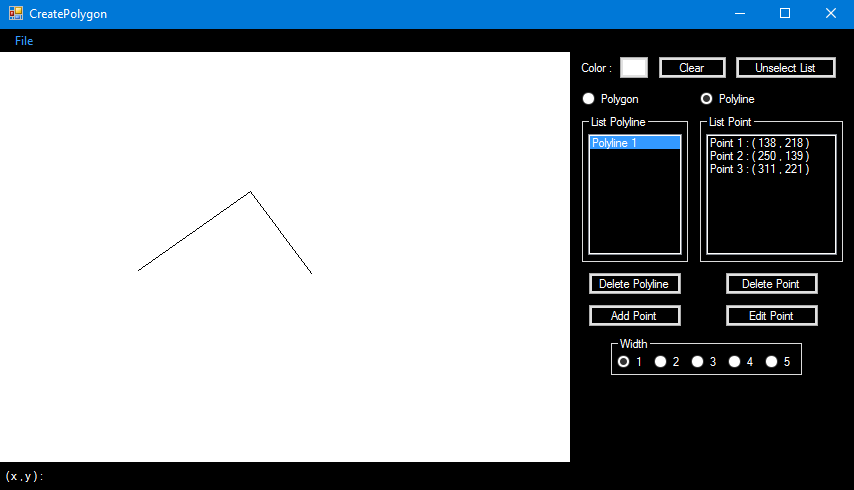
After that, if currently there is a Polygon/Polyline that is selected ( *PolyIdx <> 1* ), then the width of that Polygon/Polyline ( *CurAP.Elmt(PolyIdx).Clr* ) will be changed to the selected width and the width of Polygon/Polyline on the screen will also be updated accordingly.

1. **Evaluation of the Main Features**
2. Add Polygon



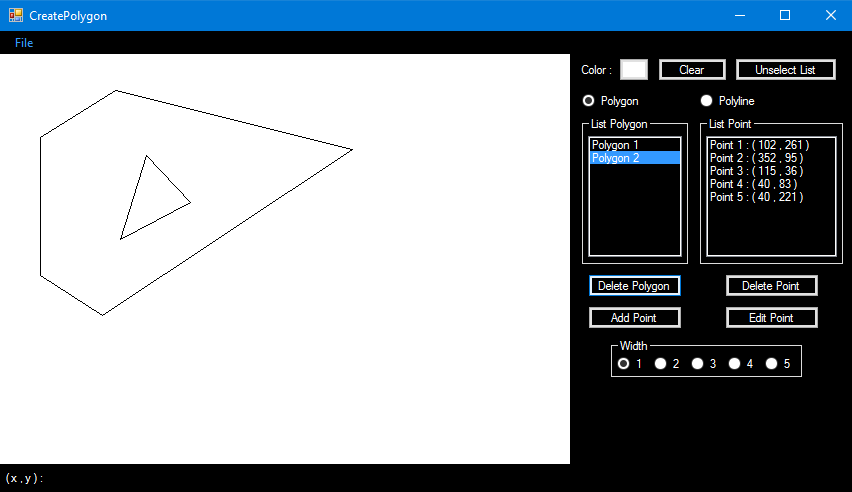
For this case, we create a polygon with 3 points and the coordinates are: Point 1 (178,161), Point 2 (246,62), Point 3 (287,152). The polygon is successfully created with the right coordinates.

1. Add Polyline

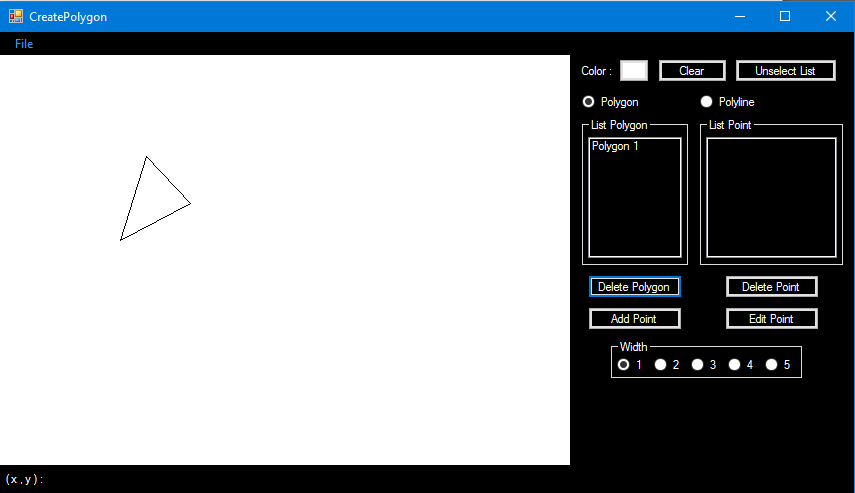


For this case, we create a polyline with 3 points and the coordinates are: Point 1 (138,218), Point 2 (250,139), Point 3 (311,221). The polyline is successfully created with the right coordinates.

