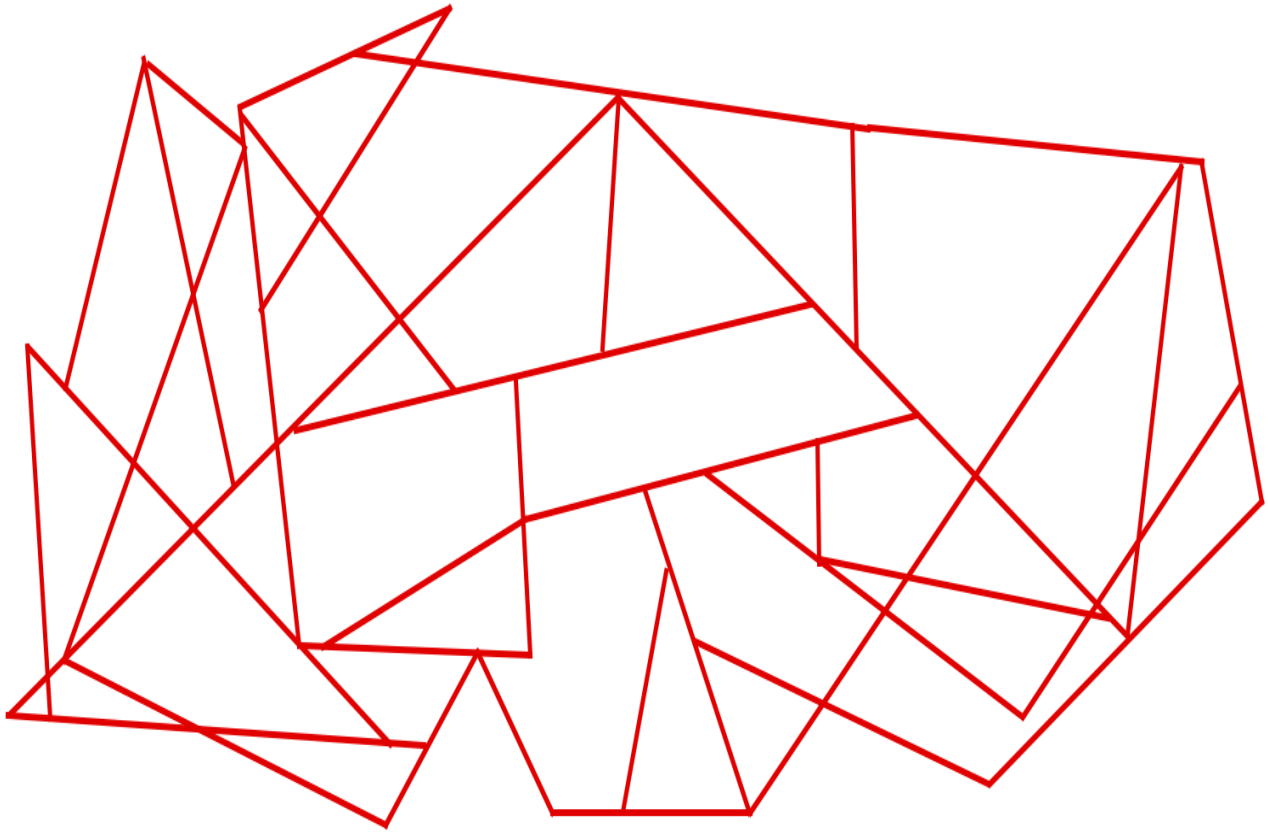
VECTOR BASED DRAWING APPLICATION

OOP Assignment 2



KARIM MOHAMED ALI

