

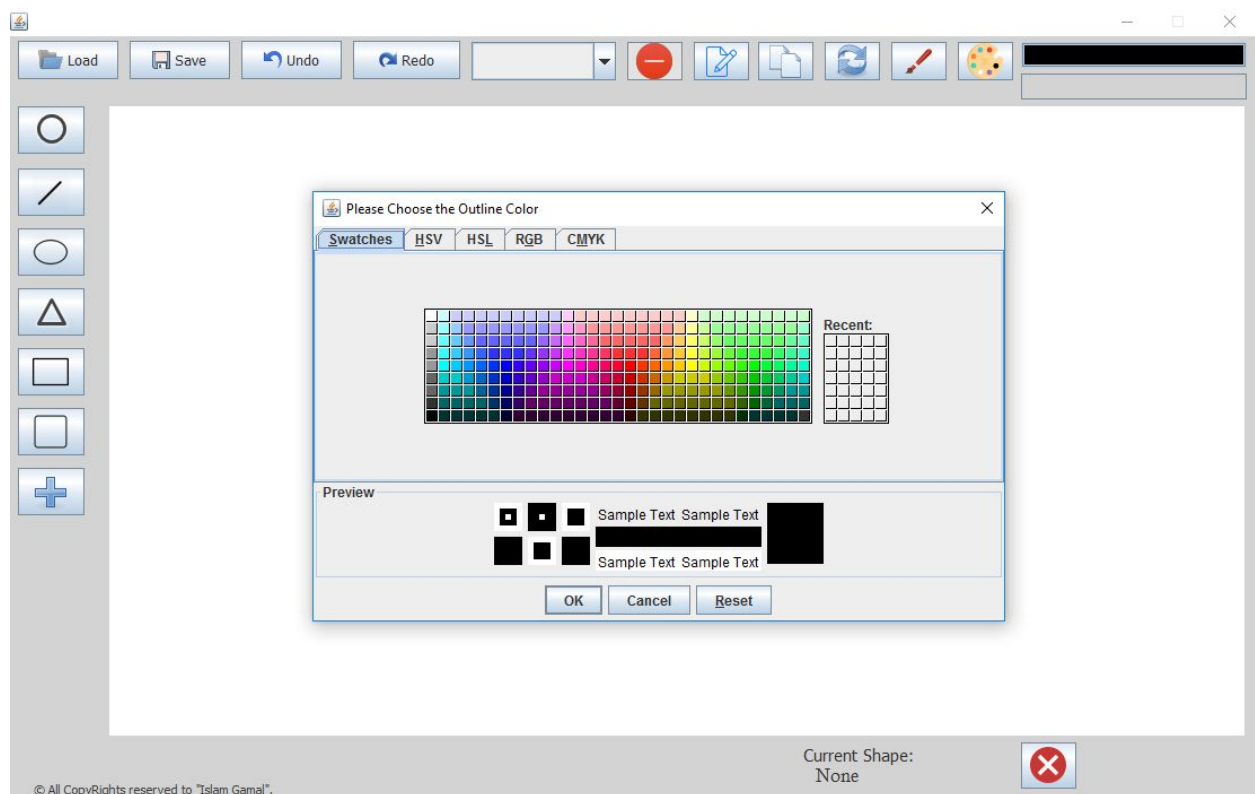
Paint

(oop)

