Java report for Graphic assignment

Since I always used some graphic editing software such as Inkscape etc. As for this coursework, I would like to design a sample draw app which can design or sketch pictures by using the JavaFx.

Background:

According to the JavaFx API https://docs.oracle.com/javafx/2/api/index.html
javafx.scene.shape which provides the set of 2D classes for defining and performing operations on objects related to two-dimensional geometry. There is a javaFx 2D shape example on line: https://examples.javacodegeeks.com/desktop-java/javafx/javafx-2d-shape-example/
Any shape that can be drawn in a 2D shape. JavaFX offers variety nodes to draw different types of shapes (lines, circles, rectangles, etc.). We can add shapes to a Scene Graph. All shape classes are in the javafx.scene.shape package.

Implementation:

As for this assignment, use the IntelliJ Idea to develop the javafx project begin with a simple with the main.java, control.java and sample.fxml to display the graphic user interface (GUI). Java part mainly to achieve the function.

First stage is to define each shape, (eg. Cicle, line and rectangle etc.), according to the javafx.scene.shape package, in the shape file including the CicleShape.java, RectangleShap.java, EraserShape.java, LineShape.java and PencilShape.java. Last three class are all use the line shape.
A Shape has a size and a position, which are defined by their properties. For example, the width and height properties define the size of a Rectangle. The radius property defines the size of a Circle. The x and y properties define the position of the upper-left corner of a Rectangle, the centerX and centerY properties define the center of a circle, the startX, startY, endX, endY define the start pint to the end point of a line etc. Shapes are not resized by their parents during layout. The size of a shape changes only when its size-related properties are changed. Shapes have an interior and a stroke. The properties for defining the interior and stroke of a shape are declared in the Shape class. In order to get properties of each value, there is a Pen.java including the getter and setter groups. In addition, all of the shapes are need use the Shape in javafx.scene.shape package, so I also create an interface PenShape to declare the Shape method for these classes to implement.

Second stage is use the shape defined in the first stage to achieve some sketch action, this stage for each shape use a class for draw action, each of these draw action class, they all extends an abstract class Draw.java for an abstract Group to use Group function declare in the package javafx.scene.Group, each of the draw function corresponded to a factory class in order to call the Pen.java (contains setter and getter groups). In these factory classes, the createPen method has been override service for different shape of action, which group the javafx Group, AnchorPane and handler (getters and setters). In order each shape draw can use the createPen method, declare a Factory Interface for them to use.

Third stage is design the Graphic user interface (GUI), like the Figure 1 shows, this interface contains two parts including the tool bar for all of too button icons to achieve functions and canvas for painting. As for this GUI design choose fxml to achieve it, it choose the FlowPane to display the tool bar and use AnchorPane to display the canvas for painting. According to the introduction of layout panes on https://docs.oracle.com/javafx/2/layout/builtin_layouts.htm
Use the label <MenuBar> to design the menu part and <MenuItem> for function save or open the files, <HBox> and <VBox> for it children design.

By the way, in this part I have found javafx has an amazing function the ColorPicker, the color picker control in the JavaFX SDK is a typical user interface component that enables users to select a particular color from the available range, or set an additional color by specifying an RGB or HSB combination, which can easily help us to solve the shape color issues. The specific introduction of ColorPicker can see in https://docs.oracle.com/javafx/2/ui_controls/color-picker.htm
Only thing to do is to get the color the user choose by ColorPicker, perhaps this is one of the advantage functions the javafx has.

