Paint Web Application

Credits: Fareeda Mohamed Ali Abouzed

I. Problem Statement

- Part One __ Geometric Shapes Data Model

Design an object-oriented model that covers the following geometric shapes: Line Segment, Circle, Ellipse, Triangle, Rectangle and Square.

- Part Two __ Drawing and Painting Application

Design and implement a GUI that allows the following functionalities for the user on all the shapes defined in Draw, Color, Resize, Move, Copy, and Delete.

Implement your application such that it would allow the user to undo or redo any action performed.

- Part Three __ Save and load Description

Provide an option in UI to save the drawing in XML and JSON file, Provide an option to load previously saved drawings and modify the shapes, Users must choose where to save the file.

II. How To Run

- Clone the repository
- Make Sure you have Angular v12, Spring Boot v2, nodejs, npm, and mysql DB on your PC.
- To install packages for Frontend, Navigate to the directory (../Paint/Frontend) and using terminal write:
- `$npm install`
- To run the Frontend, Navigate to the directory (../Paint/Frontend) and using terminal write:
- `$ng serve --open`
- The Frontend will run on (http://localhost:4200)
- To install dependencies for Backend, file pom.xml will do the job for you.
- Add in (application.properties) your mysql username and password if found.
- Add again in Class (ShapeService) in Method (restartTable) for (getConnection) Arguments the username and password.
- Make sure you have a database named (shapes).
- You could make that test through terminal using:
- `$sudo mysql -u root -p` (sudo for linux users)
- `$SHOW DATABASES;` (if you have shapes you're done otherwise)
- `$CREATE DATABASE shapes;`
- MySQL DB will run on (jdbc:mysql://localhost:3306/shapes)
- The Backend will run on (http://localhost:8080)
- Enjoy the Paint :)

III. Software Design

○ Frontend (Client Side)

The most important two packages:

- app

The app package has HTML, CSS, TypeScript files which help providing friendly UI for the user, and using **Canvas API**, drawing the shapes.

- service

The service package contacts with the Backend using **HttpClient API**

○ Backend (Server Side)

The most important two packages:

- controller

The controller package has One Single Class (Controller) that contacts with the service package from the Frontend.

