Vector Based Drawing Application.

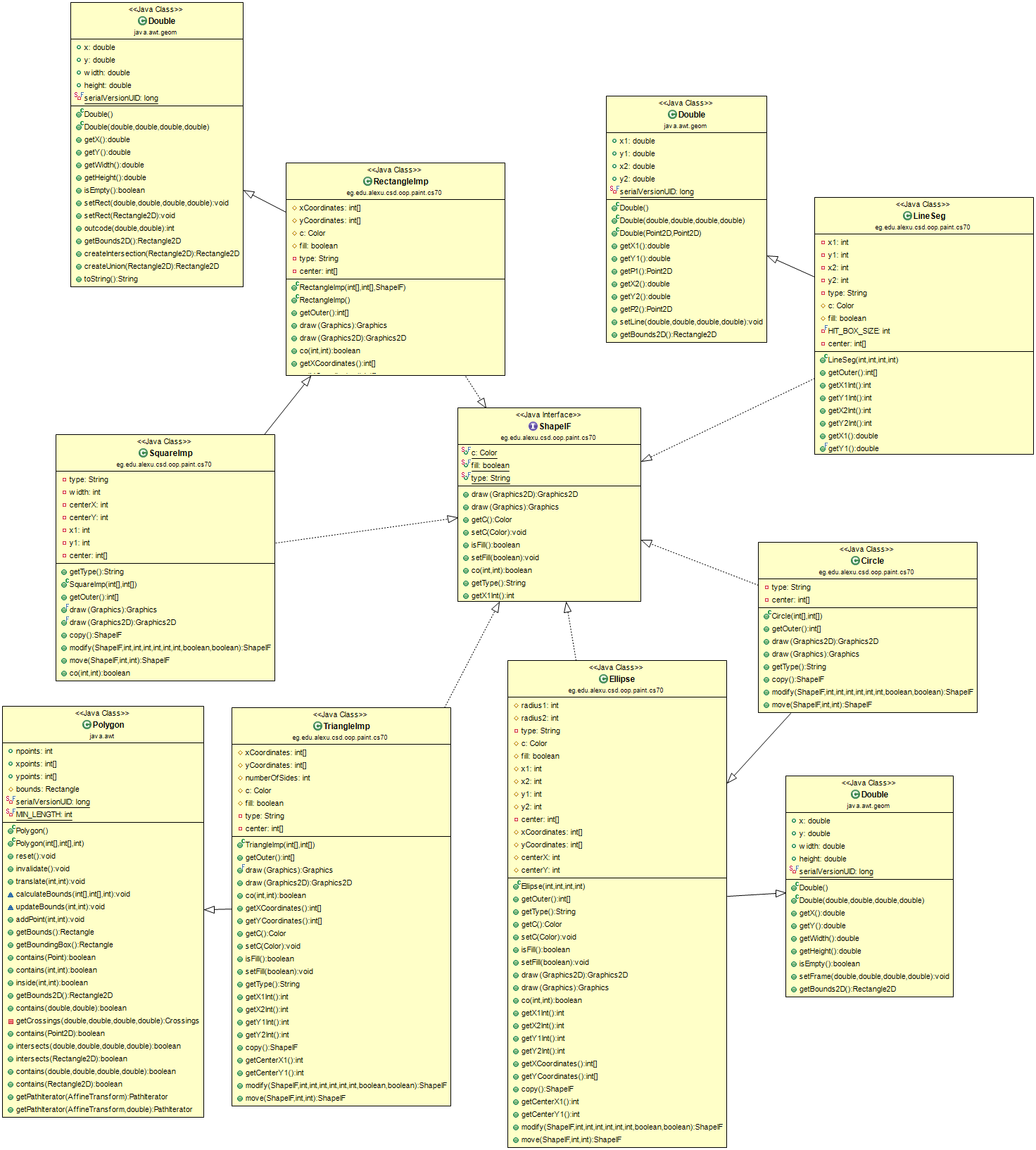
Abstract:

This application aims to help the people to draw geometric shapes such as Rectangle, Triangle, Line segment, …etc. by simple mouse clicks, also you can resize or change the colors of the shapes you draw and to undo and redo actions you made in case you want to get back to some state. This application also enables you to save the previous drawings so you can load it again and edit your drawings.

