

CS 2110: Spring 2018

Assignment A6: Paint Program

Due: See the CMS

Overview. In this assignment, you will write parts of a simple paint program. Some of the functionality you will implement is:

1. Freehand drawing and erasing,
2. Airbrushing.
3. Drawing lines.

You can work in groups of two. Form the group well before the assignment due date. Both must do something to form the group: one proposes, the other accepts. People in a group must work together. It is against the rules for one person to do programming on this assignment without the other person sitting nearby and helping. Take turns "driving"—using the keyboard and mouse.

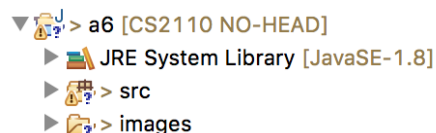
With the exception of your CMS-registered group partner, you may not look at anyone else's code, in any form, or show your code to anyone else (except the course staff), in any form. You may not show or give your code to another student in the class. You may not look at solutions to similar problems in previous semesters.

1 Installing the code

The release contains a directory `src` that contains `.java` files and directory `images` that contains `.png` files (these are images). Follow these steps to install the code in an Eclipse project:

1. Create an empty Java project.
2. Copy the two directories into the root directory of the new project. (Click overwrite when it asks if you want to overwrite director `/src`.)

Your package explorer should now look like this:



Method `main` for the program is in file `Window.java`.

2 Paint program

You can see a screenshot of a fully implemented paint program in Figure 1, where we also show it annotated with what the important GUI components represent.

- The program allows you to open image files and save in images files. See Figure 2.
- You can draw with a pencil (freehand drawing), erase, airbrush. You can draw lines and circles. You can change both the foreground color (used for drawing) and the background color (used for erasing). See Figure 3.

SAVE: The program lets you know when there are unsaved changes to the image. Label “SAVE” appears at the bottom whenever changes are made and disappears when the image is saved to a file.

You will see some more functionality by running the release code.

3 Project Structure

A significant part of your work for this assignment will be to read the release code and figure out how it works. You may need to find the relevant Java documentation on the Web and read it. Successful completion of the assignment requires you to understand well how the existing code works. Here is a brief description of classes:

Window: This class sets up the main window of the program. It handles the creation of most GUI components and responds to relevant actions.

Canvas: This class represents the image you are painting. It handles the mouse events that are relevant for drawing and performs the drawing operations.

NewImageDialog: A custom dialog for creating new blank images.

Tool: An enumeration of the available drawing tools.

Window and **Canvas** are the only classes you need to modify. **Do not change anything else in the code.** Every point inside the code where you have to change/add something has been marked for you with a `//TODO` comment.

4 Class Window

Class **Window** contains the first five TODOs; the parts that have to be implemented have been clearly marked in the release code. You will need to:

1. Implement the toolslider. When dragged up and down, it should change the sizes of the tools used on the canvas, and the current tool size should be displayed in the bottom of the window as shown in Figure 1.

2. Implement the image size label. It should display the size of the image currently on the canvas, as shown in Figure 1.
3. Implement method `getIcon`, so the foreground and background buttons display a color (as shown in Figure 1) instead of the default text.
4. Fix the foreground color so that it updates when you select a new color.
5. Fix the background color so that it updates when you select a new color.

These methods are relatively simple (a few lines of code), but you need to read and understand the code well before you attempt to make any changes.

5 Class Canvas

Class `Canvas` contains TODOs 6-10; the parts that have to be implemented have been clearly marked in the release code with a `//TODO` comments. You will find detailed instructions in the code. You will need to:

6. Modify the `mouseMoved` event handle to display the current mouse position in the window (as show in Figure 1).
7. Implement the pencil/eraser helper method `colorClick`, which draws when the pencil or eraser is used.
8. Implement the pencil/eraser helper method `colorDrag`, which draws when the pencil or eraser is used. Note: freehand pencil drawing should leave no gaps. That is, when you press and drag the mouse around, it should draw on the trace of the mouse movement.
9. Implement the airbrush. The airbrush does not have to leave a continuous trace.
10. Implement the line (note that there are three parts to this: 10a, 10b, 10c). We suggest you implement line drawing as follows: When you first press the mouse, one endpoint of the line is fixed (nothing is drawn on the image yet). When you press the mouse for a second time, the second endpoint is fixed and the line is drawn on the image.

You should visualize the “tentative line” after the first endpoint is given and before the second endpoint is given. That is, while the user is deciding on the second endpoint and moving the mouse around, they can see the line that will be drawn.

Pinned piazza note A6 FAQs contains a schedule of tasks to be done, with a separate note in each giving an explanation. With each task, we also give a suggested time to complete it.

