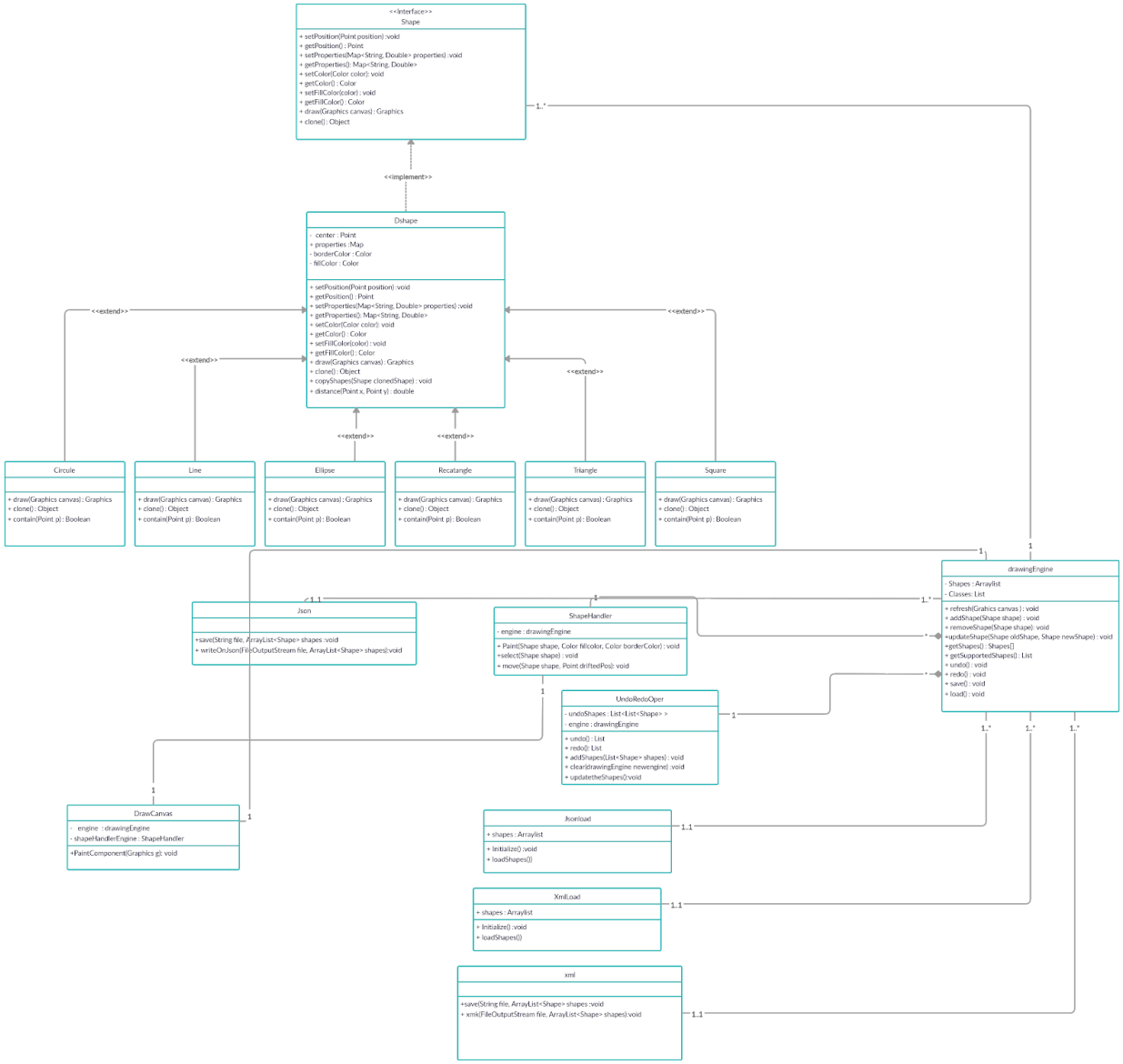
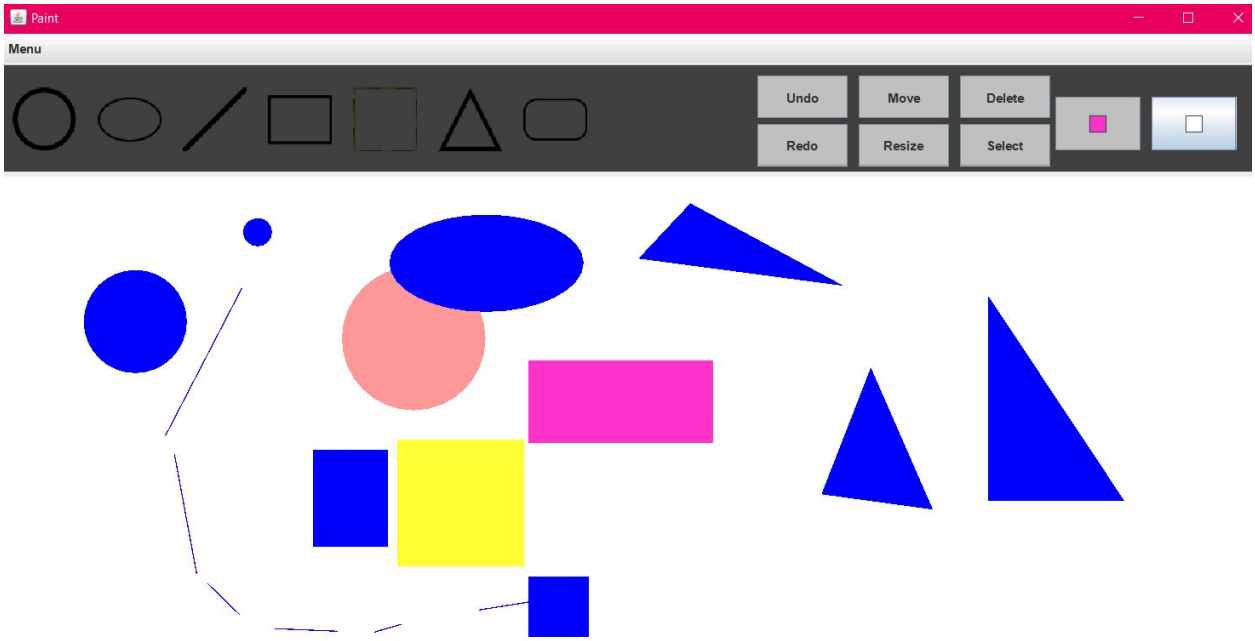
Paint