1. Delete Polygon

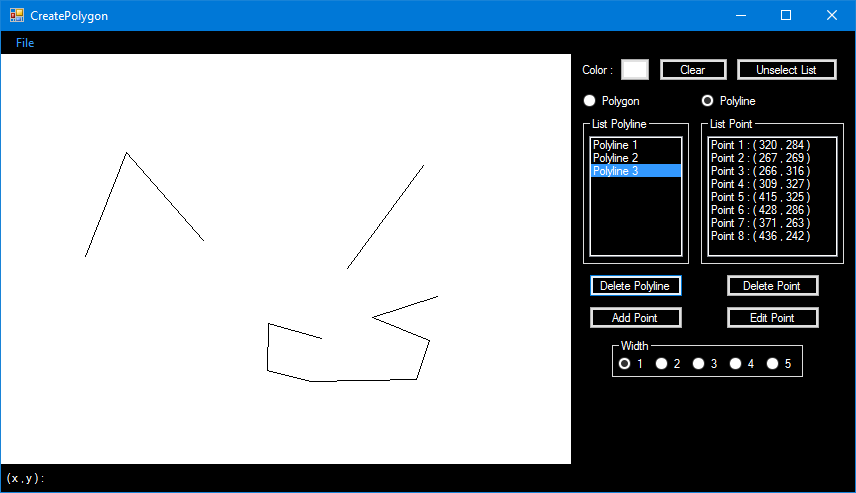


For this case, we create 2 polygons on the screen and then we delete Polygon 2.

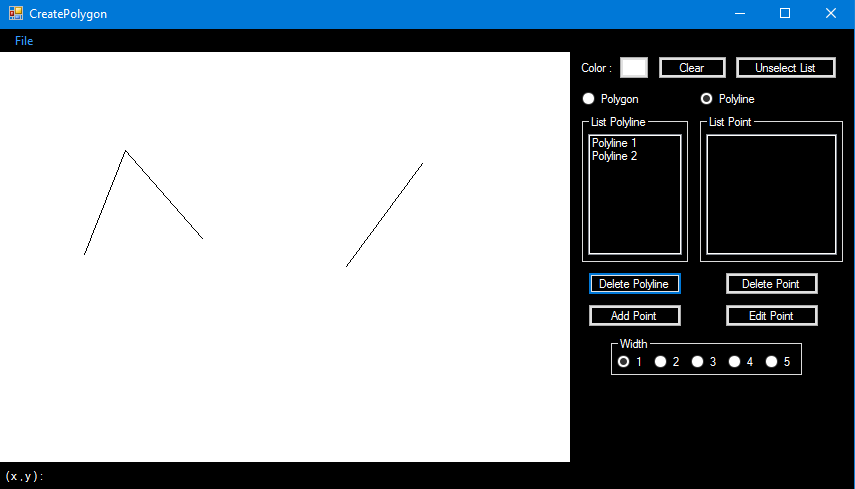


Polygon 2 has been deleted. Now, there is only 1 polygon on the screen.

1. Delete Polyline

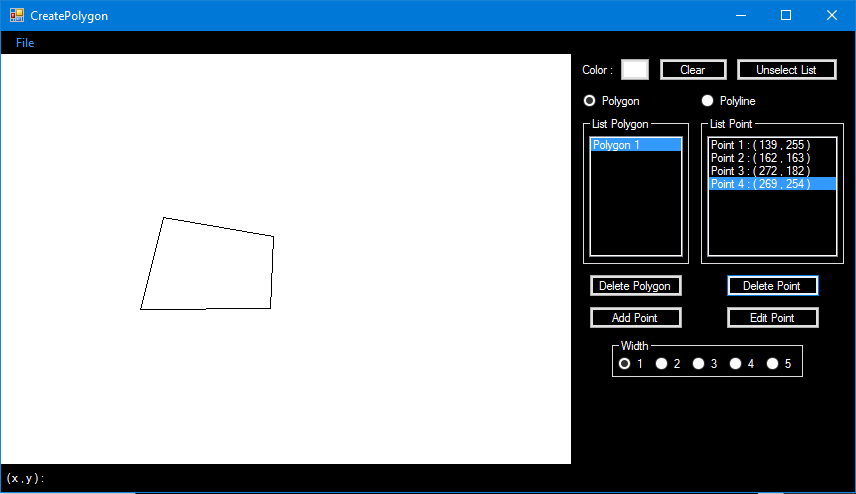


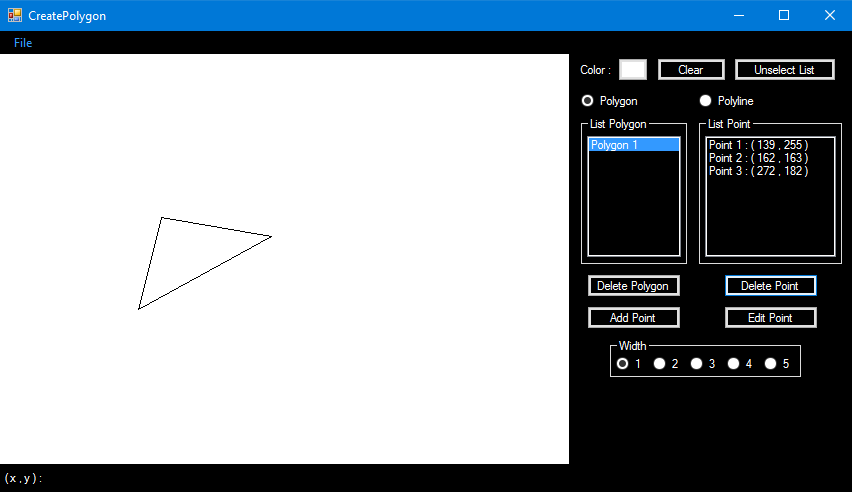
For this case, we create 3 polylines and then we delete Polyline 3.



Polyline 3 has been deleted. Now, there is only 2 polylines on the screen.