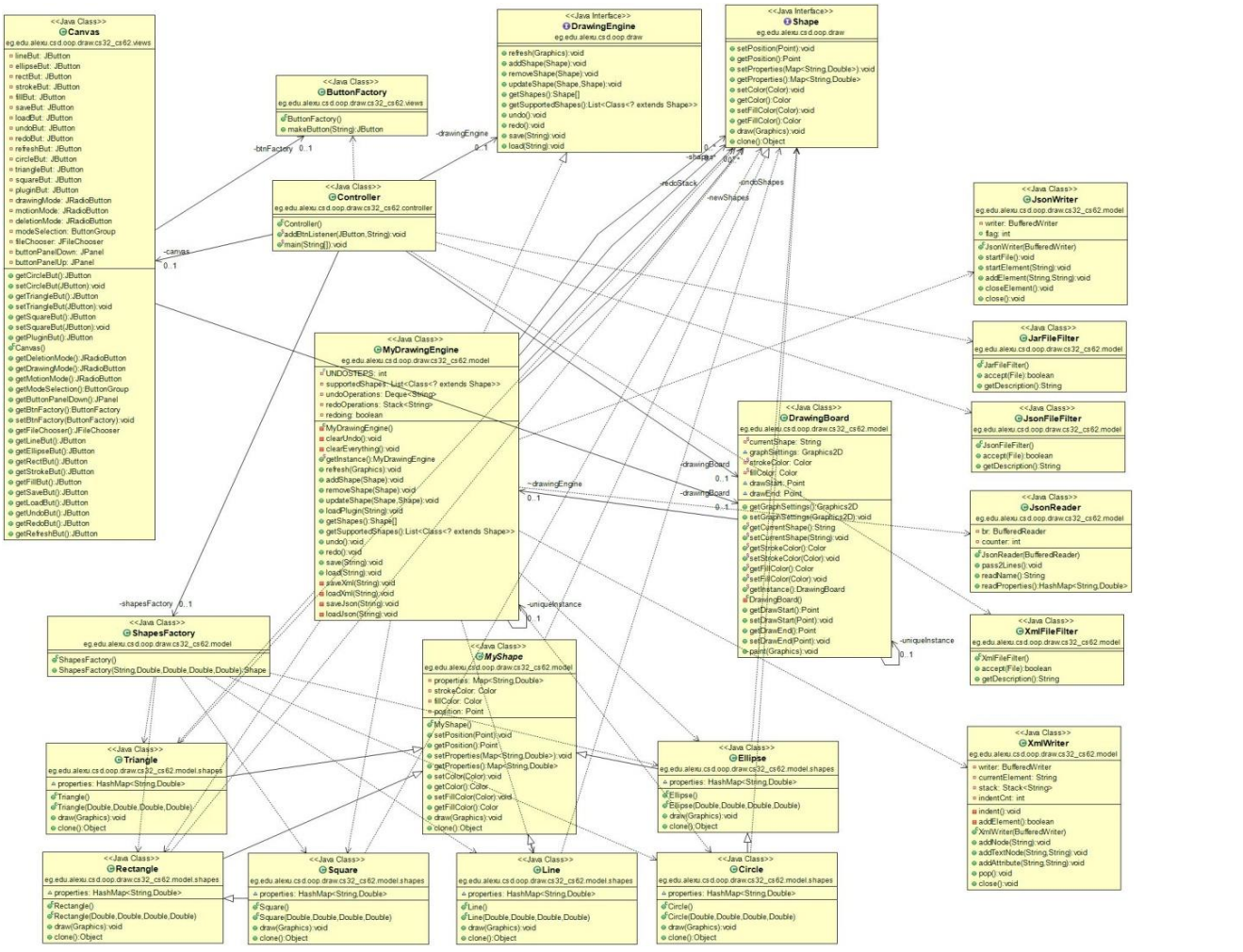
YOUSSEF ABDALLAH YOUSSEF

Application Description and Features:

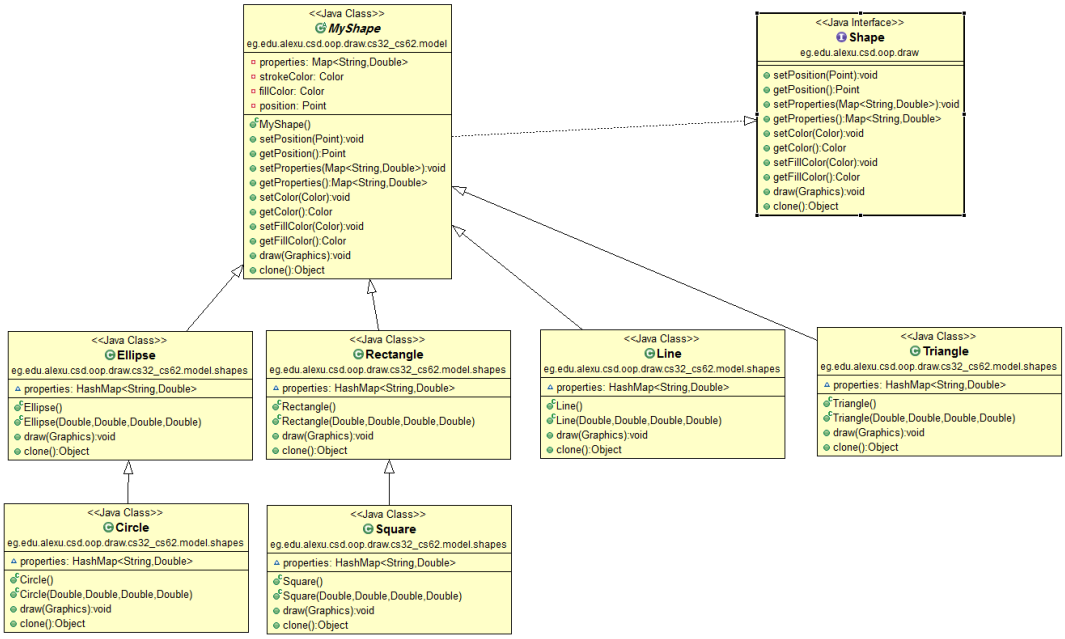
Draw many 2D geometric shapes easily (Line, Rectangle, circle, square,...) change your colors ,resize, delete and move the shapes anywhere on the drawing board with the option of doing undo and redo for a better control on your drawing. You can save your work and load it again to continue drawing. For dynamic extensions, it is possible to add new implemented shapes to your list.

UML:

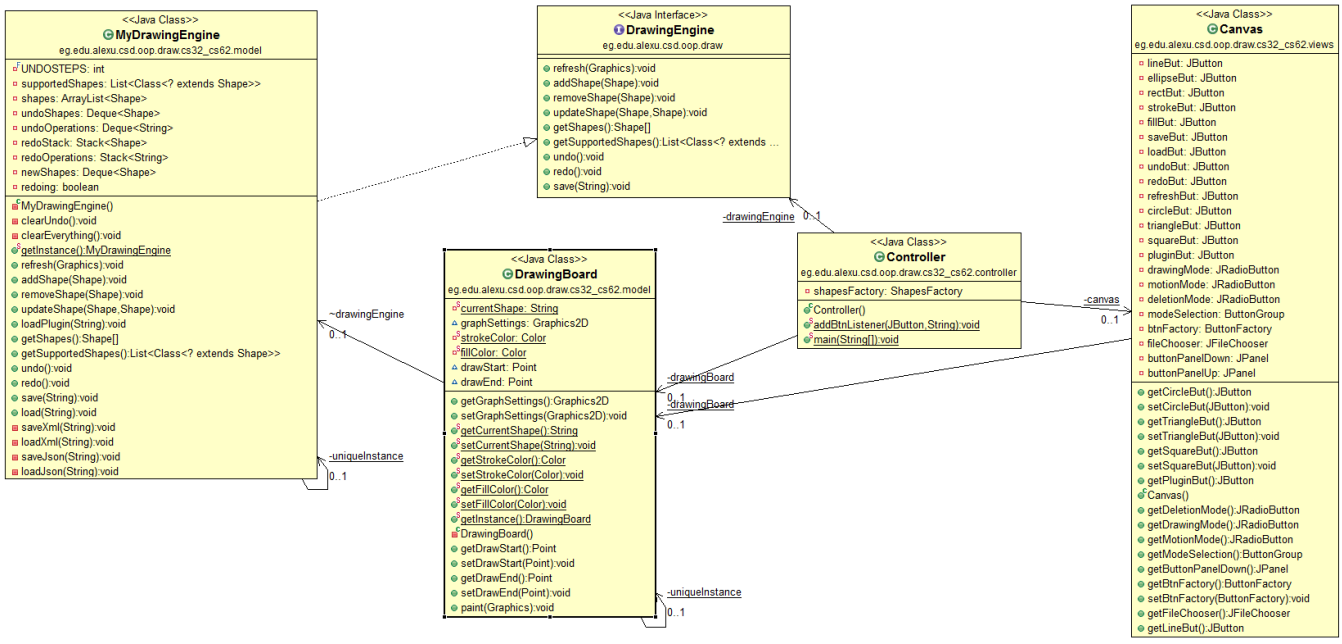
Class diagram for the full project.