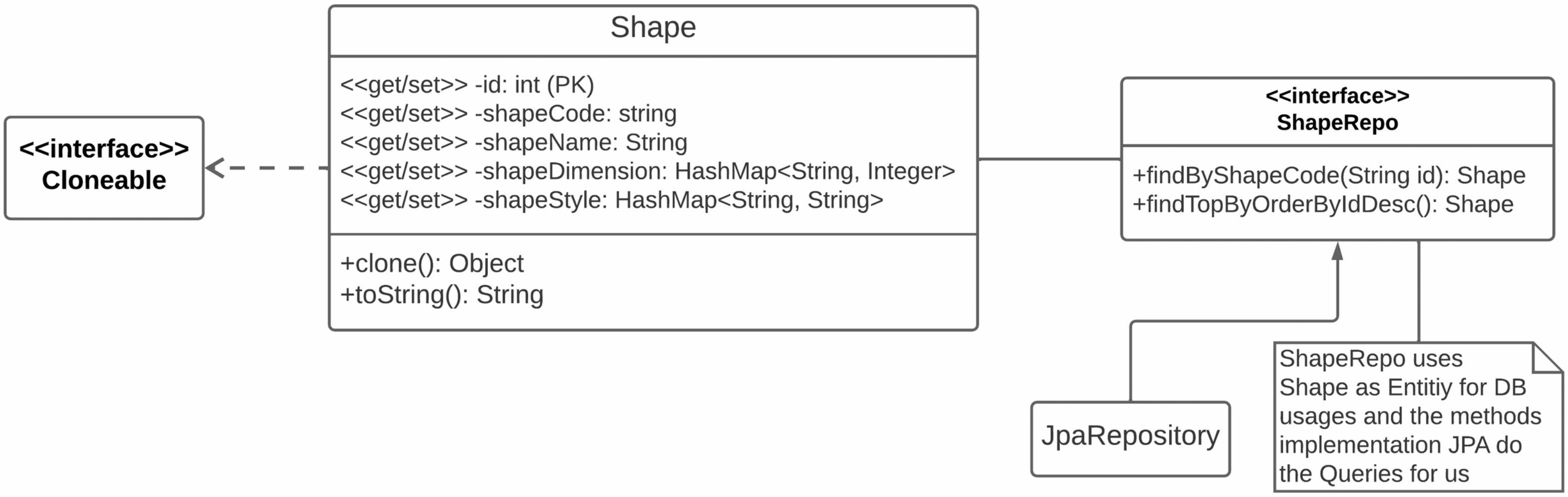
- model

Here, In the model package, We have some packages implemented using some DP that we desire to go through.

- shapes

Shape class is used as Entity for Database usage as (id, shapeName, shapeDimensions, shapeStyle, shapeCode) represented as the columns of the Table shape in shapes DB.

UML Class Diagram For Shape


Shape
<<get/set>> -id: int (PK)
<<get/set>> -shapeCode: string
<<get/set>> -shapeName: String
