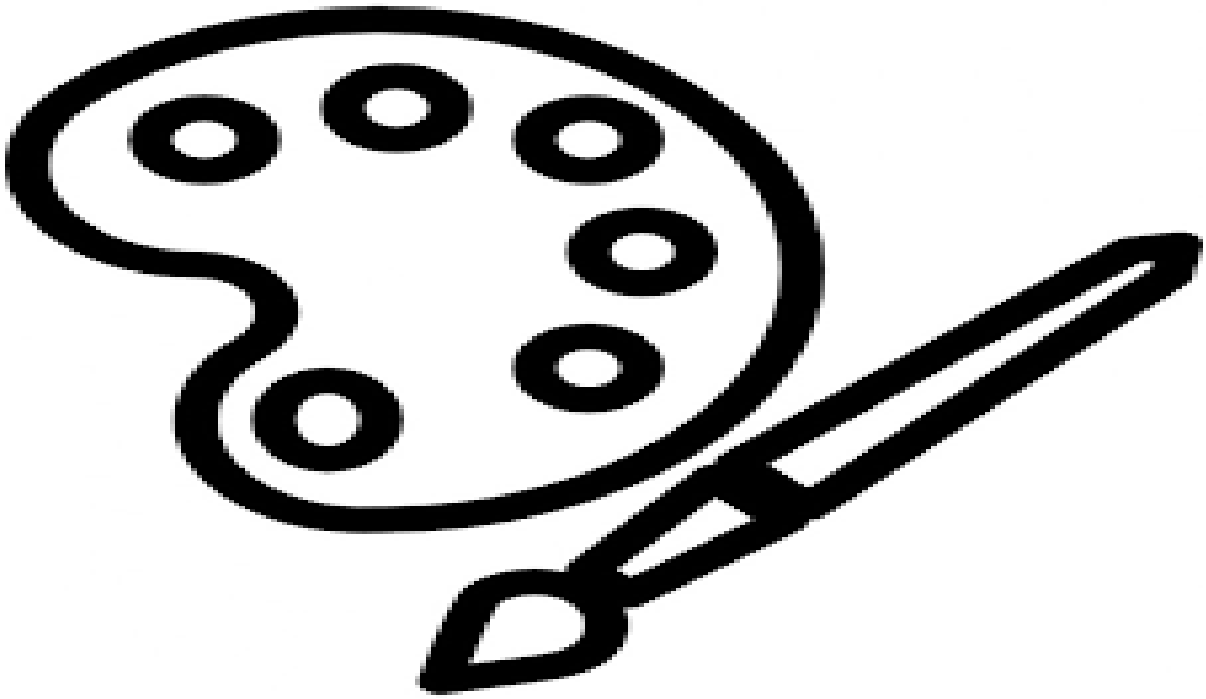
Assignment 2 : Vector drawing



Names :

Ahmed Ezzat ElMaghawry (11)

Marwan Morsy Mohamed Morsy (72)

1. Problem Statement :

Part 1: Geometric Shapes Data Model Description:

Geometric shapes belong to different groups (ex: Elliptical Shapes, Polygons, …etc). Members of these different groups are related to each other in the sense that they share common properties. In order to be able to implement an efficient and object oriented drawing application. It is essential to design a model that takes these relations into consideration. Tasks:

1- Design an object oriented model that covers the following geometric shapes: Line Segment, Circle, Ellipse, Triangle, Rectangle and Square.

2- Draw a UML Class diagram that represents your model, showing all the classes, attributes and methods.

3- Apply the concepts of inheritance and polymorphism to your design.

Part 2: Drawing and Painting Application Description:

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 a number of built in, and possibly extensible set of geometric shapes, and classically, they allow the user to undo or redo any instructions so as to make the application more usable Tasks:

1- Implement your design from part 1 in Java.

2- Design and implement a GUI that allows the following functionalities for the user on all the shapes defined in part 1: Draw Color, Resize, Move, and Delete.

3- Implement your application such that it would allow the user to undo or redo any action performed. (Optional hint: Check the "Command Design Pattern")

4- The cursor should be used to select the location of a shape while drawing it, or moving it to another location, for more accurate control on the shape parameters (ex: size), dialog boxes could be used, or you are free to implement it in a more user friendly way of your choice.

Part 3: Dynamic Application Extensions and Plug-ins

Description: The concept of dynamic class loading is widely spread in computer applications. It is an option that allows the user to extend application features at runtime. This can be easily done by the dynamic class loading capabilities that Java offers. Tasks:

1- Pick one of the Shape Classes listed in part 1, and compile it as a class library.