MyShape and the classes that extend it.



The main classes in the project.



Design:

Overview:

The design of the project was based on the MVC architectural pattern. Moreover, other design patterns were also used to develop the project such as the factory pattern and the singleton pattern.

1. Model:

In our case the model was represented by classes carrying data such as:

* Public class (MyDrawingEngine) implements DrawingEngine: has all the data needed to add, remove or update different shapes. It is also the main module in the project responsible for operations like undo, redo, save, load and importing plugins.

* Public abstract class (MyShape) implements Shape:

The main container class for all the shapes, containing all the common attributes and behaviors of different shapes such as the position, the shape' properties and the colors. Each concrete shapes (circle, triangle...) extends from this class and has the implementation for the draw() method.

* public class (ShapesFactory): used to create shapes given the shape name and the appropriate properties.

* public class (DrawingBoard) extends JComponent: the main board on which all the shapes are drawn.

* public class (XmlWriter): a helper class to save data in xml file.

* public class (JsonWriter): a helper class to save data in json file.

* public class (JsonReader): a helper class to read data from json file.

JarFileFilter, XmlFileFilter, JsonFileFilter are file filters used in the fileChooser to filter the file extensions.

2. Views:

-The actual Jframe appearing on the screen and all of its constituents such as button panels and the drawing board. Different layouts were used: The border layout for the placement of panels inside the frame and the grid layout for the placement of buttons inside the panels.

-ButtonFactory: used to create buttons based on their names, it gets the corresponding icon for each button and returns an instance of it.

3. Controller:

Performs the communication between the model and the view using mouse listeners: it fires an event when the mouse is pressed, released or dragged. This event is determined each time upon the mode selection (drawing, motion, and deletion).

Additionally, there is a mouse wheel listener used to resize the shapes.

Design Patterns:

Singleton: A creational design pattern used to create classes which we can only have one instance from such as the drawing engine, the drawing board and the canvas.

Factory design pattern: A creational design pattern that uses a template method to create objects like shapes factory class and button factory class.

