



Paint Assignment

Names:

- Mohammed Abdel-Barry Mahmoud
- Anas Salem Harby

Git repository:

- <https://github.com/AnasHarby/Paint>
- <https://bitbucket.org/Mohammed-Abdelbarry/paint>

1. Problem Description

Design of an object-oriented model for geometric shapes, with basic features of (moving, deletion, resizing... etc.).

Support of basic operation (save, load, undo, redo) and some advanced operations (JSON/XML parsing, Dynamic Class Loading).

2. Design

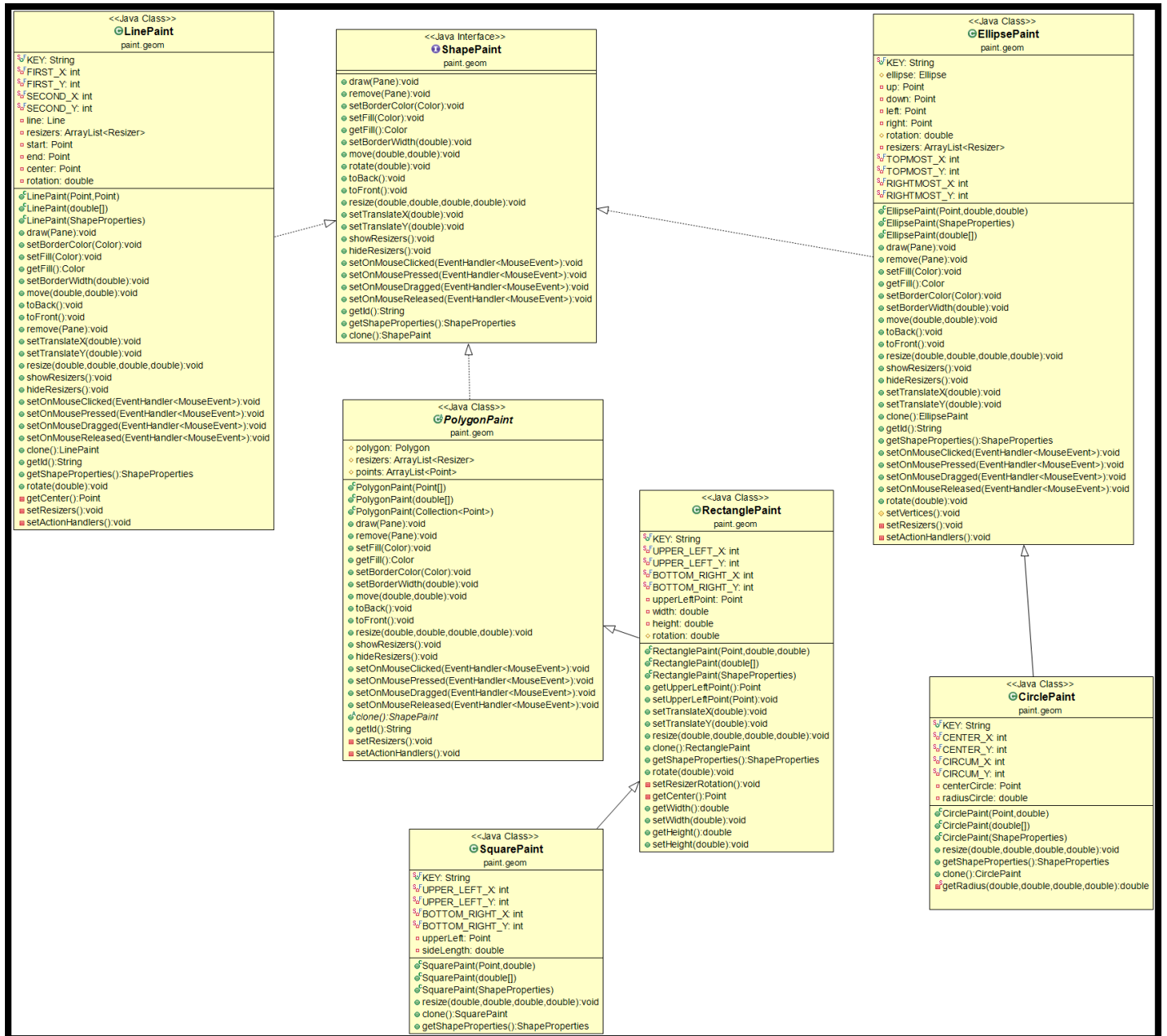
The project is divided into these packages:

- *paint.geom*:
 - Provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry.
 - Follows a hierarchy of shapes all implementing a specific interface for shapes that holds all the general specifications.
 - Consists of the classes (Shape, Line, Ellipse, Circle, Polygon, Rectangle, Square, Triangle).
 - Uses JavaFX library for interacting with the GUI.
- *Paint.geom.util*:
 - Provides some utility classes that interact with the *paint.geom* package providing extra control utilities.
 - Classes:
 - Resizer: A class that binds its properties to specific points on a shape providing the capability to resize this shape with dragging when shown on a pane in GUI.
 - ShapeController: Adds all the basic handlers for mouse events to a shape providing extra features (moving a shape with dragging, selecting a shape, resizing a shape... etc.).
 - ShapeFactory: Following factory design pattern, the class is used for the creation of objects without showing the process of creation to the user.
- *paint.geom.prop*:
 - Contains property-holders for each class in *paint.geom*, which in turn are used in the save/load process as a template for JSON/XML file types.

- *paint.data.util:*
 - Provides some utility classes for data manipulation.
 - Classes:
 - *DataHandler*: Saves/loads files to/from JSON/XML file types, follows Singleton design pattern.
 - *JsonDataParser*: Parses data to/from JSON.
 - *XmlDataParser*: Parses data to/from XML.
- *Paint.data.util.history:*
 - Provides some utility classes for saving history of operations done on shapes.
 - Classes:
 - *History*: Contains the implementation of undoing/redoing any series of *HistoryEvents*.
 - *HistoryEvent*: Behaves as a data-holder for any specific state in history, containing all of the elements (shapes) in this particular state with the access of updating any state.
 - *CurrentHistoryEvent*: Following Singleton design pattern, this class stores the current state of history with the access of updating this state.
- *paint.plugins:*
 - Consists of a single class *PluginLoader* that loads a plugin from *.class and *.jar data types.
- *paint.gui:*
 - Contains the application GUI.
 - Classes:
 - *GUIMain*: Contains the main method for starting the application.
 - *FXMLController*: Contains all the controls and interactions needed between the GUI and the previously mentioned models, providing all the features to run on the GUI.
 - The GUI follows the JavaFX schema for design of a view, which uses XML files to specify the connections and properties of each element on GUI, and uses CSS files to style any element of them.

3. UML

- *paint.geom*:











• Assumptions

- When loading a shape class from a *.jar file, the *.jar file name **must** have the same name as the class it holds.
- When loading a shape class from a *.class file the *.class file **must** be in the package directory (/paint/geom/.../*.class).
- When loading from a JSON/XML file, the shapes stored in file **must** be all previously dynamically loaded from a *.class or a *.jar file if it's not one of the generic shapes (line, ellipse, polygon).
- The button used for choosing a shape is disabled until the shape is dynamically loaded except for the generic shapes (line, ellipse).
- The pencil, eraser buttons are disabled until the freehand drawing feature is added.

• User Guide

- Main window:

- New:  creates a new project.
- Color picker and border width selector: . The color picker shows a yellow square with the hex code #e6b34d. The border width selector shows a dropdown menu with 1px selected.
- Undo/Redo: . The Undo button has a curved arrow pointing left, and the Redo button has a curved arrow pointing right.
- Shapes: . The toolbar contains icons for a line, rectangle, rounded rectangle, triangle, ellipse, and circle.
- Rotate a shape: . The icon shows a square with a circular arrow around it.
- Fill a shape: . The icon shows a square with a diagonal line from the bottom-left to the top-right.
- Pick a color of a shape: . The icon shows a pencil drawing a line.
- Remove a shape: . The icon shows a square with a horizontal line through the middle.

- Screenshots