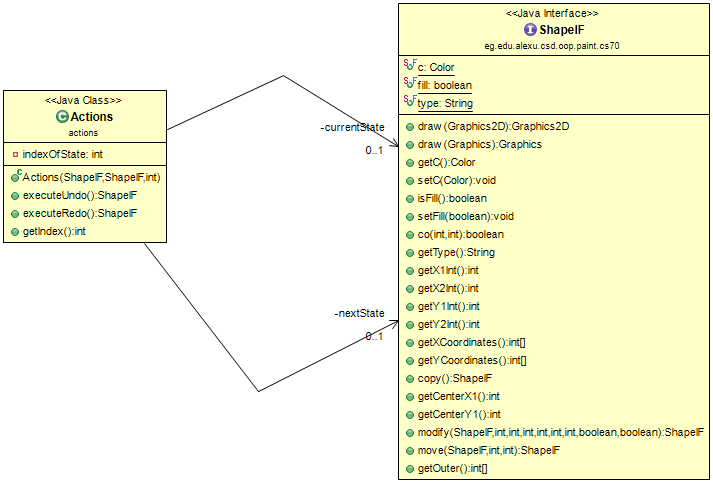
UML Diagram:

Shapes:

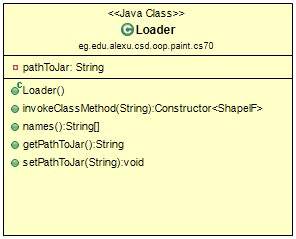
This UML describes the shapes hierarchy and the relations between them and how they inherit from each other.



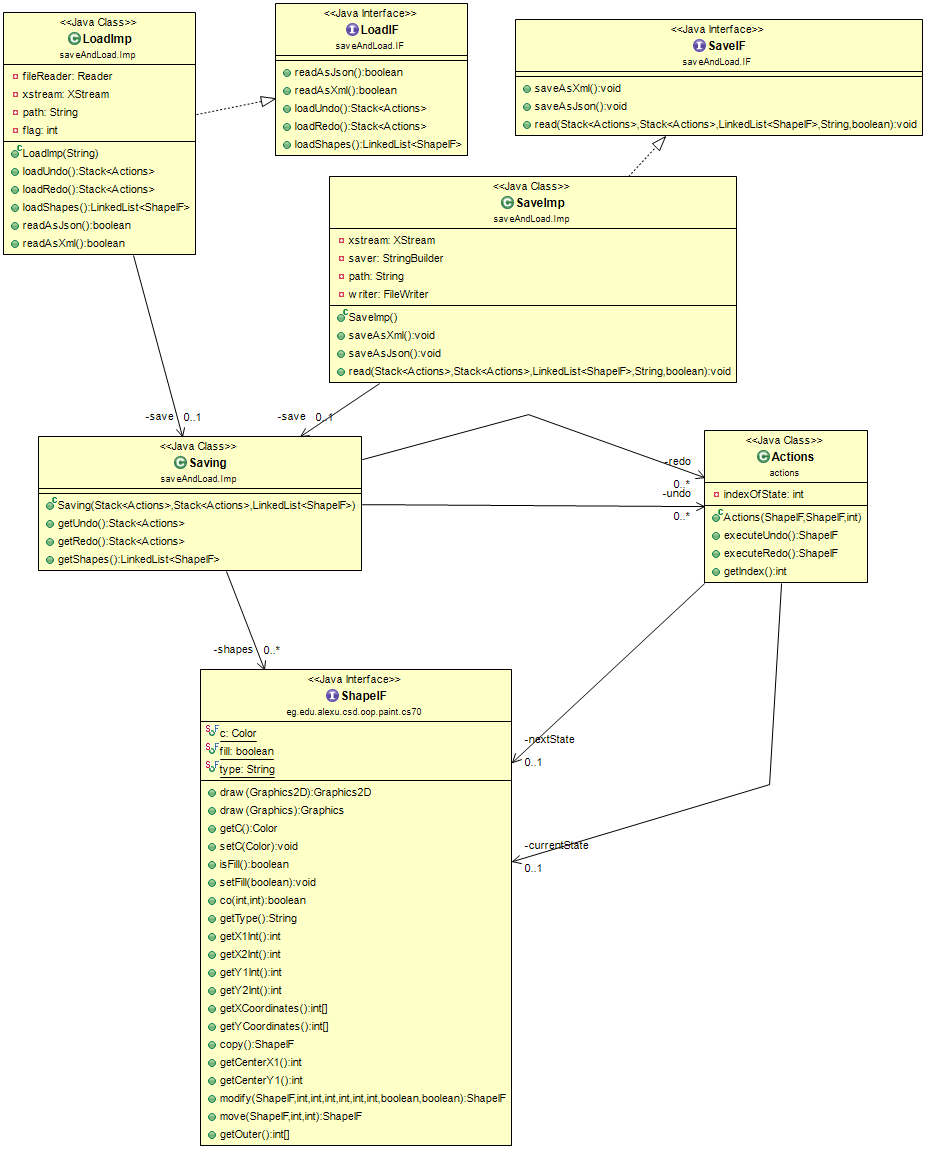
Actions:

 This class is used to save the actions made by the user such as deletion, changing color, creating, …etc. these actions save in case the user wanted to make undo or redo to get his shapes back by saving the state before the action and the state after the action.

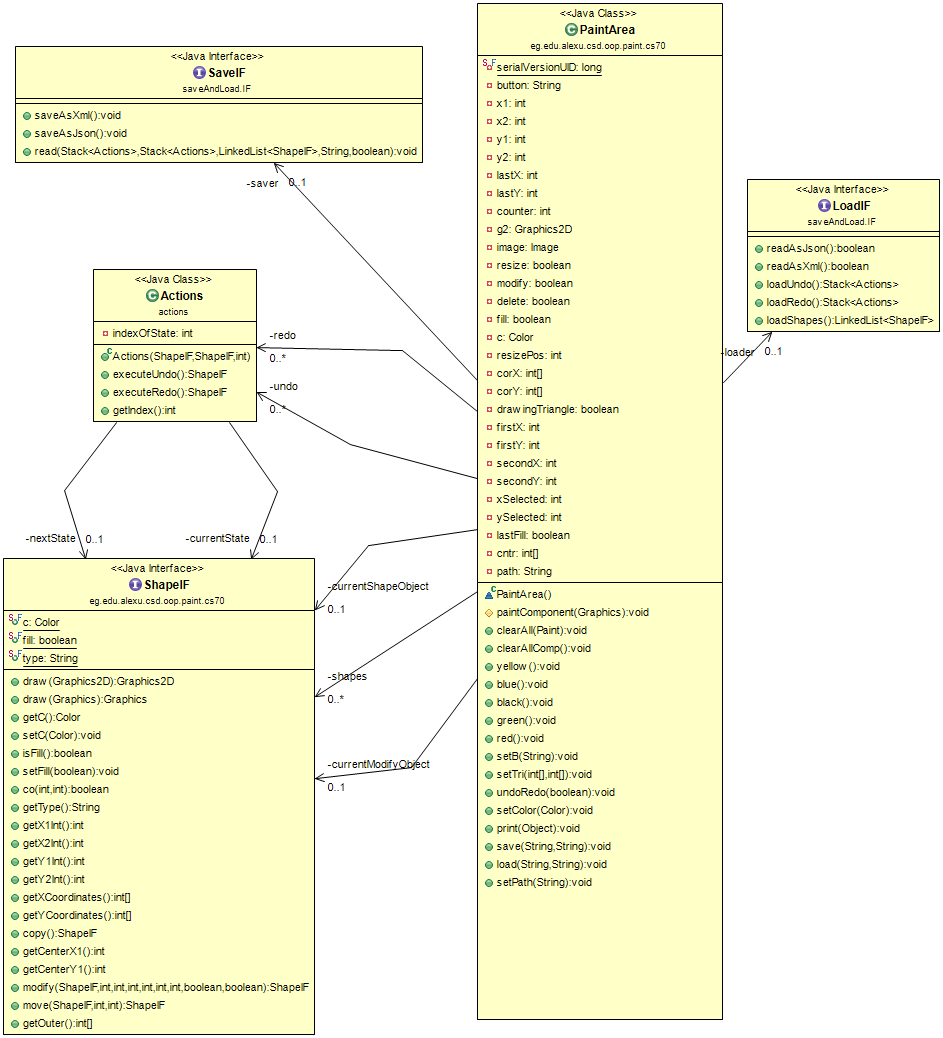
Class Loading:

 This class is responsible for loading external shapes that the user need to complete his drawing. By taking the jar that contains the new shape and load it then show the user its button.

