

Simple Paint Application

Ali Karim Fakh – 95497

Maria Nabil Afara –95492

Supervised by Dr Samar Tawbi

Table of Contents

Chapter 1: Introduction	3
1.Features:	3
Chapter 2: Business specification:	3
Chapter 3: Implementation Overview:	4
Chapter 4: UML Diagrams:	5
Chapter 5: User Manual:	8
1. Sample Runs:	9
Chapter 6: Future Work:.....	12

Chapter 1: Introduction

A simple paint Java application that allows the user to draw and color geometric shapes using simple tools.

1. Features:

- User-friendly Javafx GUI interface.
- Draw Circle, Line, Rectangle, Square, Oval , Round Rectangle, Isosceles and right triangle, Star shape, pentagram shape , heptagram shape, hexagon , pentagon ...etc. using mouse actions
- Ability to undo, redo, save and load by simple button clicks.
- Ability to zoom in and out and scroll through the canvas.

Chapter 2: Business specification:

The user can select whatever shape that he wishes to draw, with the options of selecting the stroke color, increase its size if he wants with the possibility of watching the increase of size and also the option to select the bucket and precise the color of the fill if he desires to fill the shape .After that, with his mouse pressed on the canvas he can drag it and while that he is able to watch the new positions of the shape as he move his mouse, then he leave his mouse on the position he favors.

Moreover the user can undo redo the shapes he draw and also erase them. He can save what he draw into a .png file, and he can open new file to draw on or open an existing image that takes the size of the canvas and may draw on it.

Further, the user may also select the text button and write any text in a field ,choose a font , choose size of the text and also he may change stroke and fill color as well .Then he can drag the text in any place on the canvas.

He can zoom in the canvas or out with clicking on zoom in or out or with the keyboards letters (I) for zooming-in and (o)for zooming-out, and as the canvas size increase the user can scroll it.

Finally he may exit the application.

Chapter 3: Implementation Overview:

We are using MVC architectural pattern to separate and organize the classes.

The model contains some classes related to geometric shapes and its classifications.

The view section consists of the user interface.

And the control section, comes the functionality provided to the user, from which it converts user actions into calls for other useful functions, and returns the results to do some view functions.

To allow the user to see the shape while he is dragging the mouse a temporary canvas was created and it is put with the real canvas, both, in a Stack Pane (therefore we can see the shapes on both canvas).

So while the user is dragging a shape, this shape will be displayed in the temporary canvas, but once he releases the mouse the temporary canvas is cleared and the shape is put on the real canvas.

