Application Requirements Specification

For

<Paint App>

Prepared By:

<Yuqing Gong>

Purpose and Scope Statement

The purpose of this project is to create an application for drawing. The scope of this project includes canvas creation, color selection, shape drawing, background option, template selection and file creation. Changing shape size and eraser on image are outside the scope of this project.

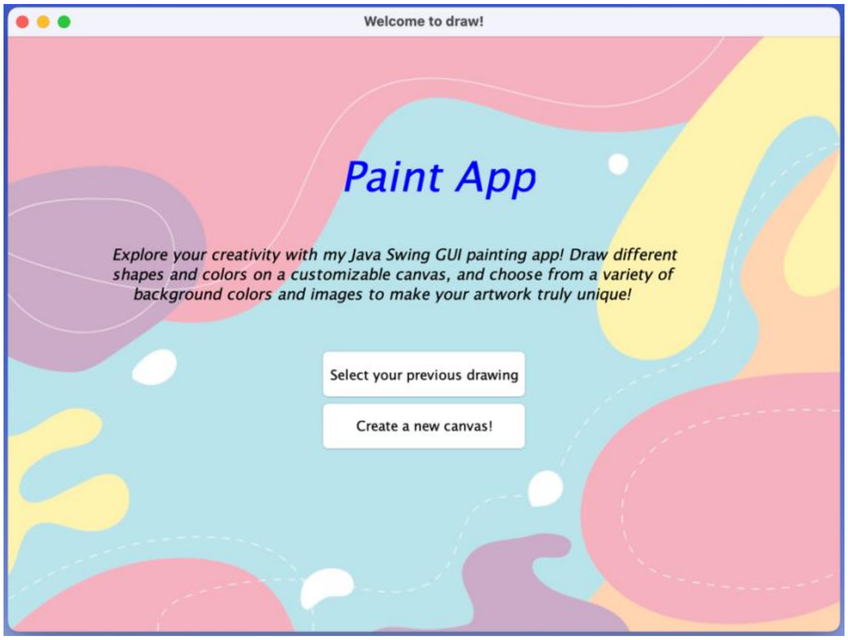
Requirements Narrative

User can perform the following functions:

1. Select from previous file
2. Create a new canvas
3. Set background color
4. Choose a brush
5. Select brush color
6. View current selected brush color
7. Change brush shape
8. View current selected shape
9. Choose an eraser
10. Change eraser shape
11. Reset canvas
12. Add background image
13. Remove background image
14. Choose background image from template
15. Draw on image
16. Save current drawing to a new file

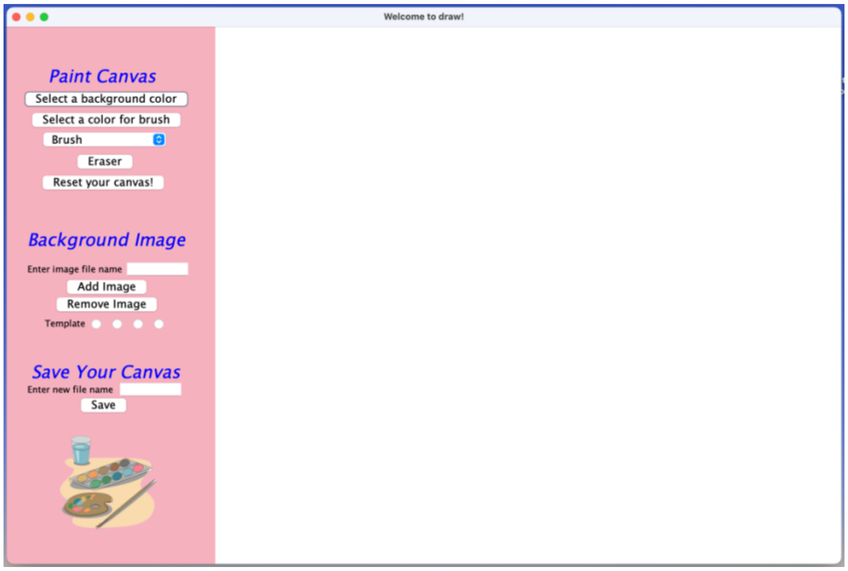
Objectives

The introduction screen for choosing from file or creating a new canvas:

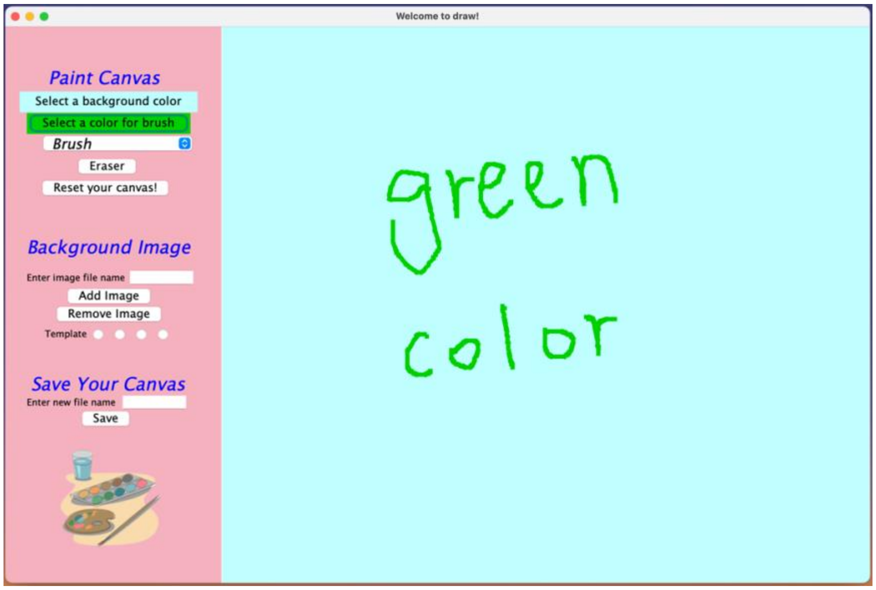


Explore your creativity with my Java Swing GUI painting app! Draw different shapes and colors on a customizable canvas, and choose from a variety of background colors and images to make your artwork truly unique!

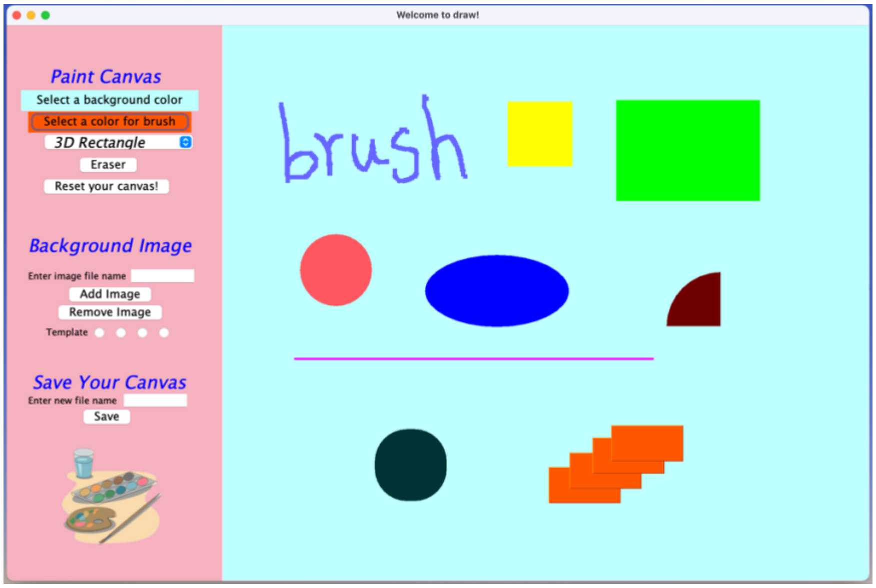
The canvas screen:



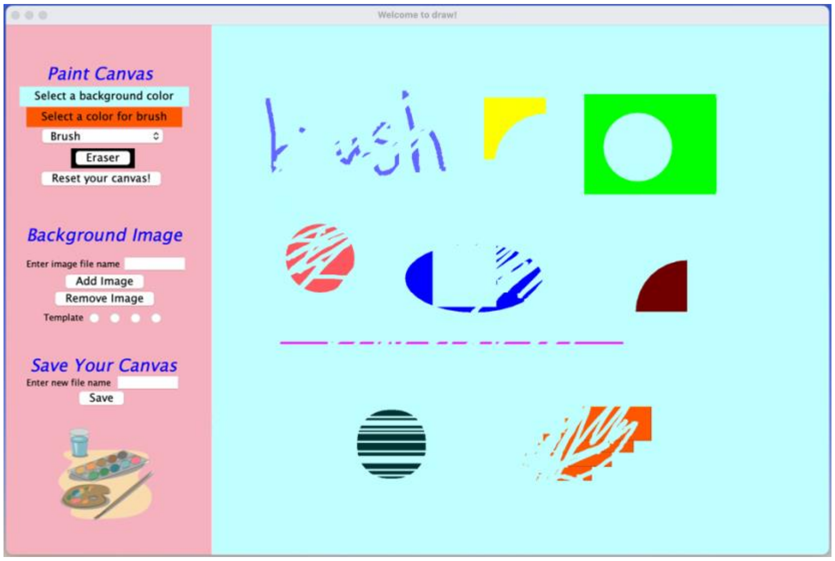
Set background color and set brush color:



Drawing shapes:

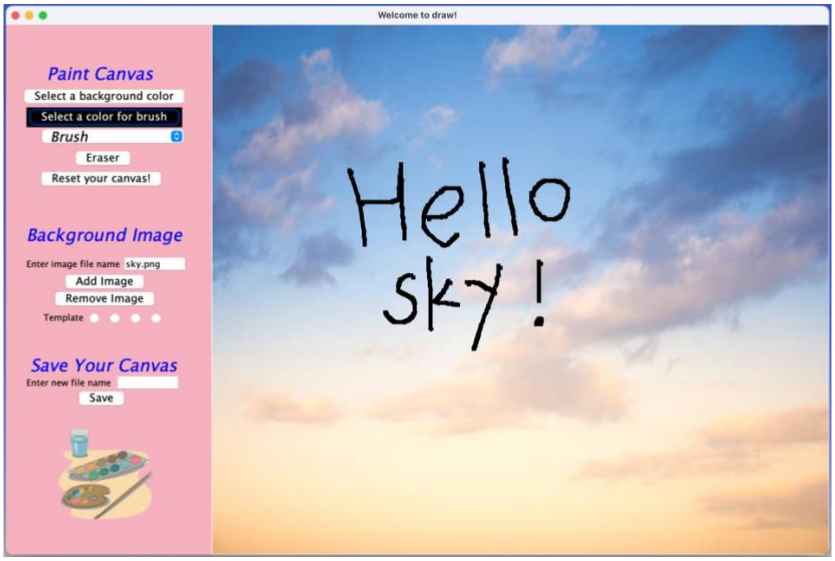


Eraser with different shapes:



Brush

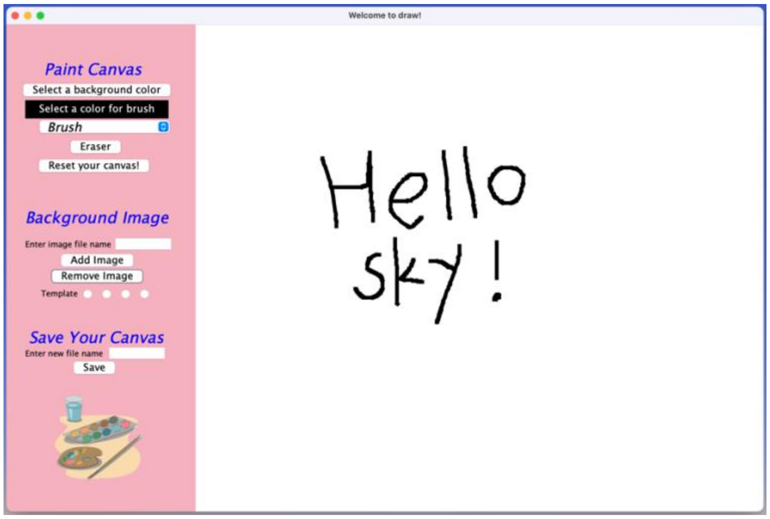
Choose a background image:



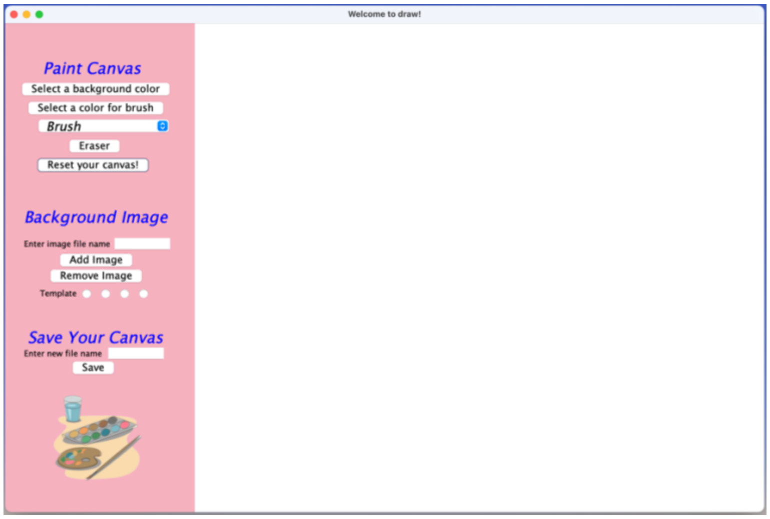
Brush

Enter image file name sky.png

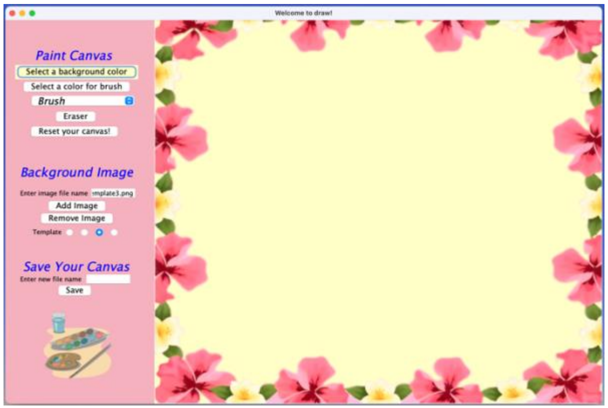
Remove current background image:



Reset canvas:



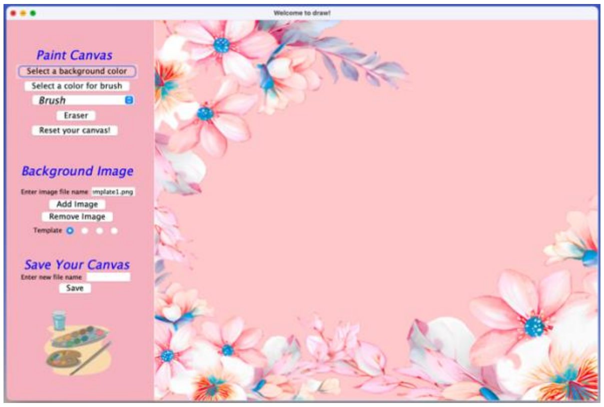
Choose canvas frame from template:



Enter image file name mplate3.png

Template ○ ○ ● ○

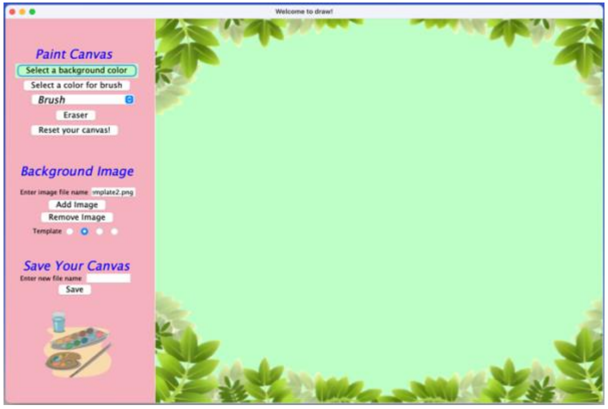
Enter new file name



Enter image file name mplate1.png

Template ● ○ ○ ○

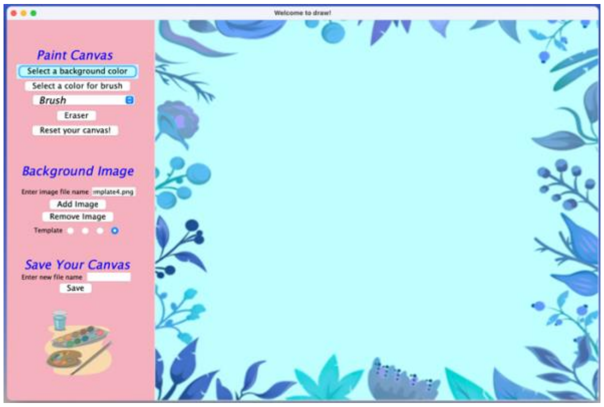
Enter new file name