Save and Load:

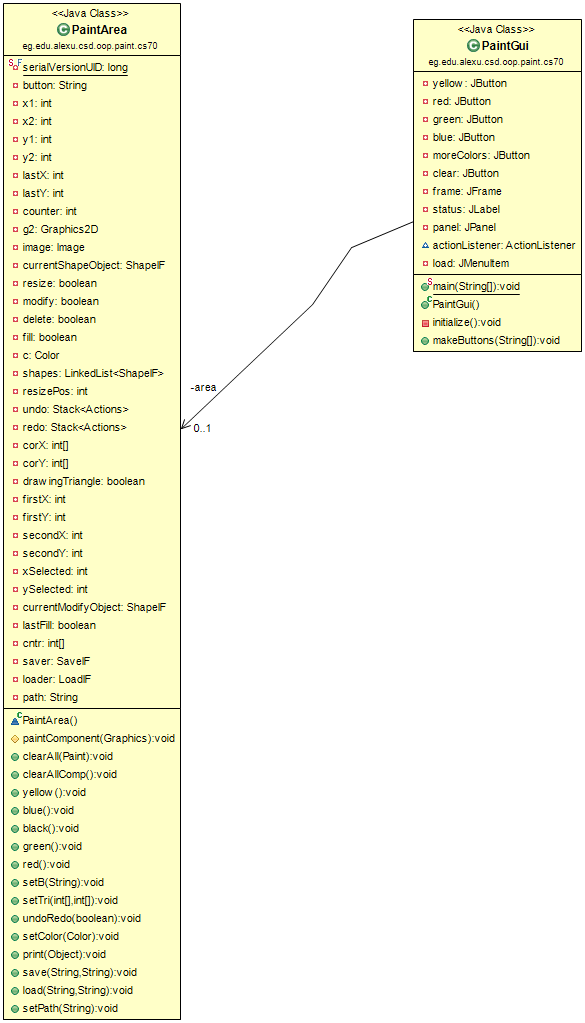
 When we made this class we aimed that the user be able to save his drawing as XML or JSON which he can afterwards load the drawing and edit as the user wants.

Main Controller:

 This class controls the flow of the application and get any request from the user and interact with him, it’s also responsible to control the main actions made by the user such as drawing, painting, loading, saving, …etc.

Graphical user interface (GUI):

That is the class which appears to the user where all logic is made behind this class. It only sends the data to the controller then the controller processes it and send the result to it appears to the user. This interface also is a user-friendly one which let the user interact with the application easily.



Design assumptions:

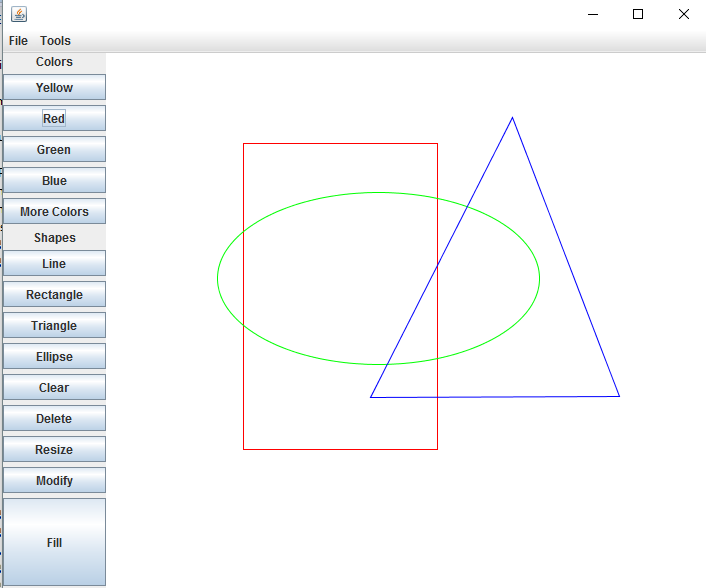
In this application we have made some assumption about how the user is going to interact and use this program.