<<get/set>> -shapeDimension: HashMap<String, Integer>
<<get/set>> -shapeStyle: HashMap<String, String>
+clone(): Object
+toString(): String
<<interface>>
Cloneable
<<interface>>
ShapeRepo
+findByShapeCode(String id): Shape
+findTopByOrderByIdDesc(): Shape
JpaRepository
ShapeRepo uses Shape as Entitiy for DB usages and the methods implementation JPA do the Queries for us

- ➢ command

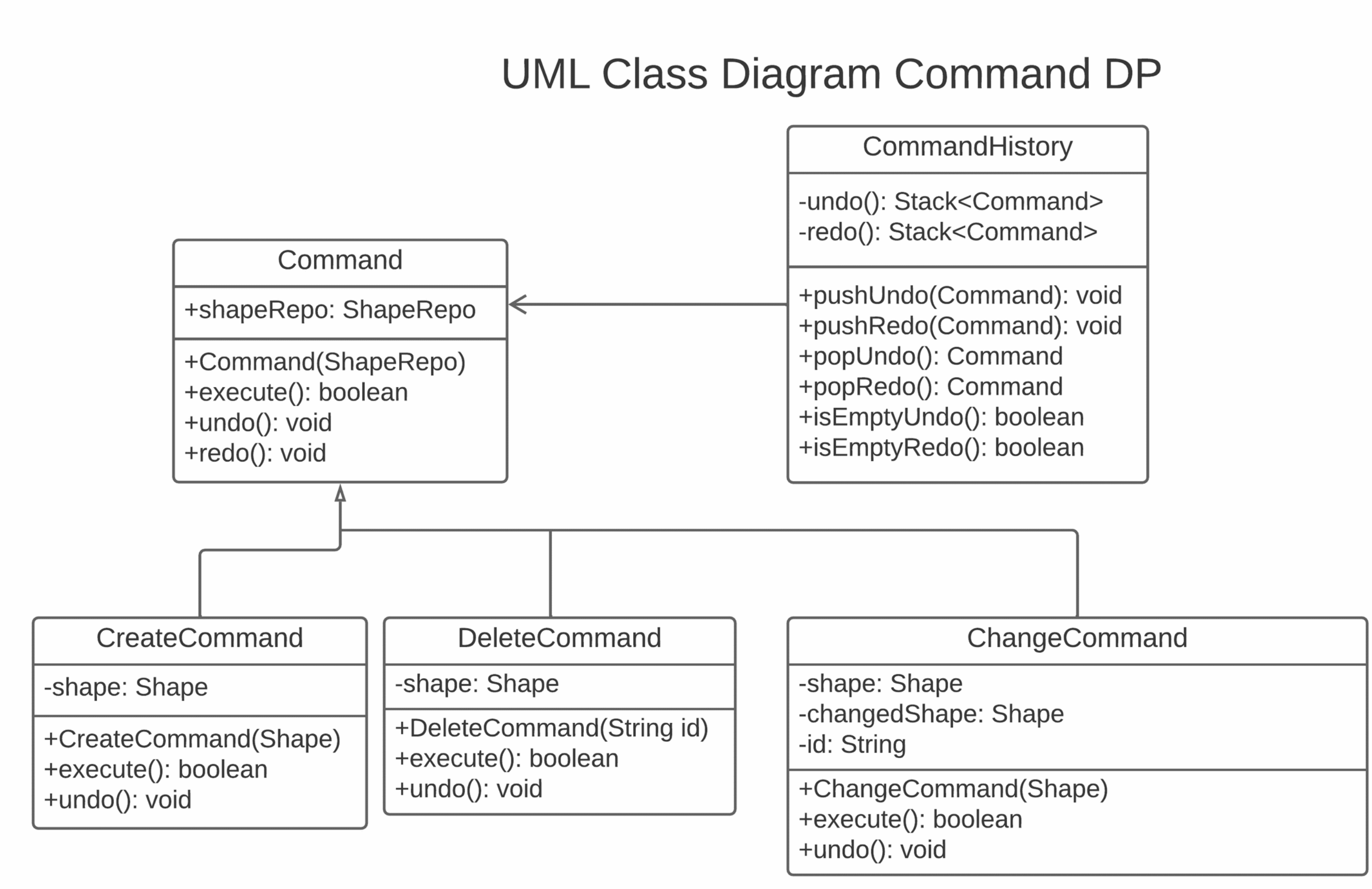
The operations Supported Like:

- Create Shape
- Delete Shape
- Copy Shape
- Change Attributes For a Specific Shape

To be undo & redo using Behavioral Pattern (Command), that turns a request into a stand-alone object that contains all information about the request.

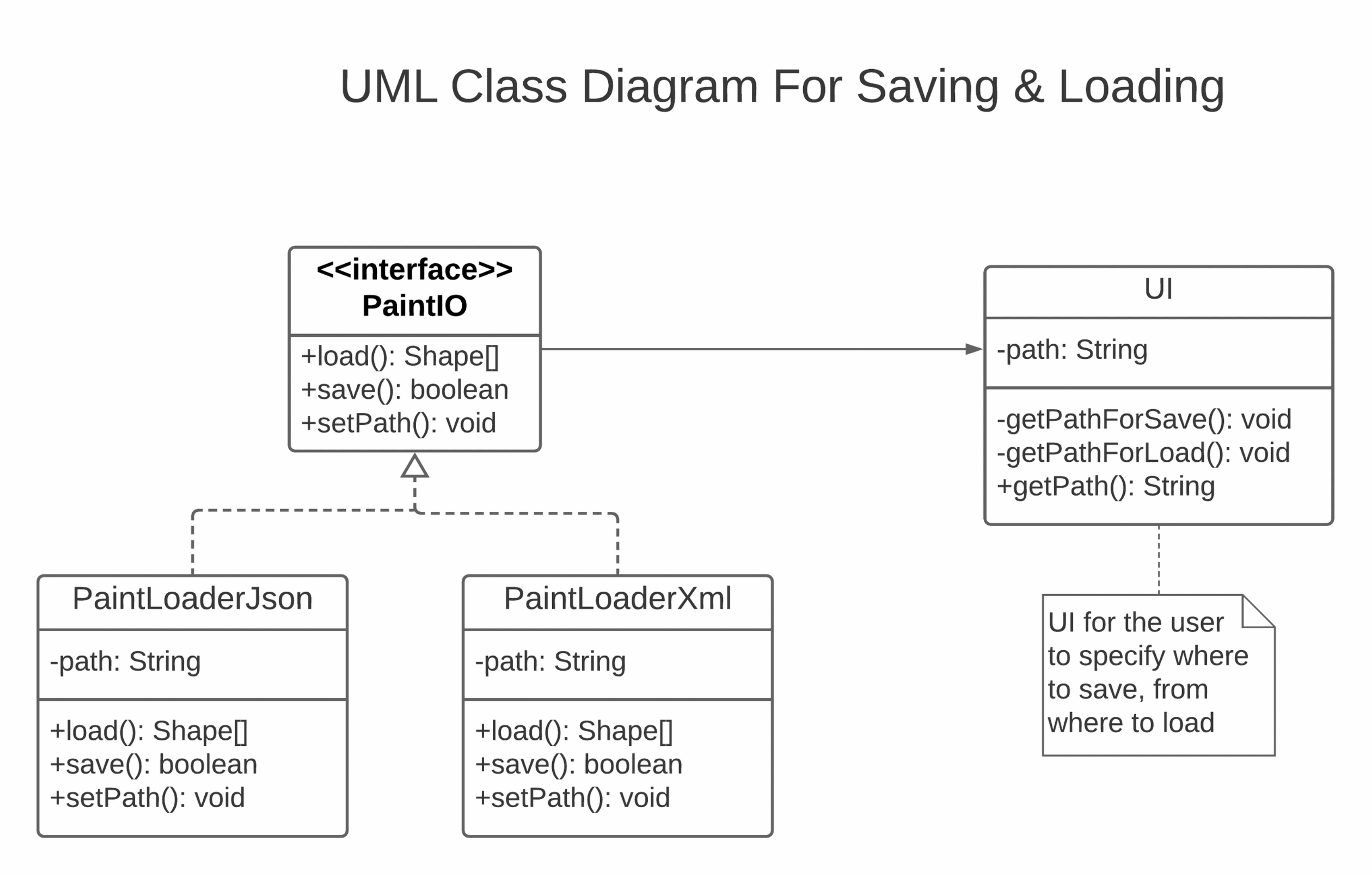
The HistoryCommand has undo & redo stack.
For undo pop the Command from undo stack, and push the Command on redo stack.