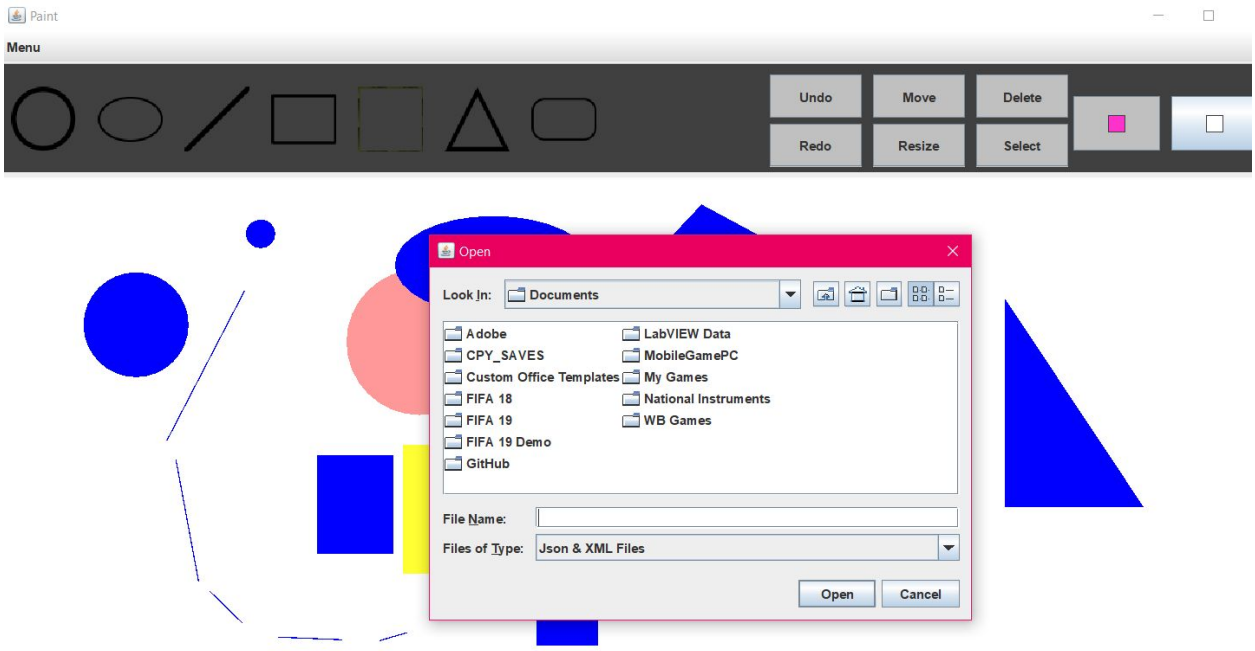
Description :

