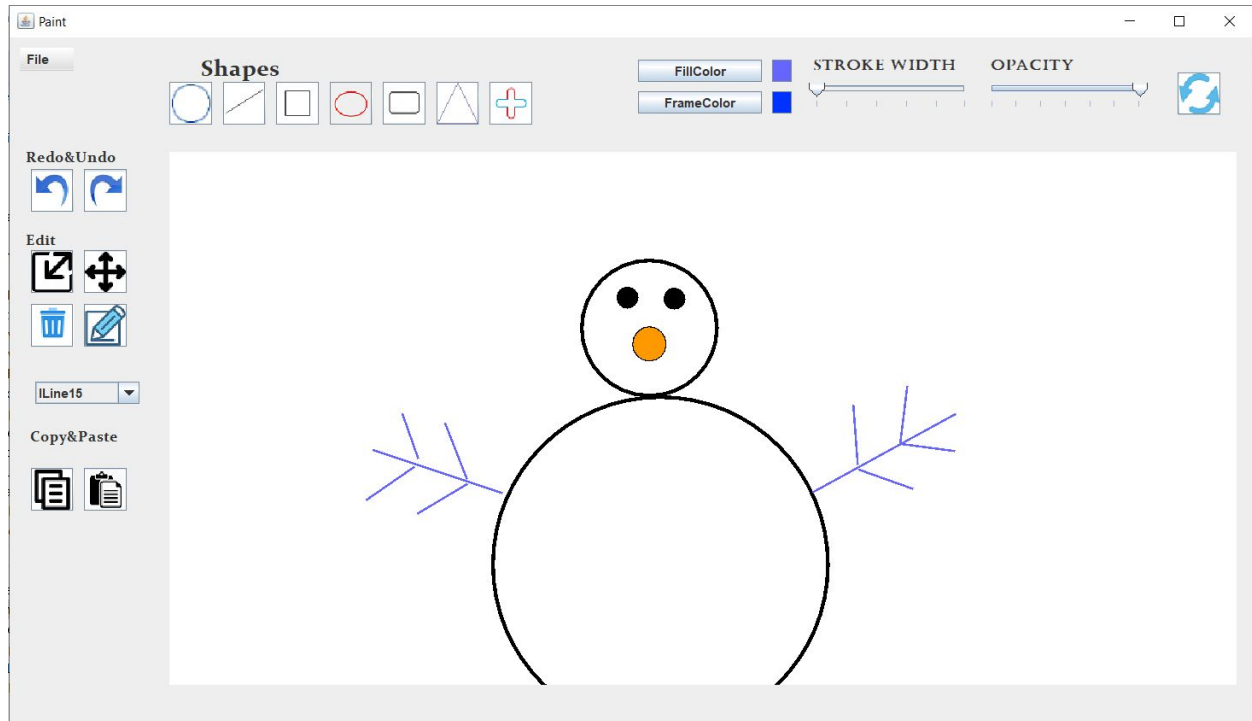Programming2

Vector Based Drawing Application

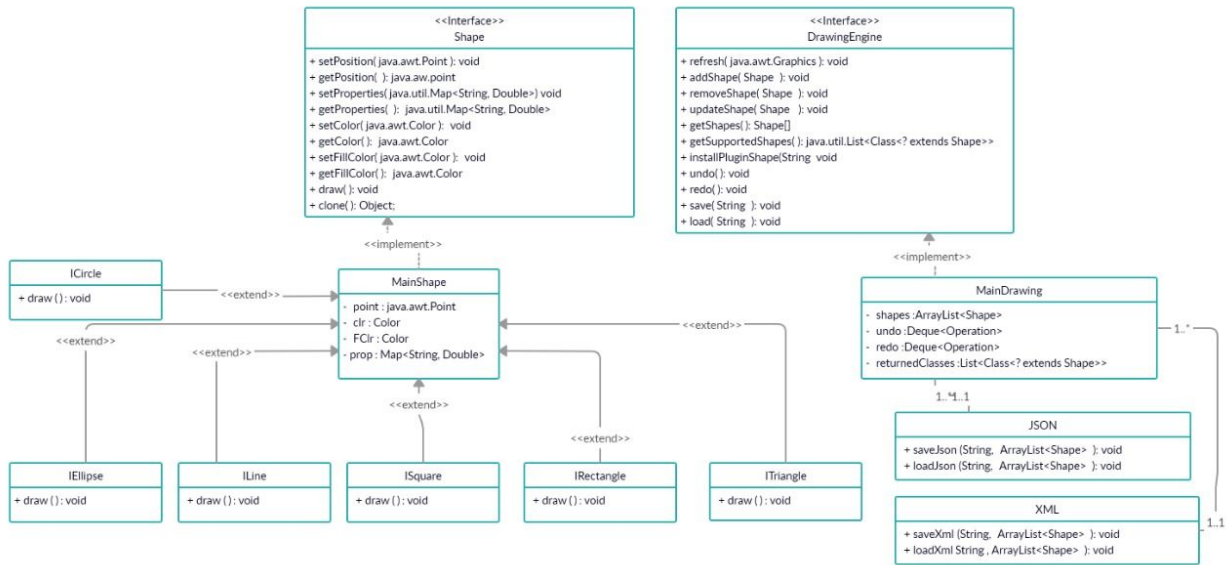Assignment report

- Khadija assem (27)
- Norhan magdi (69)

Content

- ❏ UML diagram.
- ❏ Design description.
- ❏ Snapshots of GUI.
- ❏ User guide that explains how to use your application.

UML diagram:

<<Interface>> Shape

+ setPosition(java.awt.Point): void
+ getPosition( ): java.aw.point
+ setProperties(java.util.Map<String, Double>) void
+ getProperties( ): java.util.Map<String, Double>
+ setColor(java.awt.Color): void
+ getColor( ): java.awt.Color
+ setFillColor(java.awt.Color): void
+ getFillColor( ): java.awt.Color
+ draw( ): void
+ clone( ): Object;

<<Interface>> DrawingEngine

+ refresh(java.awt.Graphics): void
+ addShape(Shape): void
+ removeShape(Shape): void
+ updateShape(Shape): void
+ getShapes( ): Shape[]
+ getSupportedShapes( ): java.util.List<Class<? extends Shape>>
+ installPluginShape(String void
+ undo( ): void
+ redo( ): void
+ save(String): void
+ load(String): void

<<implement>>

ICircle
+ draw ( ): void

MainShape
- point : java.awt.Point
- clr : Color
- FClr : Color
- prop : Map<String, Double>

<<extend>> <<extend>> <<extend>> <<extend>> <<extend>> <<extend>>

MainDrawing
- shapes :ArrayList<Shape>
- undo :Deque<Operation>
- redo :Deque<Operation>
- returnedClasses :List<Class<? extends Shape>>

1..*

1..4..1

IEllipse
+ draw ( ): void

ILine
+ draw ( ): void

ISquare
+ draw ( ): void

IRectangle
+ draw ( ): void

ITriangle
+ draw ( ): void

JSON
+ saveJson (String, ArrayList<Shape>): void
+ loadJson (String, ArrayList<Shape>): void

XML
+ saveXml (String, ArrayList<Shape>): void
+ loadXml String , ArrayList<Shape>): void

