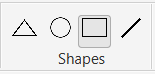
# Paint application

### **Description:**

Our paint application is a software graphics program that allows the user to draw several shapes such as **Rectangles**, **Lines**, **Squares**, **Circles**, **Ellipses**, and **triangles**.

### **Features:**

### **Drawing:**

One can draw as many shapes as he likes on one drawing board, there are 4

icons each for drawing a specific shape. A Shape is selected and the user

is required to drag the shape until acquiring the desired size.

* **Colors:**

One can also color his shapes either by choosing the colors before

drawing the shapes or by selecting the shape after drawing and

choosing a color. A Shape can be **filled** or **transparent**.

### **Copy:**

One can also copy shapes that are already drawn with the same size instead of

drawing new ones,

### **Delete:**

One can delete and shape that has already been drawn.

* **Undo and Redo:**

One can undo or redo any change that has been done on the drawing board

by pressing the **UNDO** button for undoing any action and the **REDO** button for

redoing any action.

### **Selecting, Moving, and Resizing:**

The user also has the option to select any shape and choose an action to perform on the shape, one of the actions are **MOVING** the shape to another place, another action is **RESIZING** a shape.

### **Design Overview:**

Our design consists of 3 main and 3 utility packages, the 3 mains packages are **Model**, **View**, and **Controller,** the **Model** consists of all the logical calculations necessary for drawing such as the Shape Interface the Shape factory (which creates new Objects of the required shape) and all the shapes classes.

**View** package which consists of the program friendly user interface.

**Controller** package which consists of the Drawing engine Interface and it is the connection between the **Model** and the **View.**

We’ve applied different design patterns in our design such as **MVC** architectural pattern, **Factory** design pattern, **Singleton** design pattern, **Strategy** design pattern, **Command** design pattern, and **Memento** design pattern.

### **Assumptions:**

1. The last drawn shape is always the selected shape.
2. If the program is just opened user can’t undo or redo anything.
3. There are two types of file extensions allowed in our program which are JSON and XML.
4. If the user chose to save in any other extension the saved file will be in XML format.
5. File name must contain one dot only before the extension.
6. When copying a shape, it is created on the source shape, user should drag it to the desired place.

### **Teamwork:**

* **Omar Nasr 4730**

1. Select
2. Multi Select
3. Move
4. Delete
5. Container Resizing
6. Hotkeys

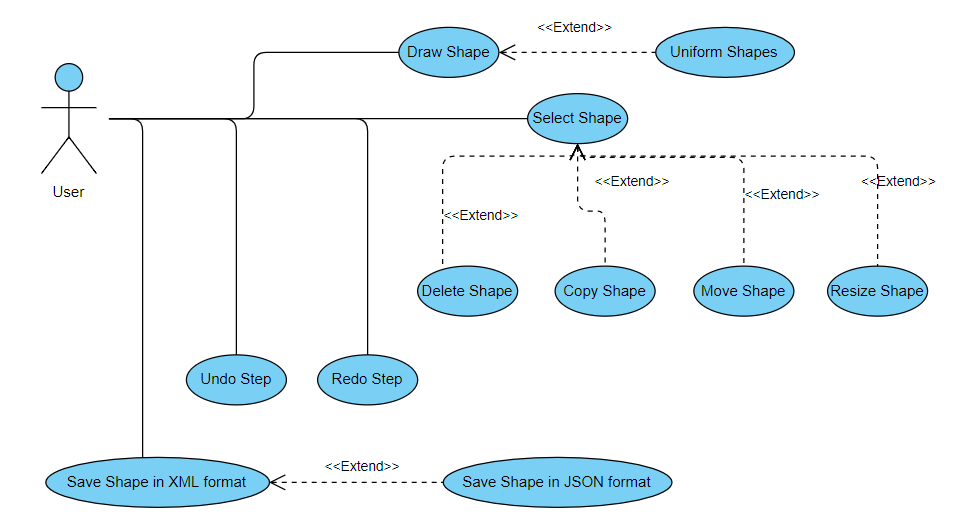
* **Mohamed Harraz 4608**
  1. Undo & Redo
  2. Save & Load
  3. Update shape properties on selection
  4. File Chooser
* **Sherif Rafik 4635**
  1. Draw Models
  2. Color
  3. Resize
  4. Copy
  5. Plugins