6 Submission

Compress exactly the following files into a `zip` file that you will then submit on the CMS:

- `README.txt`: This file should contain your name, your NetID, all known issues you have with your submitted code, and the names of anyone you have discussed the homework with (except the course staff). Also, if you want to explain something about your code that you think needs clarification, add a few paragraphs here.
- All the `.java` files needed for your program.

Do not include any files ending in `.class`.

All `.java` files should compile and conform to the prototypes we gave you. We write our own classes that use your classes' public methods to test your code. *Even if you do not use a method we require, you should still implement it for our use. Do not change the signature of any method that is in the release code.*

6.1 Grading,

Solutions will be graded on correctness, the quality of the algorithms, and style. A correct program compiles without errors or warnings and behaves according to the requirements given here and in the comments of the code. A program with good style is clear, concise, and easy to read.

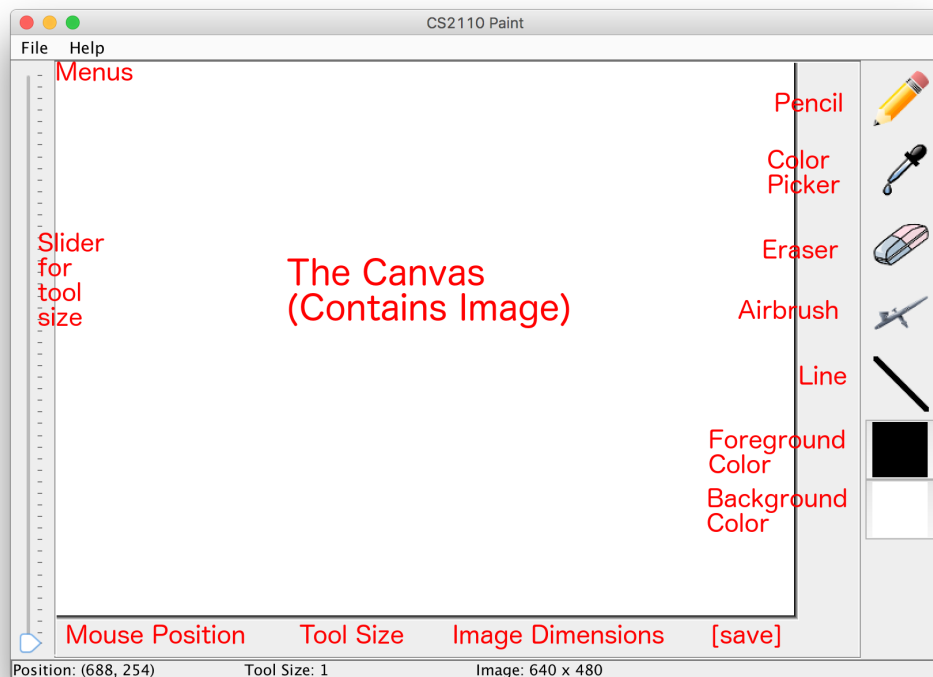
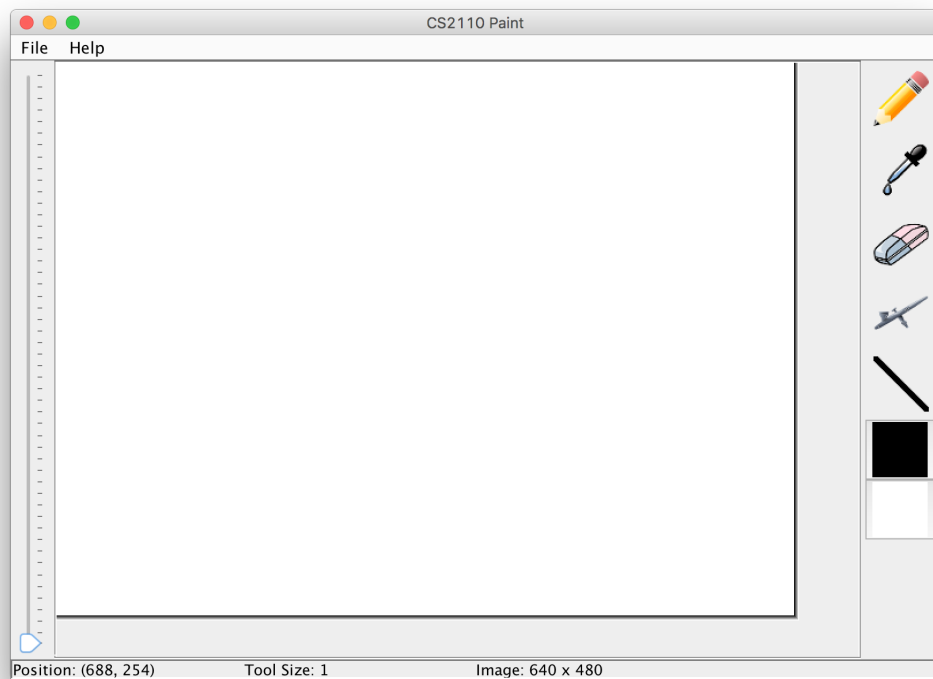


Figure 1: Fully implemented paint program.