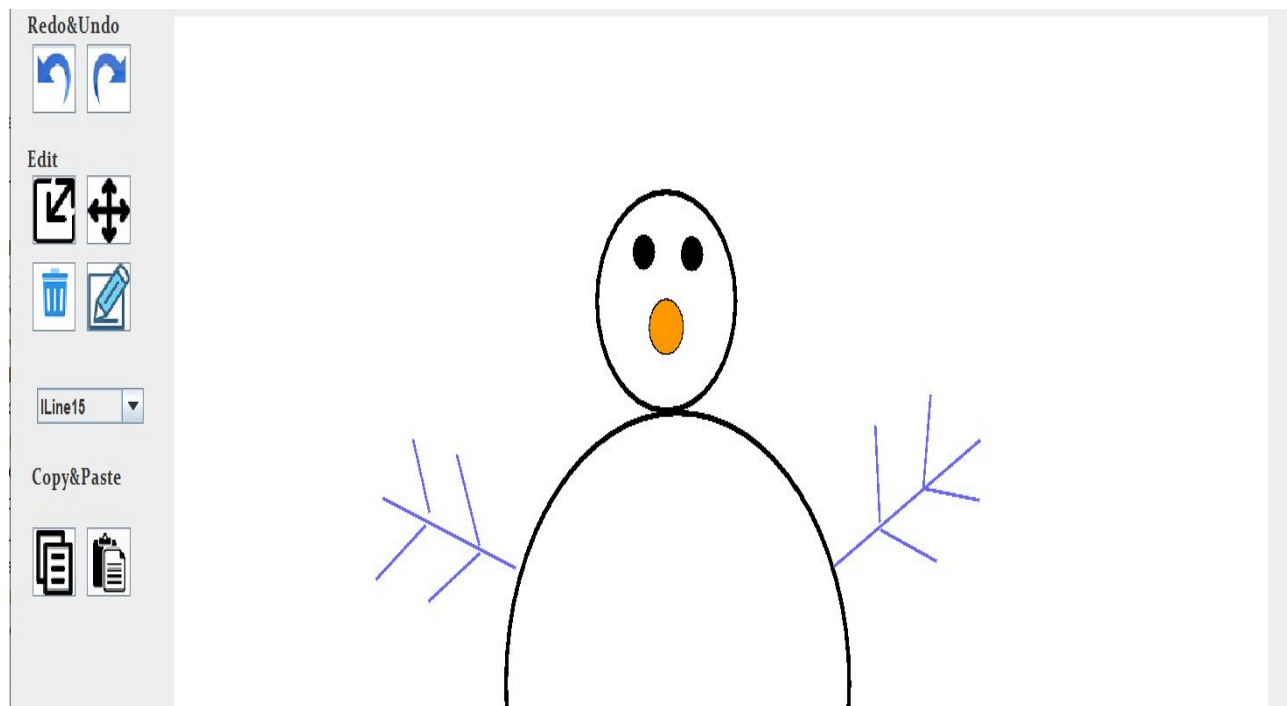