### **Data Structures:**

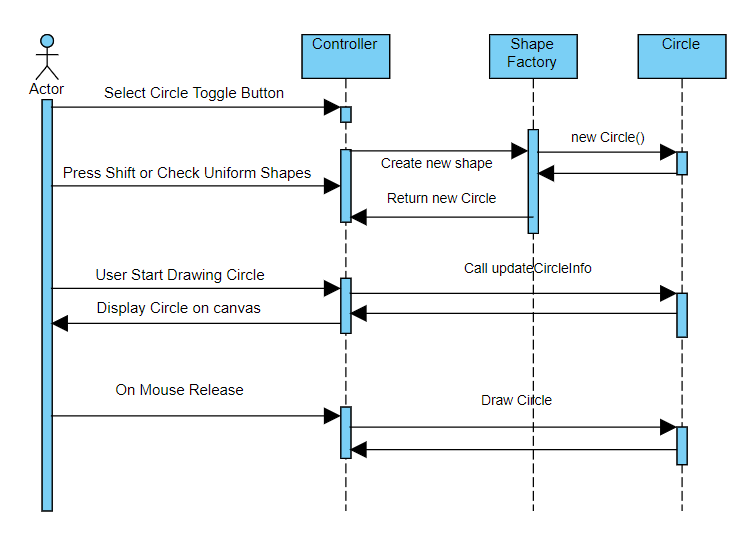
In out paint application we’ve used two types of data structures, **Array lists** and **Hash maps**.