Your Name

Aya Ashraf Saber 02

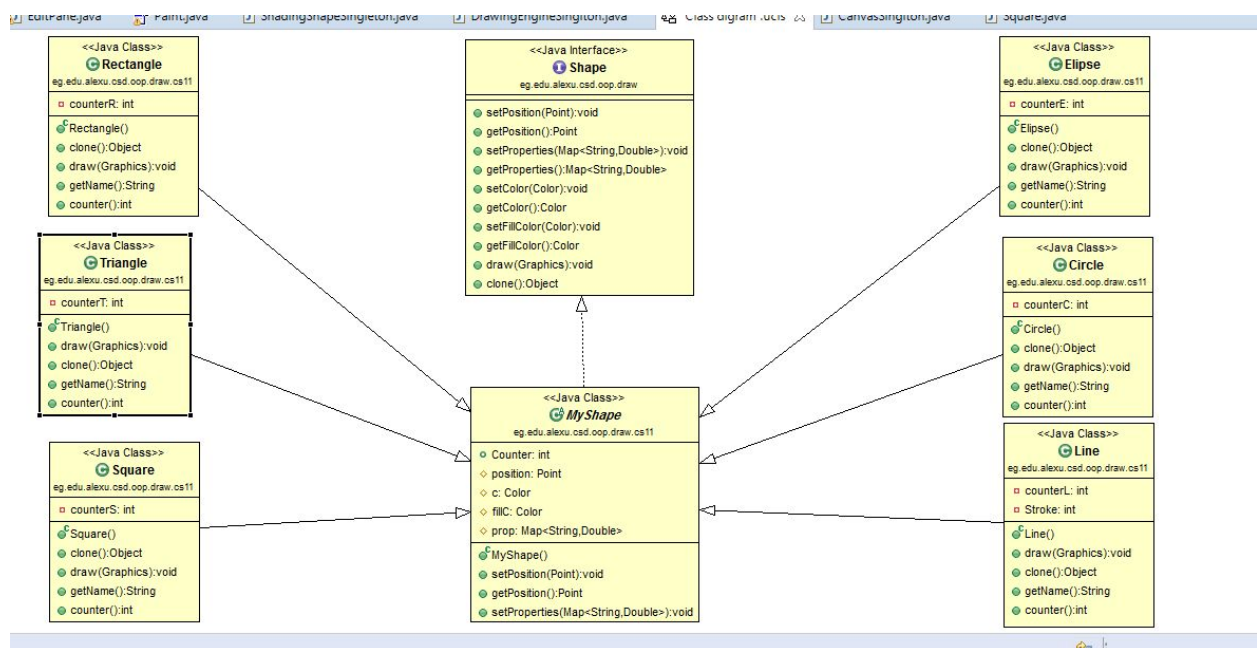
Islam Gamal 11



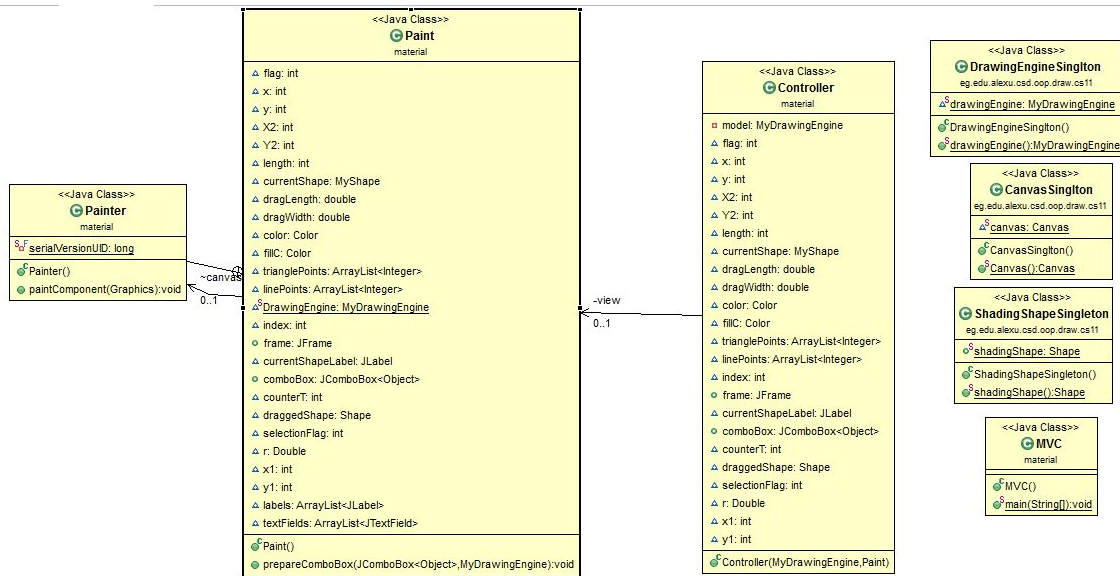
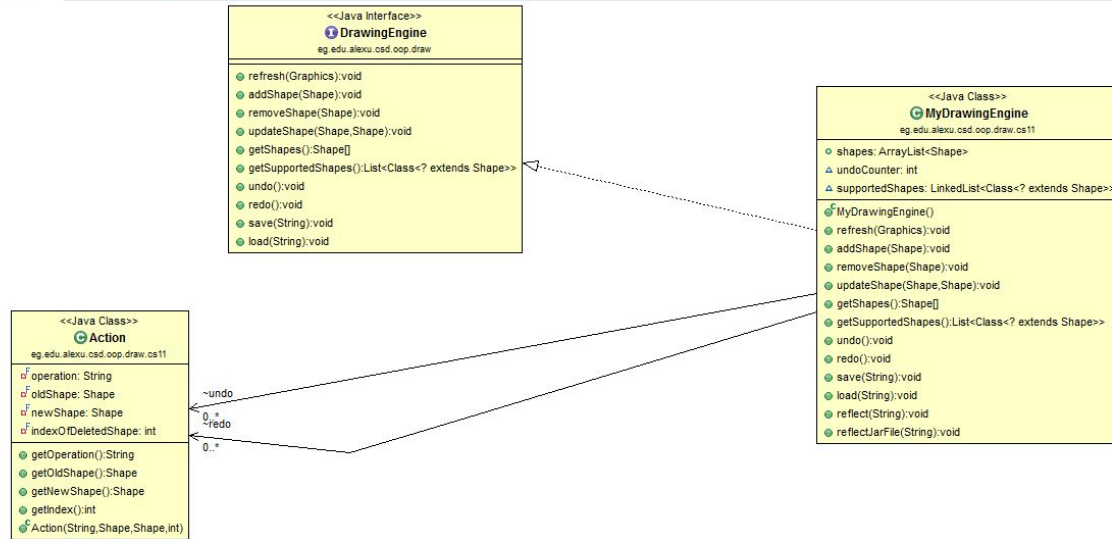
Overview