Enter image file name mplate2.png

Template ○ ● ○ ○

Enter new file name

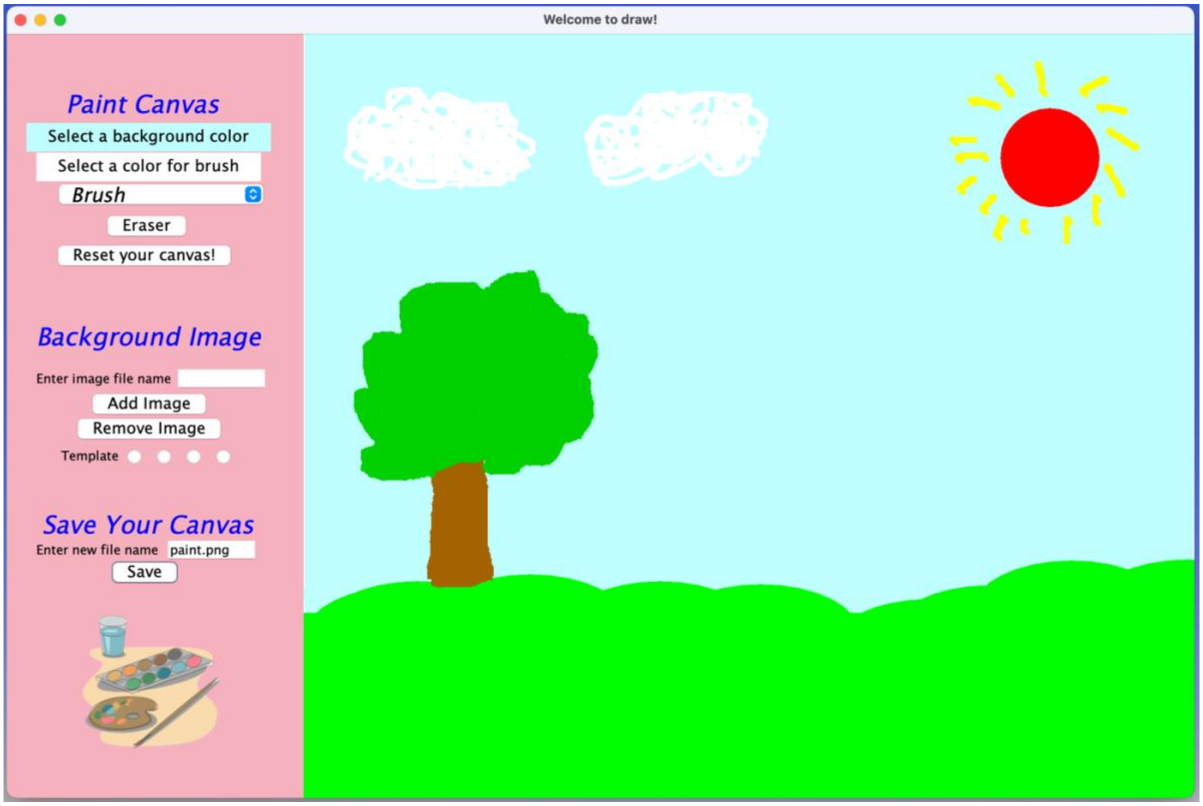


Enter image file name mplate4.png

Template ○ ○ ○ ●

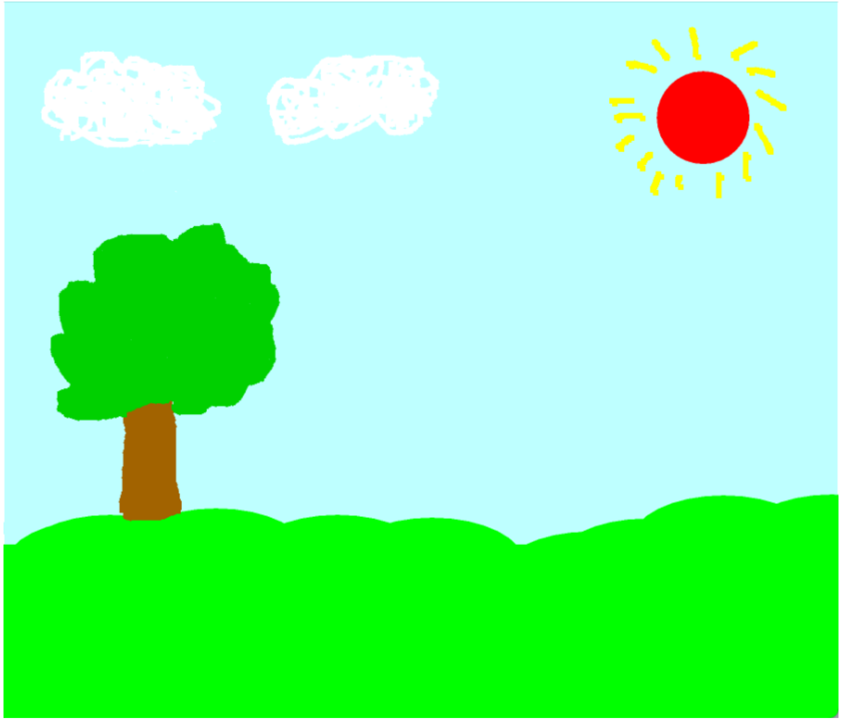
Enter new file name

Save current drawing to file:

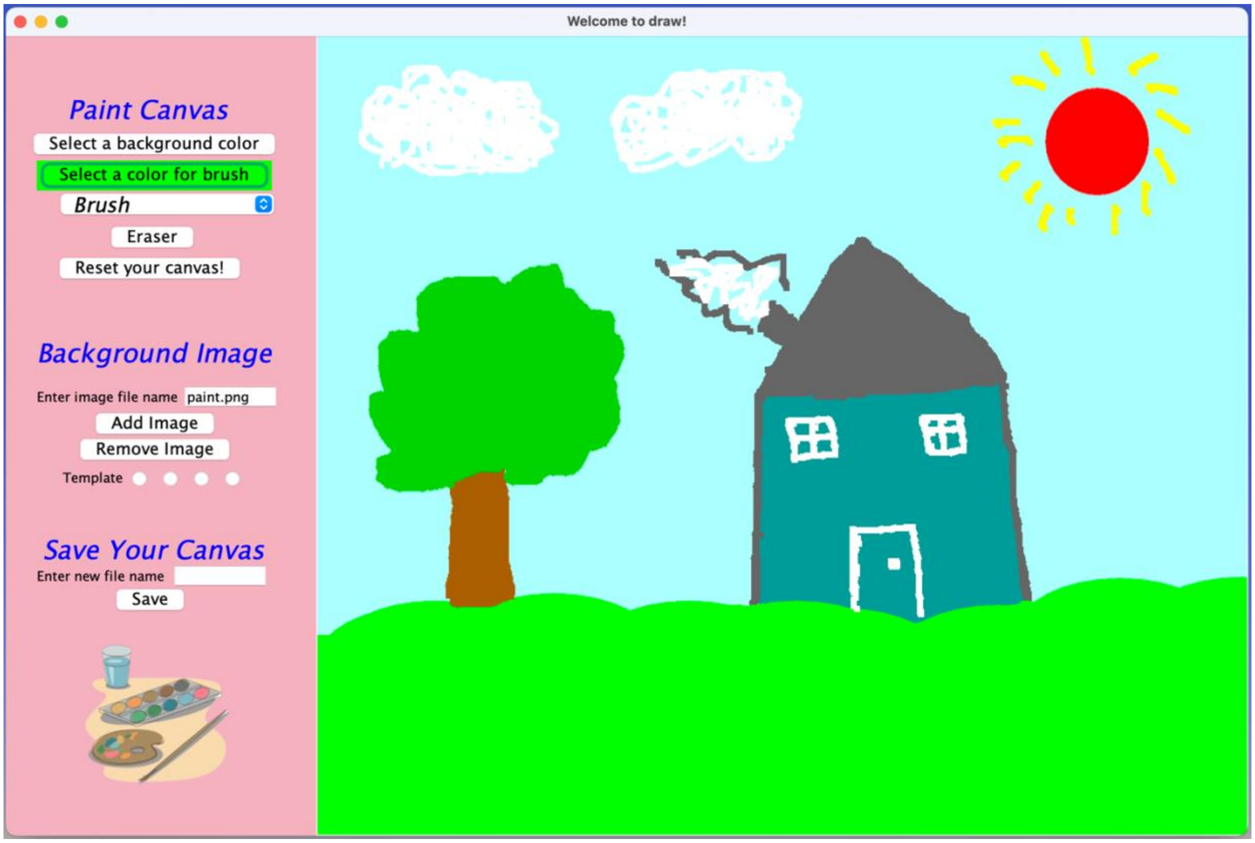


Brush

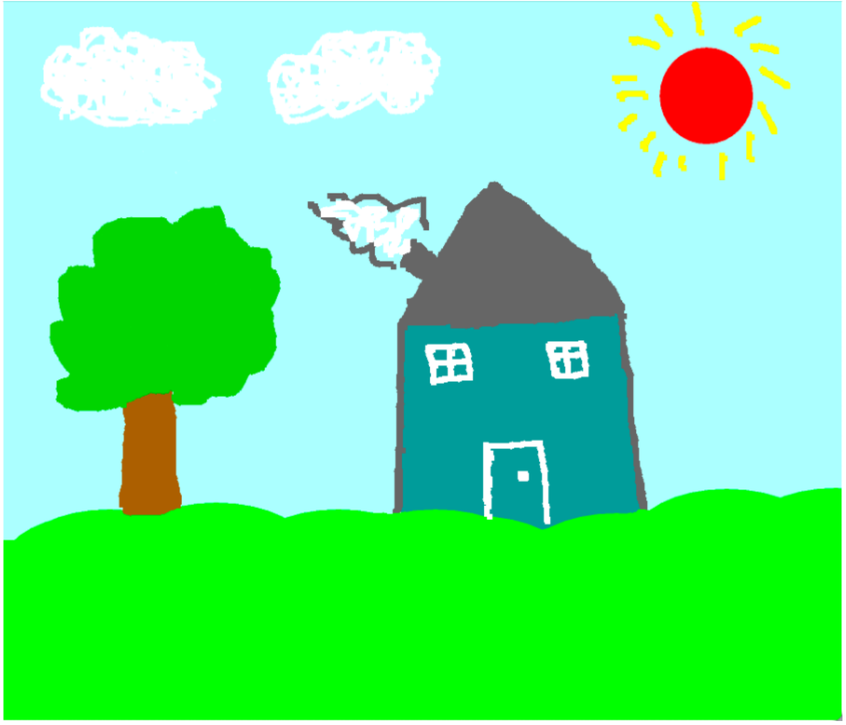
Enter new file name paint.png



Read from previous drawing file:



Enter image file name paint.png



Functional Specification

1. getSelectedFile()
2. setColor()
3. setShape()
4. setBackgroundImg()
5. removeBackgroundImg()
6. resetCanvas()
7. paintComponent()
8. fillRect()
9. fillOval()
10. getColor()
11. getX()
12. getY()

Classed Needed:

The basic classes needed for application are:

1. FileChooserScreen Class

   This class is responsible for displaying the introduction screen where users can choose from their file or choose to create a new canvas, and once users make their decision, it will lead them to the main GUI screen, which includes a Canvas object.