### **UML Diagrams:**

### **Use Case:**

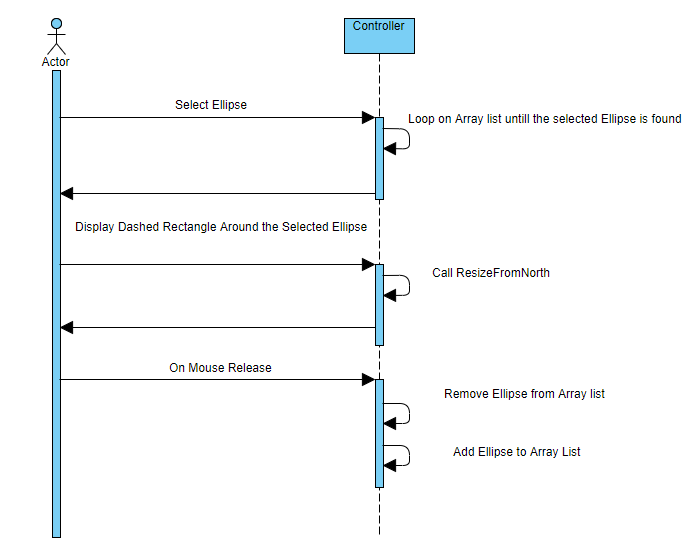


### **Drawing a circle Sequence Diagram:**

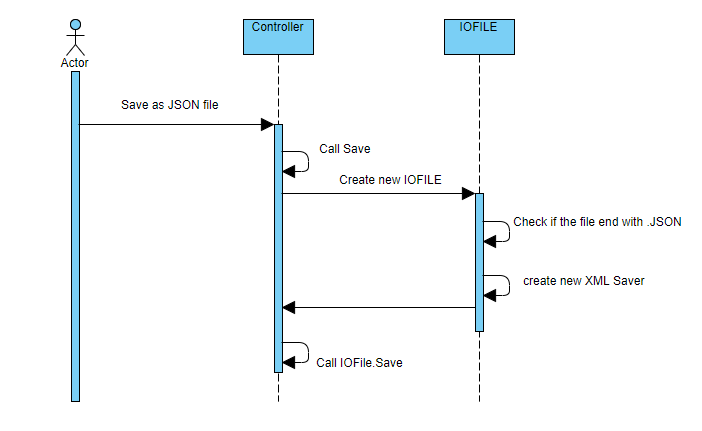


### **Editing Ellipse Sequence Diagram:**

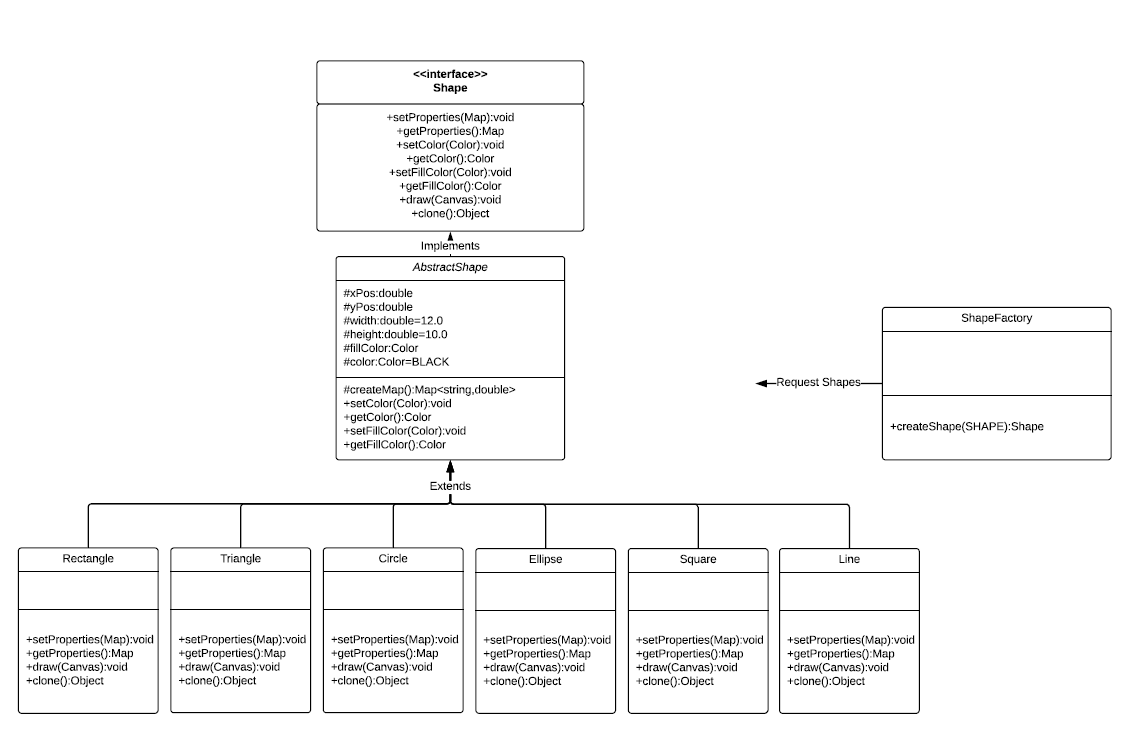
**Assuming the editing is resizing the ellipse and the user chose to resize from the north edge.**



### **Saving a file Sequence Diagram:**

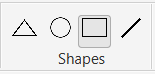
****

### **Class Diagram:**



### **User Manual:**

### **Drawing:**

One can draw as many shapes as he likes on one drawing board, there are 4 icons each for

drawing a specific shape.

A Shape is selected and the user is required to drag the shape until acquiring the desired size, If one

****wants to draw uniform shapes such as **Circles** or **Squares**, one has got two options either pressing

**SHIFT** before dragging the shape or by checking the uniform shapes check box.

### **Colors:**

One can also color his shapes either by choosing the colors before drawing the shapes or by

selecting the shape after drawing and choosing a color. A Shape can be **filled** or **transparent**,

as shown in the figure there are two color pickers one for the border of the shape and the other

for the fill color of the shape. As shown in the figure there is also a size slider by which the user can control the width of the border of the shape.

In order to draw a filled shape, the user must check the fill shapes check box.

### **Copy:**

One can also copy shapes that are already drawn with the same size instead of drawing new ones,

A shape must be selected and there is a copy button which will create a copy the user will just have

to drag it to the desired place.

### **Delete:**

One can delete and shape that has already been drawn by two options either by selecting the shape

and pressing the delete button or by selecting the shape and pressing **DEL** button on keyboard.