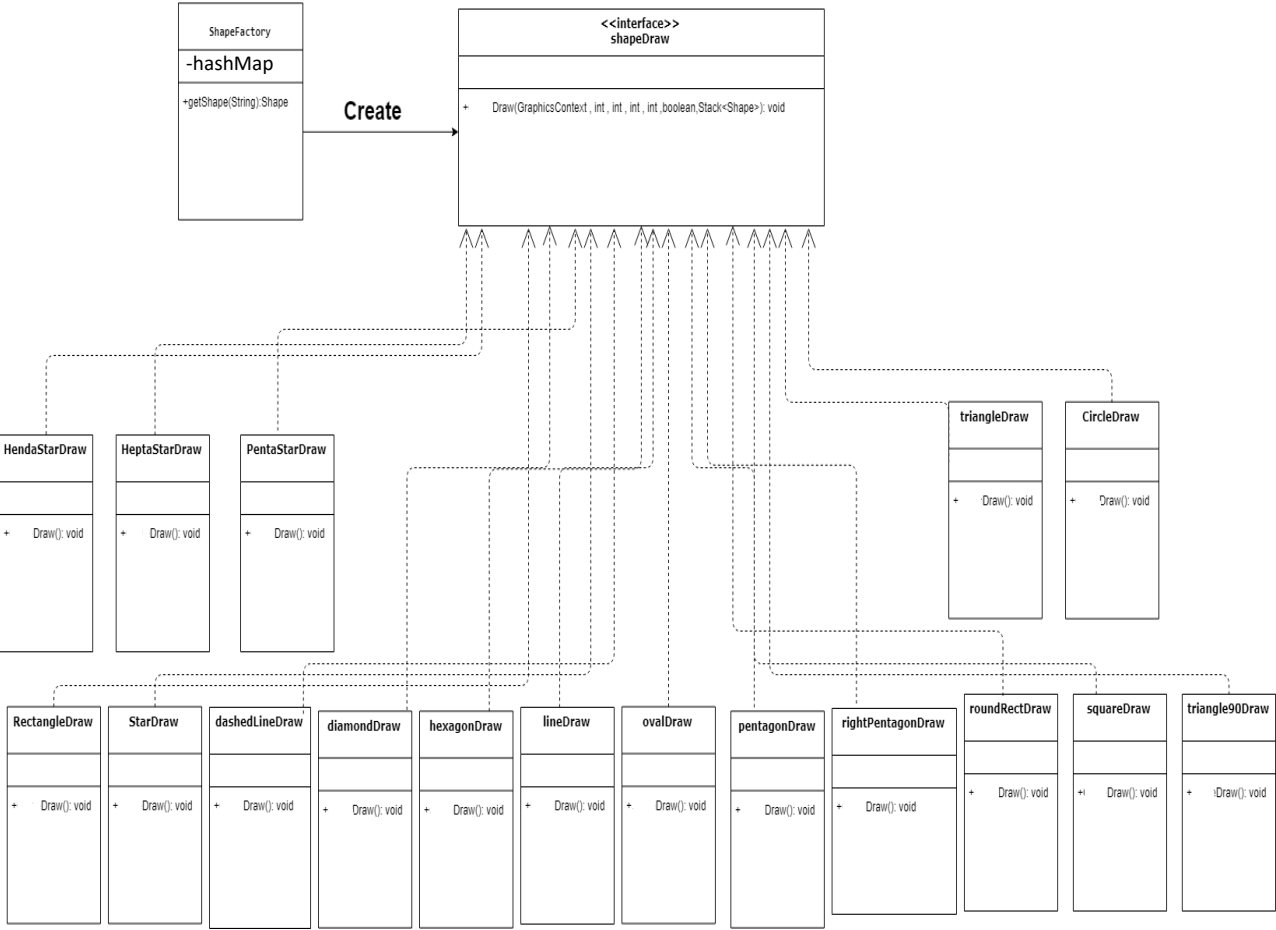
A couple of stacks are used to allow Undo and Redo functionalities (This functionalities works only with shapes, it doesn't support text drawing, erasing and images because the type of those arrays is array of shapes, for now).

But how the whole procedure works:

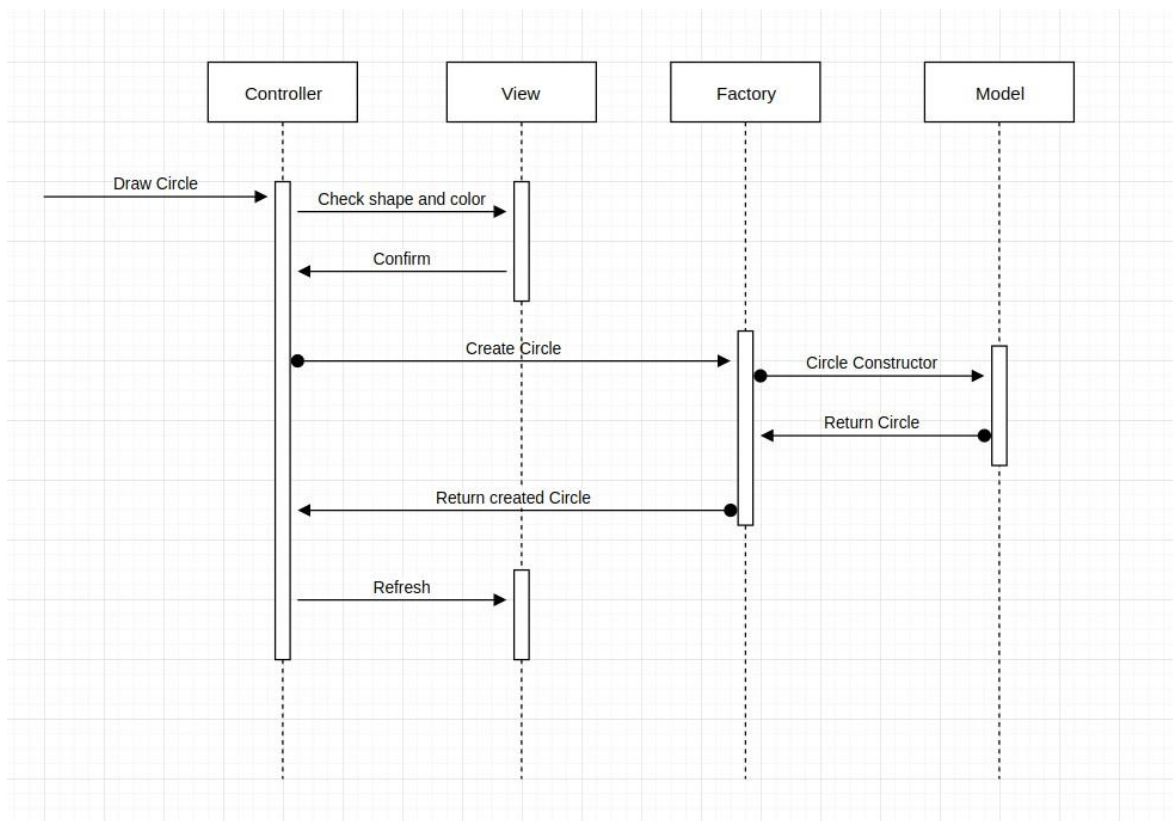
- 1) After drawing any shapes they are pushed into the undo stack.
- 2) When the undo function is called:
 - The canvas is cleared.
 - The last shape is popped from the undo stack then we put it in the redo stack.
 - Finally the rest shapes are drawn from the undo stack.
- 3) When the redo function is called:
 - The last shape is popped from the redo stack then we draw it.
 - This shape is put in the undo stack again.