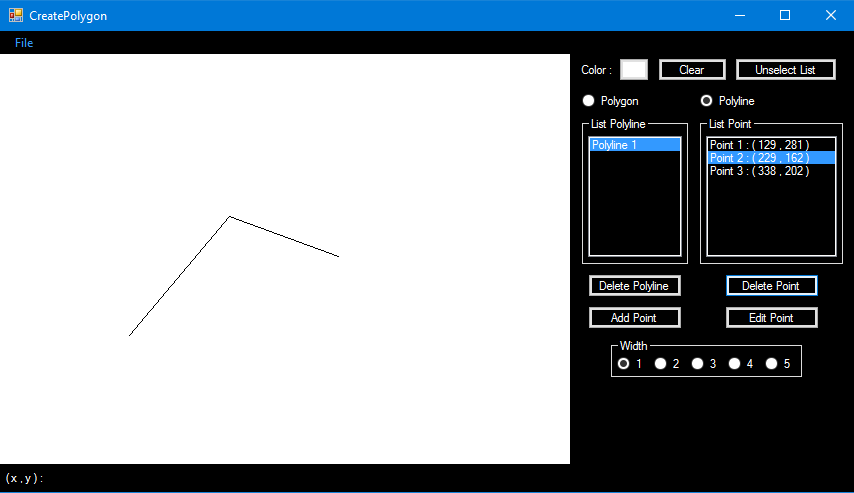
1. Delete Point of Polygon

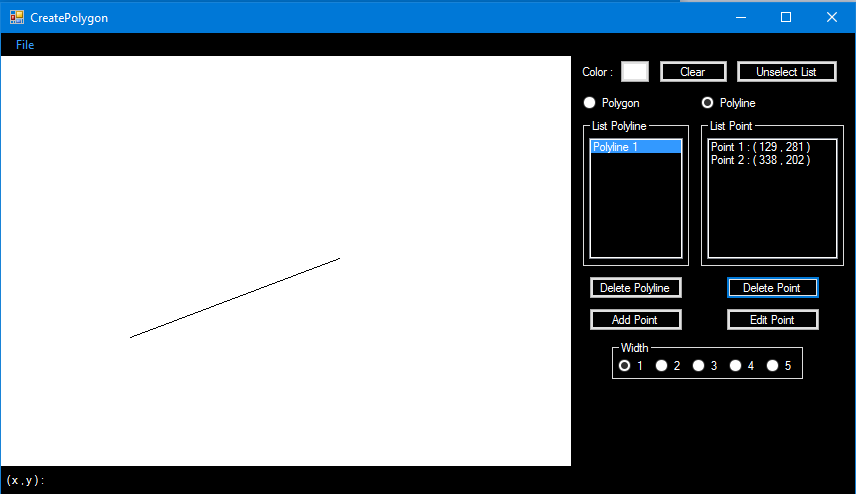
For this case, we create a polygon with 4 points and then we choose point 4 to be deleted.



Point 4 has been deleted, now the polygon only has 3 points.

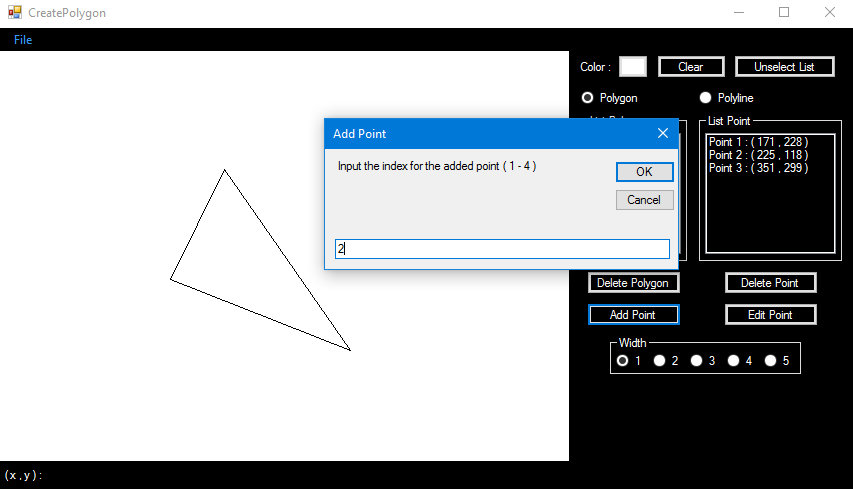
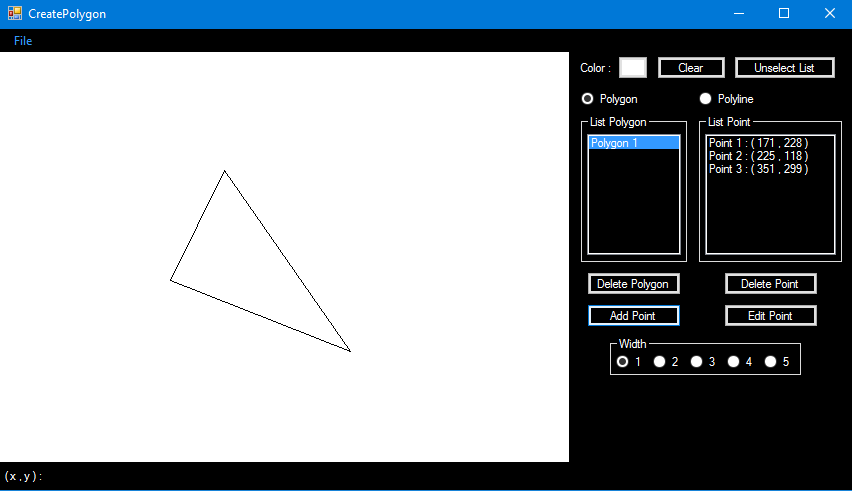
1. Delete Point of Polyline

For this case, we create a polyline with 3 points and then we choose point 2 to be deleted.