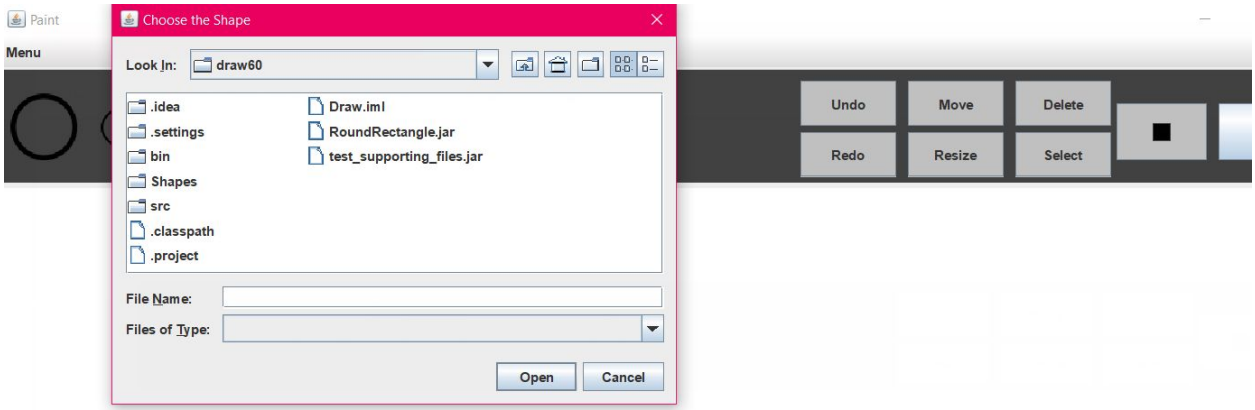
Drawing and painting applications are very popular and have a huge user base. They generally offer a big number of features that includes but is not limited to: Drawing, Coloring, and Resizing. They also include a number of built in, and possibly extensible set of geometric shapes, and classically, they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

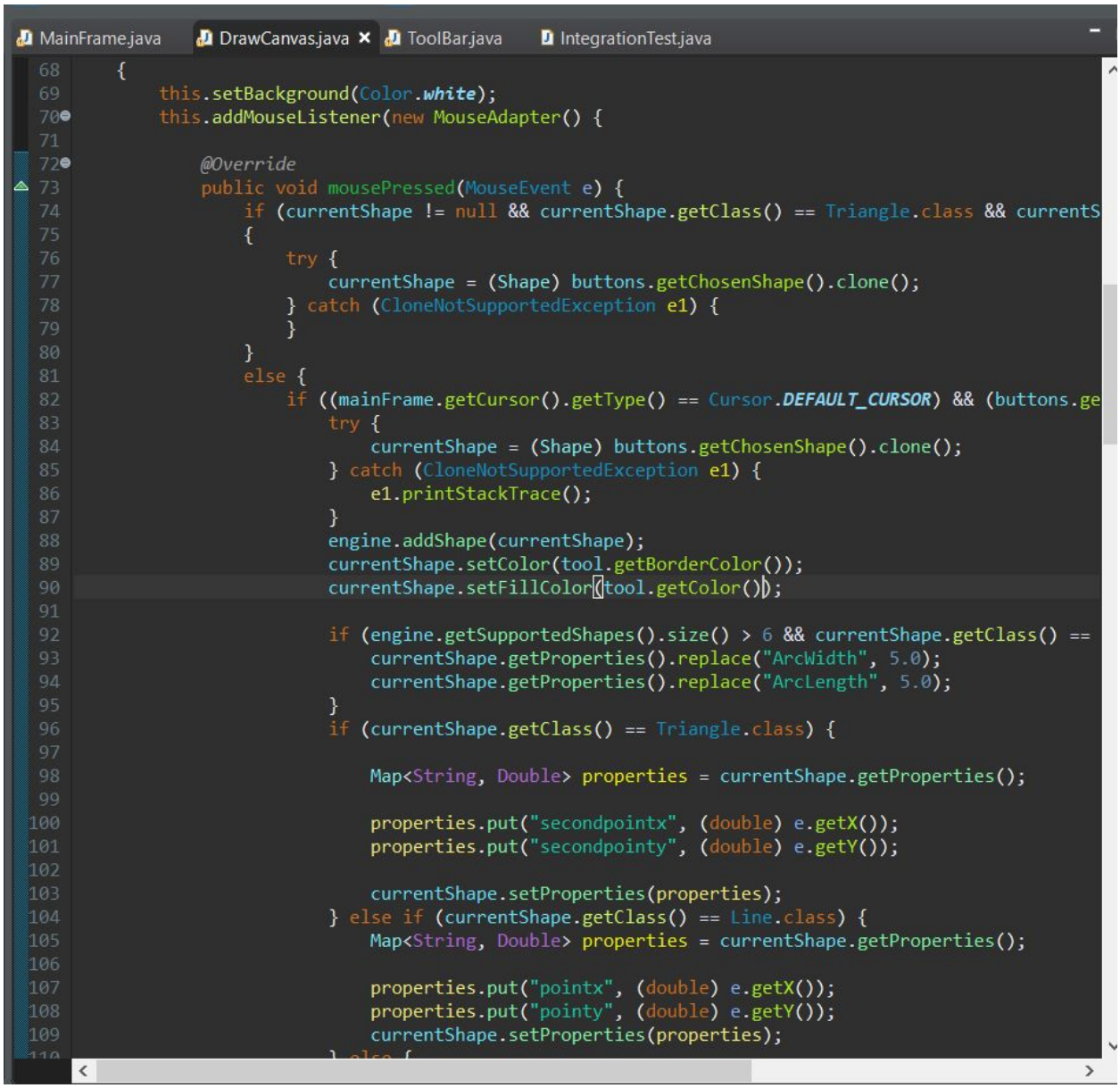
UMl Diagram :