The factory design pattern is used while calling the function that draws the shapes. The shapes are created using a Shape Factory Class that return a shape according to a string. In the other side there exists the shape interface with only one function that draw the shapes and all other shapes(Circle, triangle, star...) implements this interface.

Chapter 4: UML Diagrams:







Chapter 5: User Manual:

The interface is divided into mainly 4 parts, a set of buttons, listed on the left corner, a menu part rowed at the top, and the color selectors located at the bottom, and finally the drawing canvas in the remaining area.

In order to draw, you need to pick a shape from the left side buttons, pick the stroke-fill colors, and drag using the mouse in the canvas area.

1. Sample Runs:

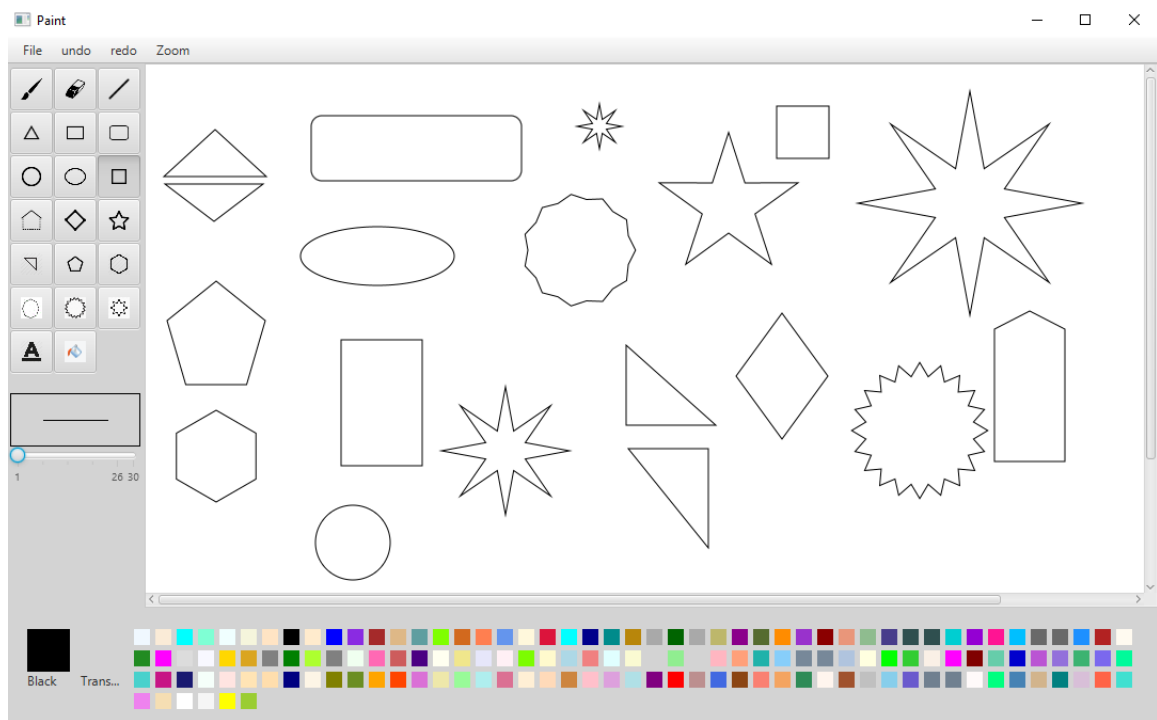


Fig1: The user selects any shape then draw on the canvas.

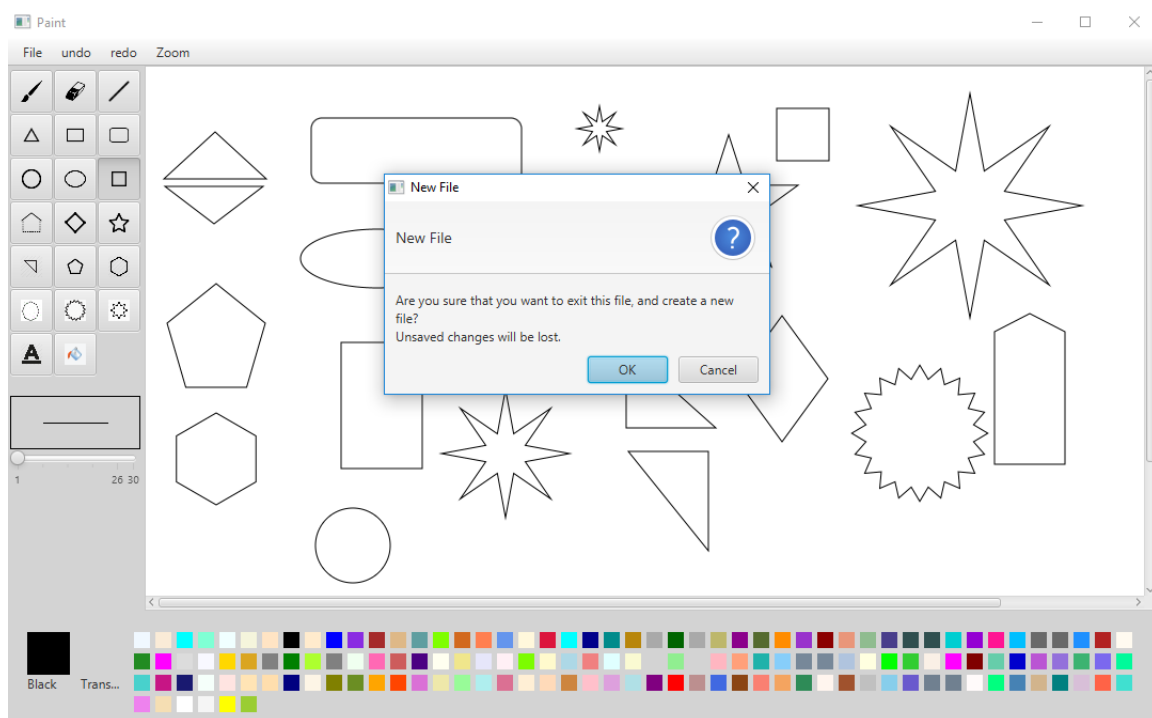
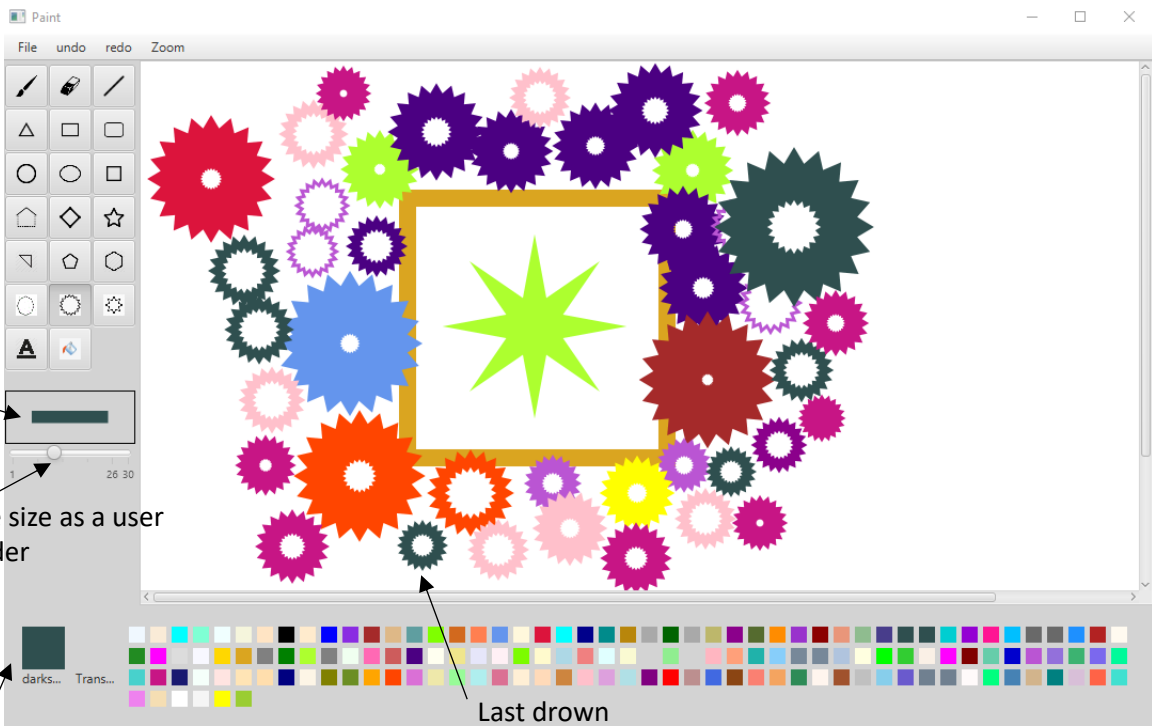