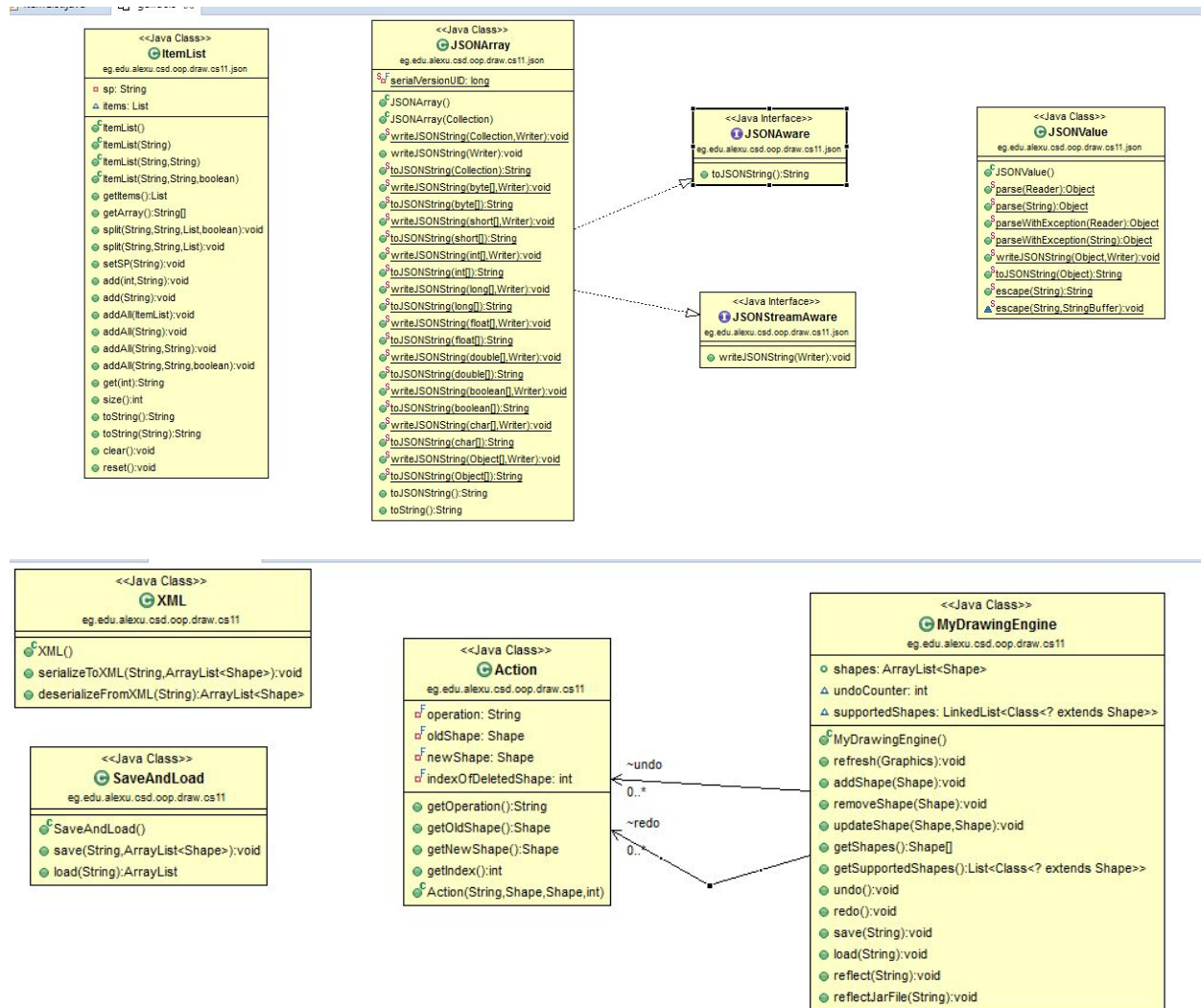
Drawing and painting applications are very popular and have a huge user base; they generally offer a big number of features that includes but is not limited to: Drawing, Coloring, and Resizing. They also include several built in, and possibly extensible set of geometric shapes, and classically, they allow the user to undo or redo any instructions to make the application more usable.