For redo pop the Command from redo stack, and push the Command on undo stack.


UML Class Diagram Command DP

- Io

The io package for save & load according to the chosen format JSON or XML.
The user has the UI Class that expose a window contains system directories to chose the place to save in, and the place to load from.

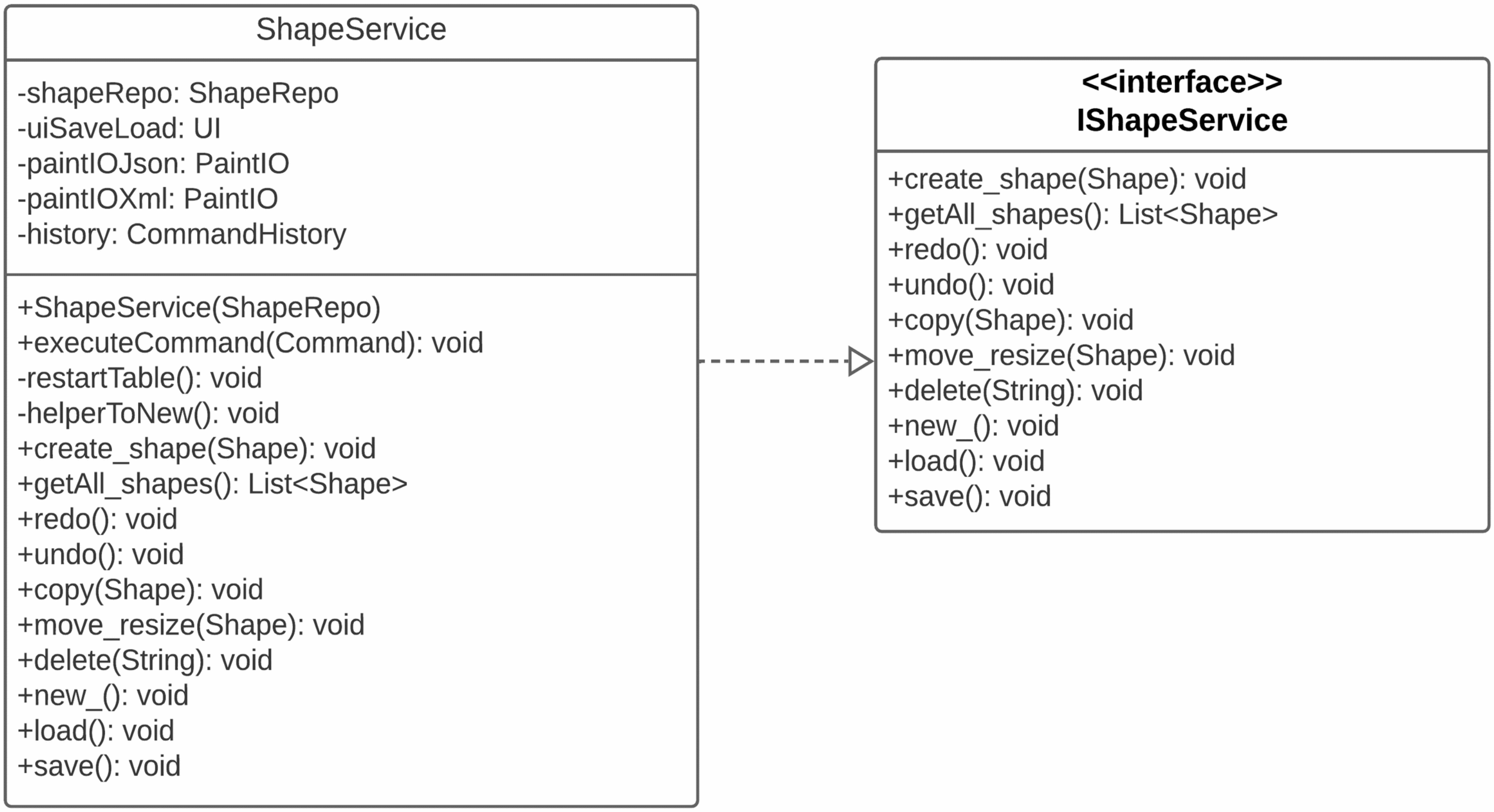


- service

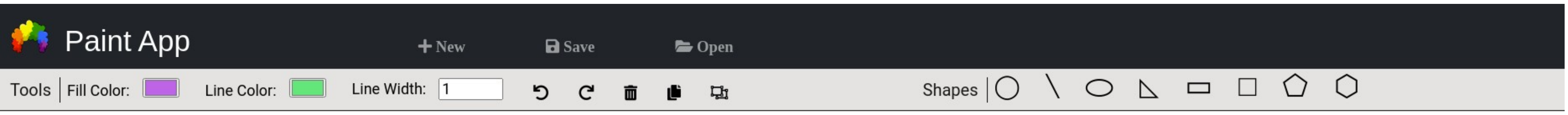
The service package for the main run as contains all the Shapes during runtime through the ShapeRepo which presents the DB and provides the service to the Controller.

ShapeService works as a Singleton Class.

UML Class Diagram Service Layer



IV. GUI Overview

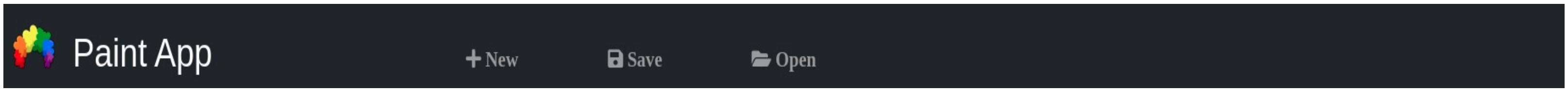


The Window consists of:

- The Main NavBar
- Tools and Provided Shapes
- The stage For drawing

V. User Manual

- The Main NavBar



The main NavBar provides three Icons:

- ➢ New

The New Icon to start a new drawing, as a restart.