### **Undo and Redo:**

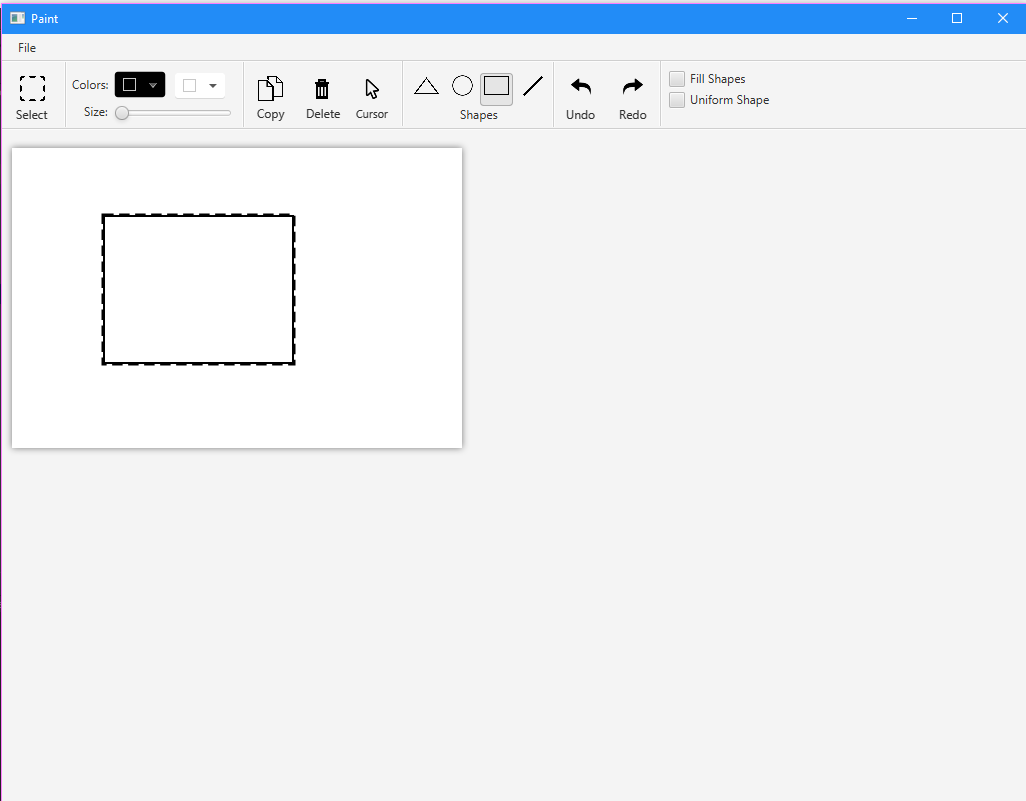
One can undo or redo any change that has been done on the drawing board by pressing the **UNDO**

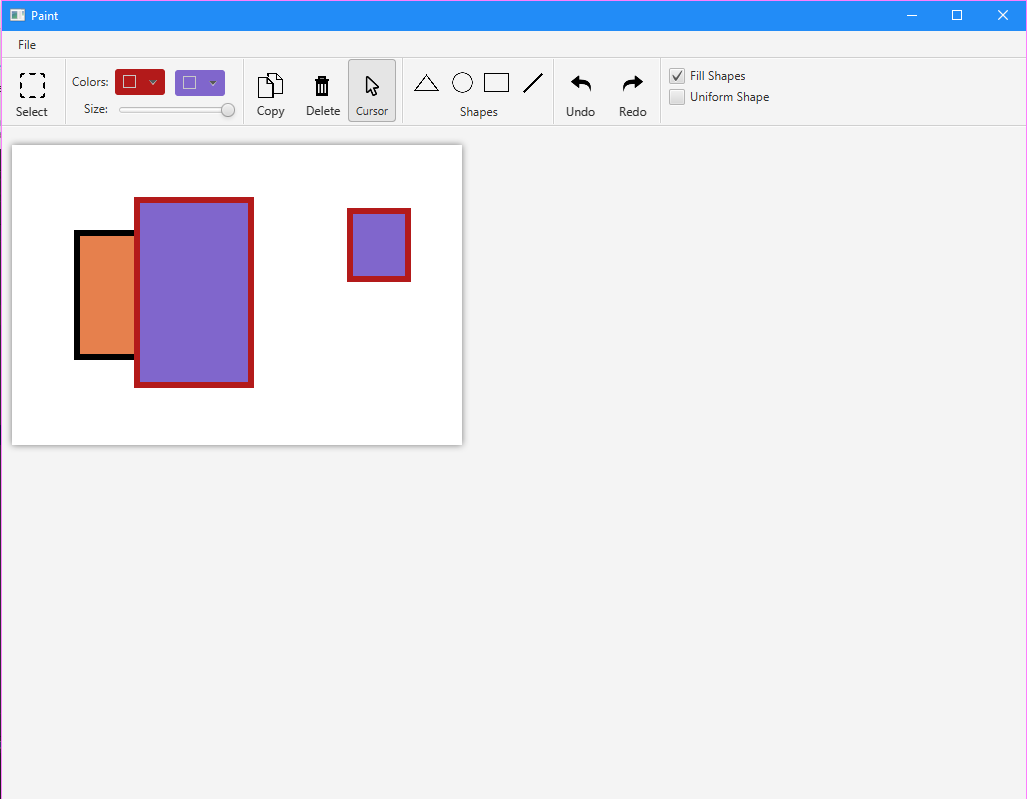
button for undoing any action and the **REDO** button for redoing any action.