UML Class Diagram



eg. drawingengine.ucis 25 Paint.java 27 Action.java





Interface class:

1. Refresh :
 - In this Function we loop in all elements in my arrayList to draw all element save on it on our canvas
2. Add Shape :
 - In this function we take the new shape from Gui and added In the Array List of shapes and update undo stack by add operation and the new shape and clear redo stack from any action.
3. Remove Shape :
 - By calling this method we remove the specific shape from my shapes list and update the undo stack by the removing shape and its position in the List and clear redo stack from any action.
4. Update Shape :
 - This get the position of old shape in the list and put the new one in specific place,
 - After that we update the undo Stack with the old shape and its position.
5. Save and Load ;
 - We save use to ways user can save xml file or Jason file :
 - XML Class contains two functions : "using java beans encoder /Decoder"
 - Jason class save file. Jason using external library.
6. Undo and redo;
 - We using two stacks one for undo and other to redo we this stack of class Action
 - Class Action hold : - Old Shape , Current Shape , Deleted Shape and its position and I Last operation.
 - I have three functions effect on undoStack {Add New Shape, Remove Shape, Update shape }
 - The redo stack will be effect when I call function undo.
 - When we have new operation we clear the redo stack from any actions.
 - We have 20 as maximum number of undo when I reached it I remove the first element of the stack;
7. Get Shapes:
 - a. Return my array list of shapes.