Fig2: The user selects File->New to create new drawing file.

Shows the exact size that the slider indicates with the stroke color chosen

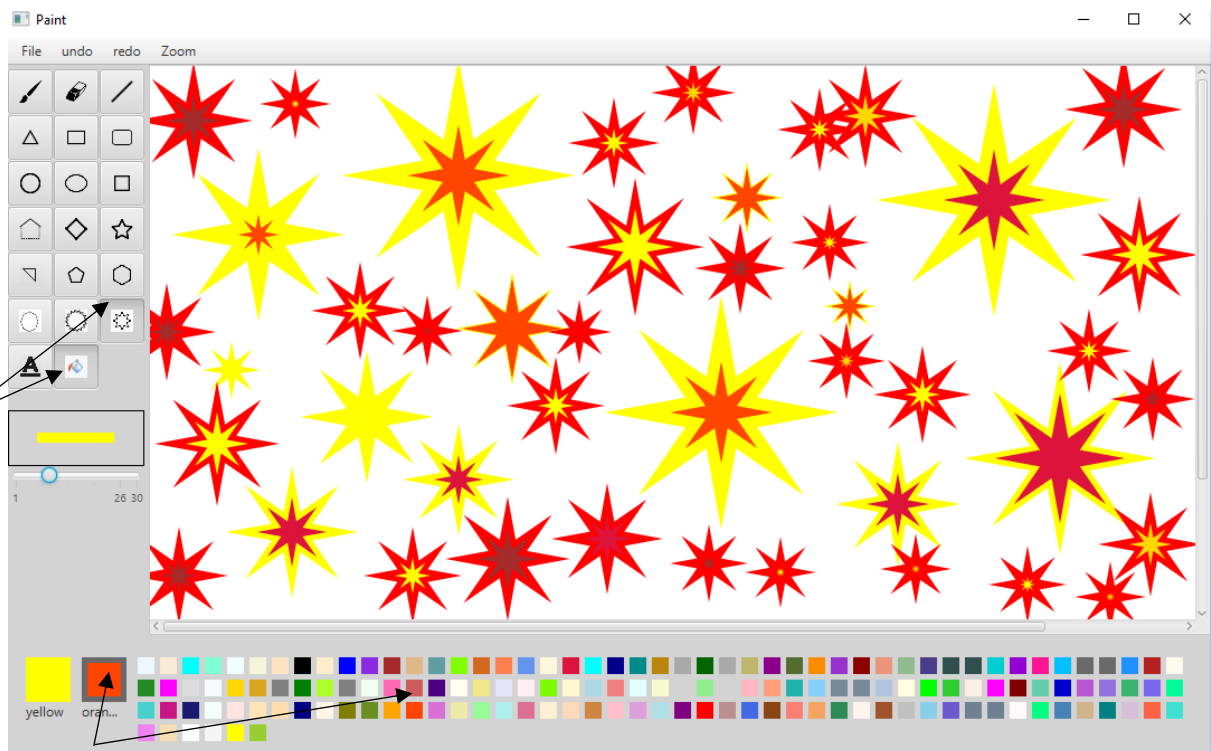
Resize stroke size as a user slides the slider



