Alexandria University
Faculty of Engineering
CSED 2025

OOP Lab 3
Report

Submitted to:
Eng. Ismail El-Yamany
Faculty of Engineering Alex. University

Submitted by:
Islam Yasser Mahmoud 20010312
Ehab Yasser Mahmoud 20010382
Marwan Yasser Sabry 20011870
Mkario Michel Azer 20011982
Faculty of Engineering Alex. University

Dec. 2022

Problem Statement:

A web-based paint program similar to that of windows whose operations are done on the server.

Features in the program:

- Free hand drawing.
- Shapes drawing (stroked or filled).
- An Eraser for deleting.
    - Any chosen option of the above 3 will have a green color to let you know what you are drawing.
- A Slider to choose the width of your drawings.
- Move drawings.
- Resize drawings.
- Rotate drawings.
- Recolor drawings
- A Select button to decide whether you want to select a drawing to do any of the above 4 operations on it or just continue drawing according to the button state enabled or disabled, you will easily know that as if it is enabled it has a green color if not a red one is showed.
- A 24-color pallet to choose the color **besides** a color picker to choose any color you want.
- 2 buttons to decide where you want to apply the chosen color in the border or the body of your shape. They take the color chosen on them to let you know which color you are using at the moment.
- An undo button. To undo any move.
- A Redo button. To redo any undone move.
- A copy button. To clone any drawing on the board.
- A clear button. To clear the board.
- Save button. To save any painting you want. By entering the path and the name at their specified places and choose whether you want your painting to be saved as JSON or XML file.
- Load button. To load any saved painting. By entering the path and the name at their specified places and choose whether you want your painting to be saved as JSON or XML file.

Required steps to run the program:

1. Extract the compressed program folder.
2. Run the back end:
    - Open the Paint_Back folder using IntelliJ IDE (recommended) or any other java IDE , or open it by simply running the pom.xml file
    - Run the PaintBackApplication on: src/main/java/com.example Paint_Back/Model.
3. Run the front end:
    - You should have NodeJS and Angular-CLI if you don’t download them.
    - Open the Paint_Front folder using VS code IDE (recommended) then copy this to your terminal npm install and press Enter.
    - Copy this to your terminal ng serve and press Enter. Then the paint program is going to run on localhost:4200 copy it to your MS edge end press Enter the

interface of the program is going to show up and enjoy 😊. (You can replace MS edge by Chrome but first you have to open launch.json in the Paint_Front from inside the VS code IDE and replace the 2 msedge written in the page by pwa-chrome and chrome **respectively.)**

Note that:

- To run the program, you have to stick to the previous steps with their order and if you restarted anyone of them for any reason you have to restart the other and put the order into consideration.
- To enjoy the full experience work on:
  - 100% zoom (or less) For(computers).
  - 80% zoom (or less) For(laptops)
- If there is a problem with the libraries copy these to the terminal of the Front end:
  - npm install primeng@14.2.1 –save
  - npm install @syncfusion/ej2-angular-base

User guide:

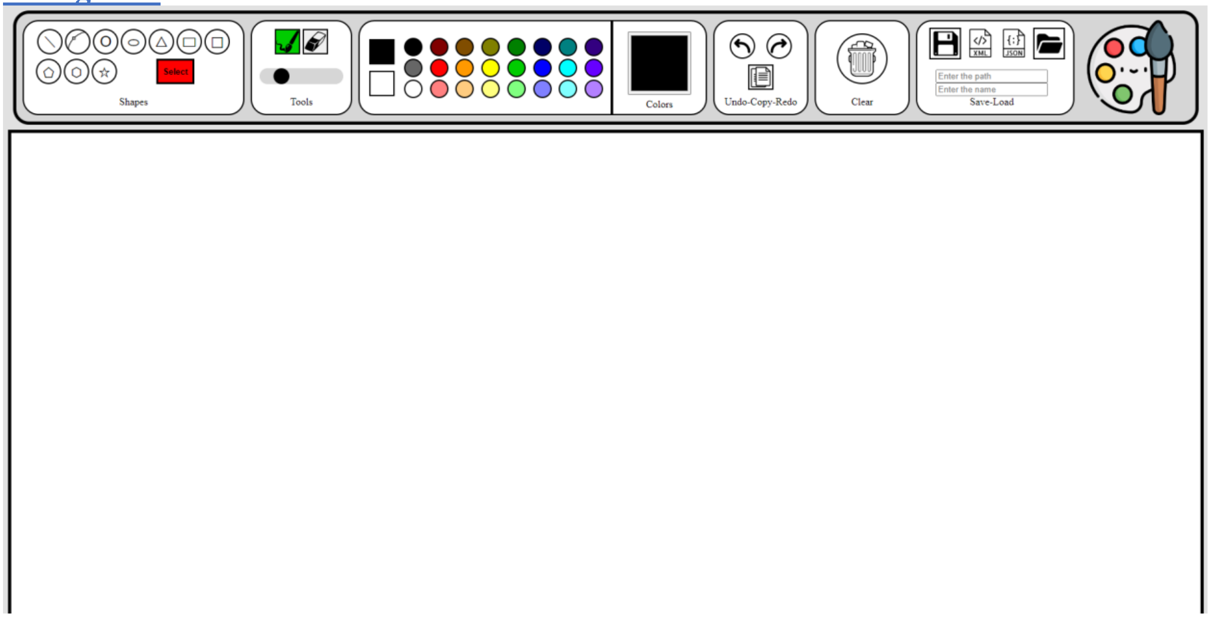


Figure 0-1: the opening interface

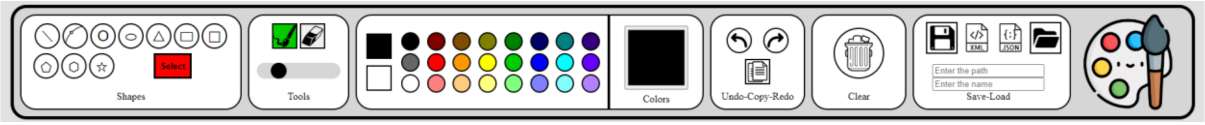
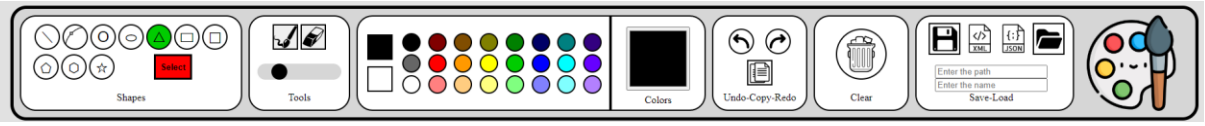