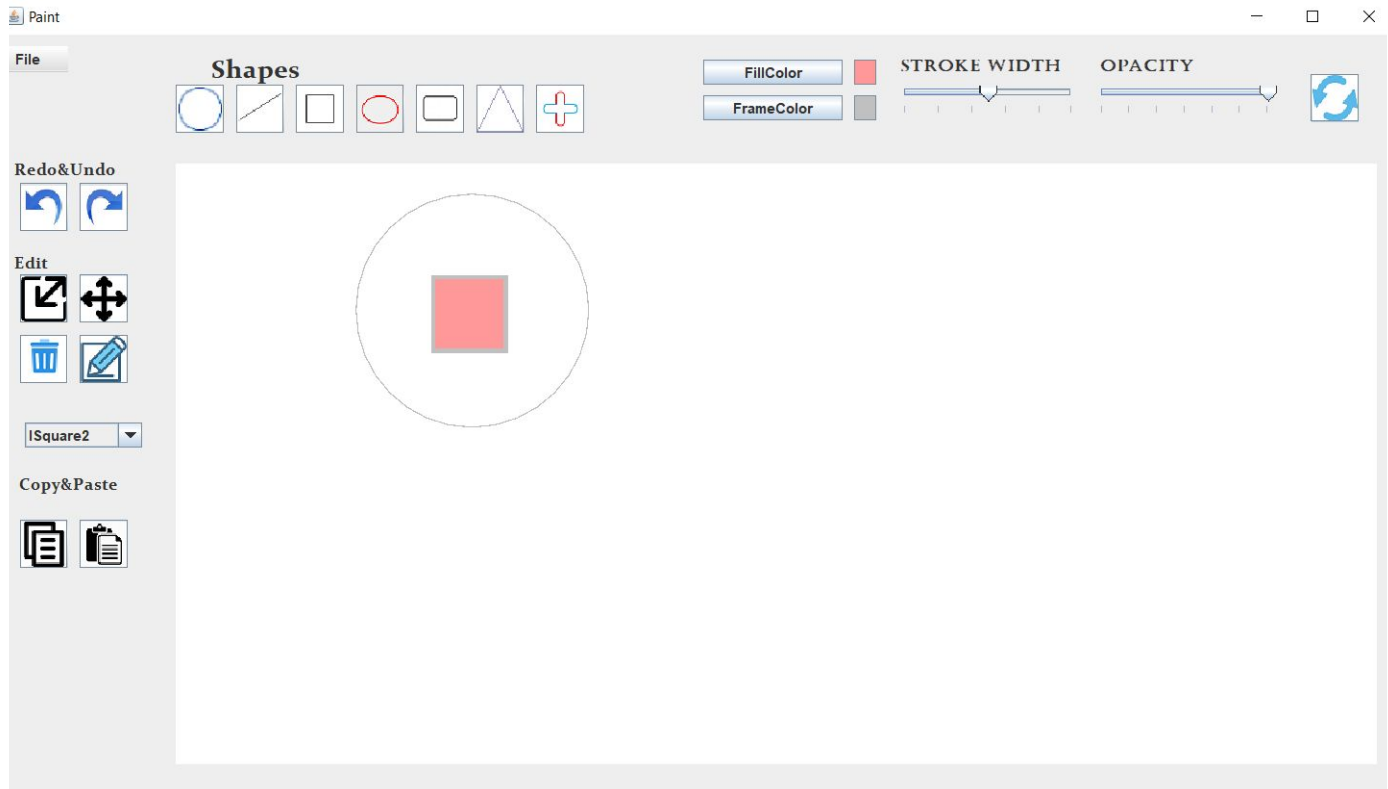1..1