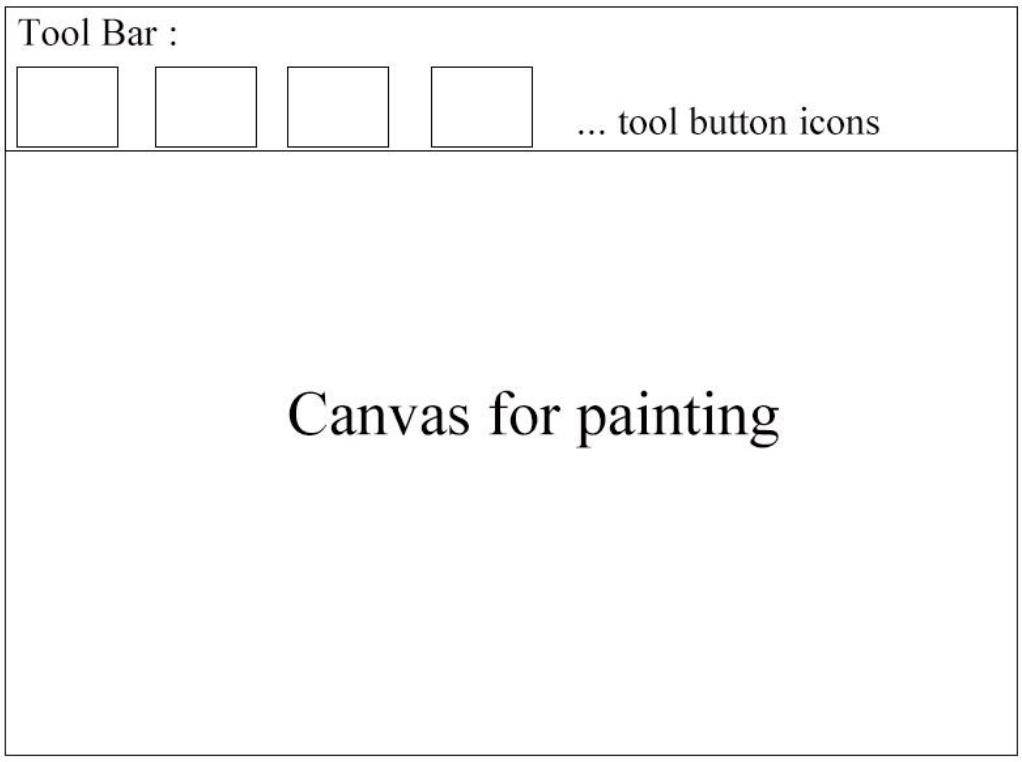


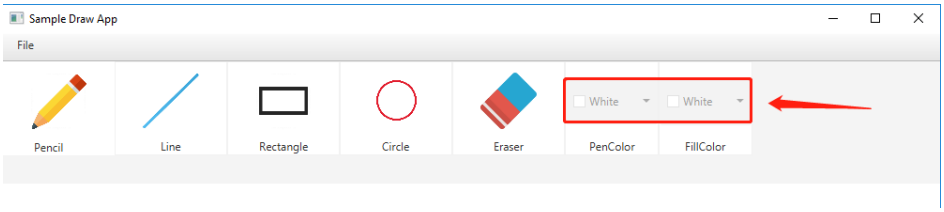
Figure 1: GUI design.

The visual rendering of this javaFx stage on Windows produced by MainStage.java and use the Fxml Loader to load the specific design resource of GUI designed in the sample.fxml and display the GUI.