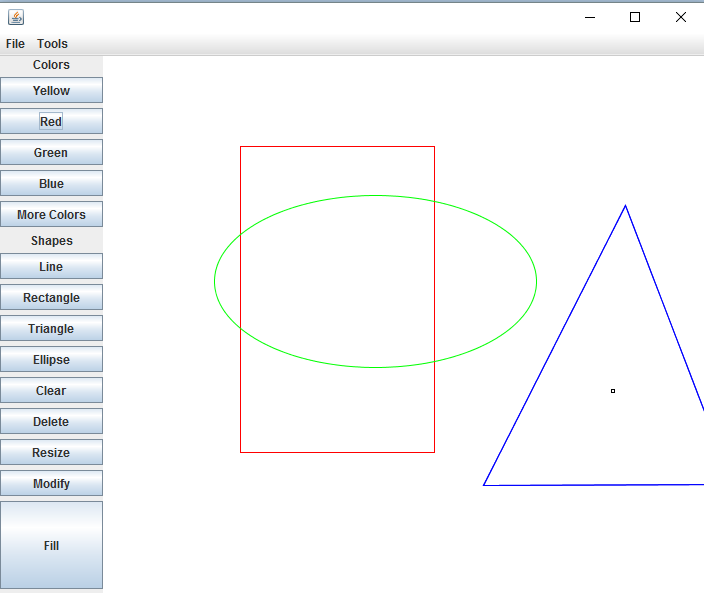
Those assumptions are:

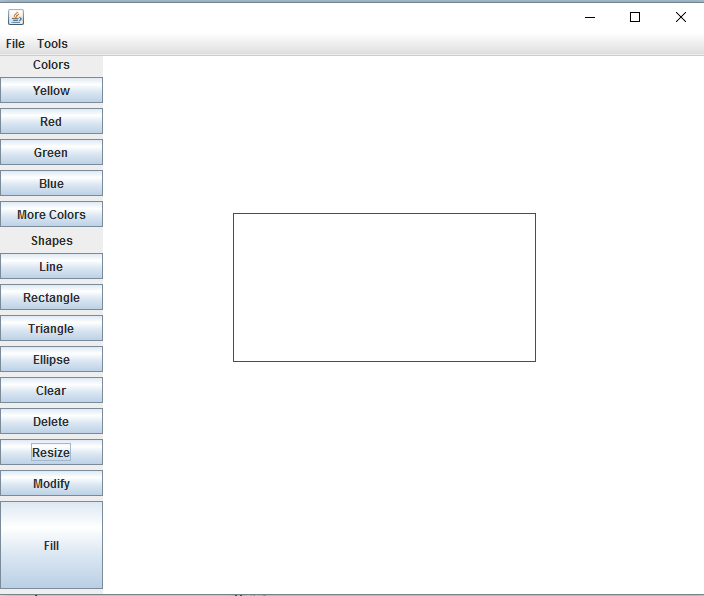
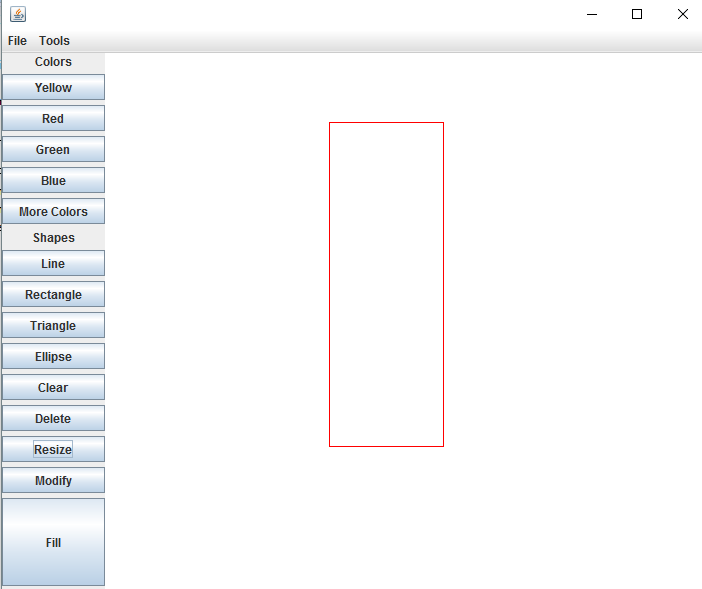
* If the user wants to draw any shape he should first set the start point or main point to start the drawing, then he drags the shape he wants to draw after that he release the mouse then he will see his shape as he wants.
* There is a special case about the triangle because the shapes as circle, rectangle, ellipse, … can be drawn using at least 2 points so we could estimate the characteristics of the shape the user wants by know only 2 points but the triangle we have to know the three point of its head in order to draw it. So we have assumed that if the user wants to draw triangle he should start with line segment to be an edge in the triangle he wants after that he release the mouse and start drawing another line which its start point is the end point of the first line segment, however after he release the mouse for the second time then the three points of the triangle become clear to us so the application is known able to draw the triangle.
* In order to move any shape, you should choose any point on it and start dragging it from a place to another.
* To resize the shape, he should first click on the resize/modify button so he enables the resizing mode then click on the shape and start dragging it until he gets the size he wants.
* Because we believe that undo and redo could save some times we have made a menu button with key accelerator such as Ctrl + Z for undo and Ctrl + Y for redo which could make it much more easily for the user.
* In save and load operations we enabled the user to save as XML or JSON so he could get his drawing back to redraw it or edit it.
* Because we believe that the drawing mustn’t be black, we added a color selector so the user could select the color he wants to make the drawing more likable for the user.
* The program starts without plug-ins each time the user opens it. Because we believe that the user always wants to see the main shapes he wants and but if he wants a specific shape he could load it and start drawing.