2- Provide an option in the GUI of your application that allows for selecting the class library file.

3- On selecting and loading the file, the isolated shape should be appended to the available list of shapes in the application.

Part 5: Move, Delete and Resize Description:

A very important feature to add is editing the painting, either by moving, deleting or resizing any object. Tasks:

1- Allow the user to select an object by clicking on it, and then start to resize, move or delete.

2- When an object is selected, its appearance should change such that the user understands that it is selected. This can be

done by a dotted rectangle that shows up around it or by showing boxes for resizing around it.

3- While resizing, show small boxes that the user can drag in order to resize the shape.

4- [bonus] Allow the user to group objects to resize, move or delete together as one object

Part 4: Save and load Description:

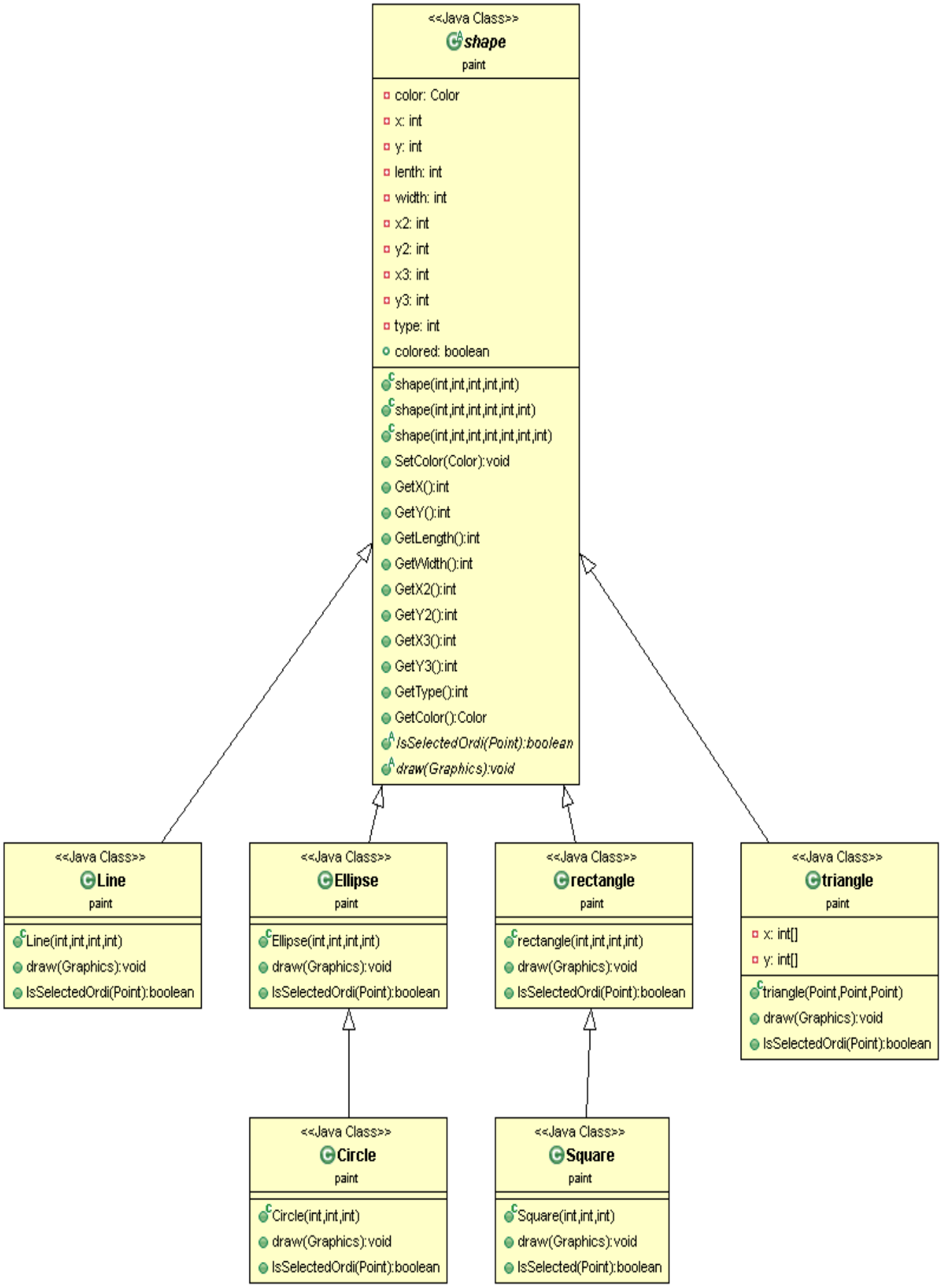
One of the main features in any paint application is saving user's drawings in a file and modifying it later. Tasks:

5- Provide an option in GUI to save the drawing in XML and JSON file (You should implement both).