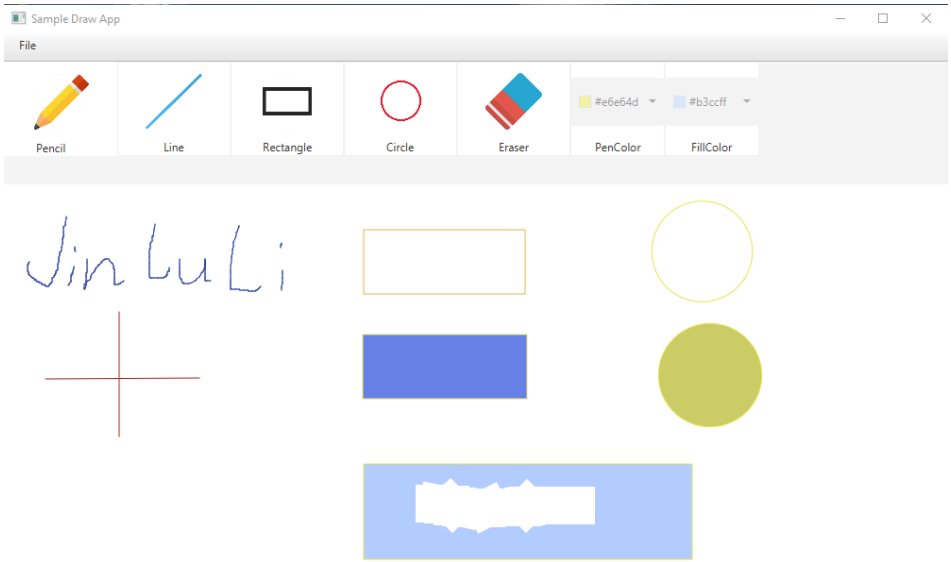
Fourth stage is to control the mouse action on the user interface, all the actions happens on the fxml interface can be get through the function (such as onAction or onMouseClicked etc) with the action name declare as “#method_name ” which method has been declared in the Controller.java with @FXML annotation. In the Controller.java part mainly including two parts file manage part and mouse manage part, as for the file management in javafx UI controls has the file chooser which can open and save files. https://docs.oracle.com/javafx/2/ui_controls/file-chooser.htm
As for the mouse manage, use the method setOnMousePressed(), steOnMouseDragged and get the mouse position (getX() and getY()).

User Manual:

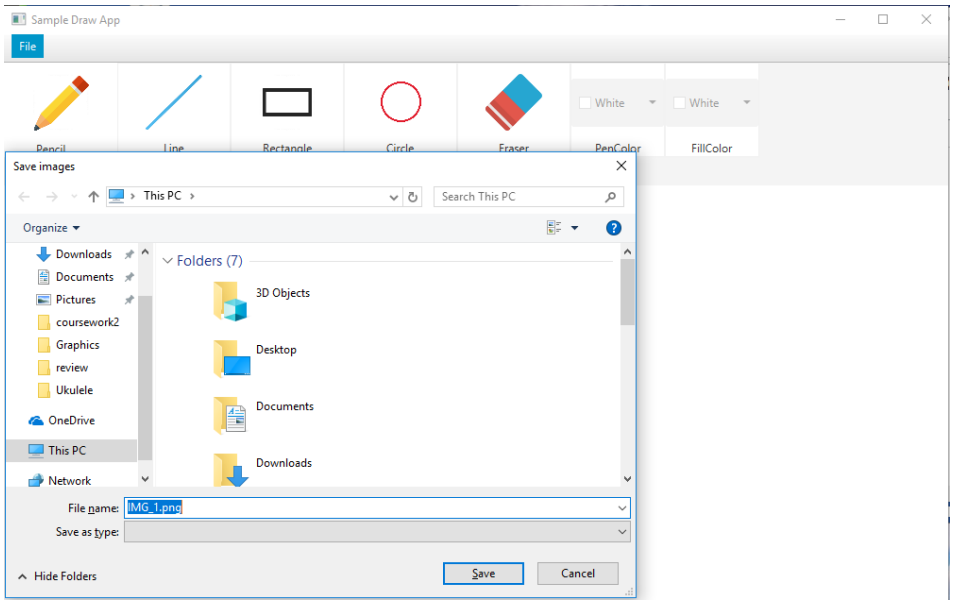
Open the Sample Draw App_JInlu Li.jar, and we can see the main interface, since the default color of ColorPicker is white the same with the background, it should choose another color first.



Basic test for all functions of the draw sketch:



This sample app can also click the File bottom to save this image or open other images with the .png extension:



One of the picture drawing by this App (Name: Home) save as .png :