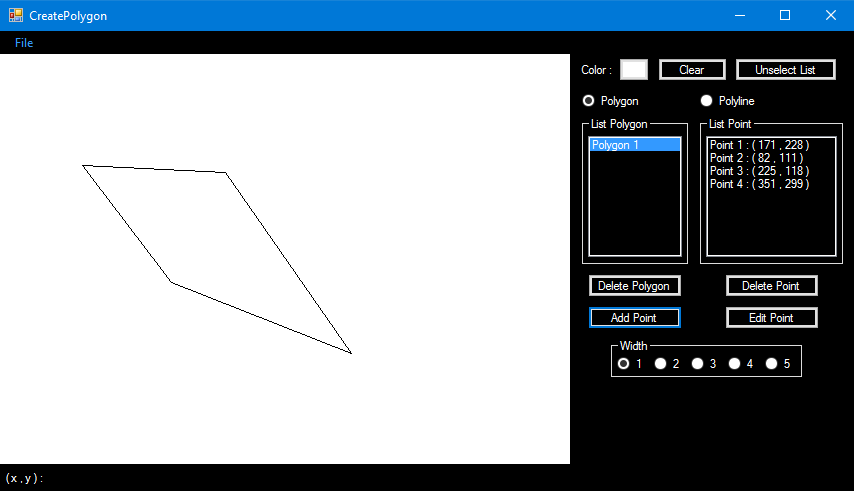
Point 2 has been deleted, now the polyline only has 2 points.

1. Add Point to Polygon

For this case, we create a polygon with 3 points.

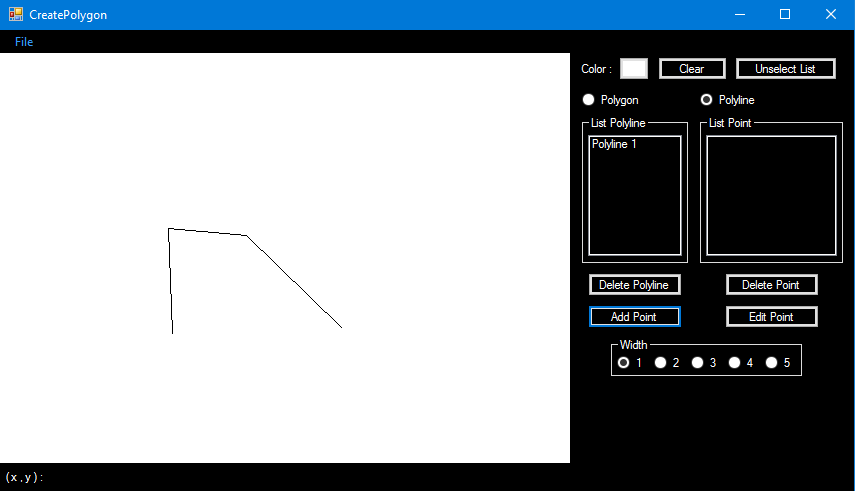


We input 2 as the index for the new point.

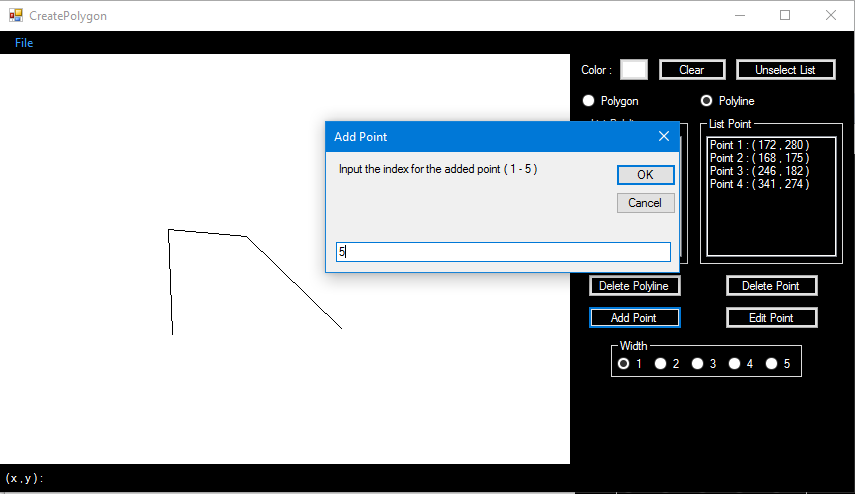


The new point has been added to the second index and now the polygon has 4 points.