6- Provide an option to load previously saved drawings and modify the shapes.

7- User must choose where to save the file.

2. UML Design :


<<Java Class>>
shape
paint
color: Color
x: int
y: int
lenth: int
width: int
x2: int
y2: int
x3: int
y3: int
type: int
colored: boolean
shape(int,int,int,int,int)
shape(int,int,int,int,int,int)
shape(int,int,int,int,int,int,int)
SetColor(Color):void
GetX():int
GetY():int
GetLength():int
GetWidth():int
GetX2():int
GetY2():int
GetX3():int
GetY3():int
GetType():int
GetColor():Color
IsSelectedOrdi(Point):boolean
draw(Graphics):void
<<Java Class>>
Line
paint
Line(int,int,int,int)
draw(Graphics):void
IsSelectedOrdi(Point):boolean
<<Java Class>>
Ellipse
paint
Ellipse(int,int,int,int)
draw(Graphics):void
IsSelectedOrdi(Point):boolean
<<Java Class>>
rectangle
paint
rectangle(int,int,int,int)
draw(Graphics):void
IsSelectedOrdi(Point):boolean
<<Java Class>>
triangle
paint
x: int[]
y: int[]
triangle(Point,Point,Point)
draw(Graphics):void
IsSelectedOrdi(Point):boolean
<<Java Class>>
Circle
paint
Circle(int,int,int)
draw(Graphics):void
IsSelectedOrdi(Point):boolean
<<Java Class>>
Square
paint
Square(int,int,int)
draw(Graphics):void
IsSelected(Point):boolean

3. design thoroughly:

The Project is Designed to be very Obvious for all Kind of people at any age , The User interface is divided to 3 parts :

1- the Left Toolbar :

which contains the Shapes which the user can draw in the panel.

Note : On opening the program the shapes which allowed for the user are the rectangle and the ellipse only.

Rectangle

Line

Ellipse

Circle

Triangle

Square

Color chooser

- The user should select firstly the shape which he want to color

- Drawing Shapes depends on dragging by mouse in the panel

2- Right Toolbar :

Which contains the other Operations which the user can do on the shapes .

Save

Load

Undo

Redo

Delete

Plugins

Select, Move, Resize

- When save button clicked there is a dialog will obtain to choose the directory which the user want to save the file in , After choosing the directory and press save then another dialog box will obtain to choose the type of the file (XML , JSON) .
- When load button clicked there is a dialog will obtain to choose the directory which the user want to load the file from.
- When the file loaded then all the previous operations will be cancelled (There is no undo or redo) as the Windows paint.
- Delete will be done when Select the shape first.
- When Plugin button clicked there is a dialog will obtain to choose the directory which the plugin is in (There is a file with the program called Additions contains all jars of the plugins)

- When Select button pressed , the mouse will have the capability to select the shape the user want by clicking on it then when the user click on the shape the shape will selected by drawing (4 small rectangles for rectangle , square, Ellipse and Circle or 3 small rectangles for Line and triangle) to move (by the central rectangle or resize by other rectangles .