8. Get Supported Shapes:

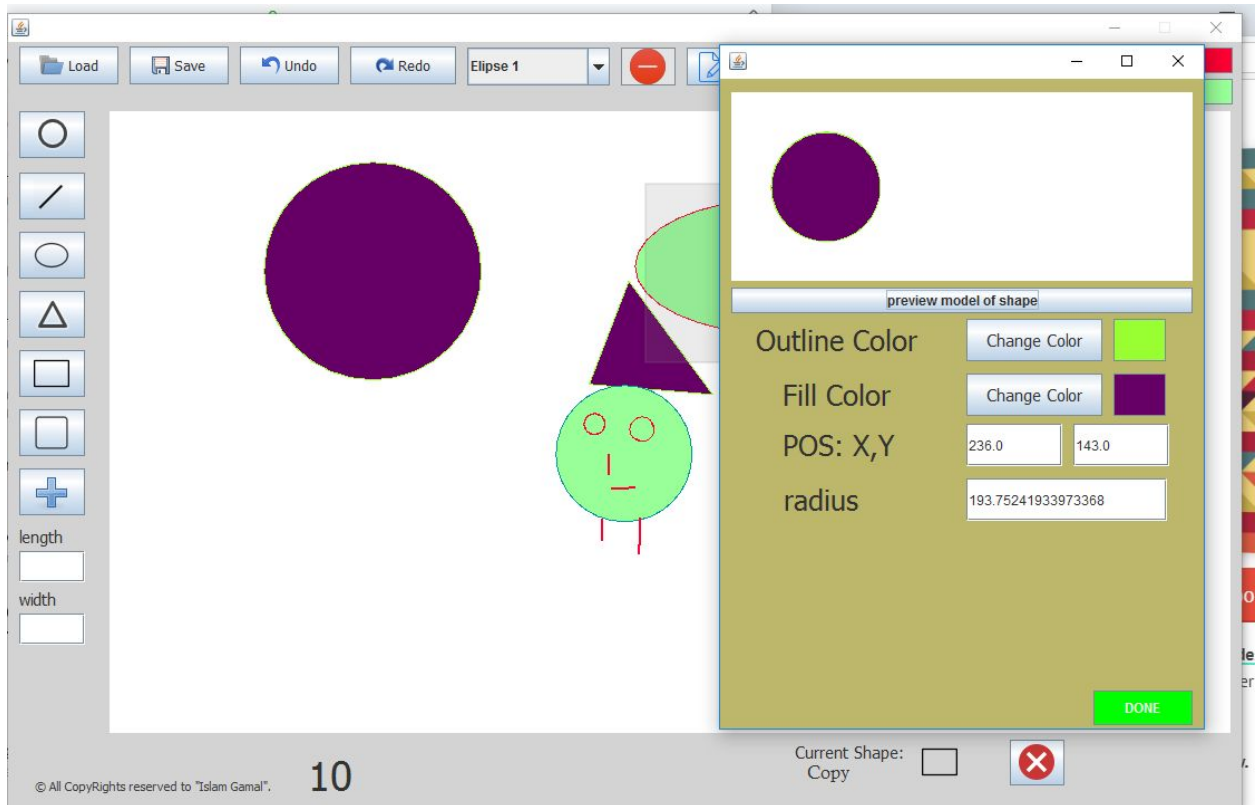
- We add all class in which can be drawn in my gui by default we have 6 classes but the user can add new jar contain class in my program in the run time I have a array list of supported shape and we return it after calling function.