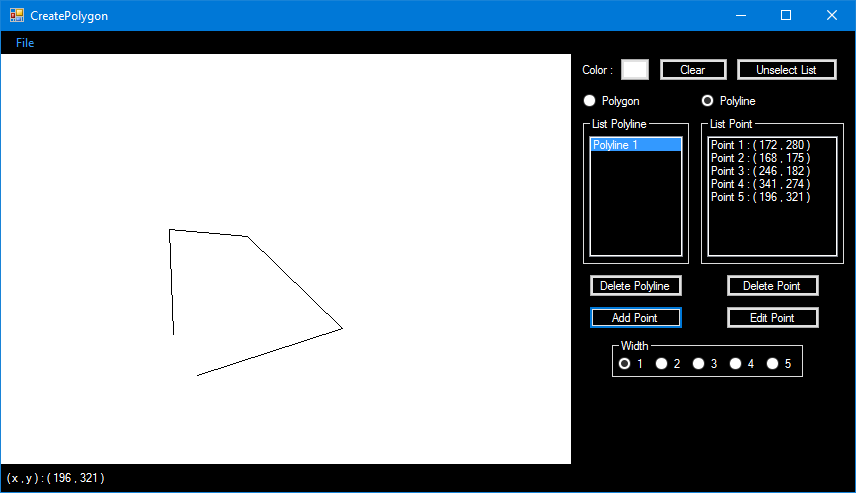
1. Add Point to Polyline



For this case, we create a polyline with 4 points.



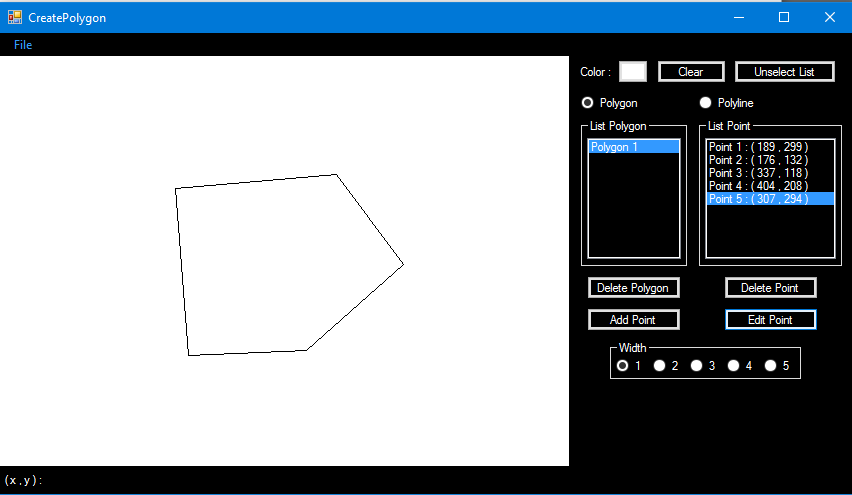
We input 5 as the index for the new point.



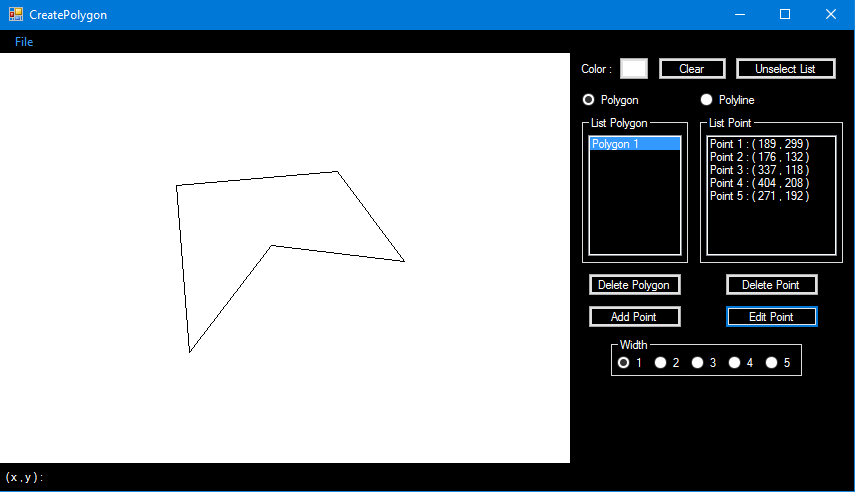
The new point has been added to the fifth index and now the polyline has 5 points.

1. Edit Point of Polygon

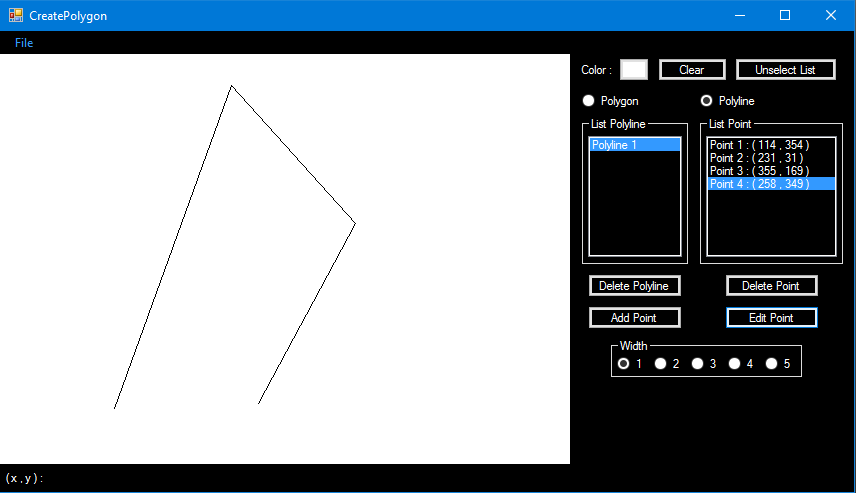
For this case, we create a polygon with 5 points and then we choose Point 5 to be edited.