<<Interface>>
Shape
+ setPosition(Point position):void
+ getPosition() : Point
+ setProperties(Map<String, Double> properties) :void
+ getProperties(): Map<String, Double>
+ setColor(Color color): void
+ getColor() : Color
+ setFillColor(color) : void
+ getFillColor() : Color
+ draw(Graphics canvas) : Graphics
+ clone() : Object
1..*
<<implement>>
Dshape
- center : Point
+ properties :Map
- borderColor : Color
- fillColor : Color
+ setPosition(Point position) :void
+ getPosition() : Point
+ setProperties(Map<String, Double> properties) :void
+ getProperties(): Map<String, Double>
+ setColor(Color color): void
+ getColor() : Color
+ setFillColor(color) : void
+ getFillColor() : Color
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ copyShapes(Shape clonedShape) : void
+ distance(Point x, Point y) : double
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
Circule
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ contain(Point p) : Boolean
Line
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ contain(Point p) : Boolean
Ellipse
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ contain(Point p) : Boolean
Recatangle
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ contain(Point p) : Boolean
Triangle
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ contain(Point p) : Boolean
Square
+ draw(Graphics canvas) : Graphics
+ clone() : Object
+ contain(Point p) : Boolean
drawingEngine
- Shapes : Arraylist
- Classes: List
+ refresh(Grahics canvas) : void
+ addShape(Shape shape) : void
+ removeShape(Shape shape): void
+updateShape(Shape oldShape, Shape newShape) : void
+getShapes() : Shapes[]
+ getSupportedShapes() : List
+ undo() : void
+ redo() : void
+ save() : void
+ load() : void
Json
+save(String file, ArrayList<Shape> shapes :void
+ writeOnJson(FileOutputStream file, ArrayList<Shape> shapes):void
ShapeHandler
- engine : drawingEngine
+ Paint(Shape shape, Color fillcolor, Color borderColor) : void
+select(Shape shape) : void
+ move(Shape shape, Point driftedPos): void
UndoRedoOper
- undoShapes : List<List<Shape> >
- engine : drawingEngine
+ undo() : List
+ redo(): List
+ addShapes(List<Shape> shapes) : void
+ clear(drawingEngine newengine) :void
+ updatetheShapes():void
DrawCanvas
- engine : drawingEngine
- shapeHandlerEngine : ShapeHandler
+PaintComponent(Graphics g): void
Jsonload
+ shapes : Arraylist
+ Initialize() :void
+ loadShapes()
XmlLoad
+ shapes : Arraylist
+ Initialize() :void
+ loadShapes()
xml
+save(String file, ArrayList<Shape> shapes :void
+ xml(FileOutputStream file, ArrayList<Shape> shapes):void