Figure 0-2: the toolbar

Fig 0-2: 1) As you can see in the toolbar the brush button is green (by default) indicating that the selected item to draw with now is the brush and you can choose any other item from the shapes and the eraser. Like the following:



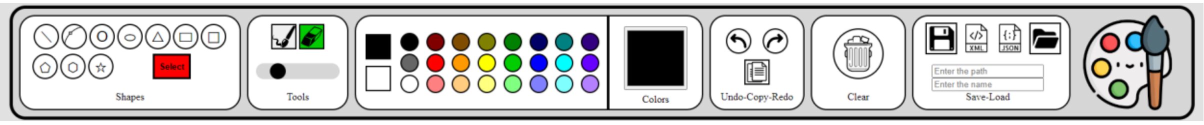
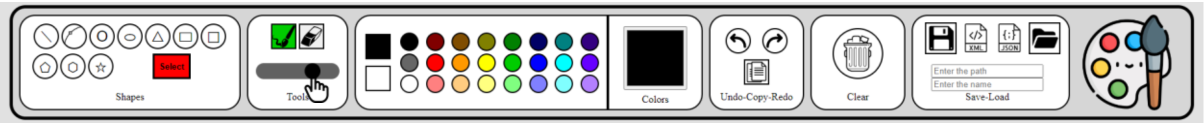


Fig 0-2: 2) we can also see in the tools section a slider it is responsible for choosing the desired width by just moving the thumb. Like the following:



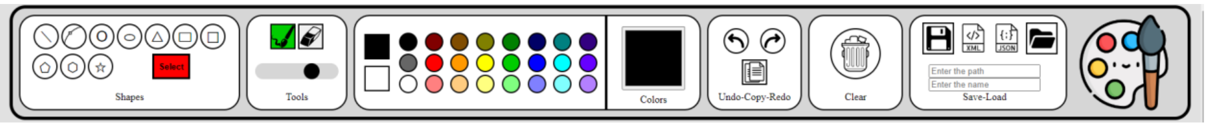


Fig 0-2: 3) we can also see a select red button it indicates whether the selection of items is enabled(green colored) or not to perform MOVE,RESIZE,ROTATE,COPY on them. Like the following:

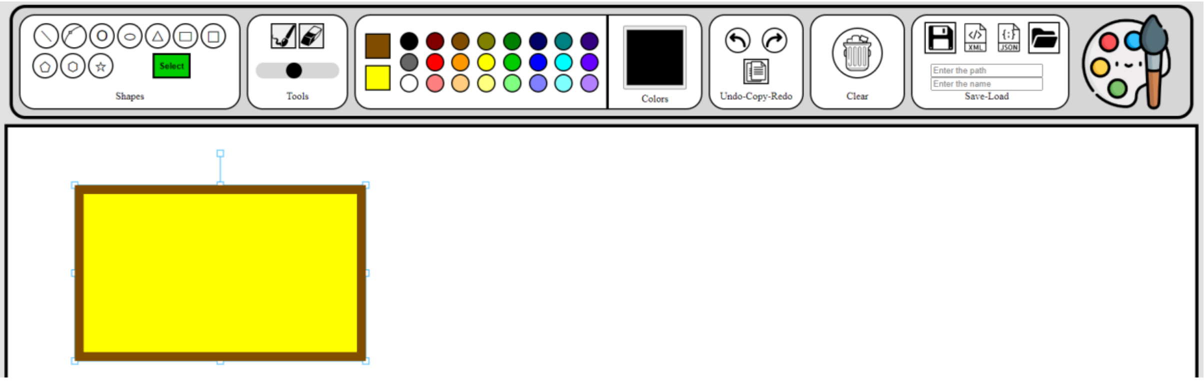
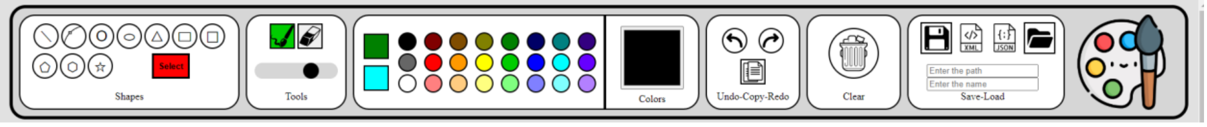
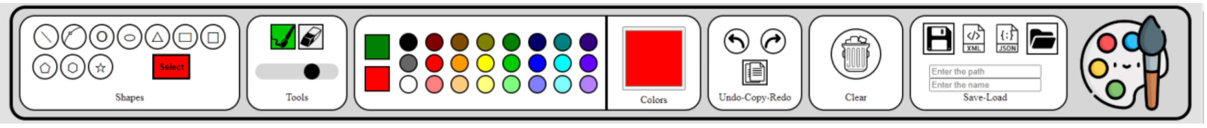


Fig 0-2: 4) we can also see a colors section consists of 2 big buttons one for the border color and the other is for the fill color and a 24-color palette that you can choose from it by just clicking on the desired color and a color picker that you just click on it, and it will open to choose any color you want. All these color choices are applied to the part button that you want. Like the following:

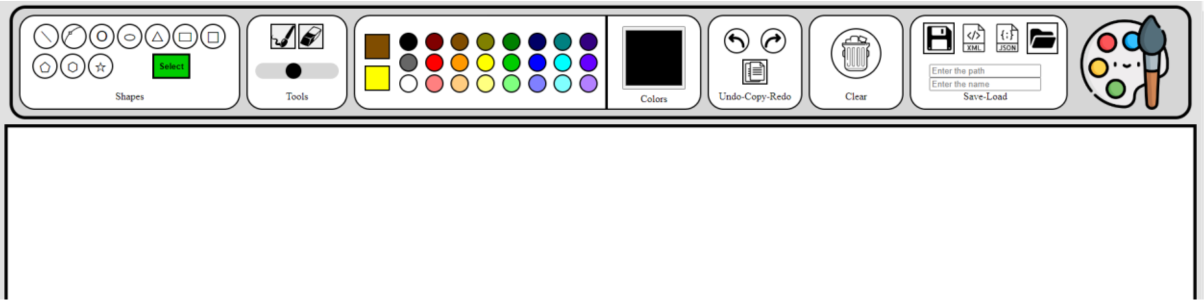


Using the palette

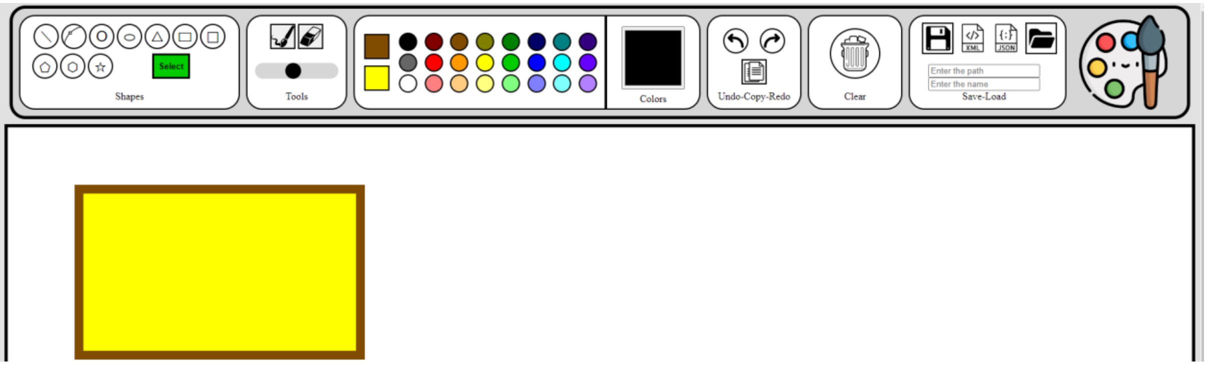


Using the color picker

Fig 0-2: 5) we can also see an undo button the undo any action and a redo button the redo any undone action. Like the following:

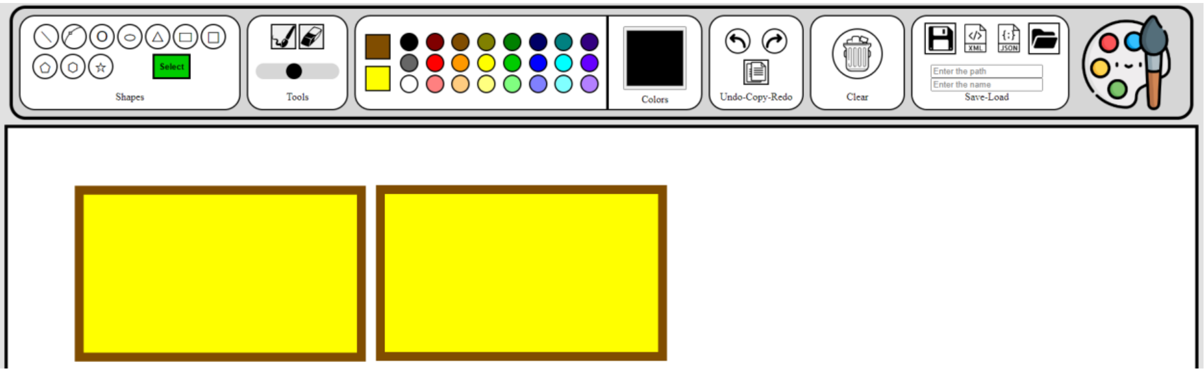


we undone the rectangle that we have drawn



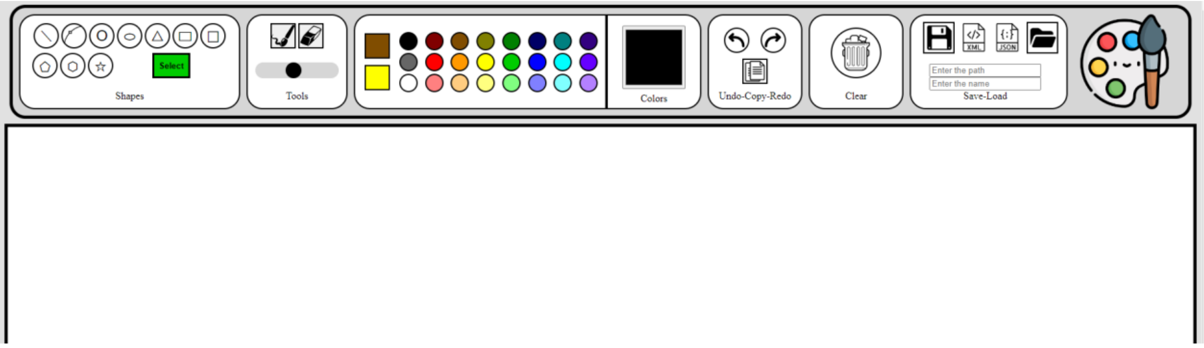
We redone it back

Fig 0-2: 6) we can also see a copy button that make copies of the items that we have drawn. Like the following:



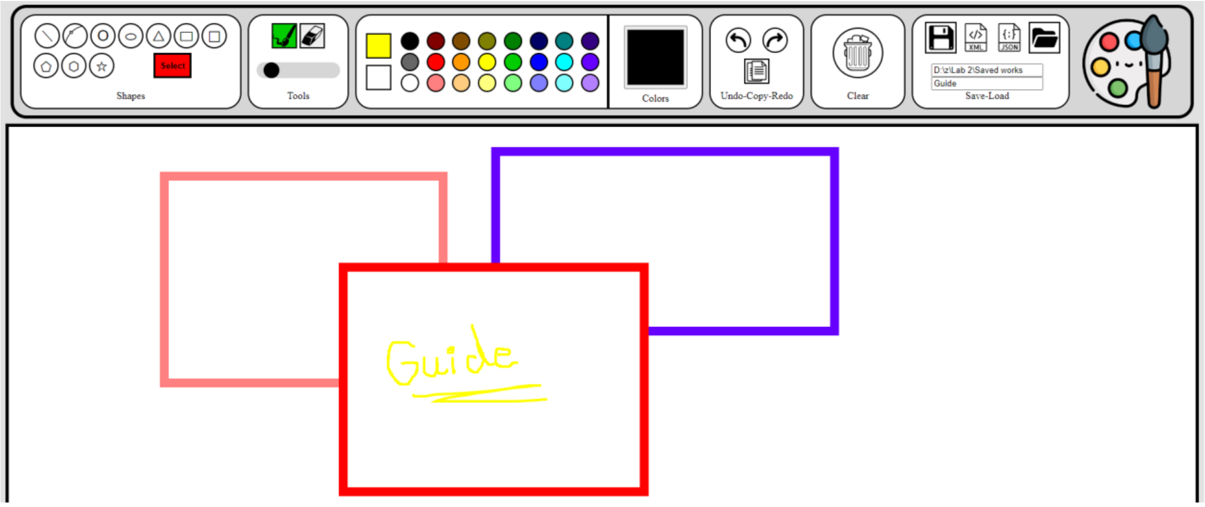
We have made a copy of it

Fig 0-2: 7) we can also see a clear button that clears all the items that we have drawn. Like the following:



We have cleared the board

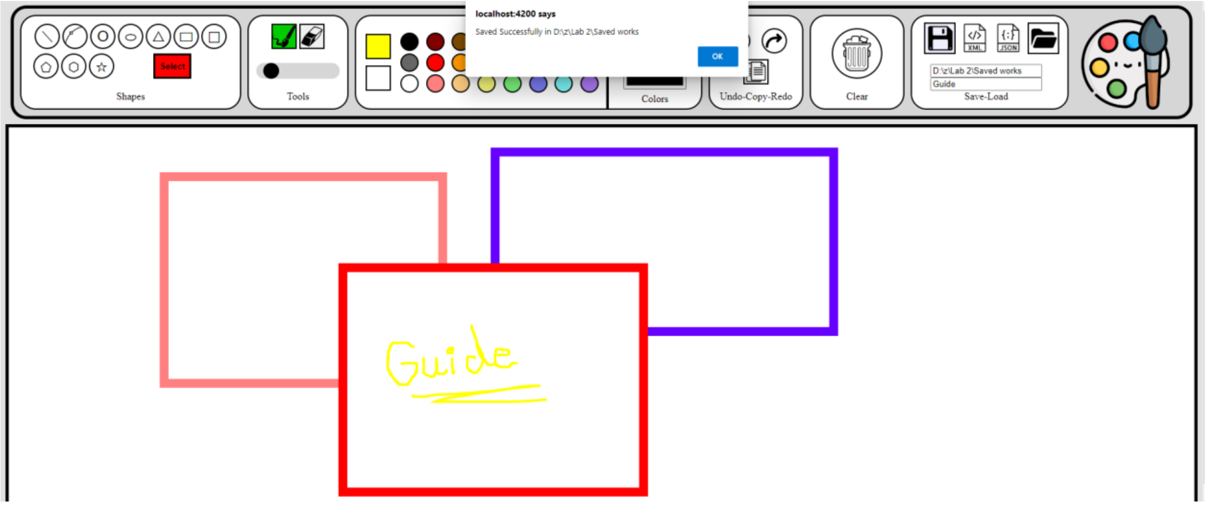
Fig 0-2: 8) we can also see a Save button and a Load, both of them works by just Entering the path and name and choose one of the 2 extensions XML or JSON and press save or load. Like the following:



D:\z\Lab 2\Saved works

Guide

We have made a work and entered the name the path and the extension we want.