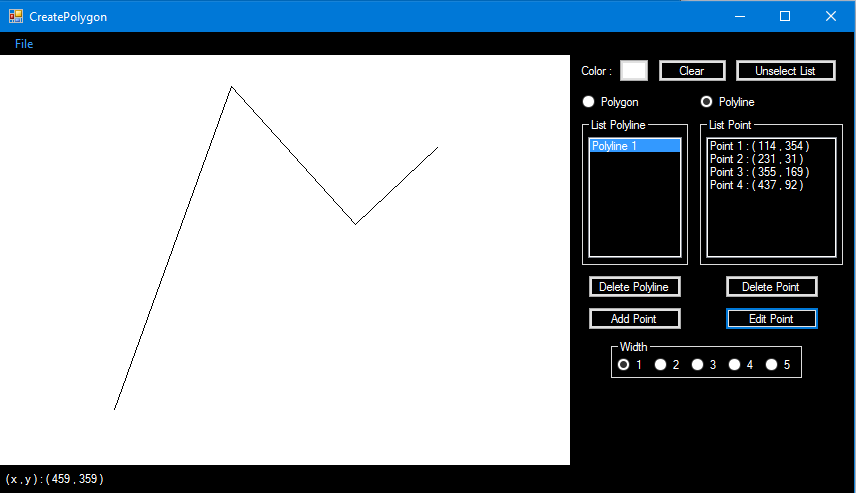


Point 5 has been edited, now the coordinates of point 5 has been changed.



1. Edit Point of Polyline

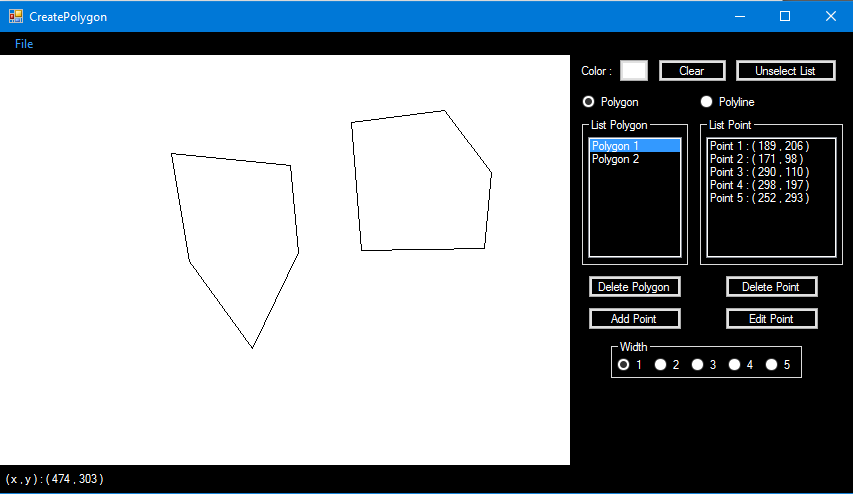
For this case, we create a polyline with 4 points and then we choose Point 4 to be edited.



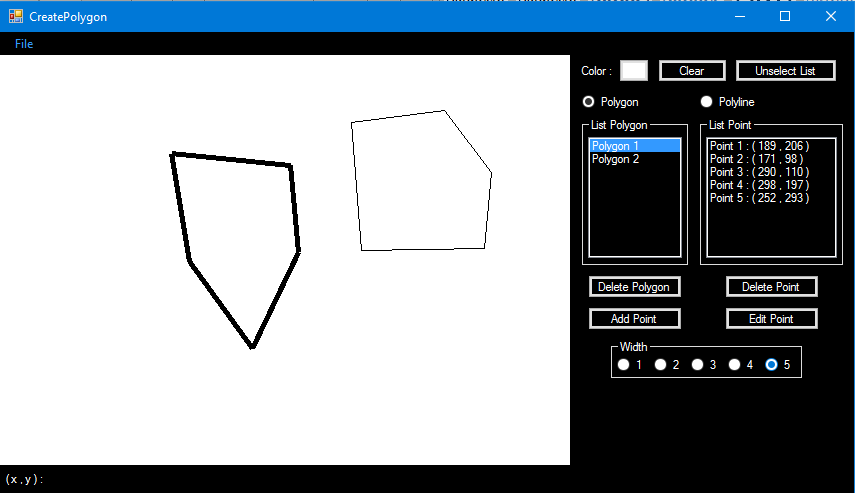
Point 4 has been edited, now the coordinates of point 4 has been changed.

1. Change the Width of Polygon/Polyline

For this case, we create 2 polygons with the width of 1 and then we choose Polygon 1.

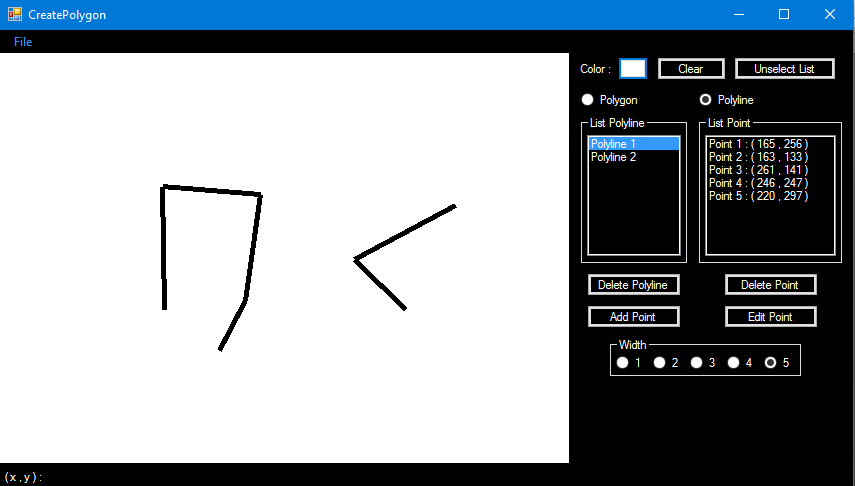


We change the width to 5 and now the width of Polygon 1 has been changed to 5.