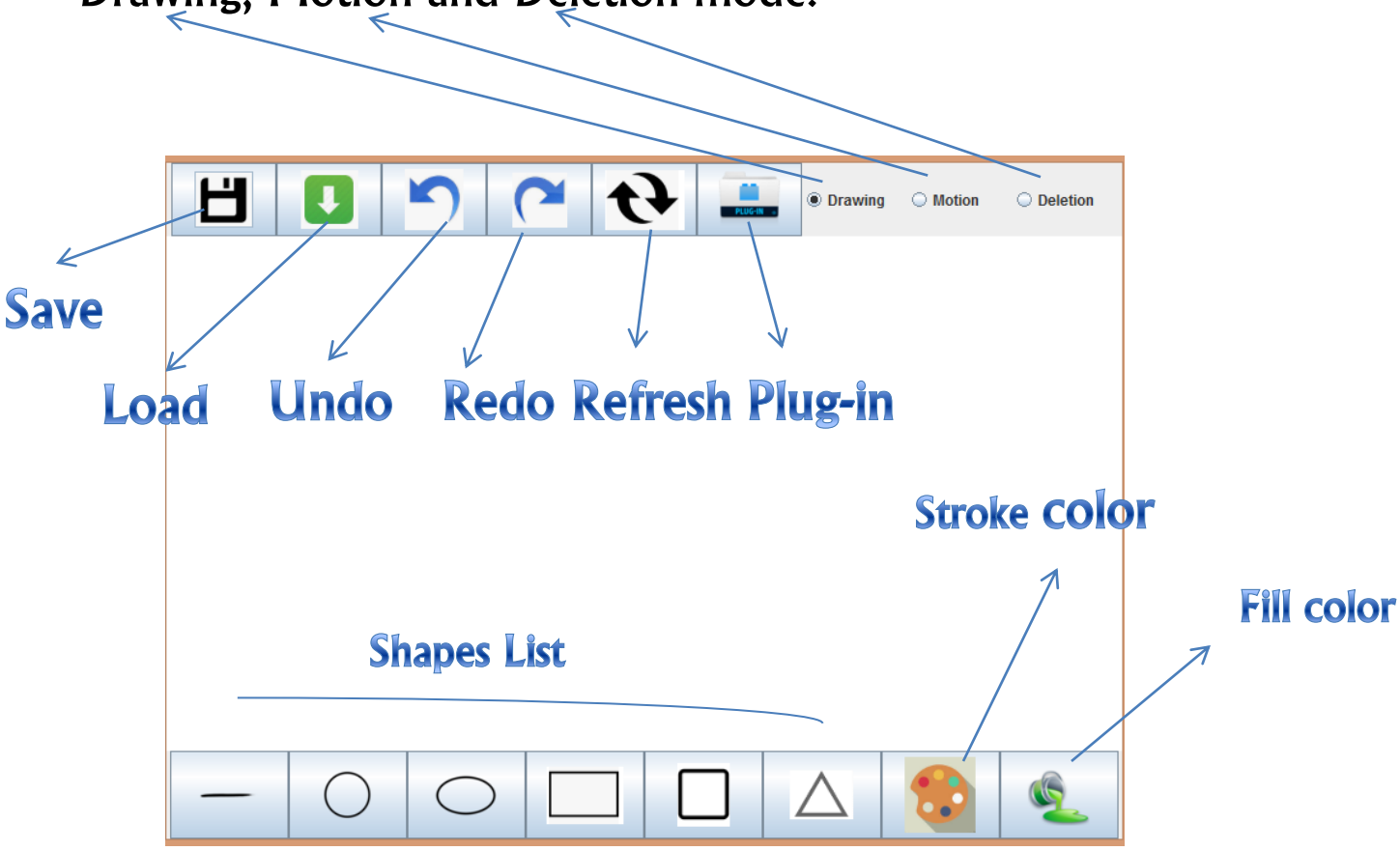
Interface pattern: defines the signature operations of an entity and sets the minimum communication boundary between two entities. Used interfaces: Shape interface and Drawing engine interface. With the help of these interfaces, we managed to dynamically load the class of round rectangle.

Event Listener pattern: a predefined method of the listener is invoked by the producer when an event is fired. It is commonly used in the java libraries such as button listeners and mouse listeners.

User Guide:

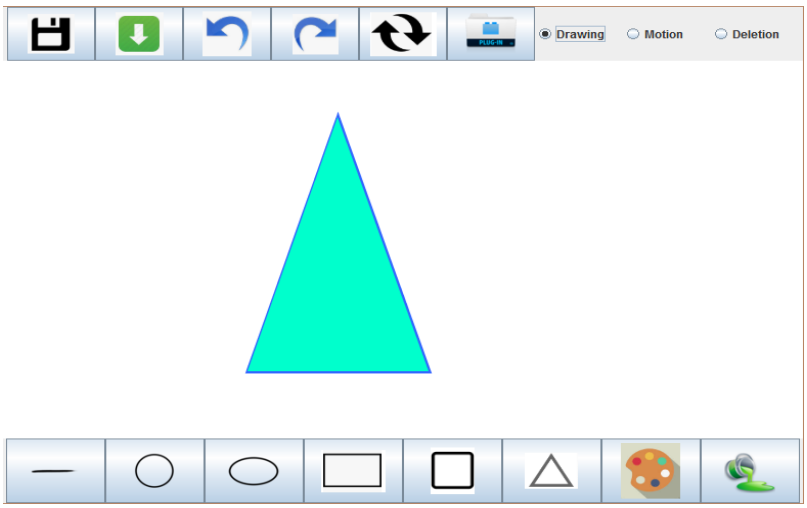
The application works on three main modes:

Drawing, Motion and Deletion mode.