Design description:

2 interfaces are implemented

- ➔ **Shapes:** it contains functions to set and get position,frame color,fill color and properties of current shape.
- ➔ **DrawingEngine:** it contains functions to add , remove , update and draw shape and allows user to redo ,undo , save and load actions.
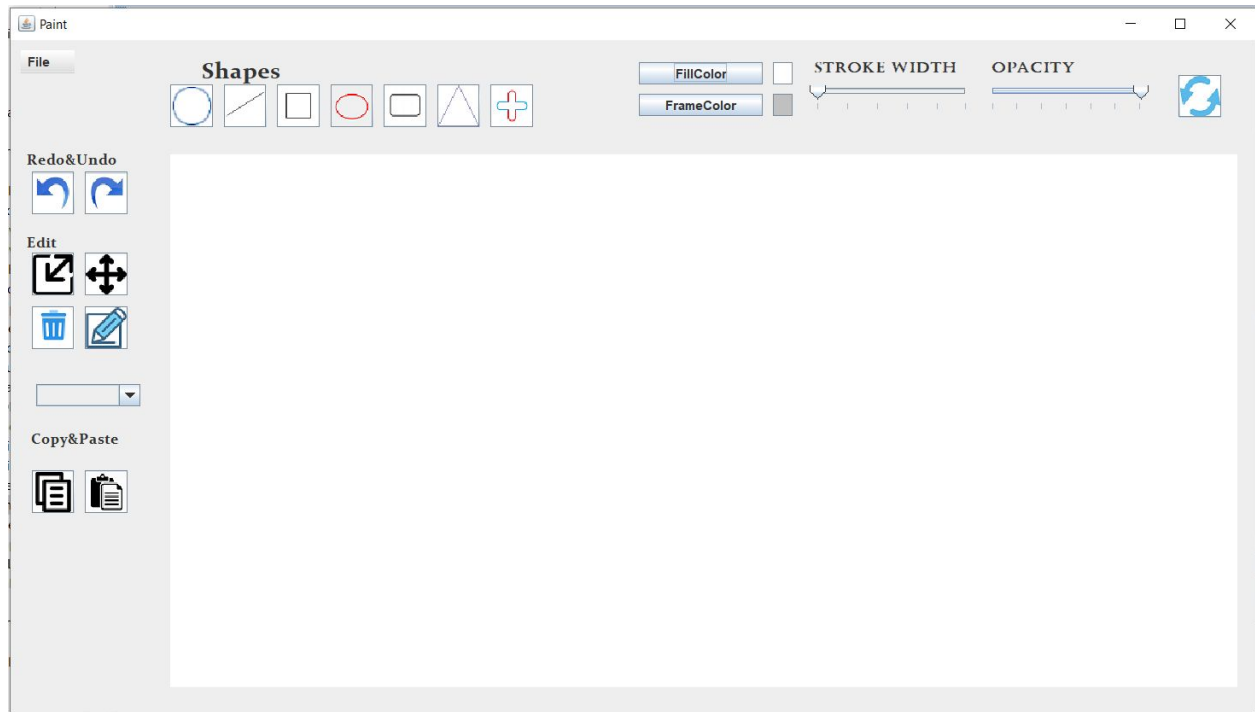
GUI is divided into two parts

- ➔ **Paint:** it extends DrawingArea and implements GUI action listener.
- ➔ **DrawingArea:** it extends JPanel and allow the area to draw in.

Snapshots of GUI:

File

Shapes

FillColor
FrameColor

STROKE WIDTH   OPACITY

Redo&Undo

Edit

ISquare2

Copy&Paste

Redo&Undo

Edit

ILine15

Copy&Paste

User guide:

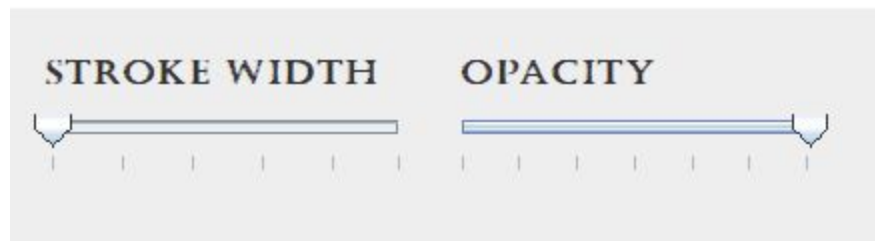- **First look when you open the programme**