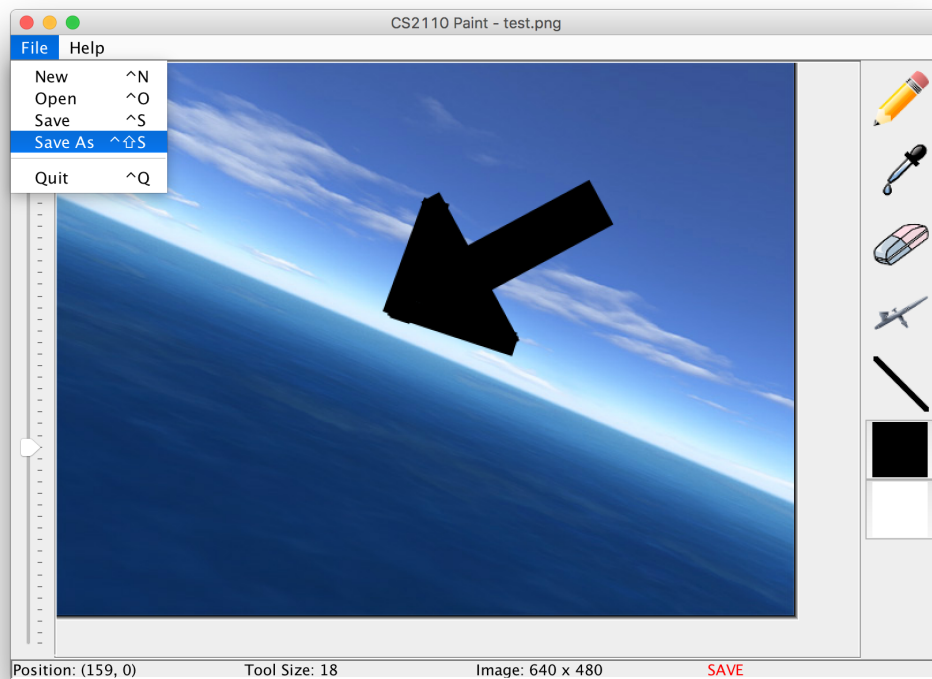
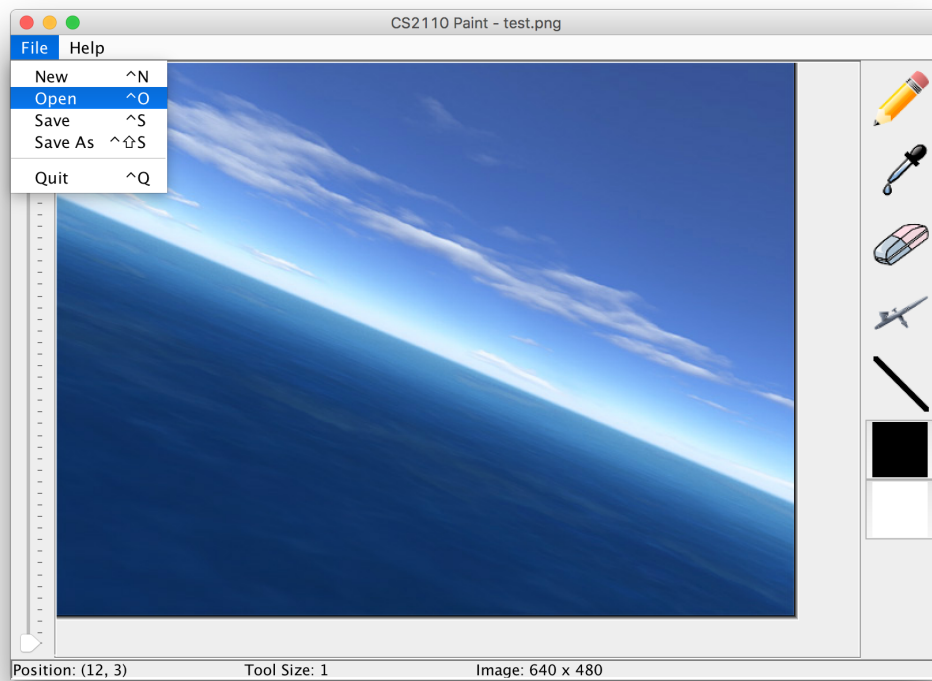


Figure 2: Open and save image files.