9. My Shape:

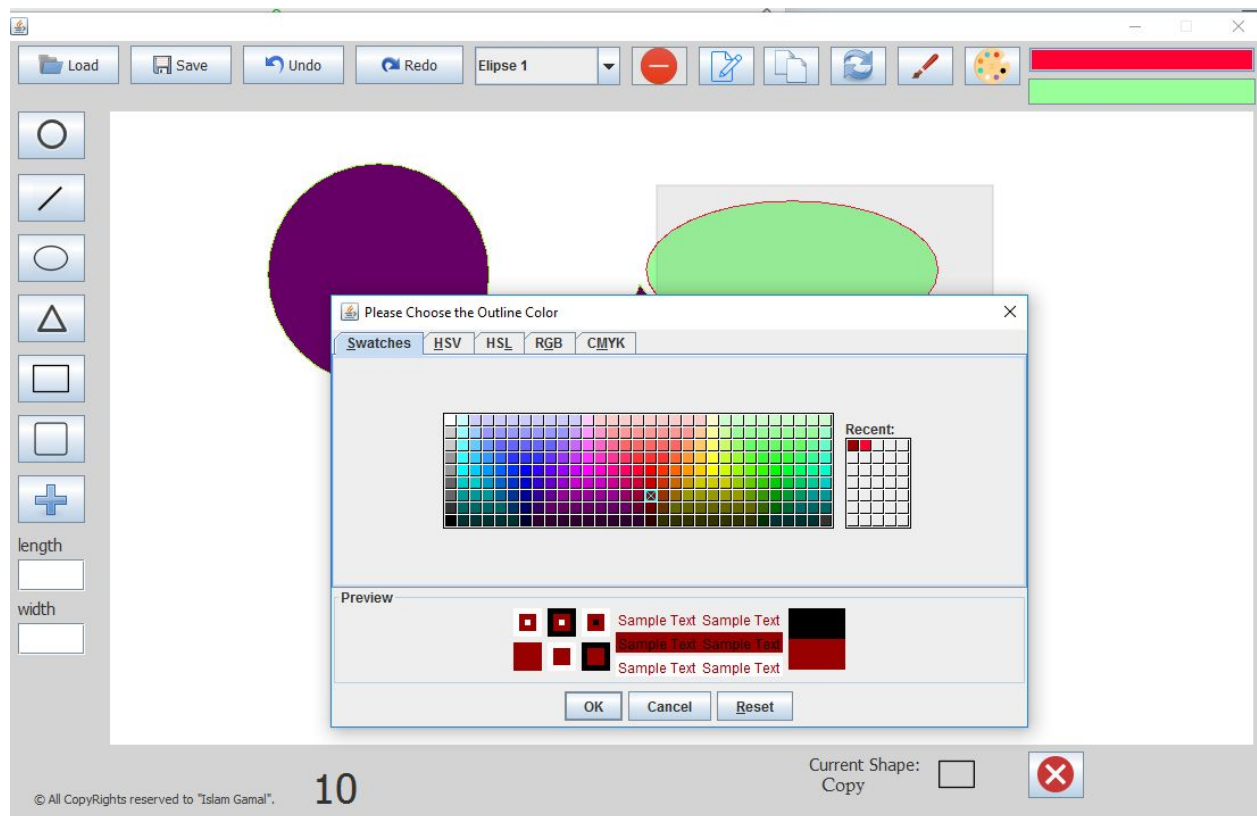
- Every shape in my program extend abstract class My shape which inherit from the Shape interface.
- Every shape has its own constructor which contain its properties which is different from one shape to another.
- Every shape implements its own draw method which use java 2D graphics.
- Every Shape implements its own clone method.
- And the common method between the shapes are implemented in MyShape class Like Set and get color , set and get position , set and get hash map of properties and set and get position of the first point of the shape in canvas.

Sample Run

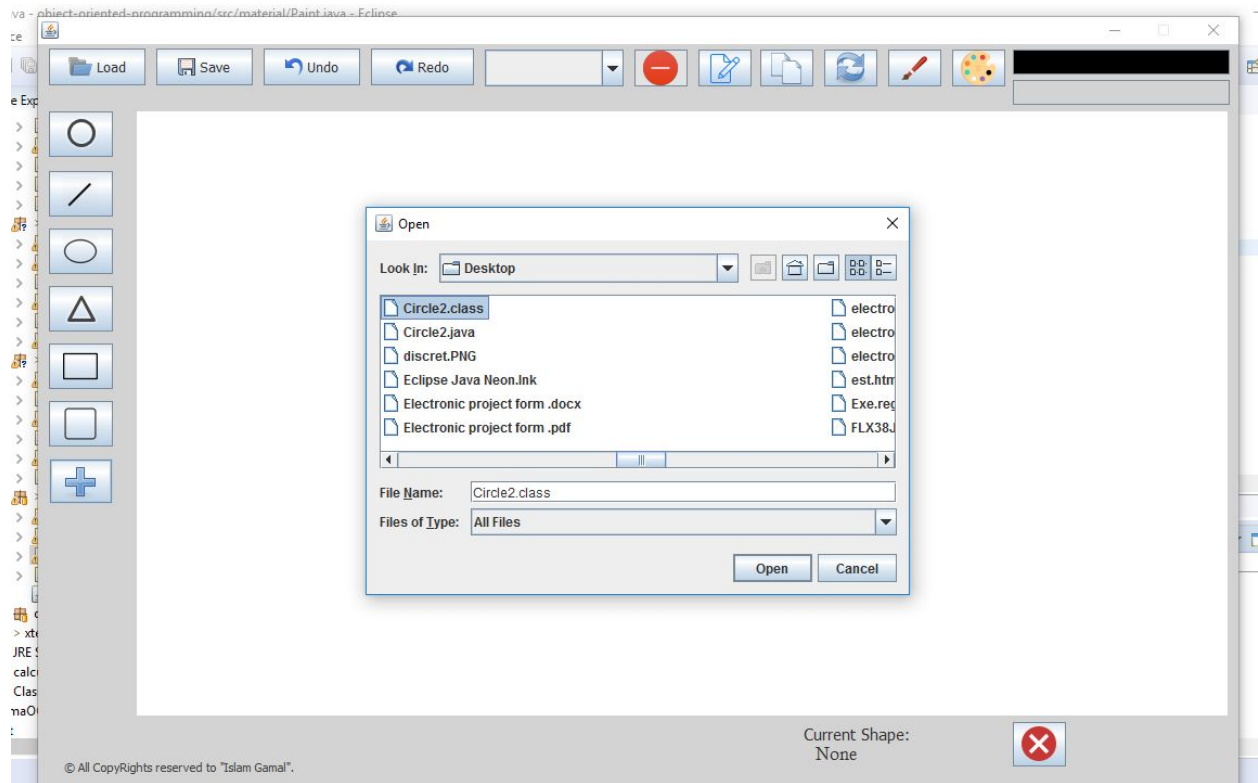
I. Update Shape:



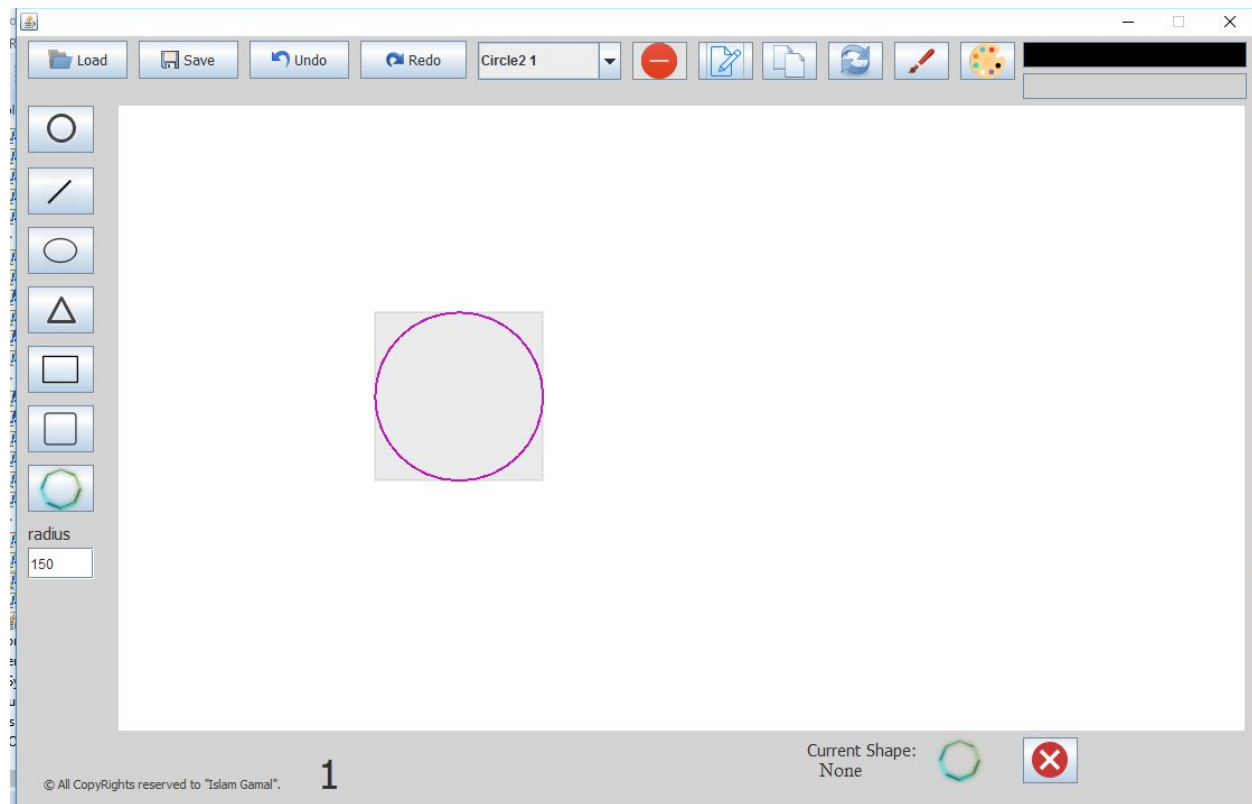
II. Set outLine color :