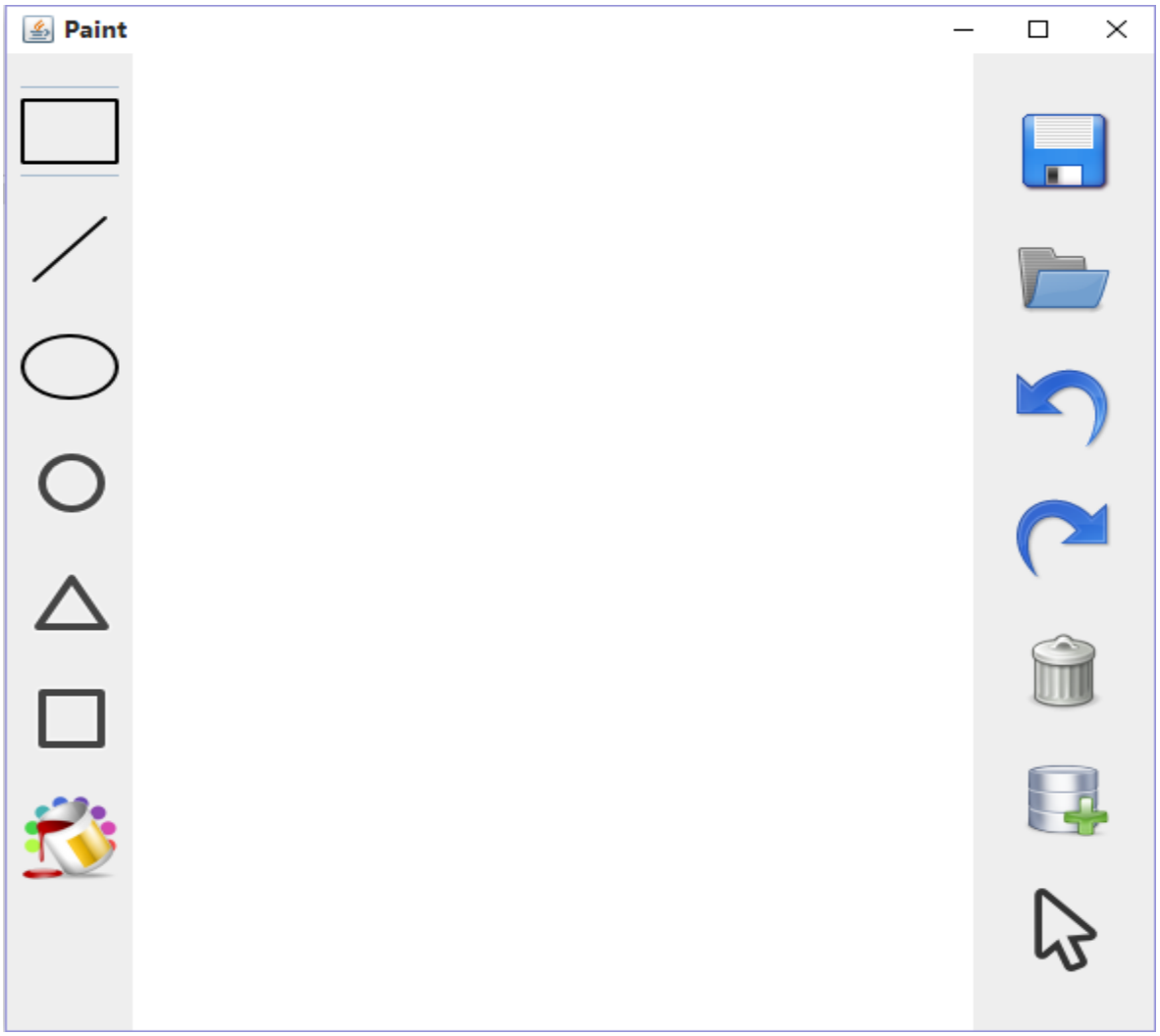
3- Central Panel :

The Space which the user can work in .