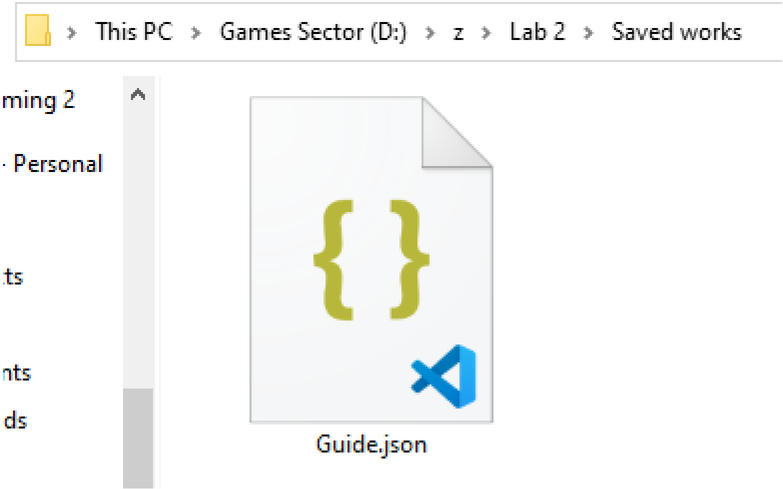
localhost:4200 says

Saved Successfully in D:\z\Lab 2\Saved works

D:\z\Lab 2\Saved works

Guide

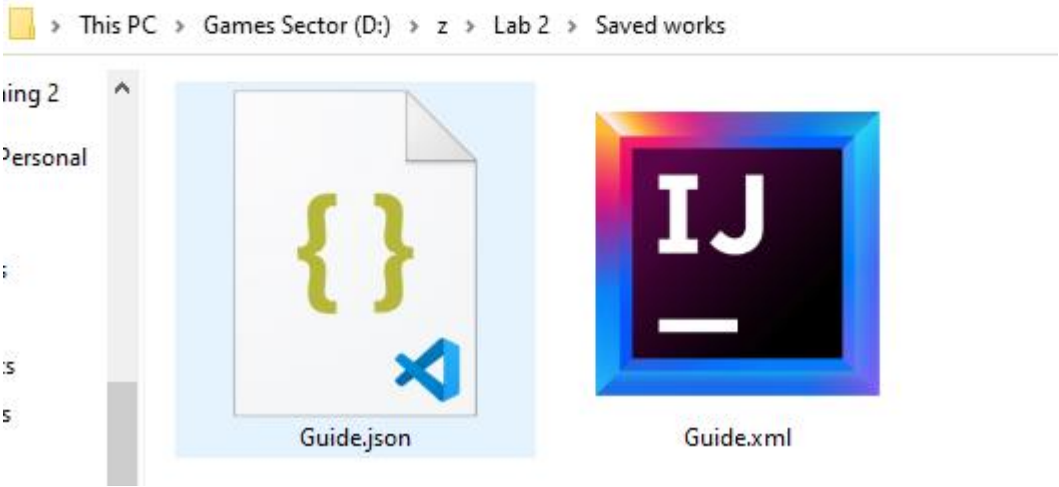
after pressing save



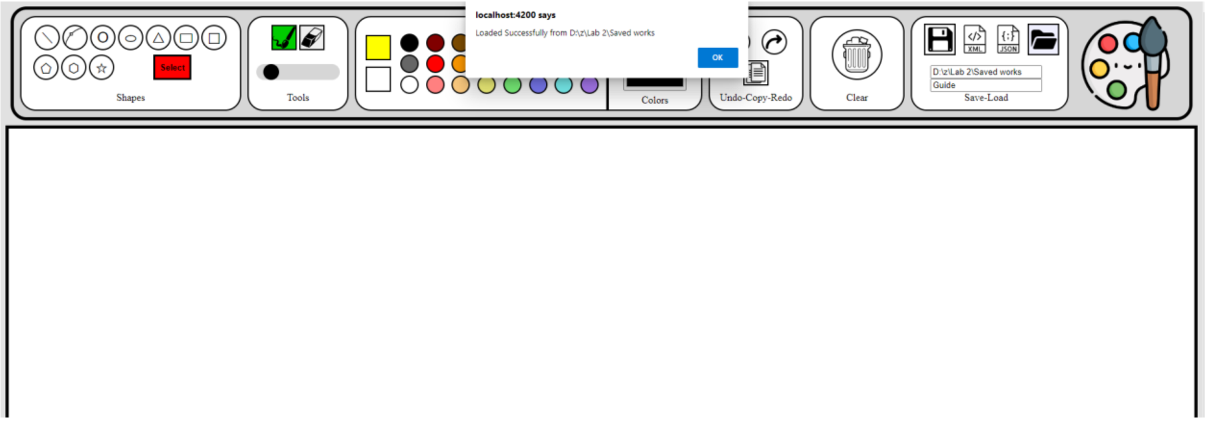
This PC > Games Sector (D:) > z > Lab 2 > Saved works

Guide.json

The file saved successfully



If we changed the extension to XML, we would get another file.

