Quote Painter

1. Project overview

My program functions as a basic drawing application which also provides a challenge in the format of a timed challenge to draw a random prompt (pulled from zen quotes). A problem that this program solves is providing a fun prompt to draw when you're bored and have no idea what to draw.

2. Code break down

-The LaunchPage class:

The LaunchPage class isn't very exciting. It is a JFrame that contains mainly just JLabels as text to introduce the user to the application. It contains a button which leads to another JFrame or window: The SelectionPage Object.

-The SelectionPage Class:

This class is a JFrame which contains 3 radial buttons that correspond to the minutes you want draw for. The value of minutes gets passed onto a new PaintGUI object.

-The PaintGui Class:

-The PaintGUI constructor:

The PaintGUI class is the backbone of the UI
This is the constructor:
-sets the time
-sets up the menu items
-sets up the JFrame
-and calls a very important method addGUI

-addGUI( ):

The addGUI method creates a JPanel with a SpringLayout. Which acts a container for most of the elements such as other objects of different classes. It creates and adds the Quote object, TimerBar Object, Canvas Object,
The Functionality calls on Canvas object's methods:
-creates undo button
-creates a Jslider for brush size
-creates a JColorChooser
-eraser checkbox
-undo button
-reset button

-getCanvas( ):

Just a getter method for the canvas object created

-savePanel( ):

Lets you choose a file location in file explorer and save the canvas' image as a png.

-loadImage( ):

Lets you select a file in file explorer to import into your canvas. Calls on a method in canvas object

The Canvas Class:

-Canvas constructor:

This constructor does a lot, the first notable part of this is it's mouse listeners which tracks your mouse movement. Whenever you click you create a custom object the ColorPoints object is created with a color and x and y coordinates. These ColorPoints are added to a ColorPoints List called currentPath. This list represents a line (which is made out of smaller lines interpolated between points to make a smooth continuous line like a S curve). And there is a list of currentPaths called allPaths which represent all of the lines on the canvas.

The constructor also creates an **Indicator object** which is to show your brush size as your mouse pointer. It is a JPanel which follows your mouse' xy. Its transparent and redraw the brush indicator based on color and stroke size. Updates the xy and color (using update indicator) in the mouseMoved listener method.

The constructor also makes a **ColorHistory Object**. Which is a small JFrame that cycles between 5 colors which are buttons that can be pressed on to reselect recents colors that have just been used. The average color of this history can be returned too. The mouse released listener adds the color via the addColor method in ColorHistory

-set color:

sets color

-reset canvas:

clears the canvas

-paintComponent:

to ensure the lines don't disappear when canvas is updated it redraws all of allPath and imported image

-isEraser:

toggles brush color to white

-setBrushSize:

sets brush size

-Undo:

Delete most recent currentPaths from allPaths and reloads

setImage:

draws image of import

addColor

history: helps add color history

returnColor history:

returns Color history

-The Quote class:

Uses's Ms. Turin's getData method to pull a JSONarray from zen quotes. Stores the first element in a JSONobject and gets the string of the object to input into a JLabel which is by extension put into the PaintGUI as mentioned previously.

-The TimerBar class:

The constructor creates a JProgressbar that starts from the max value. Then in the countDown method it counts the time down. When it reaches 0 it gives you a JOptionPane message (which gives you your average color from color history), asks you to save your image, and closes all windows to loop back into the Launchpage.

3. Features Implemented

✔Base Project (+88%)
Drawing app
✔Statistics / ML / Basic Computations (+6%)
Calculate the average color
✔ GUI (+2%)
Java Swing to build a drawing app GUI
✔ Save/Load (+2%)
Ability to save and load a PNG
= 98%

4. Outputs Example

5. What did I learn?

-I learned how to use Swing for GUI
-Got experience for making larger projects
-Learned about some of the drawing features in Swing
-How to save and load to files using Swing
-How to deal with a basic API