Selecting this stroke color box then chose a color that fills it

Fig3: Size and color of stroke

Selecting a shape and the bucket button allows you to fill the shape with any color



Select the fill color box and choose the color to fill then draw

Fig4: Fill with Color

Choosing the text button allows you to enter a text, may choose a font and your colors and size and drag it in the canvas



Fig 5 : Text and Font.

Choosing File in the menu then open will allow the user to select an image and load/open it into the canvas

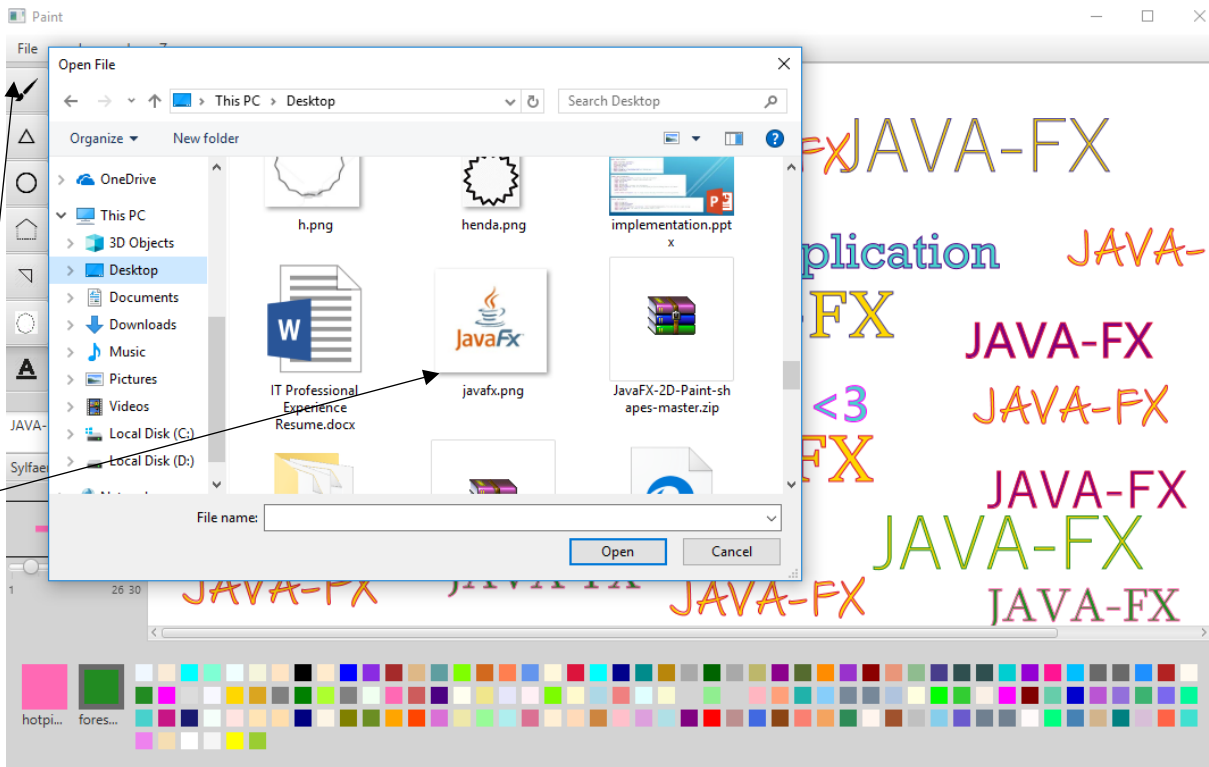


Fig 6: Open Image.