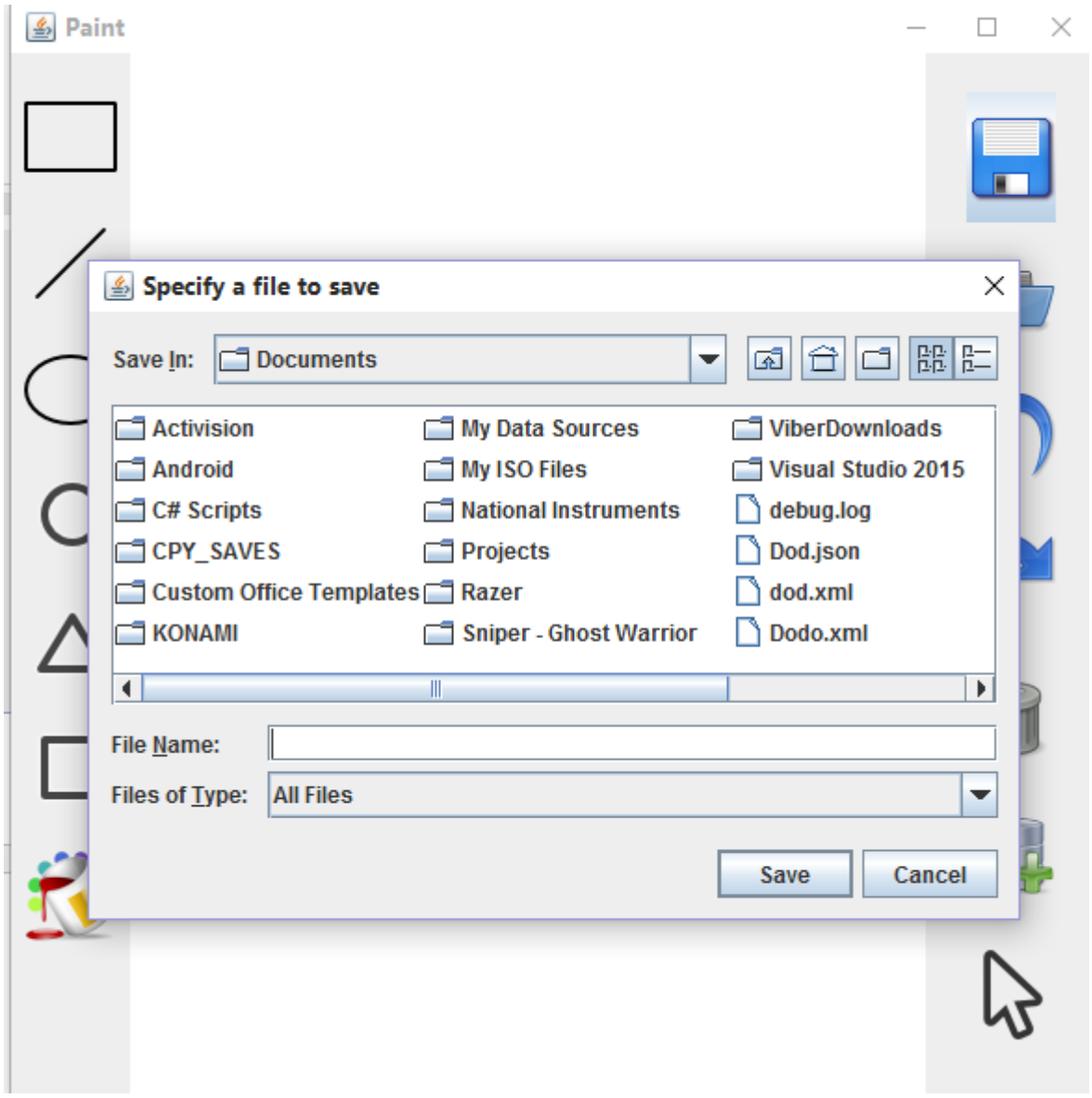
Note: The frame is resizable So the user can maximize the frame , the tools will resize with .

4. Snapshots :

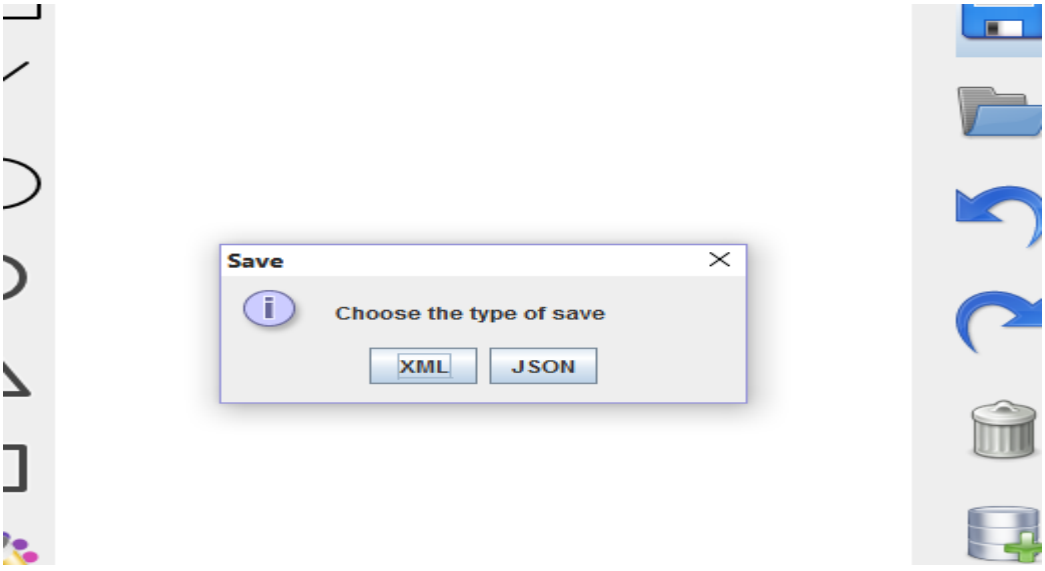
- The Gui :



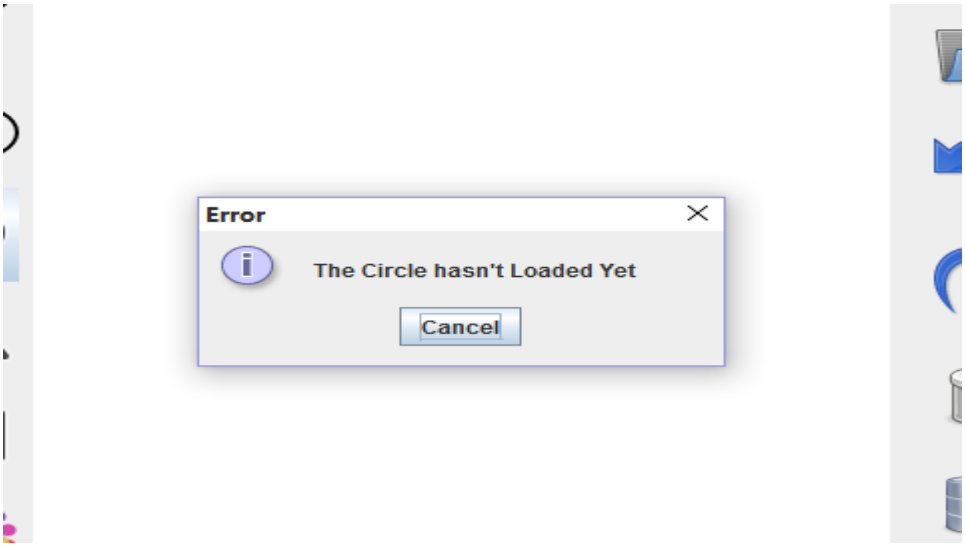
- File choser for save , load , Plugin :



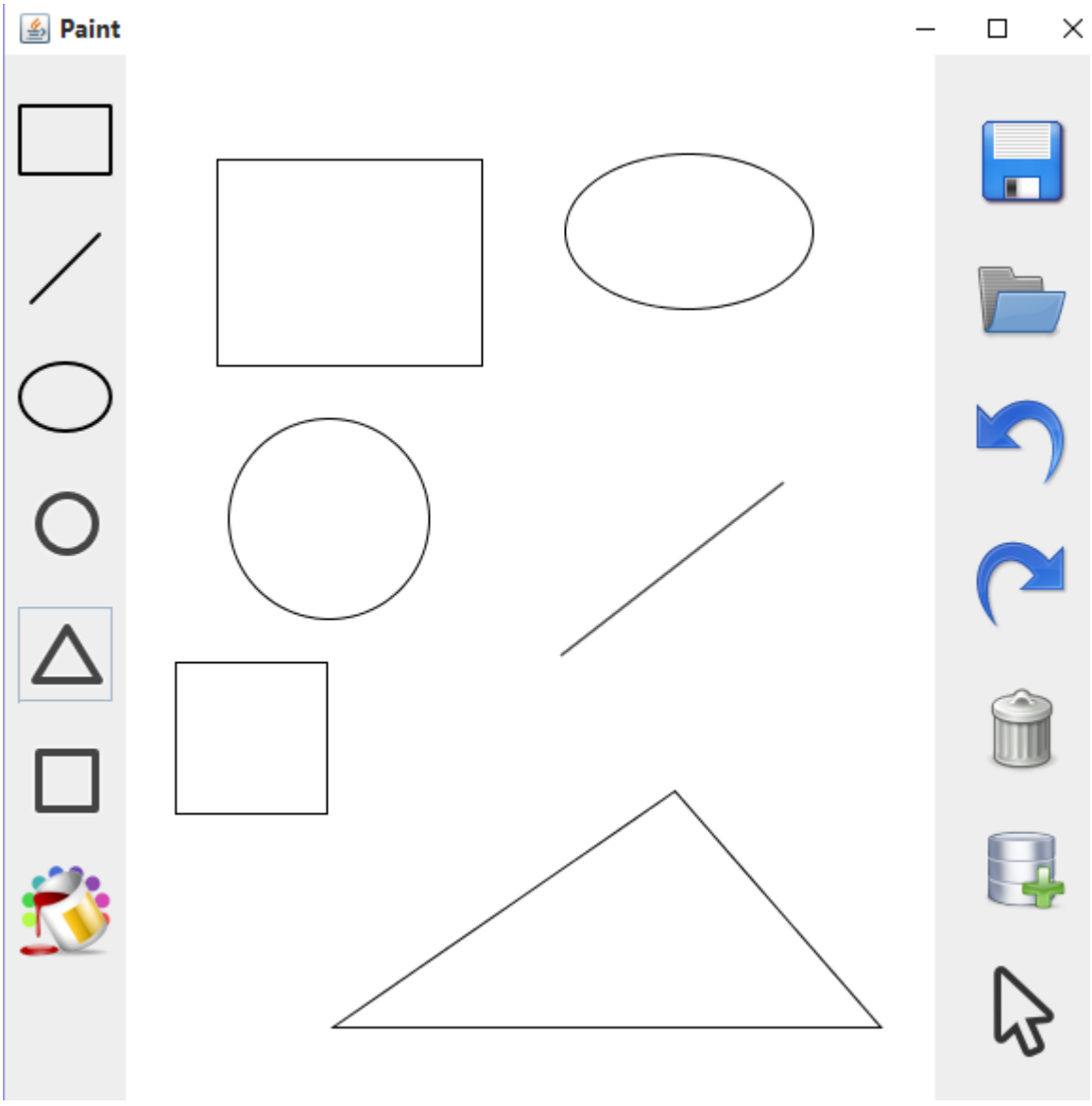
- The Save type :



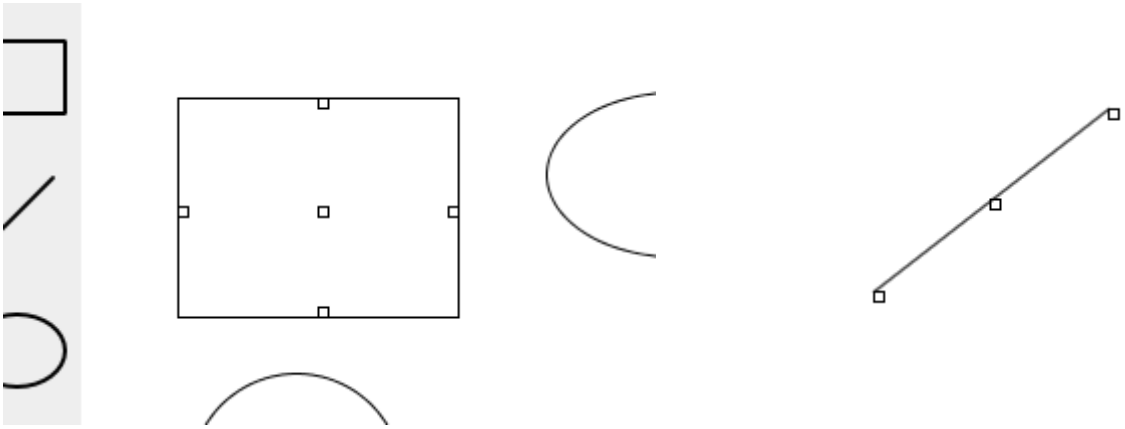
- The dialog which obtain when clicking on Not Existence shape :



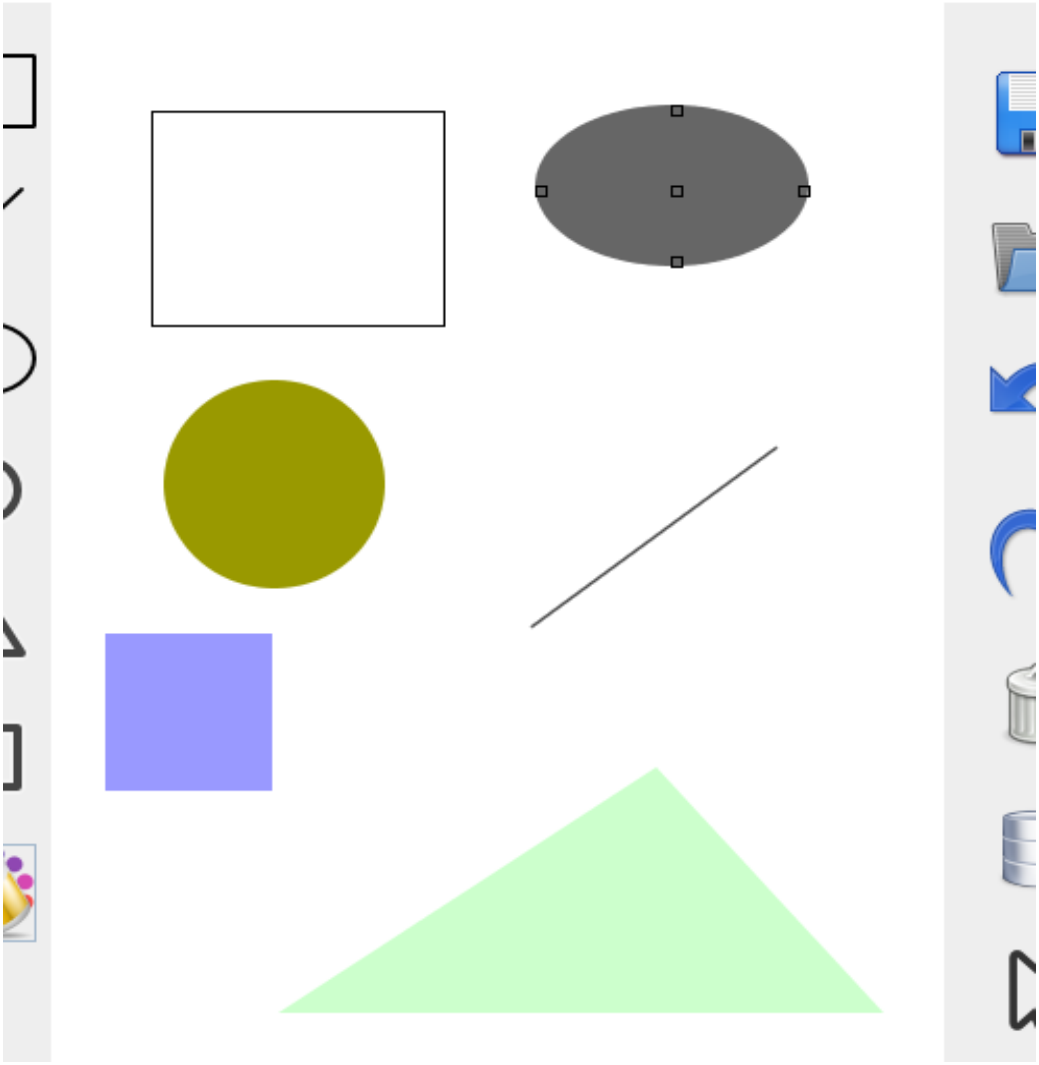
- Shapes :



- Selected Shape :



- Colored Shape :



5. Design Dictions :

- The Program have a Custom Panel which we have made to serve our design , Because we need some functions from the panel to work as we need when we need to make the gui more easier .

- Custom Action List which used for undo and redo where the undo and redo are implemented using stacks . Every action the user make is stored in actionDone object which consist of the shape before the action and the shape after the action and it's index in shapes array list . Each action is added into the undo Stack .when the user clicks undo the old shape appears and push the new shape into the redo stack .

- Save and Load Classes which saves the shapes and handle if the shape is null or not to save it in the file (null Because of the undo and redo) and load the shapes at the load class Note that we only used one external library to save Jason , Another Note : we save Jason as a format of Jason .

- ClassLoader : The program load the external class using a jar files of these classes after that Detect which class is chosen by its name then identify its constructor which initialized as null , the program knows if the shape is loaded or not using its constructor if the constructor is null then it hasn't selected yet else then the shape is loaded and can use it .

- Resizing function : It is a function which will call when the user press Select button then select the shape ,the program will define the shape type (Using GetType) which defined in the in the Super class (Shape) but there is a problem there is only 4 types in the super class which means there is two types are lost (type 1 for Rectangle and Square , type 2 for Ellipse and Circle , type 3 for Line and type 4 for triangle) the program could beat this issue by the properties of the shapes (the square its width equals its length as same as the circle) so the program could define the shapes correctly after that the function takes the shape and the type of the shape then do operations on it to Put the small rectangles which the user can resize and move from it .

- In every class of the shapes there is function (IsSellected(point)) which takes a point and return true or false , Contains the equation of each Shape to identify if the shape selected or not .

- In the Super Class there is 3 Constructor of the same name but different number of parameters which serves the principle of polymorphism

6 . Code Process :

- Firstly we used 6 Array list of 6 different types of the shapes to draw in the panel , it was easy to do and more easier to save and load and delete , but using class loader change all the code , the program have one arraylist of shapes which contains all the shapes in the panel and the program could handle the shapes using its type in the super class.

- At save and load firstly we used one ArrayList of ArrayList of Shapes , and in it the programe saves arraylists of each state (Every operation the user made will recorded in a new array list contains all the previous operations but will be a big load on the program , so we decided to use a stack of shapes but it was a little hard , finally we used two stacks to undo and redo of type action done (A custom class we made)