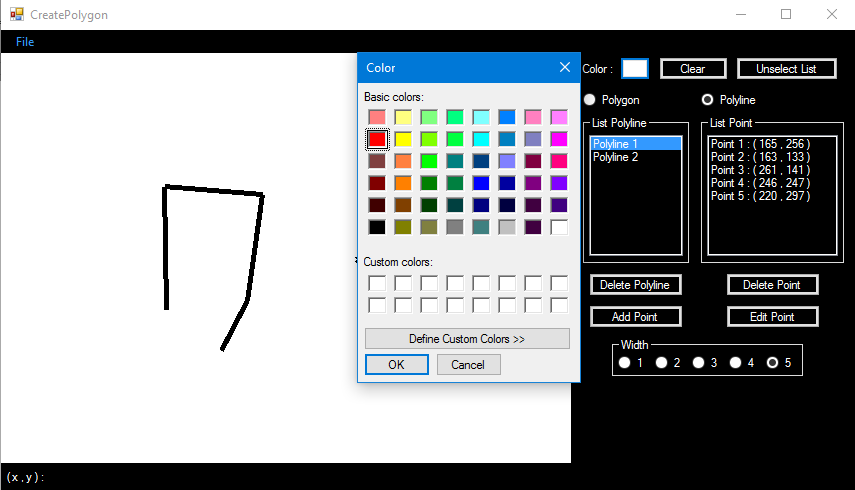


1. Change the Color of Polygon/Polyline

For this case, we create 2 black polylines and then we choose Polyline 1.



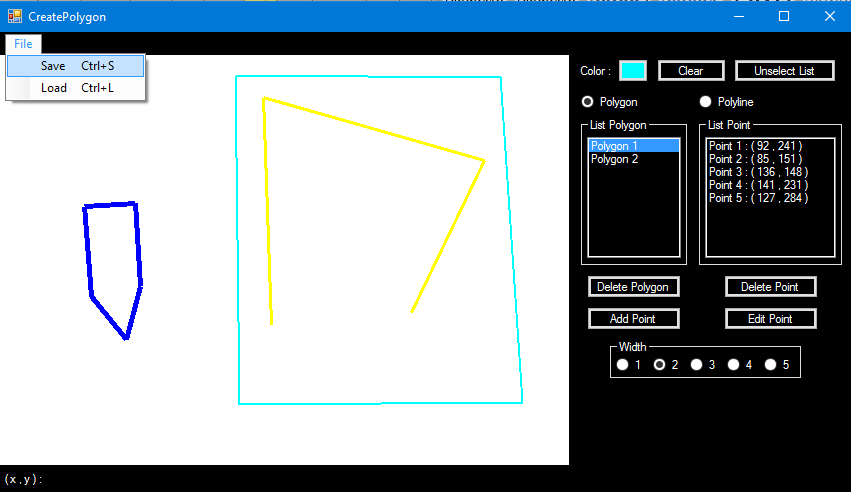
We choose red as the new color.

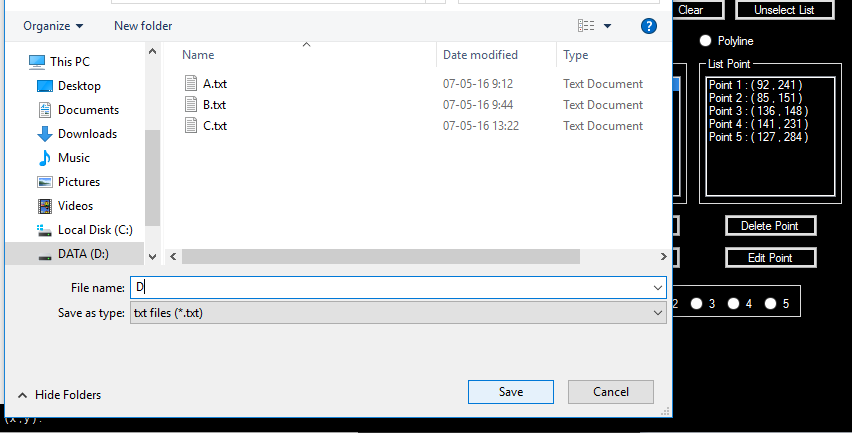


Now the color of Polyline 1 has been changed to red.