When you are on the default mode of Drawing, pick the shape you want to draw from the shapes list, choose a color to fill the shape and another one for the stroke.

Click on the drawing board and drag your shape which will appear after releasing the mouse.



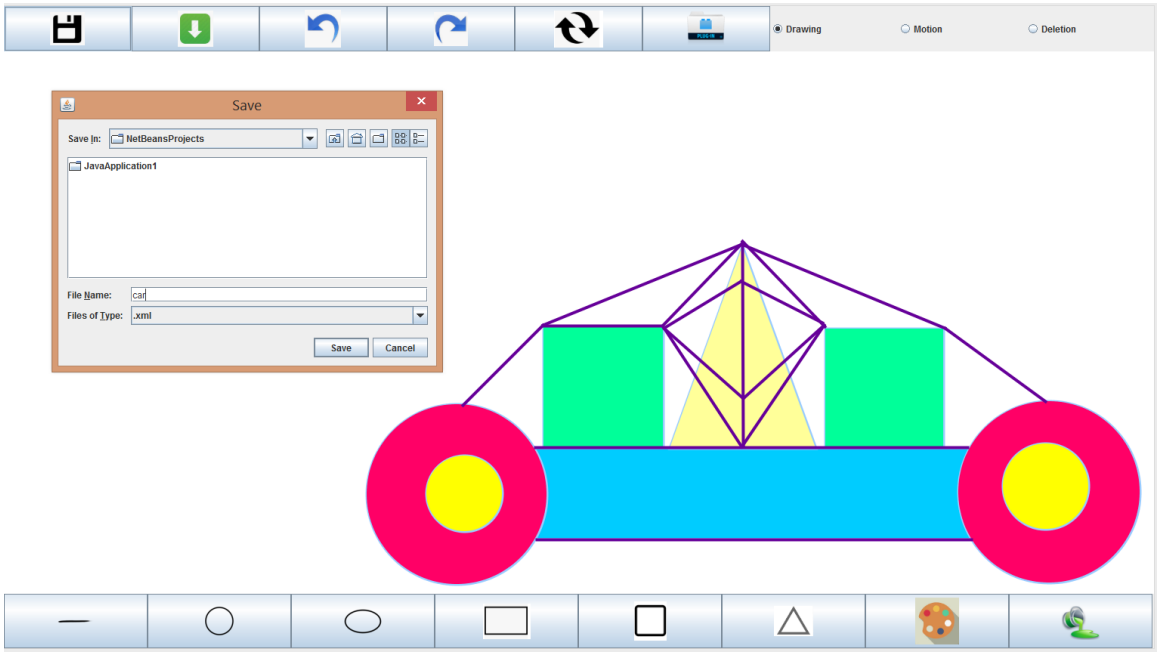
Click “undo” to remove your last drawn shape and “redo” to get it back (up to 20 times undo and redo).

To resize your shape, just put the cursor behind the shape and resize it using the wheel of your mouse (if you are using a touchpad, zoom in and out to resize the shape).

To move the shape, change the mode to motion mode, click behind the shape and drag it to the new position you want.

To delete a shape, move to the deletion mode and just double click behind the shape you want to delete.

If you want to save your work, click the save icon and choose the place to save the file in. You have to name your file first and choose the extension of the file (xml or json).



Load it by clicking the load button and choose the file from the same place.

To use the plug-in option, click the plug-in button and choose the jar file that has the new shape.

Then a new button will be added to the shape list the loaded shape