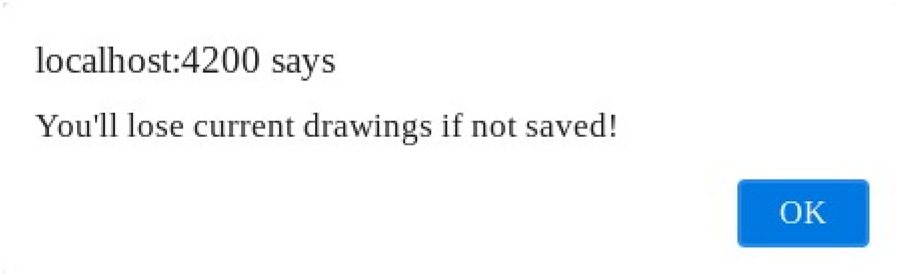
- ➢ Save

The Save Icon to save the current drawing, a window will appear to choose a path to save in and one of the provided format JSON, or XML.

- ➢ Open

The Open Icon to load a saved drawing, a window will appear to choose a path to load from one of the provided format JSON, or XML.

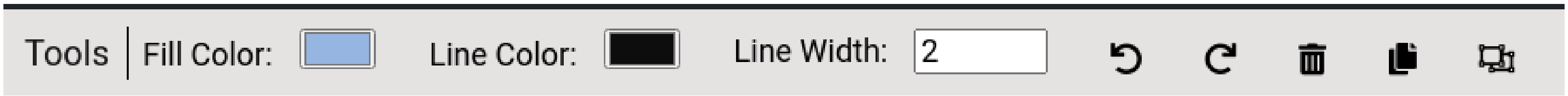
Trying to Load a drawing when one is already on stage the user will get an alert:



localhost:4200 says

You'll lose current drawings if not saved!

- Tools



Tools | Fill Color: Line Color: Line Width: 2

The Tools bar provide:

➢ Fill Color & Line Color

The Input Fill Color & Line Color to pick a color from HTML picker, just a Left click to appear and you could choose a specific color using RGB format.

➢ Line Width

The ranger Line Width to choose from (1 : 100), left click for the cursor to appear then write the width you need.

➢ Undo

The Undo Icon to undo the last operation.

➢ Redo

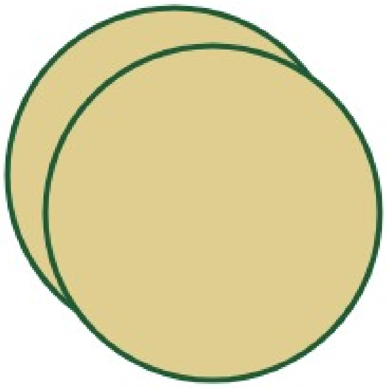
The Redo Icon to redo the last undo.

➢ Delete

The Delete Icon to delete the selected shape.