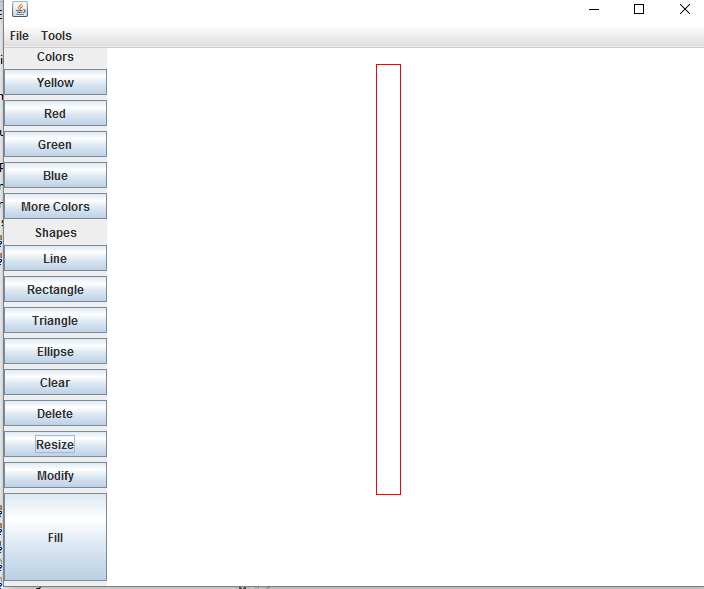
Screen shots:

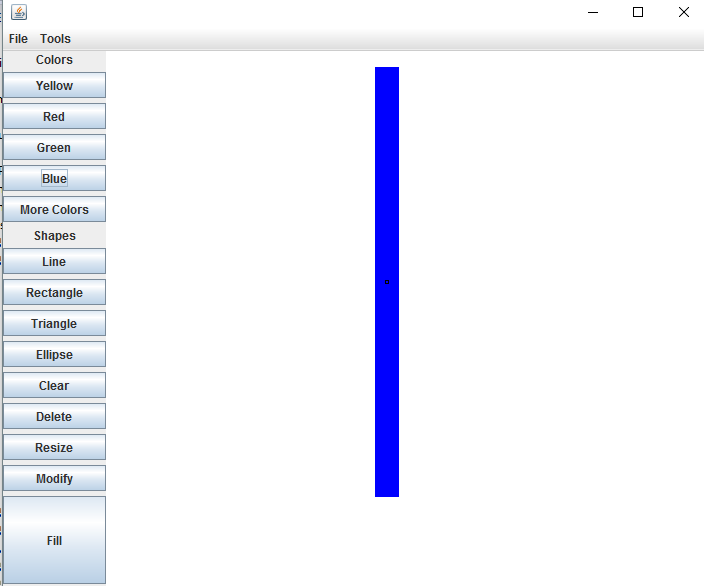
Drawing Shapes:

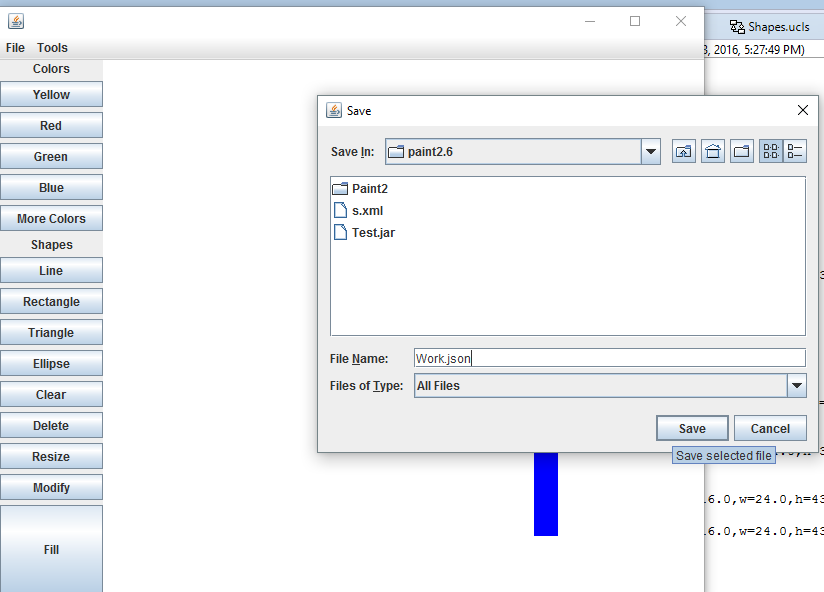


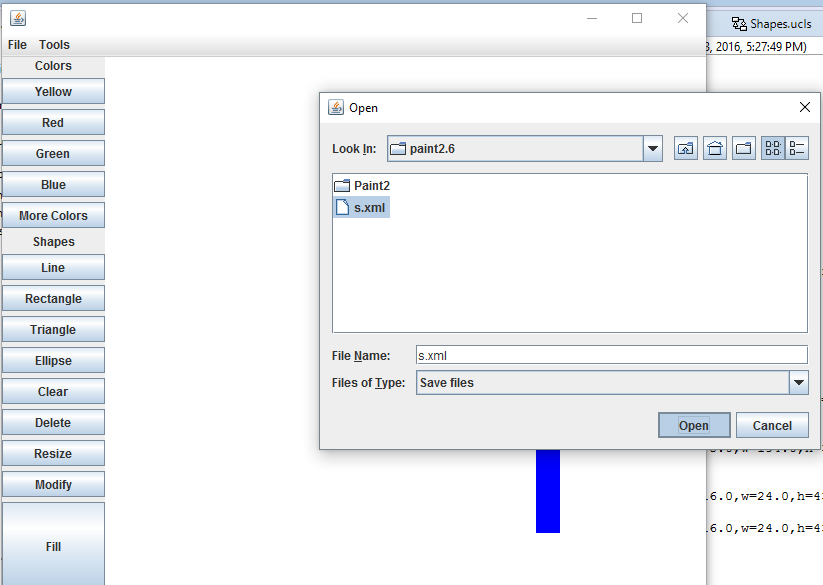
Moving Shapes.

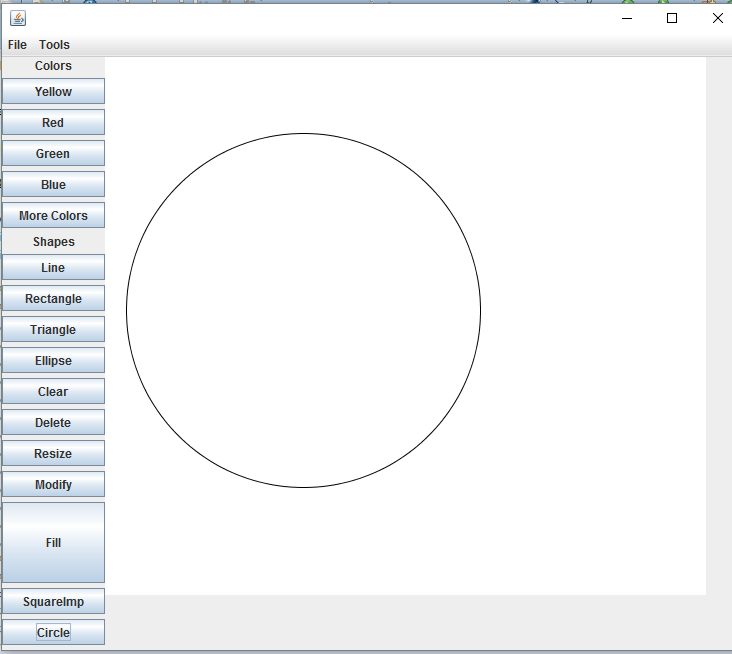
Resizing Shapes:



Fill with Color:

Saving and loading:



Dynamic Loading:

