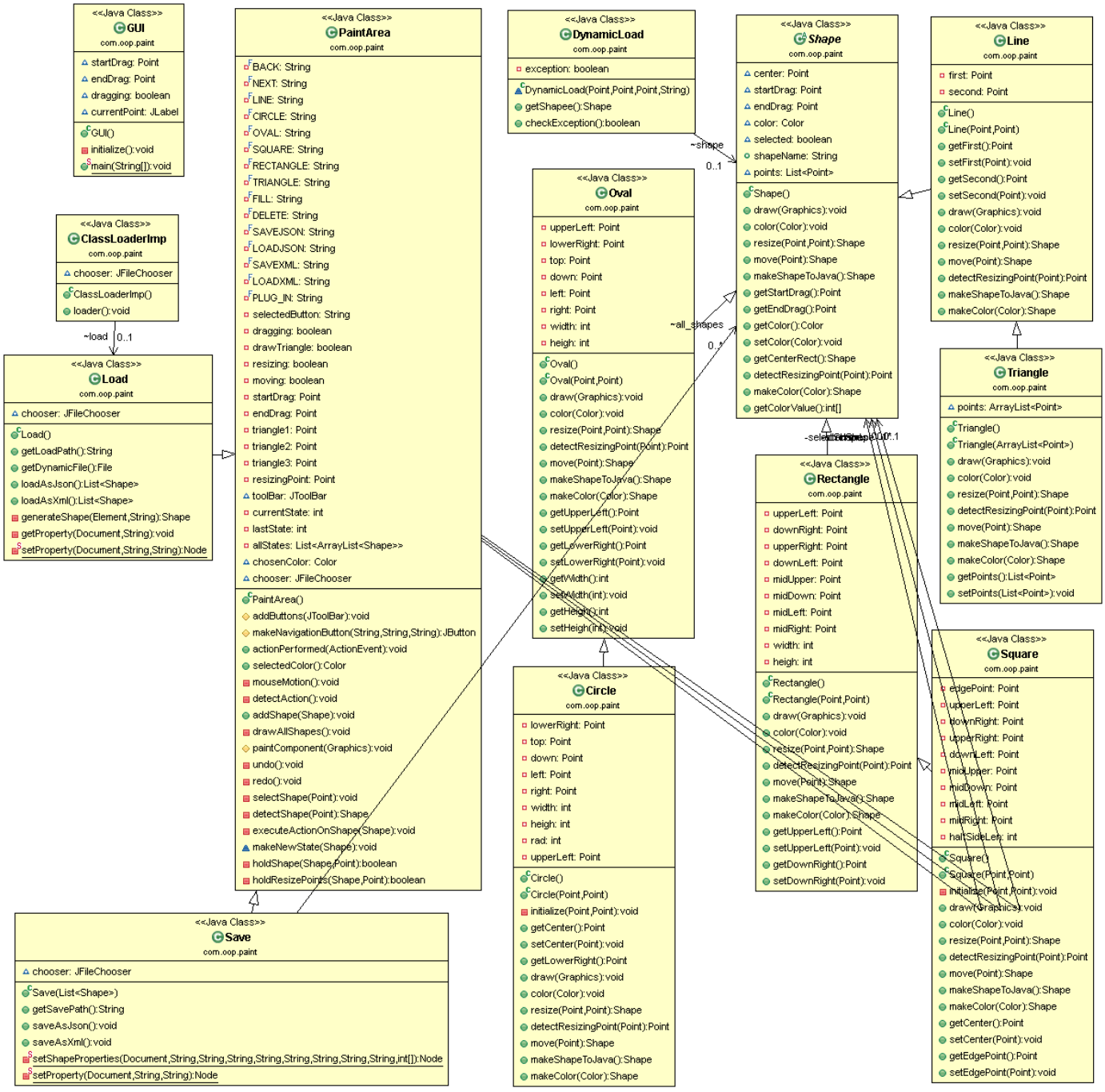
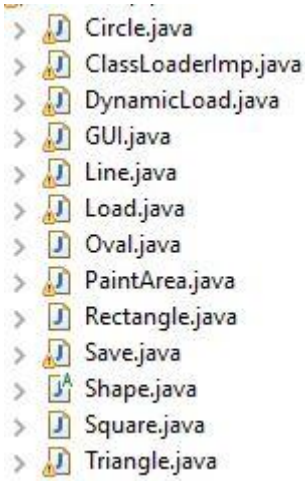
UML Diagram :