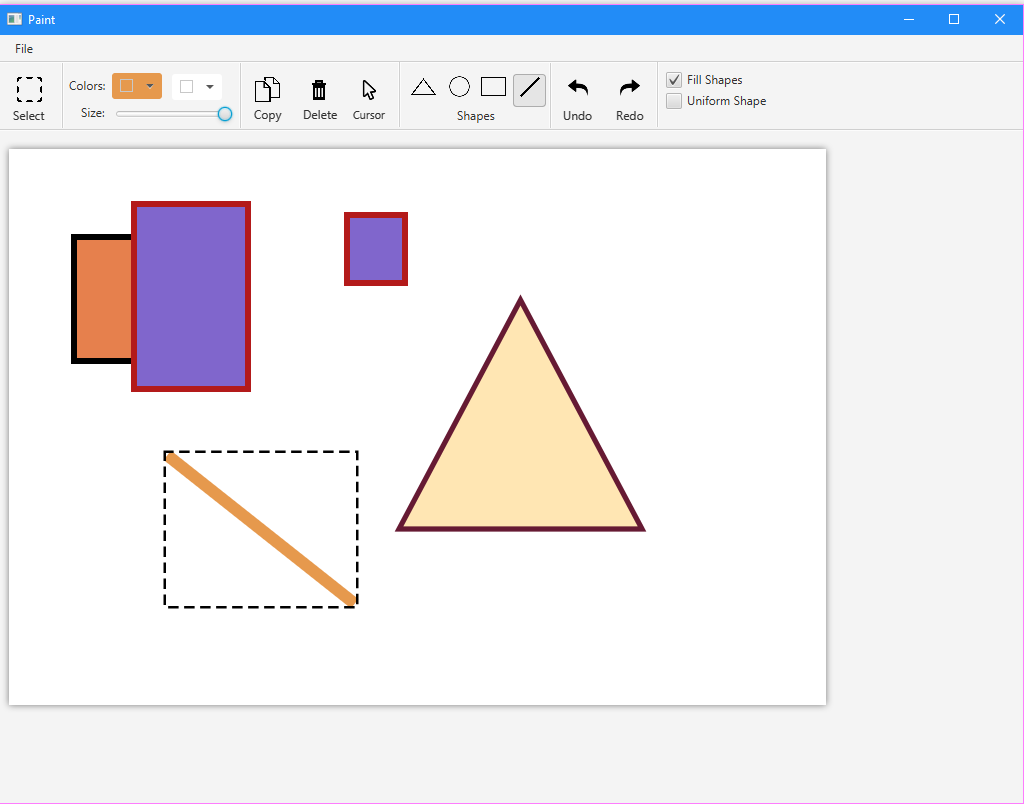
### **Selecting, Moving, and Resizing:**