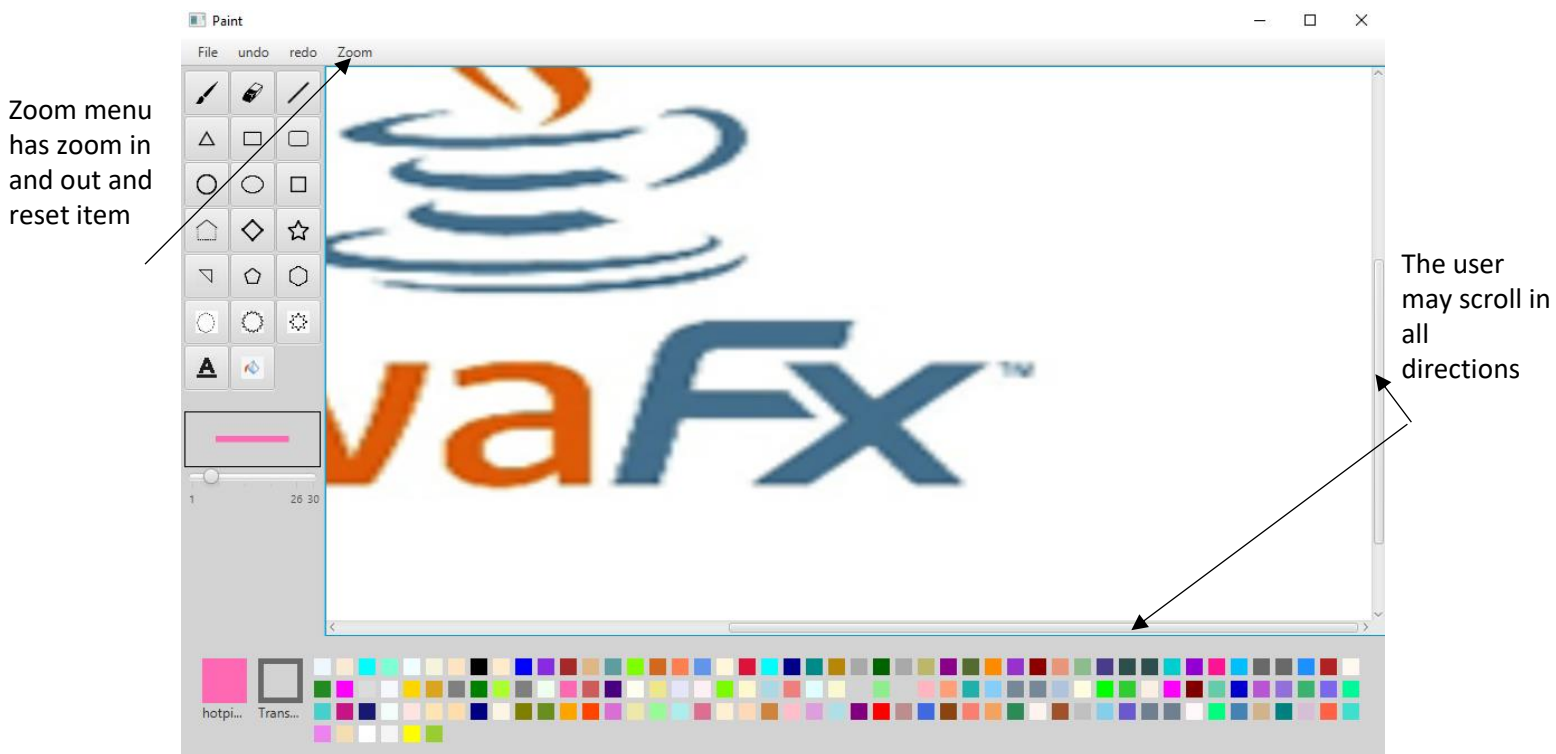


Fig 7: Zooming and scrolling.

Chapter 6: Future Work:

We must mention that we hoped we could implement much more functionalities in our app but unfortunately time didn't help us quit enough. But however there's some of what we planned and couldn't let it be ready for now:

1-For future work we plan to let the shapes work with drag and drop properties , that means that dragging the shape in the canvas and resizing it while the focus is still on it while the mouse clicks on the rest of the canvas the shape is drawn and can no longer be resized or dragged.

2-Secondly, we plan to make undo redo property work for everything, resizing rotating, images, eraser...