<<Java Class>>
GUI
com.oop.paint
startDrag: Point
endDrag: Point
dragging: boolean
currentPoint: JLabel
GUI()
initialize():void
main(String[]):void
<<Java Class>>
PaintArea
com.oop.paint
BACK: String
NEXT: String
LINE: String
CIRCLE: String
OVAL: String
SQUARE: String
RECTANGLE: String
TRIANGLE: String
FILL: String
DELETE: String
SAVEJSON: String
LOADJSON: String
SAVEXML: String
LOADXML: String
PLUG_IN: String
selectedButton: String
dragging: boolean
drawTriangle: boolean
resizing: boolean
moving: boolean
startDrag: Point
endDrag: Point
triangle1: Point
triangle2: Point
triangle3: Point
resizingPoint: Point
toolBar: JToolBar
currentState: int
lastState: int
allStates: List<ArrayList<Shape>>
chosenColor: Color
chooser: JFileChooser
PaintArea()
addButtons(JToolBar):void
makeNavigationButton(String,String,String):JButton
actionPerformed(ActionEvent):void
selectedColor():Color
mouseMotion():void
detectAction():void
addShape(Shape):void
drawAllShapes():void
paintComponent(Graphics):void
undo():void
redo():void
selectShape(Point):void
detectShape(Point):Shape
executeActionOnShape(Shape):void
makeNewState(Shape):void
holdShape(Shape,Point):boolean
holdResizePoints(Shape,Point):boolean
<<Java Class>>
DynamicLoad
com.oop.paint
exception: boolean
DynamicLoad(Point,Point,Point,String)
getShapee():Shape
checkException():boolean
~shape
0..1
<<Java Class>>
Shape
com.oop.paint
center: Point
startDrag: Point
endDrag: Point
color: Color
selected: boolean
shapeName: String
points: List<Point>
Shape()
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
move(Point):Shape
makeShapeToJava():Shape
getStartDrag():Point
getEndDrag():Point
getColor():Color
setColor(Color):void
getCenterRect():Shape
detectResizingPoint(Point):Point
makeColor(Color):Shape
getColorValue():int[]
<<Java Class>>
Line
com.oop.paint
first: Point
second: Point
Line()
Line(Point,Point)
getFirst():Point
setFirst(Point):void
getSecond():Point
setSecond(Point):void
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
move(Point):Shape
detectResizingPoint(Point):Point
makeShapeToJava():Shape
makeColor(Color):Shape
<<Java Class>>
Oval
com.oop.paint
upperLeft: Point
lowerRight: Point
top: Point
down: Point
left: Point
right: Point
width: int
heigh: int
Oval()
Oval(Point,Point)
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
detectResizingPoint(Point):Point
move(Point):Shape
makeShapeToJava():Shape
makeColor(Color):Shape
getUpperLeft():Point
setUpperLeft(Point):void
getLowerRight():Point
setLowerRight(Point):void
getWidth():int
setWidth(int):void
getHeigh():int
setHeigh(int):void
~all_shapes
0..*
<<Java Class>>
ClassLoaderImp
com.oop.paint
chooser: JFileChooser
ClassLoaderImp()
loader():void
~load
0..1
<<Java Class>>
Load
com.oop.paint
chooser: JFileChooser
Load()
getLoadPath():String
getDynamicFile():File
loadAsJson():List<Shape>
loadAsXml():List<Shape>
generateShape(Element,String):Shape
getProperty(Document,String):void
setProperty(Document,String,String):Node
<<Java Class>>
Triangle
com.oop.paint
points: ArrayList<Point>
Triangle()
Triangle(ArrayList<Point>)
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
detectResizingPoint(Point):Point
move(Point):Shape
makeShapeToJava():Shape
makeColor(Color):Shape
getPoints():List<Point>
setPoints(List<Point>):void
<<Java Class>>
Rectangle
com.oop.paint
upperLeft: Point
downRight: Point
upperRight: Point
downLeft: Point
midUpper: Point
midDown: Point
midLeft: Point
midRight: Point
width: int
heigh: int
Rectangle()
Rectangle(Point,Point)
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
detectResizingPoint(Point):Point
move(Point):Shape
makeShapeToJava():Shape
makeColor(Color):Shape
getUpperLeft():Point
setUpperLeft(Point):void
getDownRight():Point
setDownRight(Point):void
<<Java Class>>
Circle
com.oop.paint
lowerRight: Point
top: Point
down: Point
left: Point
right: Point
width: int
heigh: int
rad: int
upperLeft: Point
Circle()
Circle(Point,Point)
initialize(Point,Point):void
getCenter():Point
setCenter(Point):void
getLowerRight():Point
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
detectResizingPoint(Point):Point
move(Point):Shape
makeShapeToJava():Shape
makeColor(Color):Shape
<<Java Class>>
Square
com.oop.paint
edgePoint: Point
upperLeft: Point
downRight: Point
upperRight: Point
downLeft: Point
midUpper: Point
midDown: Point
midLeft: Point
midRight: Point
halfSideLen: int
Square()
Square(Point,Point)
initialize(Point,Point):void
draw(Graphics):void
color(Color):void
resize(Point,Point):Shape
detectResizingPoint(Point):Point
move(Point):Shape
makeShapeToJava():Shape
makeColor(Color):Shape
getCenter():Point
setCenter(Point):void
getEdgePoint():Point
setEdgePoint(Point):void
<<Java Class>>
Save
com.oop.paint
chooser: JFileChooser
Save(List<Shape>)
getSavePath():String
saveAsJson():void
saveAsXml():void
setShapeProperties(Document,String,String,String,String,String,String,String,String,int[]):Node
setProperty(Document,String,String):Node

Design Description:

paint Design consists of 13 classes ; 6 for shapes and , 1 for Shape essential , 2 for save and load , 2 for Dynamic loading and 2 for essential GUI .



Shape class has many attributes and methods which all shapes extend from as [center : which is the centre of the shape , startdrag and enddrag which used in paint area to detect mouse actions points and take action on them].

Mainly PaintArea is where all actions occurs .
it contains a tool bar which has all the buttons [back and next , shapes , fill , delete , and save options ..]
all buttons added using the `addButtons` , which use `makeNavigationButton` to choose the icon of the button or show a text if the icons not found …etc
then return the button which add buttons put on the toolbar.
toolbar has many options which accessed using final keys , when the button selected , we take the necessary action on it in the method `actionPerformed`

Save and Load :
Save and Load using Json and XML done by its implementation
XML and Json Implementation go as all data have been saved in strings and gotten in strings also which is the essential formal method of saving data in json and xml.

Draw :
First we detect which shape the user wants to draw using the buttons then make him drag and release the shape after that we draw it using Graphics2D library in java which supports this operation after calculate the needed parameters to draw it .

Vector Drawing

Color :
To color a Shape we make the user select the shape then chose which color he wants to draw it by using color chooser in java then we use fill function to color the shape with the chosen color .

move:
To move a Shape we make the user select the shape then we make the corner points of the selected shape visible to show the shape to the user then he can drag the center point and move the shape to another location using dragging from center point .

Resize :
To resize a Shape we make the user select the shape then we make the corner points of the selected shape visible to show the shape to the user then he can drag from any point of these point to resize the shape .

Undo :
We save all the states of the program for example when we launch the application at first we are at state zero which means initial state and then draw a shape so our state is 0ne now and so on , when the user pressed the undo button we go one step back to previous state and draw it on the screen .

Redo :
We save the states by the same way in undo , when the user pressed the redo button we go one step forward to next state and draw it on the screen .

Dynamic Application Extensions and Plugins :
To add a plugin the user should clicked on the plugin button then he select the class file he wants to load then we change the button which is related with the selected class to be enabled to make the user able to draw this shape and use it

User Guide :

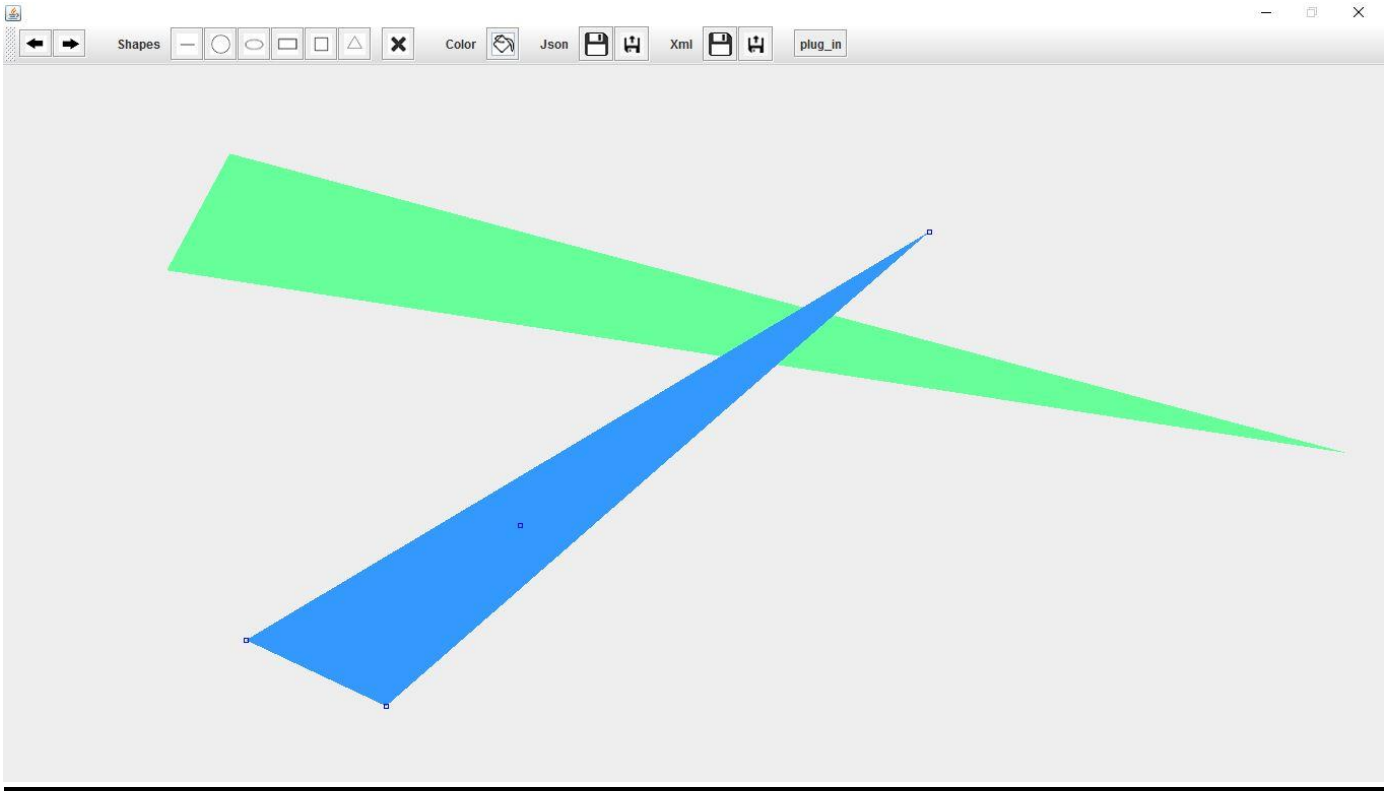
on opening the program user will get a window with tool bar containing all buttons he can use .
back and next button , Shapes to select a shape and draw it using clicking and dragging but triangle should drawn by clicking on the screen three times to select the locations of the three points .

delete button can be used agter selecting a shape and click on it

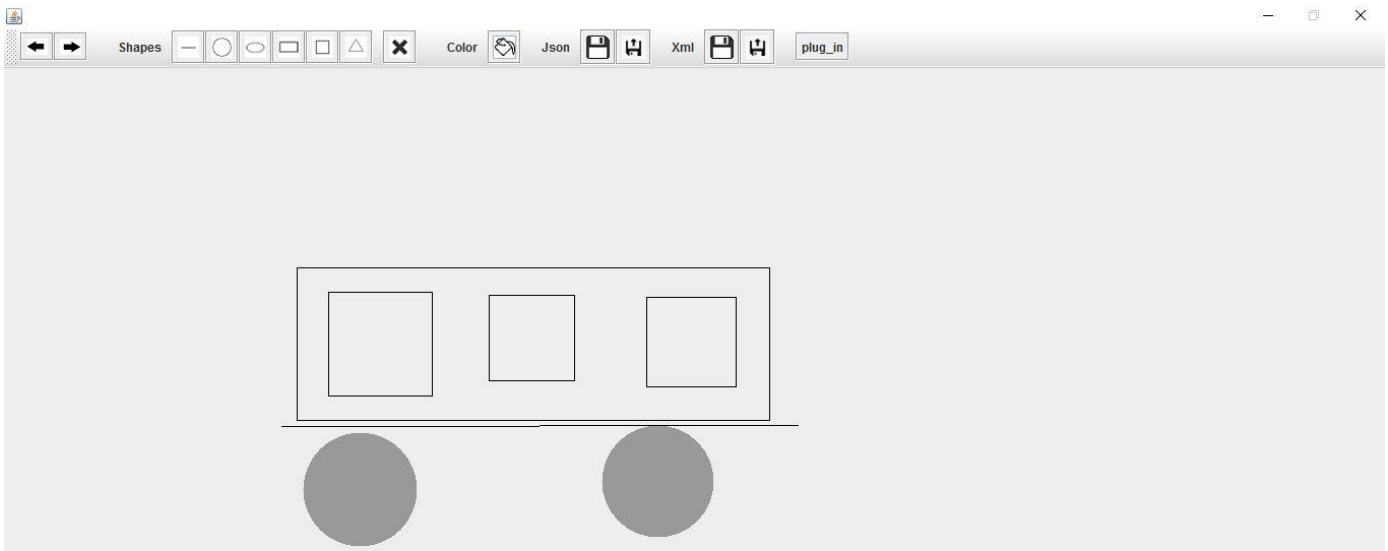
fill button will show a plate of colors , choose a color and click ok to colorize the selected shape .

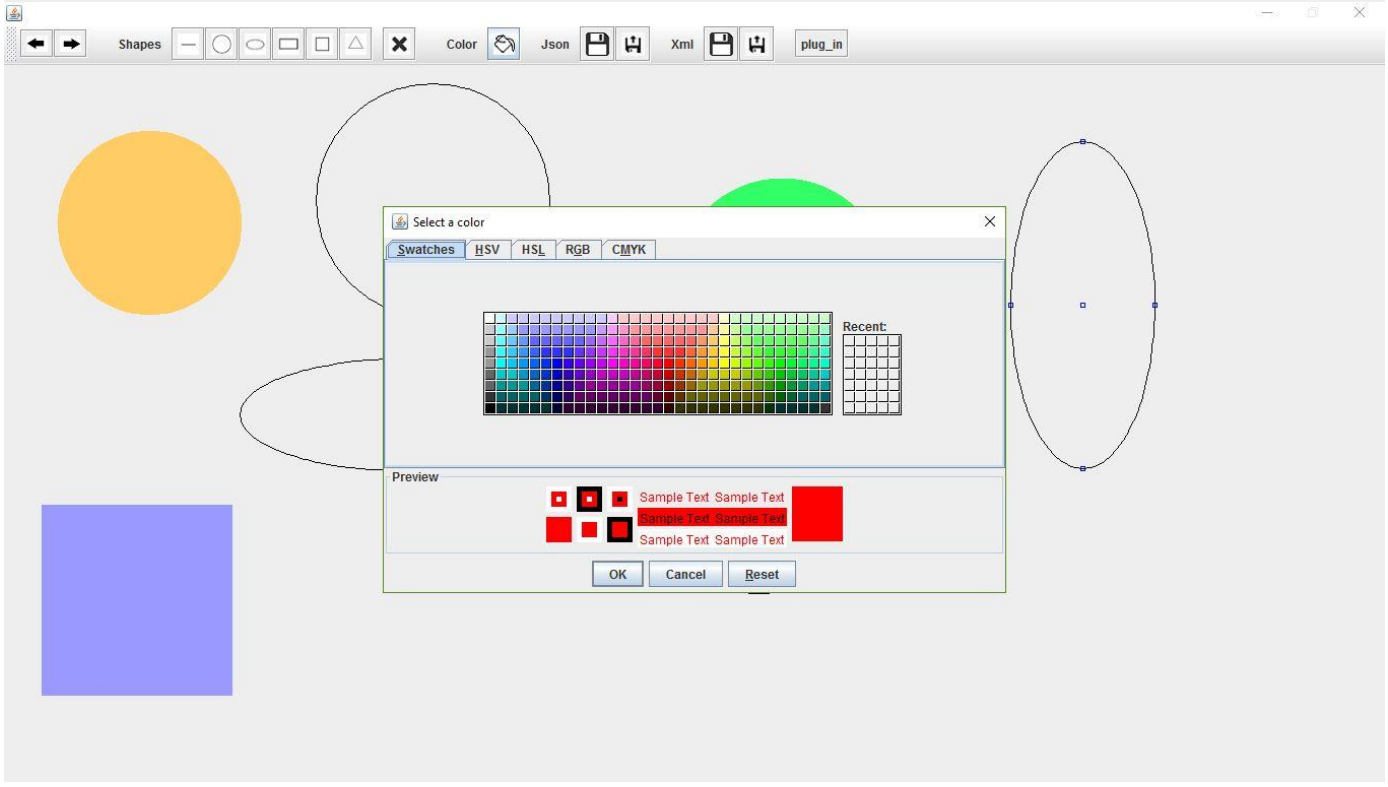
save and load we have two options , save and load as JSON or as XML you can select the file name to save or file path to load easily .

Snap Shots :

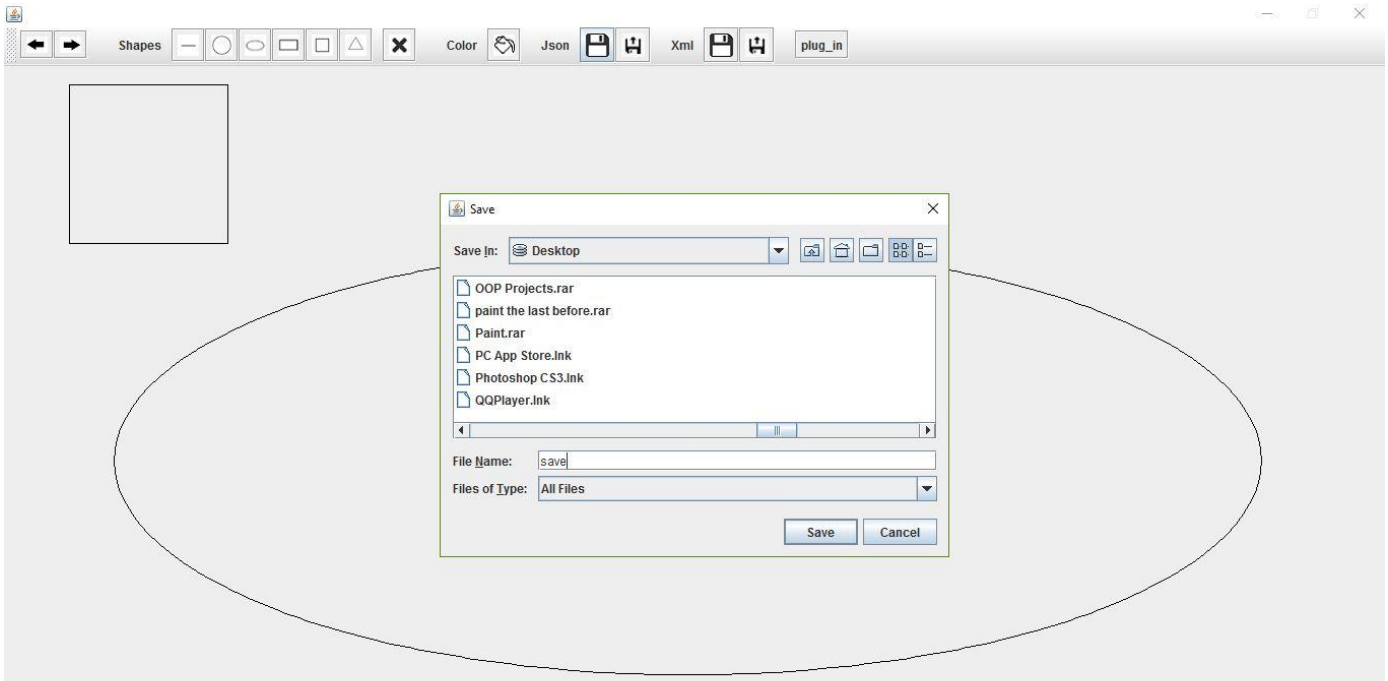


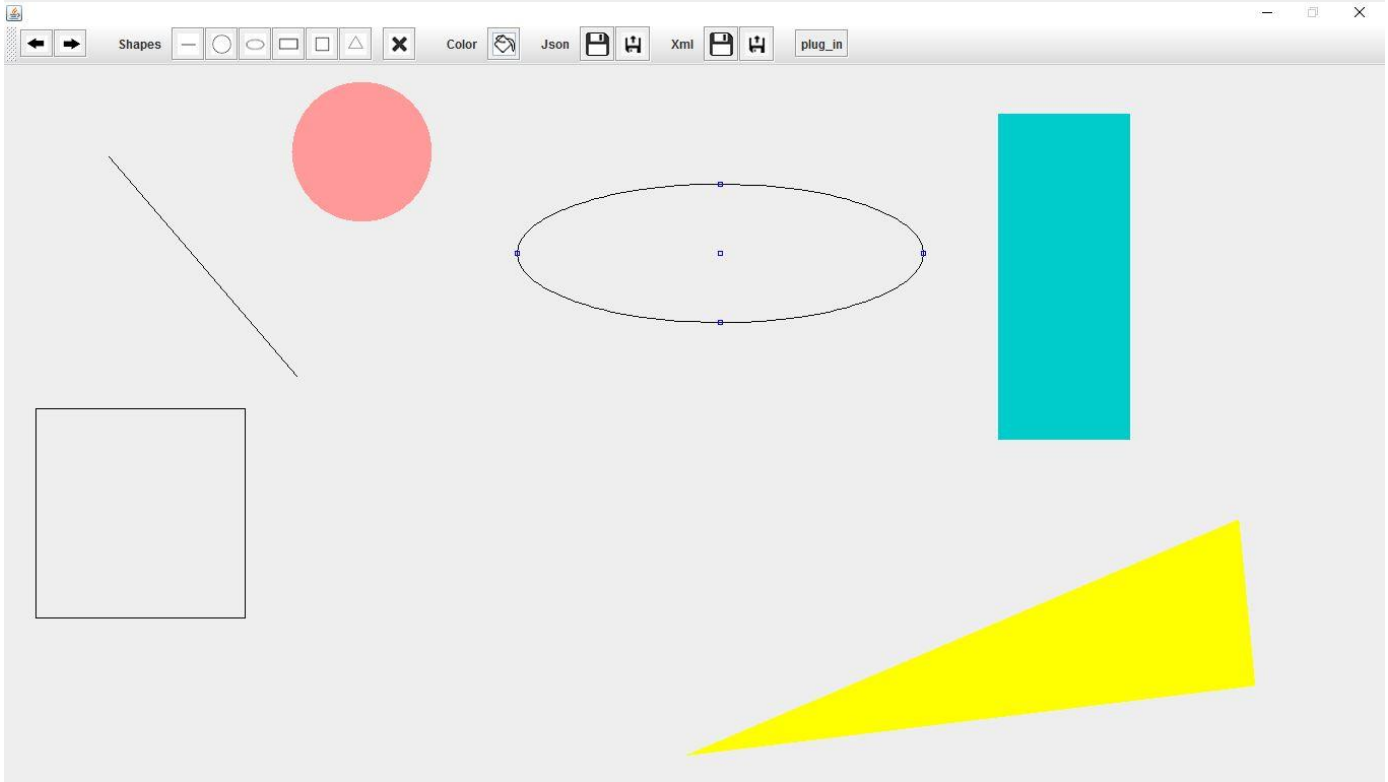
Vector Drawing





Vector Drawing



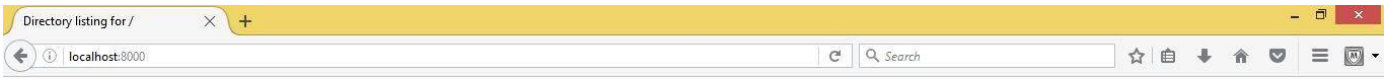


Design Assumptions :

1- Triangles drawn by clicking on the screen to specify three points.

2- After install plugin , a copy of the class taken so removing it in the runtime dosen't affect of the program running.
3- Selecting the shape from the intersection point will get me the first drawn shape.
4- If uncontained shaped loading a message error appear asking to istall the plug in

Bonus : JNLP



Directory listing for /

- launch.html
- launch.jnlp
- lib/
- Paint.jar
- README.TXT

Test page for launching the application via JNLP

Launch