- **Choose a shape to draw (it's available to load your own class)**

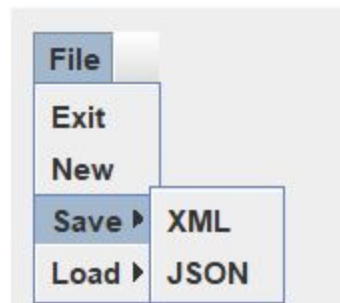- *Choose frame color and fill color*
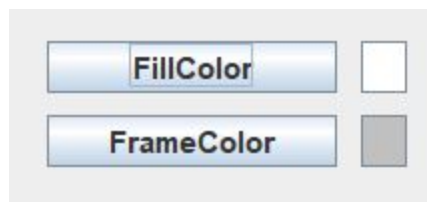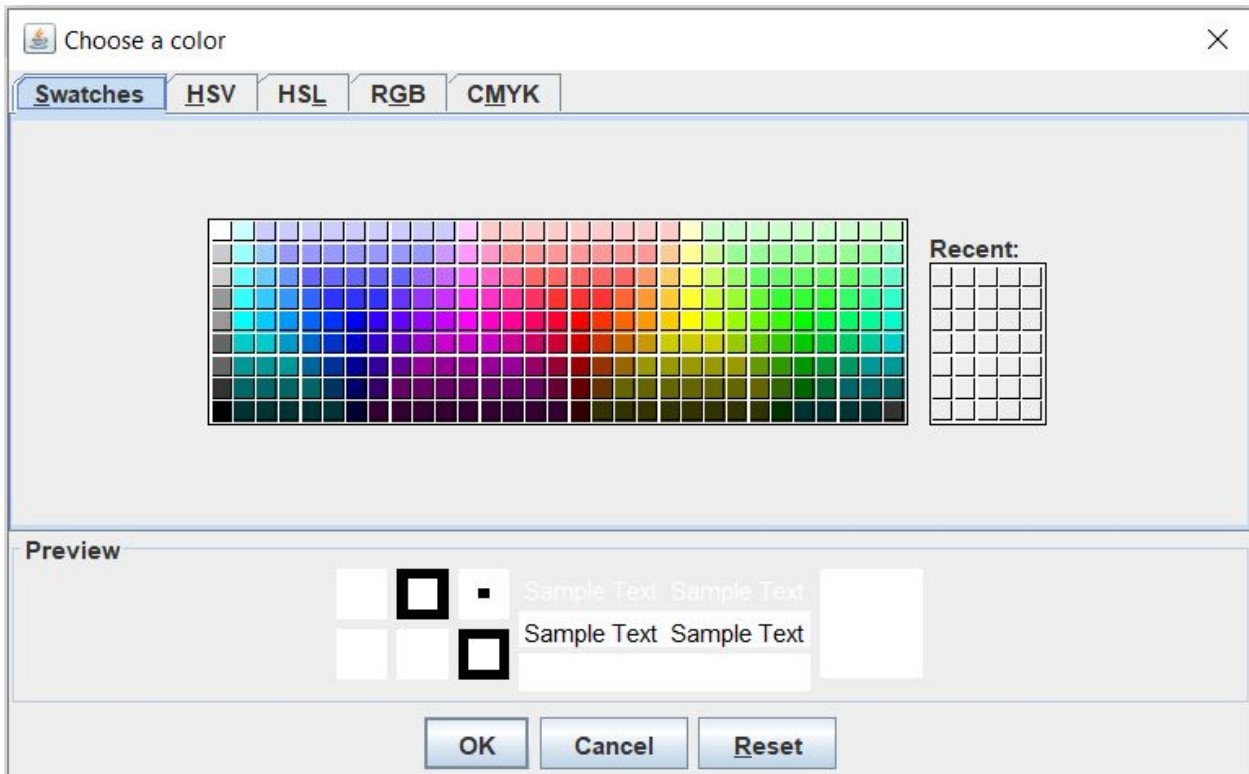
- *Change frame width and opacity*
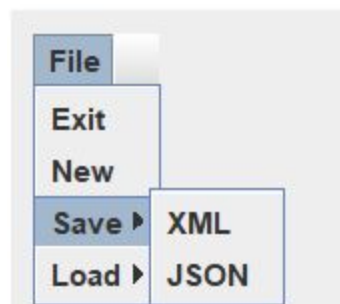

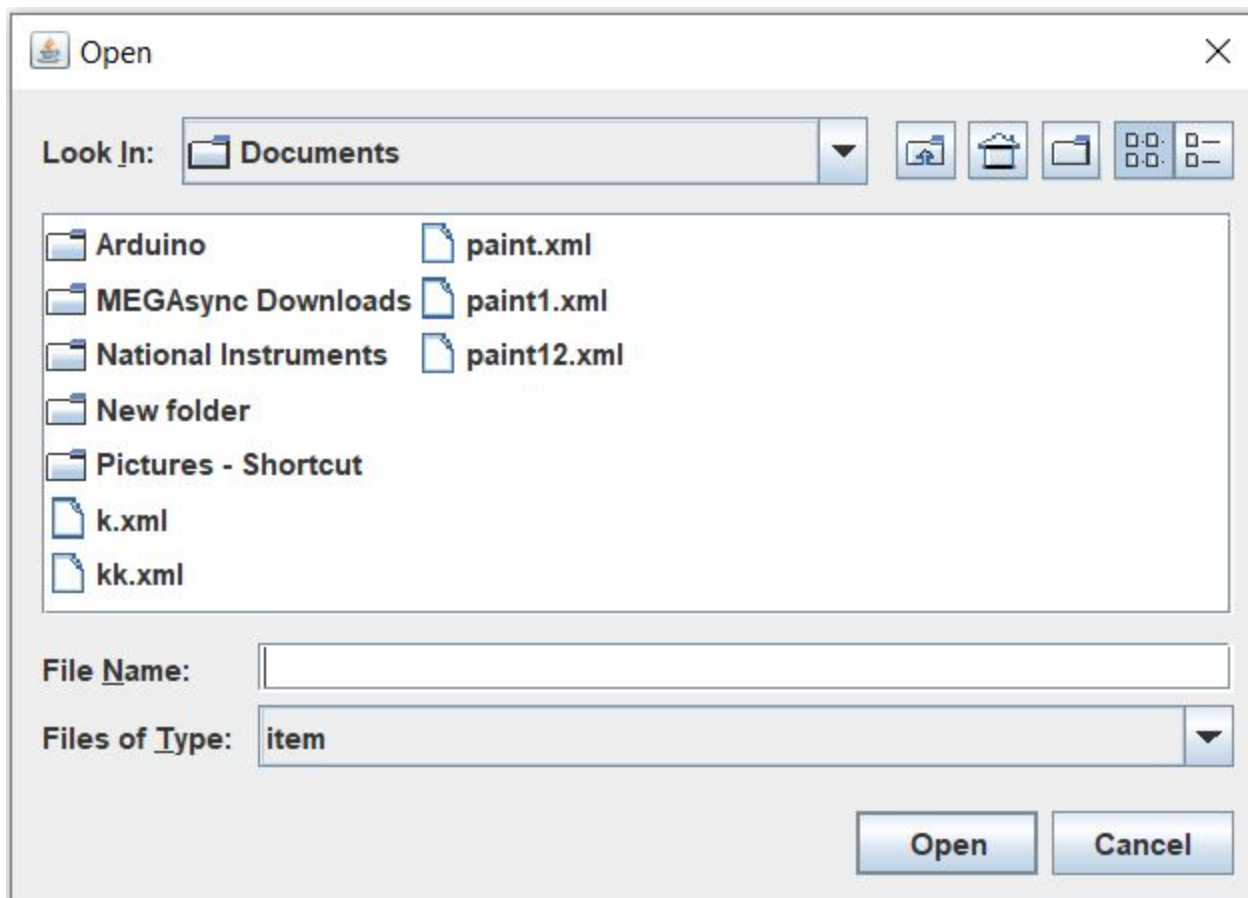- *Edit shapes (delete,move,resize,edit)*

- **Redo & undo**
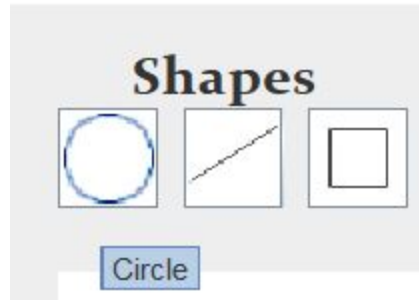

- **Exit and create new file**

- **Save and load**

- **Copy & paste**

- *On hovering over the button its name/function appears*

- *Refresh drawing area*