➢ Copy

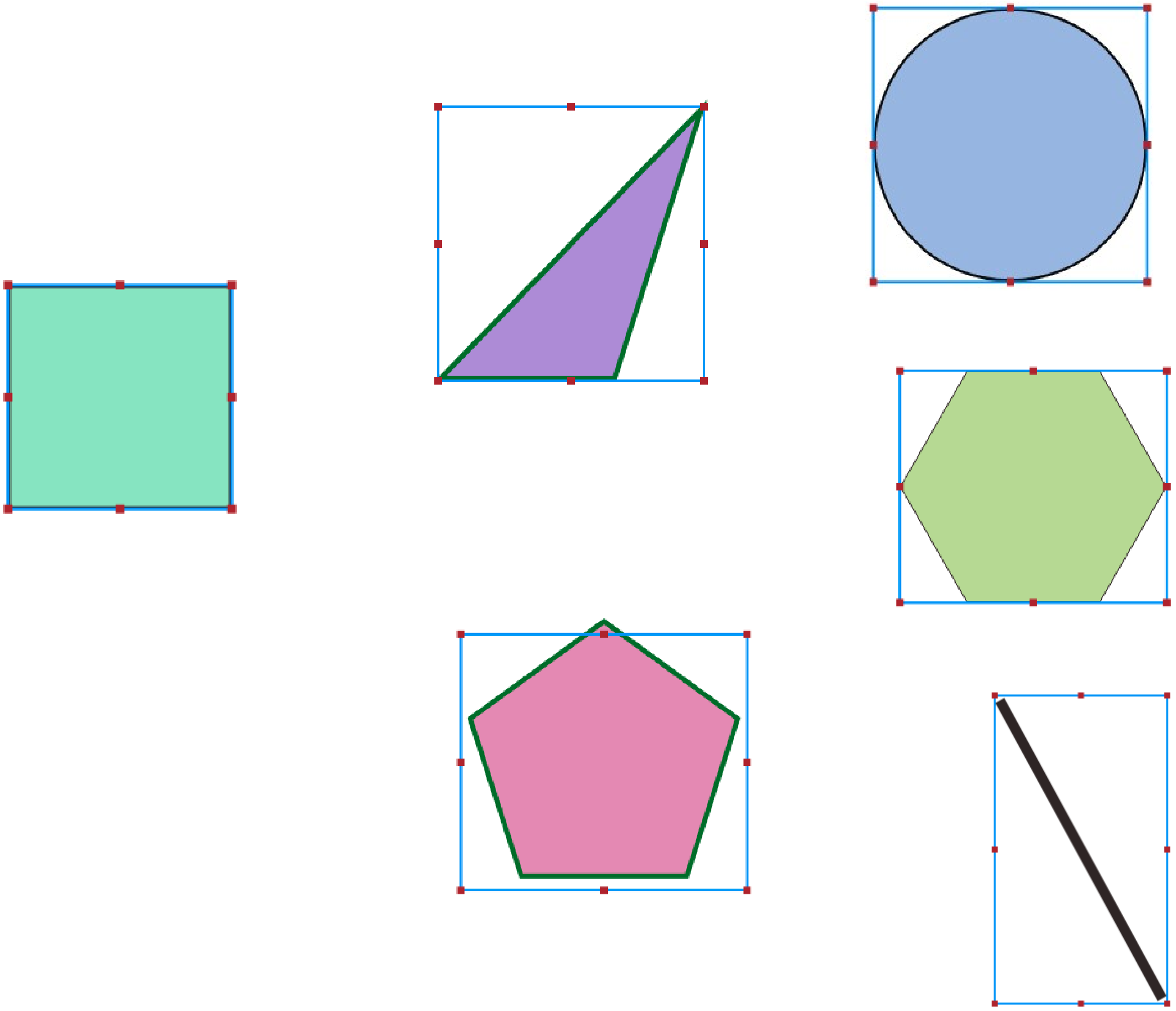
The Copy Icon to copy the selected shape, the copied shape will appear at 20 distance from the selected shape.



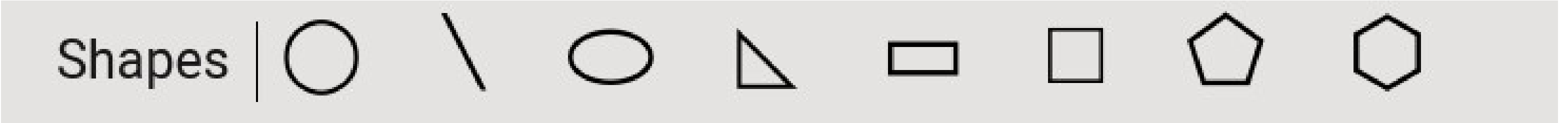
➢ Move & Resize

The Move & Resize Icon actually does some stuff.

- Left Click on the Move & Resize Icon.
- Left click on the desired shape.
- Box of Selection will appear around the selected shape.
- The Selected Shape could be dragged to the required place.
- The box with eight handlers to resize.
- The cursor is shaped for each one of the eight handlers to provide friendly resizing.
- The selected shape’s attributes (Line Width, Fill Color, Stroke Color) could be changed.



- Shapes



The Shapes Bar provides the available shapes for the user:

- Circle
- Line
- Ellipse
- Triangle
- Rectangle
- Square
- Pentagon
- Hexagon

Left Click on the required shape.
The first click on the stage is the starting point, then while mouse is moving the shape is shaped until the mouse is up.
When mouse is up, the shape will be drawn as a final.