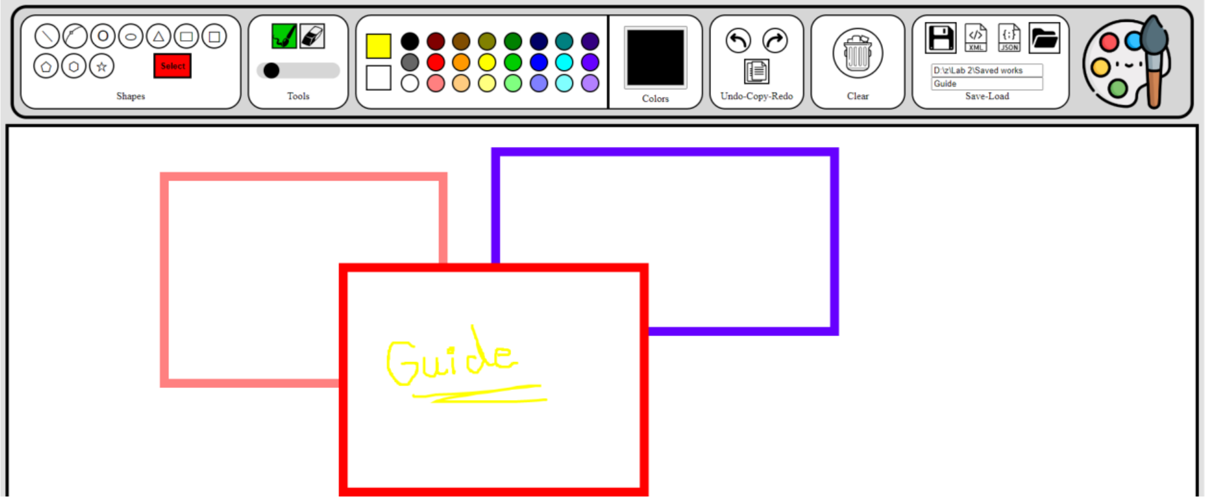

localhost:4200 says

Loaded Successfully from D:\z\Lab 2\Saved works

D:\z\Lab 2\Saved works

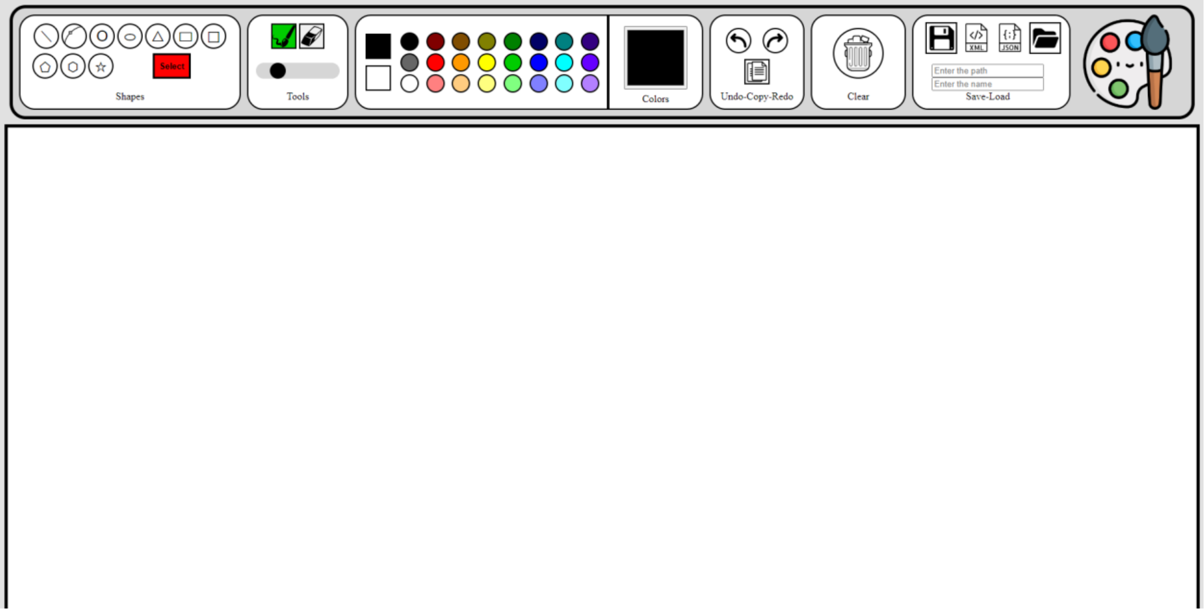
Guide

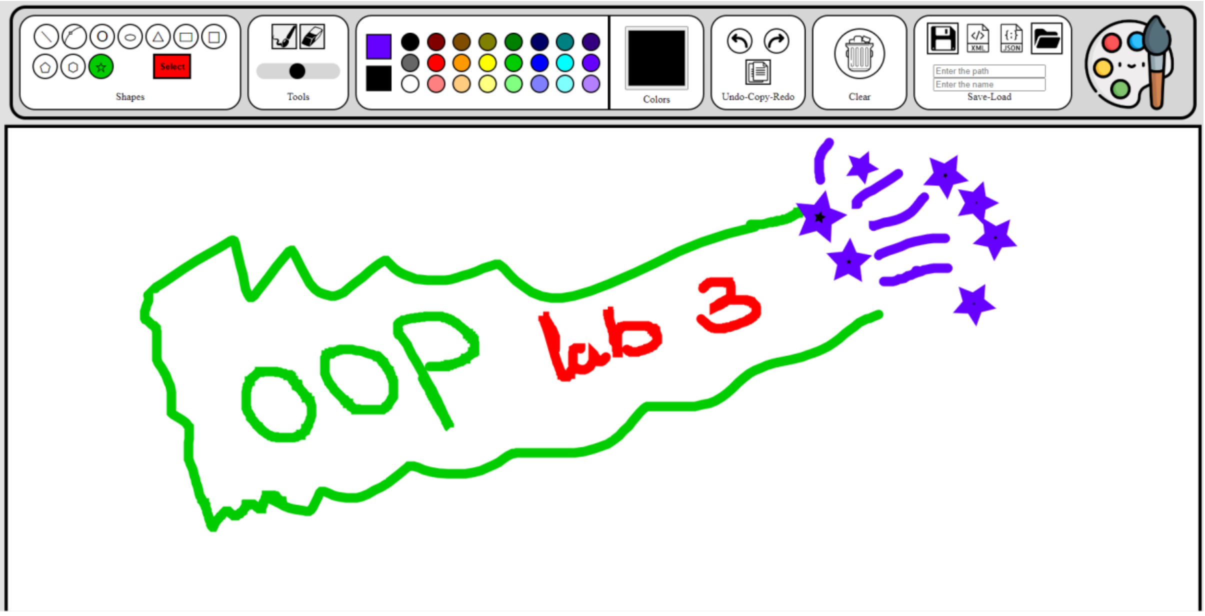
After clearing the board if we clicked on load, we will get this alert then:

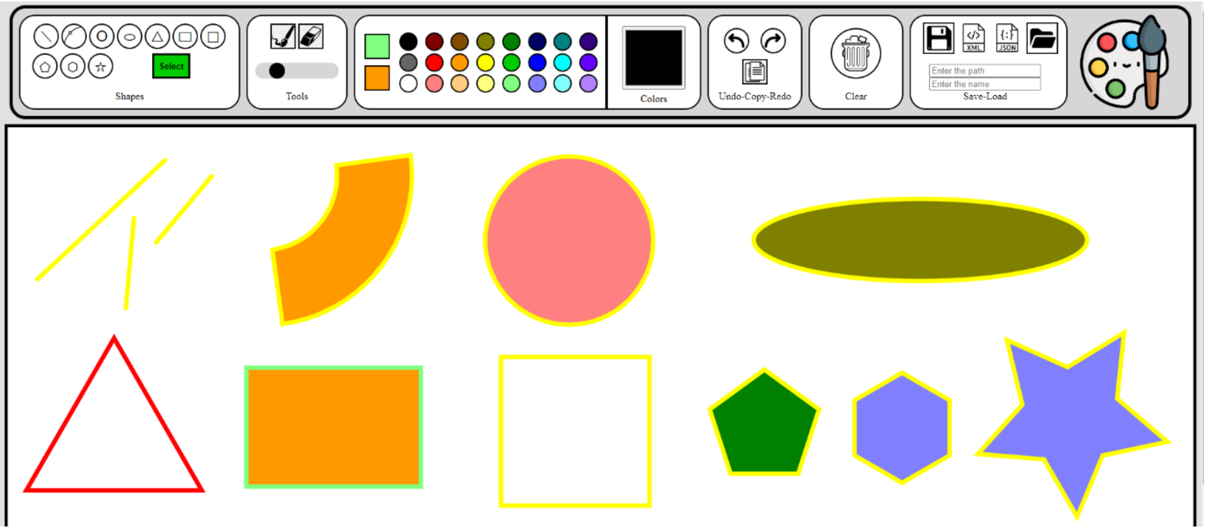

D:\z\Lab 2\Saved works

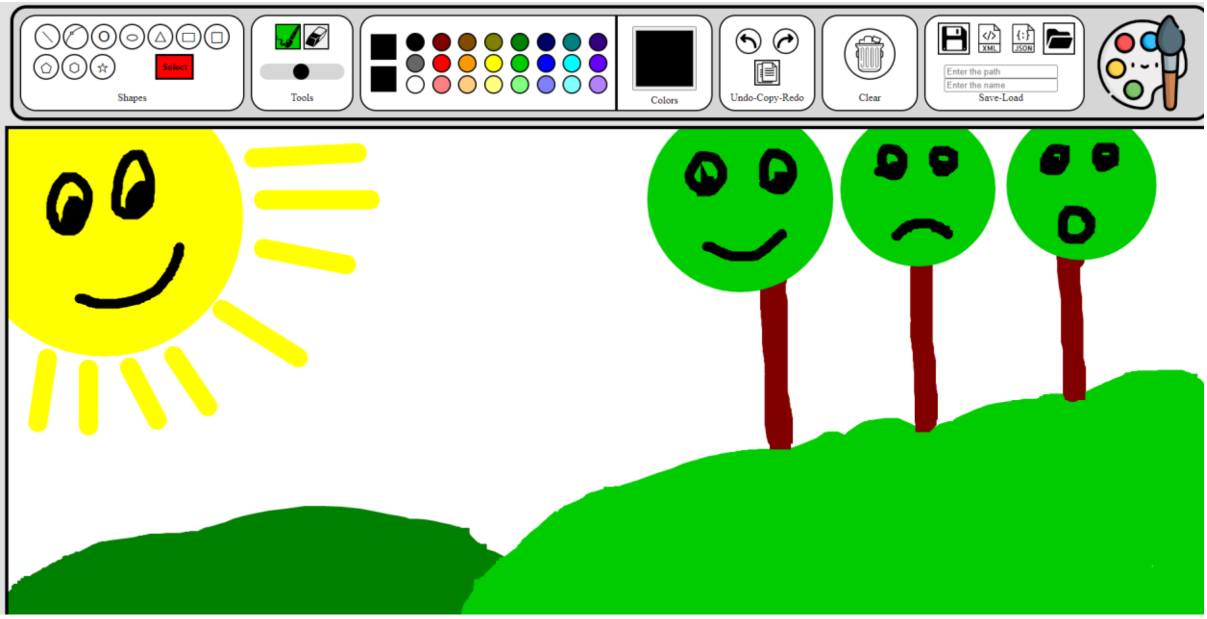
Guide

Snapshots of the UI:

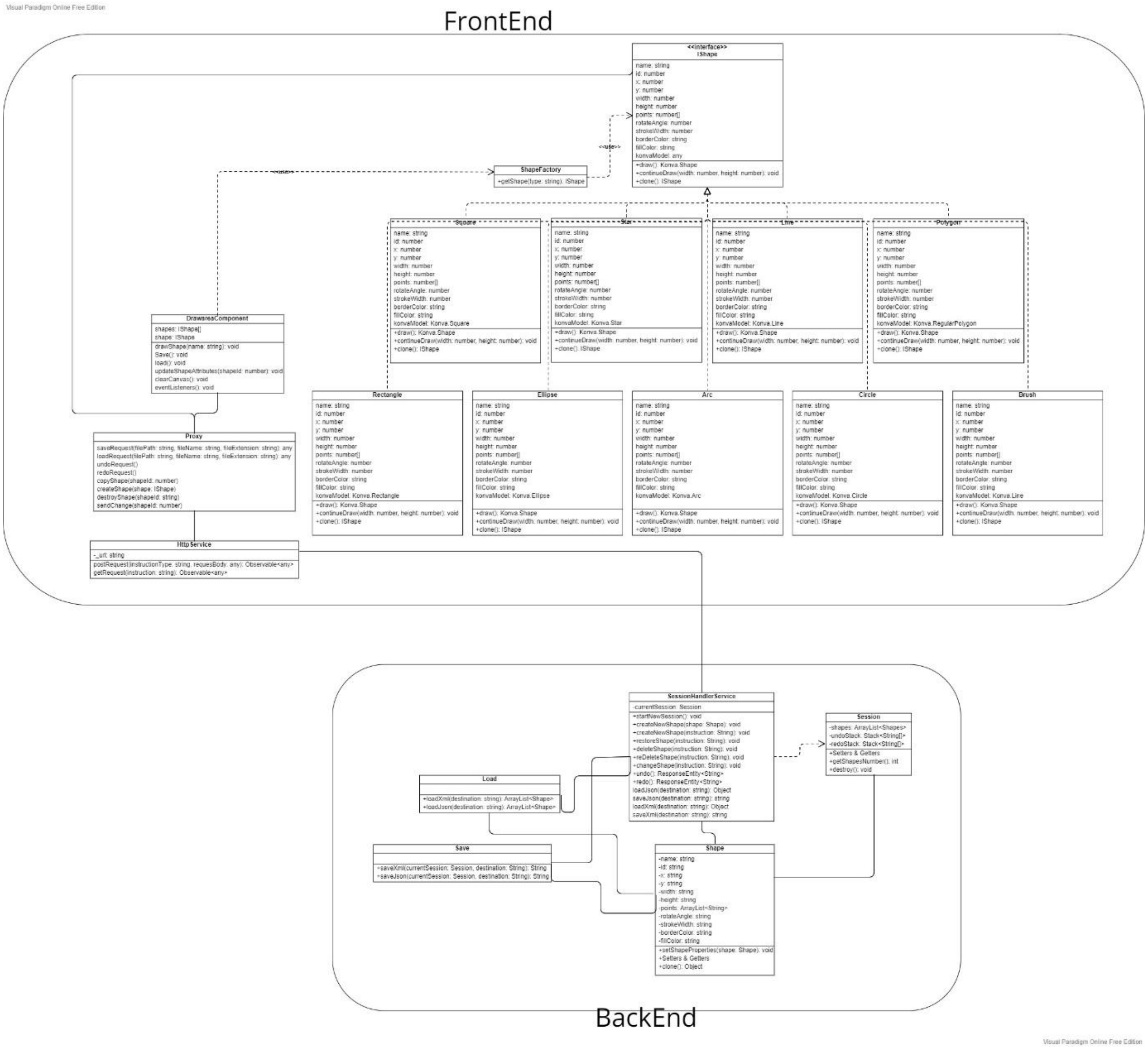








Class diagrams:



You can download it form here for a better quality:

https://drive.google.com/drive/folders/1p0ALGYk8jWETAYuA4ISDHf5z0p3P9grC

How the design patterns are applied:

ShapeFactory:

We separate the implementation of the draw function of each shape by using a shapeFactory which has a function getShape() which returns an IShape type where we instantiate an object from ShapeFactory class then call getShape() function by passing as a parameter the name of the desired shape to be drawn.

ProtoType:

We use it to clone an existing object i.e. (Shape) with all the existing properties instead of creating it from scratch by implementing our clone() function in the FrontEnd and in the BackEnd by making Shape class implements cloneable class.

Proxy:

We use Proxy Design pattern in the FrondEnd where it is responsible for sending and receiving Requests to/from HttpService which in order sends post and get requests to the server side and returns responses to the Proxy to handle them.

Design decisions and assumptions:

Front-end:

- We put the shape factory on the front end.
- We put ethe proxy on the front end.
- We made 3 components to separate the work (including the main app component).
- We moved the data between the components using the parent-child relation and vice versa.
- The clear cannot be undone.
- You must enable the select button to move, resize, rotate, or recolor the drawing.
- If the select button is disabled, these features won’t work, and the selected shape would be drawn instead.
- All shapes are drawn with the current selected attributes(stroke width, border color, fill color) from the toolbar, if a shape is drawn with a certain attribute, it can be changed lately by enabling the select button and clicking on the shape then choosing the desired new attributes while the shape is selected.
- The copy of the shape can be done by the same manner.
- Save and load asks the user the paths in which he wants to save or load the file.
- The default shape of the drawing app is the brush.
- The default stroke width is 7.
- The default border color is black.

- The default fill color is white.
- The default saving extension is json.
- The default of the select button is disabled.
- After loading a saved file no undo or redo can be done on the loaded drawings. Which means that we can continue drawing and perform undo and redo on the new work.
- We have discovered a bug that helped me in the 2-parts color indicator as in angular there are nothing called interpolation in CSS, so I tried to make it inside the HTML by using style attribute and it gives a problem, but it works properly.
- For each change done on any shape, the change is sent to the server side to be stored in the Undostack

Back-end:

- We used dependency injection In the SessionHandlerService class where we made a constructor and inject an object from the class Session.
- Each action is represented by an Array of strings stored in a stack used for undo called undoStack. When the user performs an action the reverse of the action (which is what would cancel the effect of that action) is pushed into the undoStack.
- For each undo action the reverse of that action is to be stored in another stack called redoStack to be used later if the user wanted to make redo
- When the user makes redo, the action is popped from redoStack, and the reverse of that action is pushed to the undoSatck. When the user undos or redos the action is popped from the stack and sent to the client side where a proxy object translates that action into the canvas.
- The user should enter the full path which desired to be saved in and loaded in also he should write its name and specify whether it is json or xml extension.
- If the user enters a wrong path or a wrong file name within the process of save & load the web will pop a message including an error has occurred.
- The points of the brush(freehand) are stored in the Backend in an ArrayList of points which contains very large number of points.
- When a shape is deleted, it's marked as deleted and not actually deleted from the session. This is to avoid recreating shapes when undoing after a deletion action. The mark is a Boolean called 'deleted'. After setting this Boolean, the reverse of the action is pushed into the undoStack.
- redelete method is a helper method which can't be called from the client side, and it does the same thing as the delete method, but it doesn't push any action onto the undo or redo stacks. It is used during the undoing or redoing process.
- restore method is a helper method which can't be called from the client side, and it sets the 'deleted' Boolean of the shape to false, so the shape is redrawn. It is used during the undoing or redoing process.
- copy method accepts the id of the shape to be copied and uses the clone method to clone the shape and push it into the Arraylist of shapes