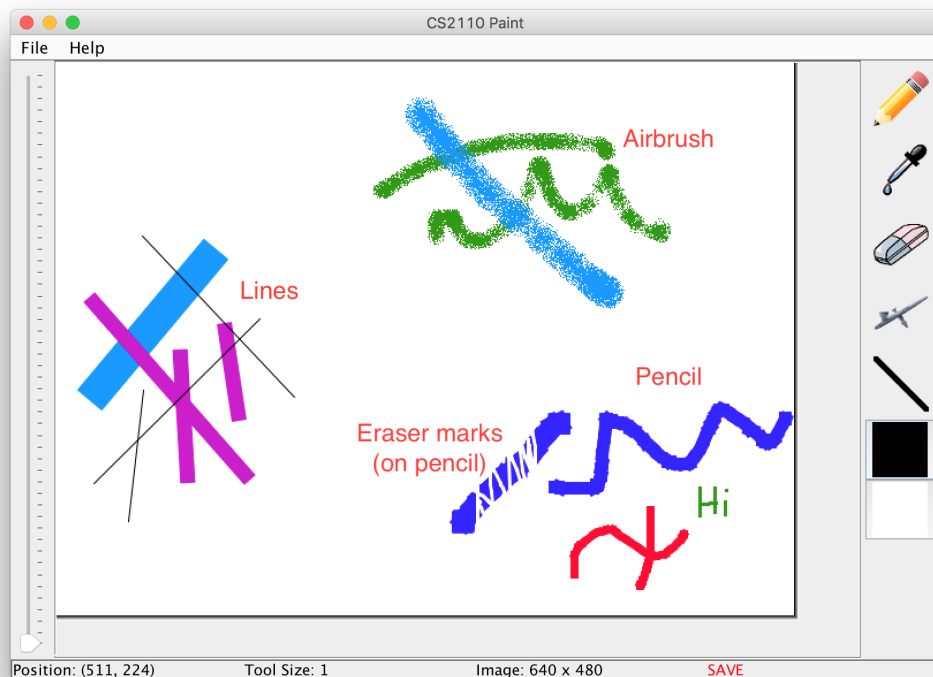
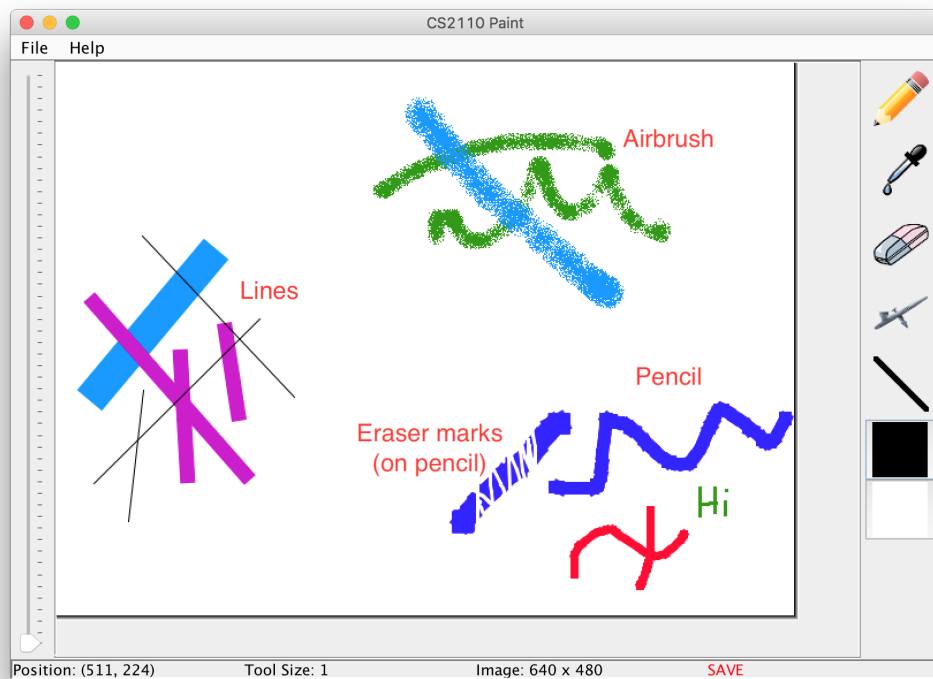


Figure 3: Drawing.