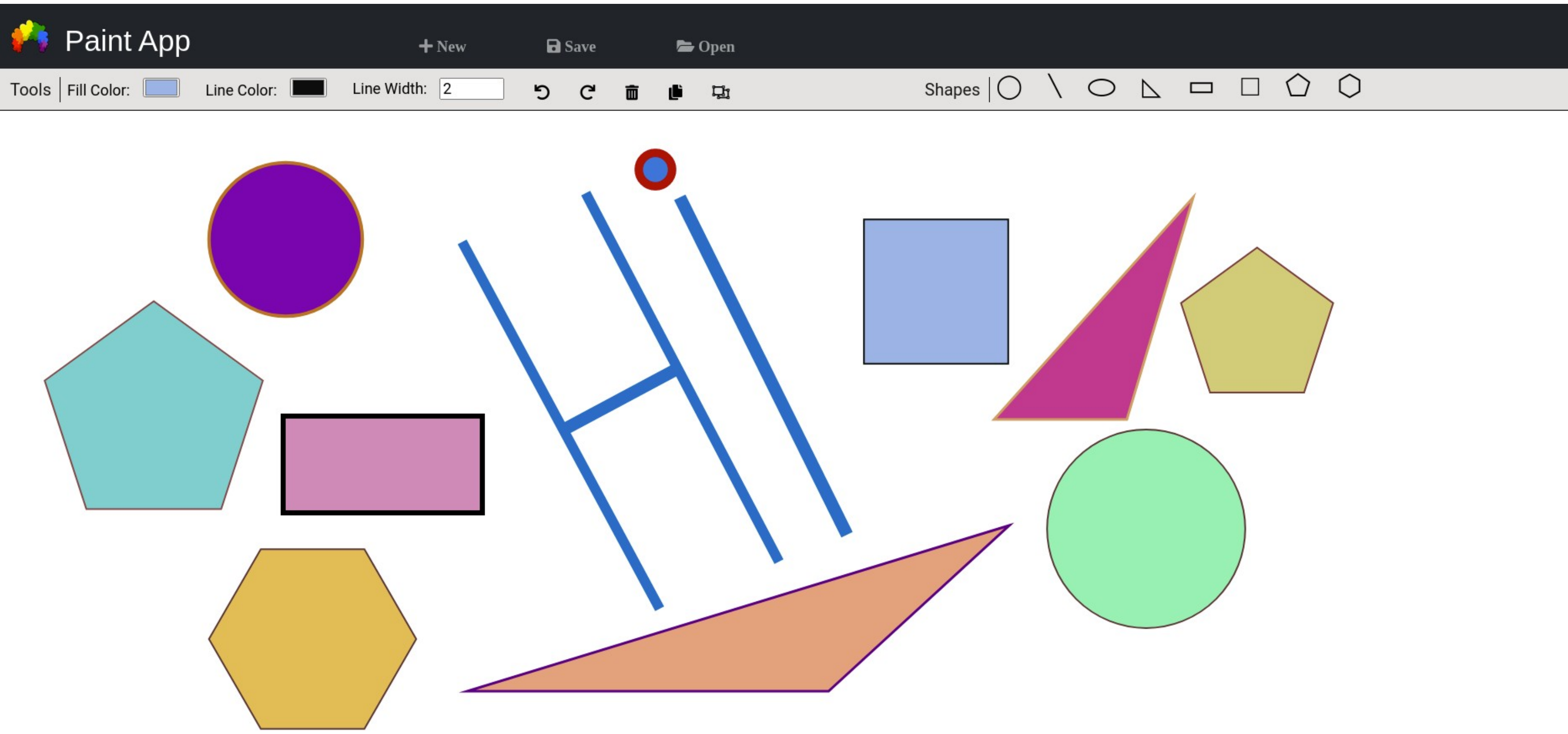
If any operation is done, then you have to left click again on the required icon for shapes.

- Keyboard Event Listeners

The application provides some keyboard Event Listeners for the basic operations:

- Ctrl + z – undo
- Ctrl + y – redo
- Ctrl + c – copy
- delete – delete
- Ctrl + s – save
- Ctrl + o – load
- Ctrl + n – new

VI. After Drawing



Line Width: 2