2. Canvas Class

   This class is responsible for the creation of canvas. It includes methods for drawing on canvas, selecting different shapes, setting colors, setting background image, removing background image, resetting canvas.

3. Point Class

   This class is responsible for holding current location (where the mouse is pressed) and current selection of color.

4. GUI Class

   This class is responsible for displaying a Canvas object and some drawing tools. A group of GUI components will be created.

Apart from these classes, there will be some built-in Java classes, including:

1. MouseAdapter Class

   This class object is created in Canvas Class, where the mousePressed(), mouseReleased(), mouseDragged() methods will be implemented and Point objects will hold the mouse locations and colors. It is part of the Java AWT library.

2. Graphics Class and Graphics2D Class

   This class object is created in Canvas Class that provides a simple API for drawing basic shapes such as lines, rectangles, and circles. The Graphics2D is an extension of the Graphic Class for a more advanced API for drawing more complex shapes and mages. It is part of the Java AWT and Swing libraries.

3. SpringLayout Class

   This class object is created in both the introduction screen and the canvas screen. It is a class in the Java Swing library to manage the layout of components.

4. ActionEvent Class

   This class object is responsible for a user's action on a GUI component, and it is part of the Java Swing library.

5. File Class

   This class object is responsible for working with files, in our case, we will need it to read the image file. It is part of the Java Standard library.

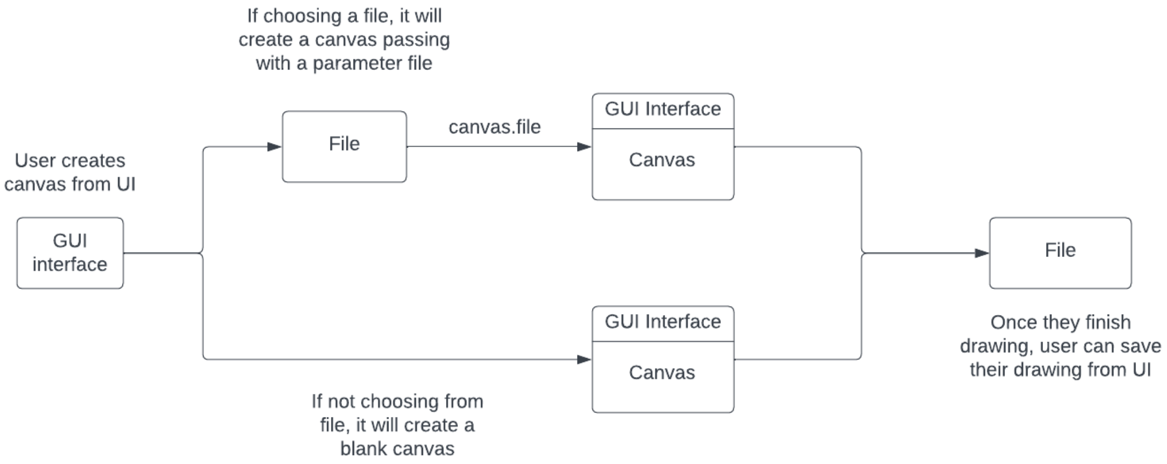
6. ImageIO Class

   This class is needed when write the current canvas to file.

Technologies Needed:

Java Swing, Java Abstract Window Toolkit, GUI

Logic Specification


If choosing a file, it will create a canvas passing with a parameter file
User creates canvas from UI
GUI interface
File
canvas.file
GUI Interface
Canvas
GUI Interface
Canvas
File
Once they finish drawing, user can save their drawing from UI
If not choosing from file, it will create a blank canvas

Concerns

This is only a desktop application and need to be improved with more advanced Java frameworks and tools.