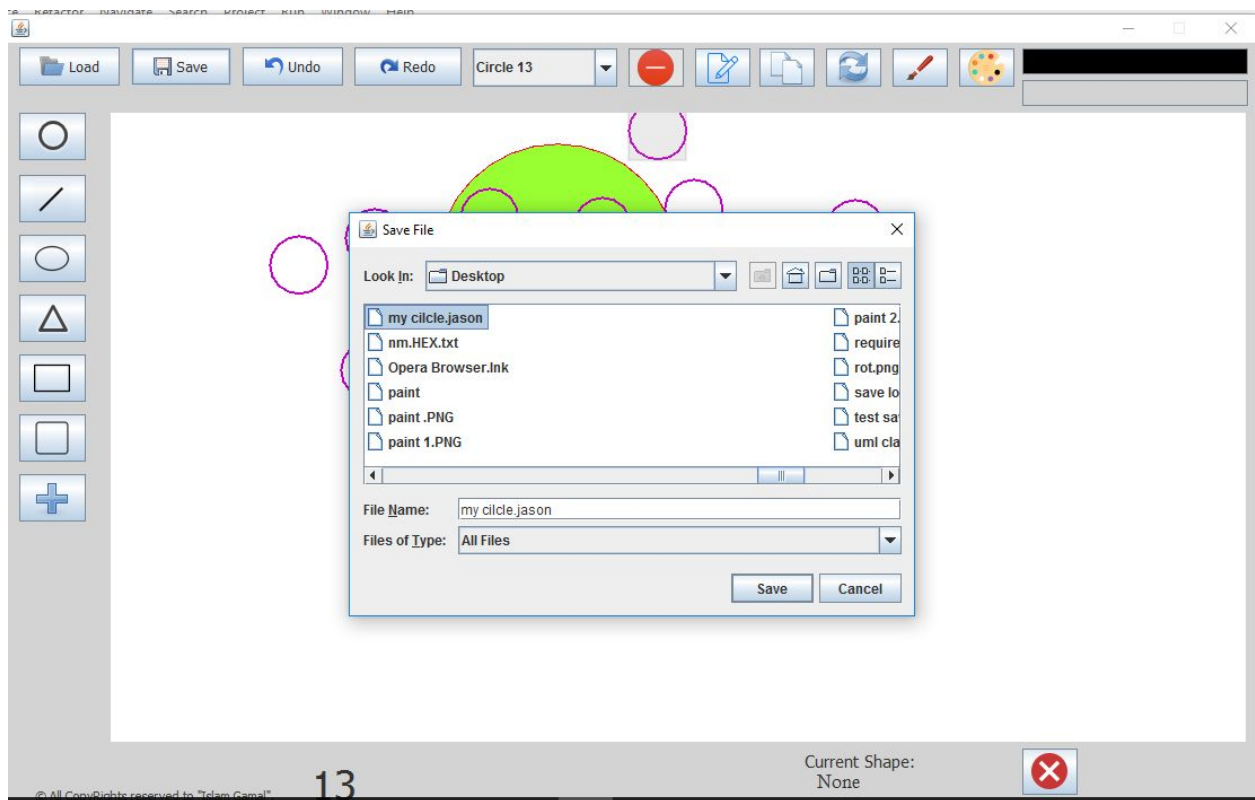
III. Dynamic load class :



IV. After Dynamic Loading :



V. Saving Paint :



VI. User Guide :