Paint
Menu
Undo
Move
Delete
Redo
Resize
Select


Paint
Menu
Undo
Move
Delete
Redo
Resize
Select
Open
Look In:
Documents
Adobe
CPY_SAVES
Custom Office Templates
FIFA 18
FIFA 19
FIFA 19 Demo
GitHub
LabVIEW Data
MobileGamePC
My Games
National Instruments
WB Games
File Name:
Files of Type:
Json & XML Files
Open
Cancel


Paint
Menu
Undo
Move
Delete
Redo
Resize
Select
Choose the Shape
Look In:
draw60
.idea
.settings
bin
Shapes
src
.classpath
.project
Draw.iml
RoundRectangle.jar
test_supporting_files.jar
File Name:
Files of Type:
Open
Cancel


MainFrame.java
DrawCanvas.java
ToolBar.java
IntegrationTest.java
{
this.setBackground(Color.white);
this.addMouseListener(new MouseAdapter() {
@Override
public void mousePressed(MouseEvent e) {
if (currentShape != null && currentShape.getClass() == Triangle.class && currentS
{
try {
currentShape = (Shape) buttons.getChosenShape().clone();
} catch (CloneNotSupportedException e1) {
}
}
else {
if ((mainFrame.getCursor().getType() == Cursor.DEFAULT_CURSOR) && (buttons.ge
try {
currentShape = (Shape) buttons.getChosenShape().clone();
} catch (CloneNotSupportedException e1) {
e1.printStackTrace();
}
engine.addShape(currentShape);
currentShape.setColor(tool.getBorderColor());
currentShape.setFillColor(tool.getColor());
if (engine.getSupportedShapes().size() > 6 && currentShape.getClass() ==
currentShape.getProperties().replace("ArcWidth", 5.0);
currentShape.getProperties().replace("ArcLength", 5.0);
}
if (currentShape.getClass() == Triangle.class) {
Map<String, Double> properties = currentShape.getProperties();
properties.put("secondpointx", (double) e.getX());
properties.put("secondpointy", (double) e.getY());
currentShape.setProperties(properties);
} else if (currentShape.getClass() == Line.class) {
Map<String, Double> properties = currentShape.getProperties();
properties.put("pointx", (double) e.getX());
properties.put("pointy", (double) e.getY());
currentShape.setProperties(properties);

```
        return Shapes.toArray(ar);
    }

    @Override
    public List<Class<? extends Shape>> getSupportedShapes() {
        return Classes;
    }

    @Override
    public void installPluginShape(String jarPath) {
        try {
            JarFile jarFile = new JarFile(jarPath);
            Enumeration<JarEntry> e = jarFile.entries();

            URL[] urls = { new URL("jar:file:" + jarPath +"!/") };
            URLClassLoader cl = URLClassLoader.newInstance(urls);

            while (e.hasMoreElements()) {
                JarEntry je = e.nextElement();
                if(je.isDirectory() || !je.getName().endsWith(".class")){
                    continue;
                }
                String className = je.getName().substring(0,je.getName().length()-6);
                className = className.replace('/', '.');
                Class loadClass = cl.loadClass(className);
                if (Shape.class.isAssignableFrom(loadClass))
                {
                    Classes.add((Class<Shape>)loadClass);
                }
            }
        } catch (ClassNotFoundException | IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void undo() {
        try
        {
            this.Shapes = (ArrayList<Shape>) this.undoRedoOper.undo();}
        catch (NullPointerException | CloneNotSupportedException e)
        {
```

MainFrame.java | DrawCanvas.java | ToolBar.java | IntegrationTest... | drawingEngine.j... | DShape.java

```java
package eg.edu.alexu.csd.oop.draw.Shapes;

import eg.edu.alexu.csd.oop.draw.Shape;

public abstract class DShape implements Shape {
    private Point center;


    public Map<String, Double> properties;
    protected String[] ShapeProperties = {"Width", "Length"};
    private Color borderColor;
    private Color fillColor;



    public DShape()
    {
        center = new Point();
        properties = new HashMap<>();
        borderColor = Color.blue;
        fillColor = Color.blue;
        properties.put("Width" , (double) 5);
        properties.put("Length" , (double) 5);
    }
    @Override
    public void setPosition(Point position) {
        this.center.setLocation(position.getX(), position.getY());
    }

    @Override
    public Point getPosition() {
        return this.center;
    }



    @Override
    public void setProperties(Map<String, Double> properties) {
        this.properties = properties;
    }

    @Override
    public Map<String, Double> getProperties() {
```