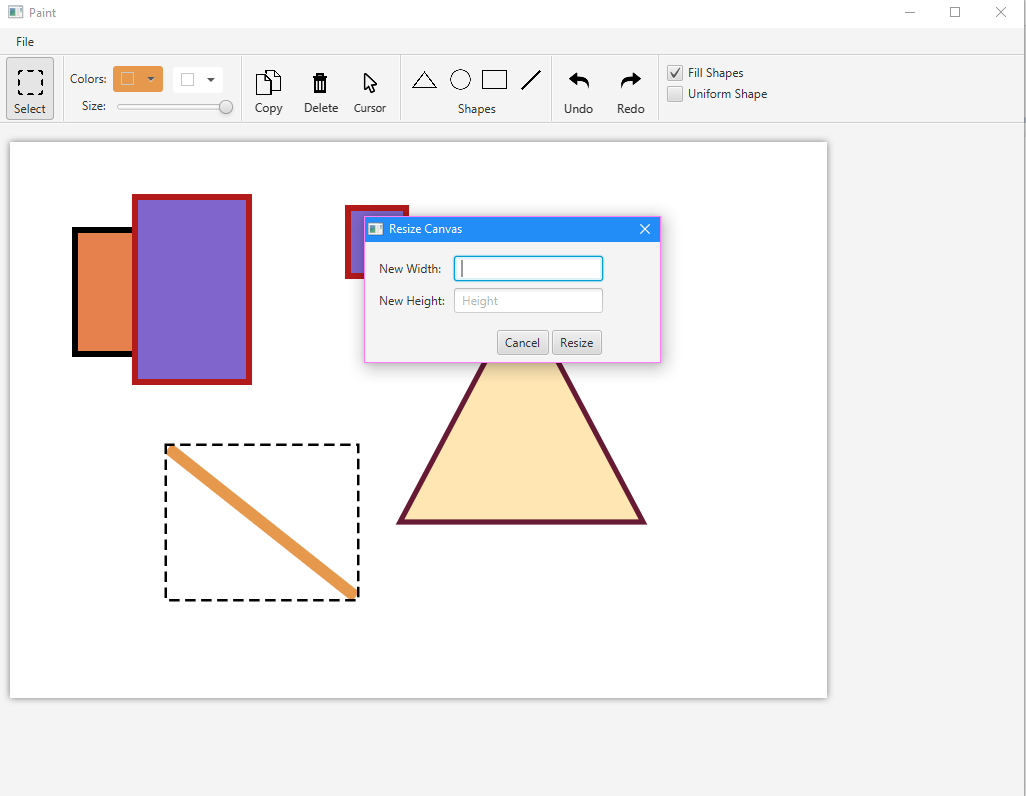
The user also has the option to select any shape and choose an action to perform on the shape, one of the actions are **MOVING** the shape to another place, this can be done by selecting the shape and holding it down and dragging it to the desired place. Another action is **RESIZING** a shape, this can be done by selecting the shape and hovering over any of its sides and dragging until acquiring the desired size.

### **Sample runs:**









### **References:**

<https://docs.oracle.com/javase/tutorial/deployment/jar/jarclassloader.html>

<https://stackoverflow.com/questions/28764893/creating-instances-of-loaded-classes>

<https://stackoverflow.com/questions/17132365/classcastexception-when-try-to-use-loaded-class>

<https://stackoverflow.com/questions/8648793/java-isfile-isdirectory-without-checking-for-existence>

<https://www.geeksforgeeks.org/clone-method-in-java-2/>

### **Description of important functions/Modules:**

We have got 10 packages, 3 of them are MVC, the rest are Memento, Command, Invoker, File Chooser View, IOFILE, Utility, Main.

**Memento:**

1. Care Taker
2. Fixed Stack
3. Memento
4. Originator
5. Undo – Redo

**Command:**

1. Change Shape Properties
2. Command
3. Copy Shape
4. Delete Shape
5. Create Shape

**Invoker:**

1. Controller Invoker
2. Invoker Interface

**IOFILE:**

1. File Extension Interface
2. IOFILE
3. Json saver and Loader
4. XML saver and loader

**Main:**

1. Main Class

**Utility:**

1. Container Resizer
2. Utility Class