1. Save information of Polygon/Polyline to a Text File

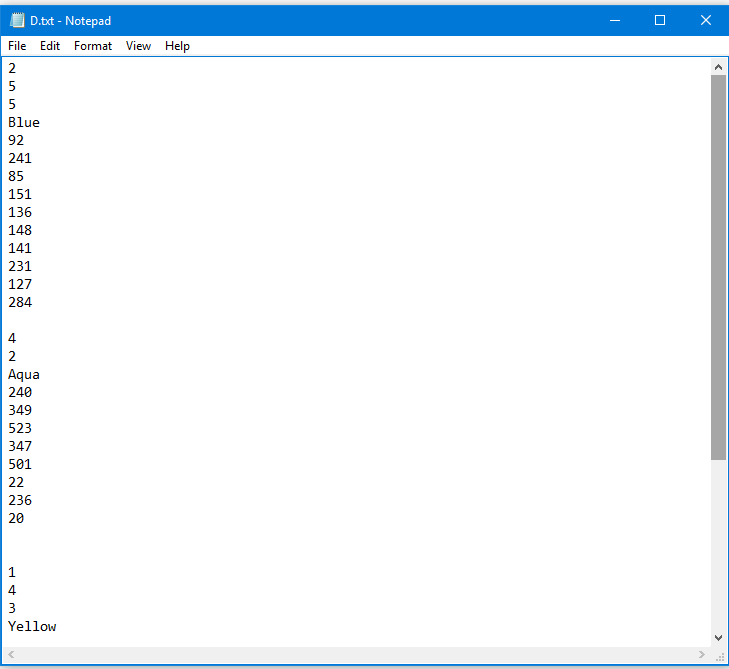
For this case, we create 2 polygons and 1 polyline and then we click save.



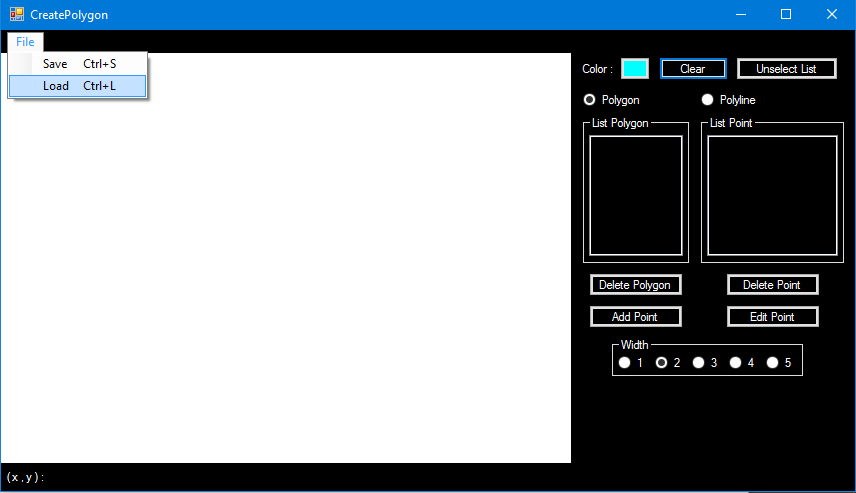


A save window will appear, then we input the name of text file (D) and we click save.

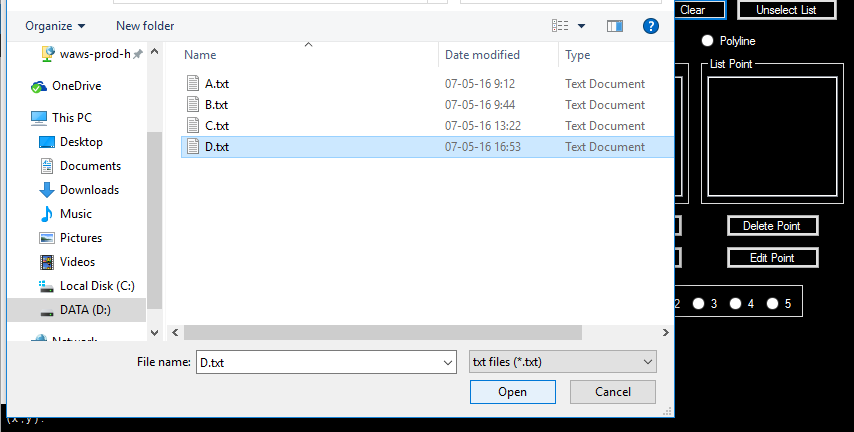
This is the content of “D.txt” text file that has been created, which store all the information of each polygon and polyline on the screen.



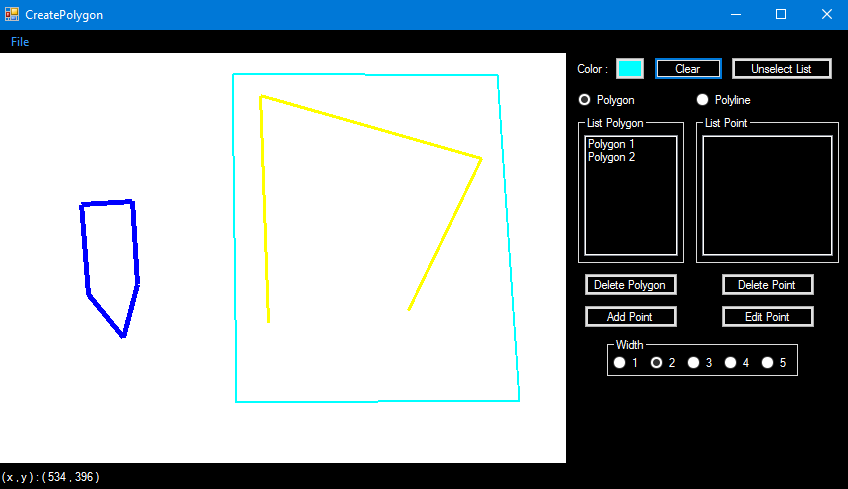
1. Load information of Polygon/Polyline from a Text File



For this case, we have nothing on the screen and then we click load.



A load window will appear, then we click the targeted text file (D.txt) and we click open.



Now the polygon and polyline has been loaded to the screen, based on the information from the chosen text file.