CS 2110 — Fall 2017

Assignment A5: Paint Program

Due: See the CMS

In A5, you will write parts of a simple paint program. You will implement:

1. Freehand drawing, erasing, airbrushing.

2. Drawing lines and circles.

0 Instructions

0.1 Grading, Partners, Academic Integrity, etc.

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.

Partners. 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.

Academic integrity. 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.

A5 FAQs. Pinned piazza note A5 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.

Timely work. You will get extremely frustrated and tired and worn out if you try to do this in the two days before the deadline. You won't have time to think, to ponder, to debug, to ask questions of the consultants. Do your best to adhere to the schedule we suggest.

Time spent. Please keep track of the time you spend on A5. We'll ask for it, as usual, at the top of one of the files you submit.

1 Installing the code

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

1. In Eclipse, create a new Java project called, say, `a5` and drag the `.java` files into the project's `src` folder.

2. Place the image files into the project by dragging them over the project name in the Eclipse Package Explorer.

Method main for the program is in file `PaintGUI.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:

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

`DrawingPanel`: 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.

`PaintGUI` and `DrawingPanel` 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.
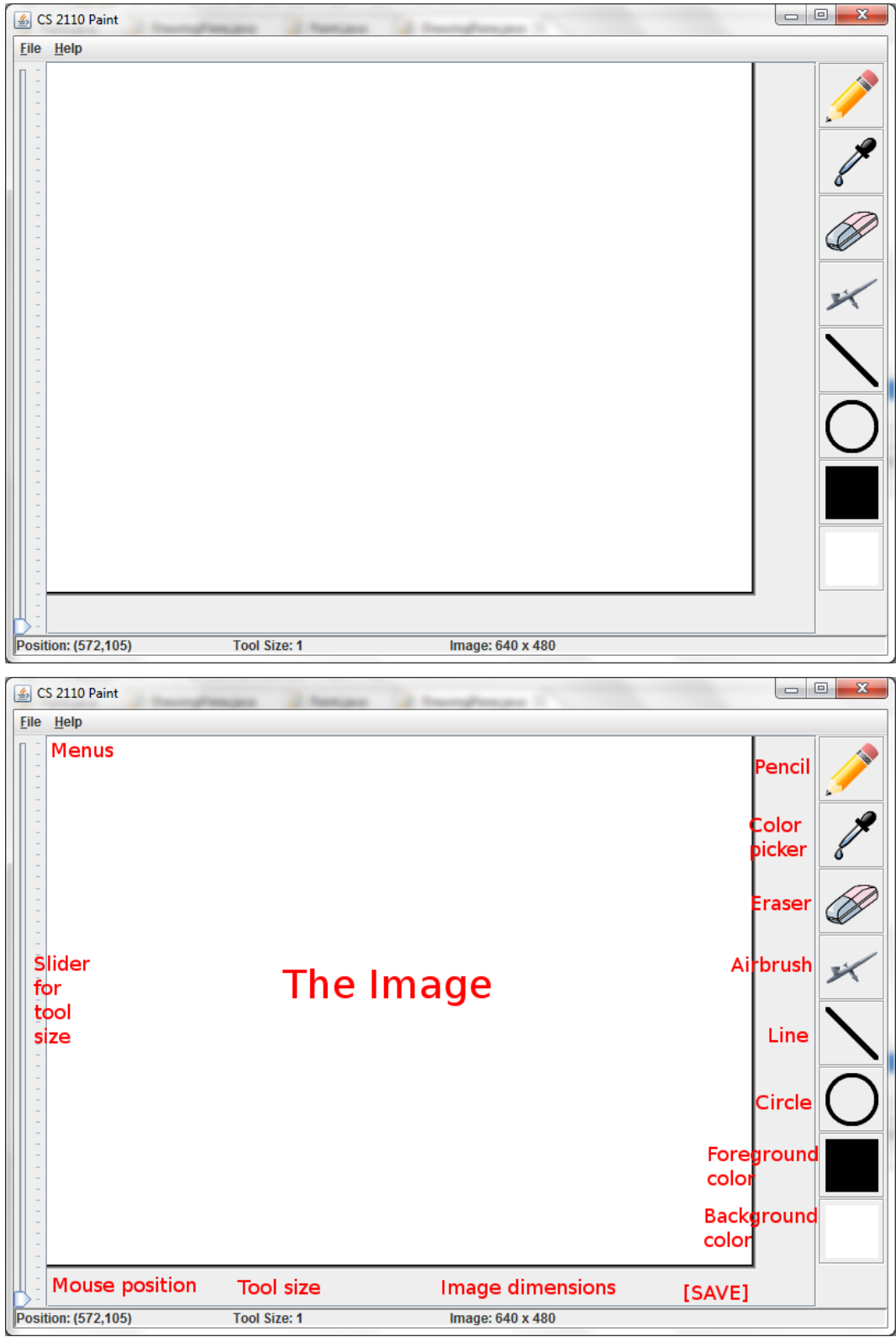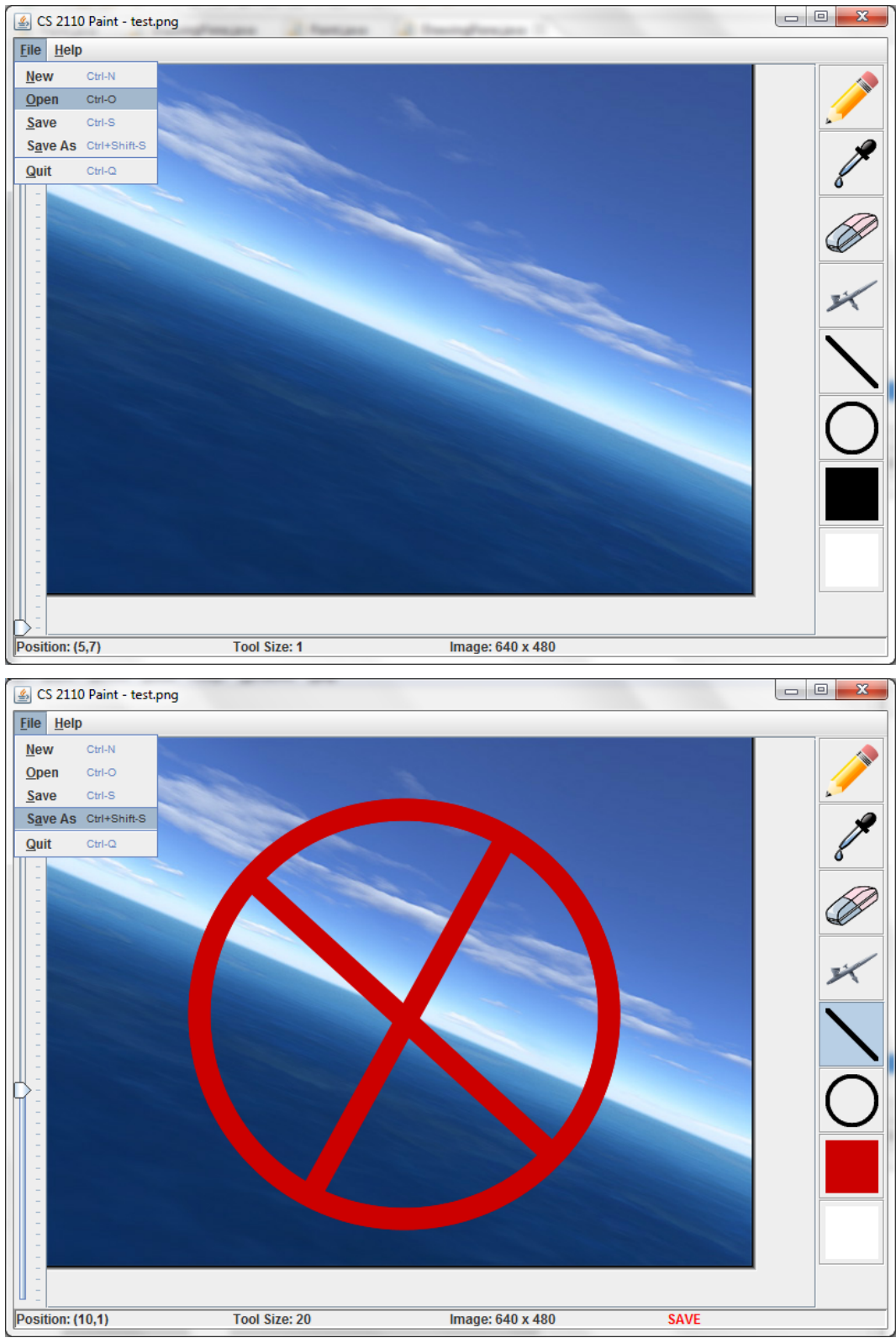
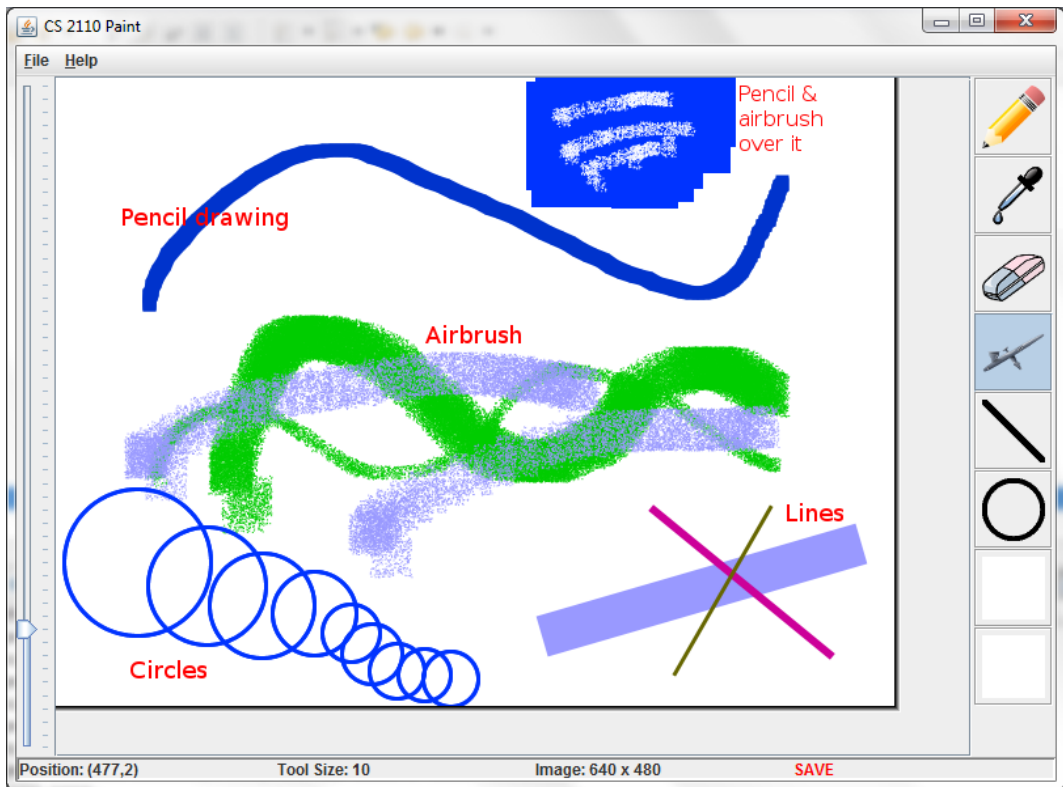Figure 1: Fully implemented paint program.
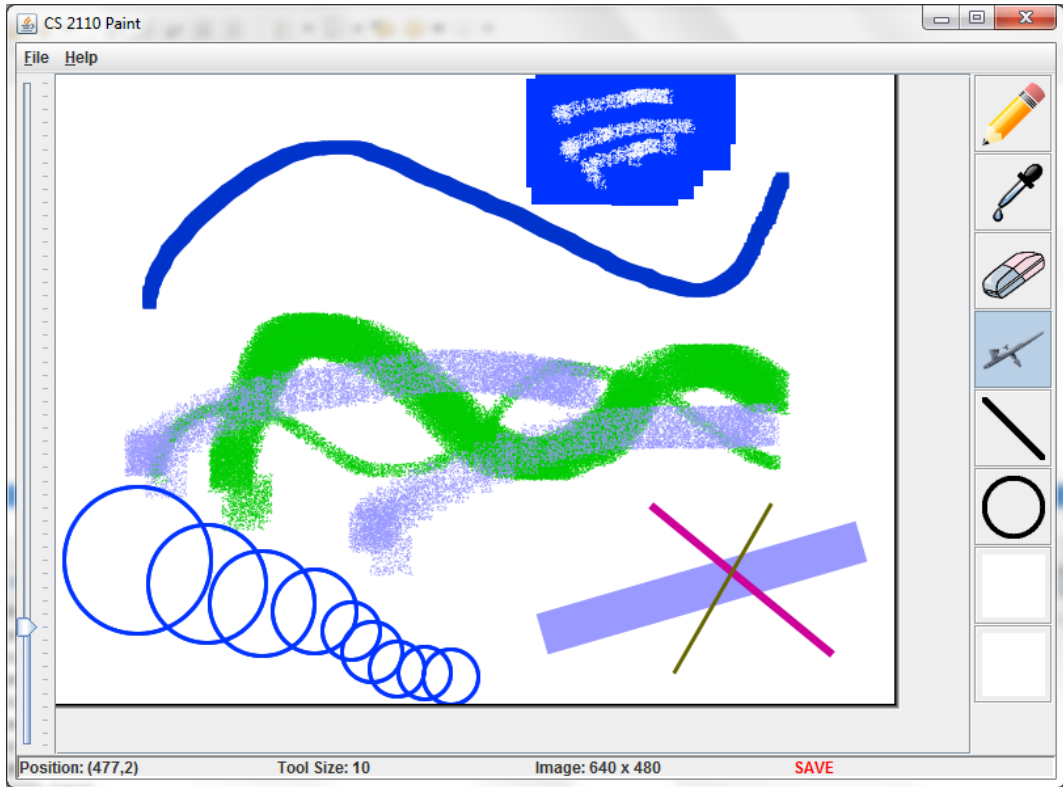
Figure 2: Open and save image files.

Figure 3: Drawing.

4 Class `PaintGUI`

The parts that have to be implemented have been clearly marked in the release code. There is very little to be done here and it is very straightforward, but you need to read and understand the code well before you attempt to make any changes. Look at the pinned Piazza note A5 FAQs for full instructions and guidance.

5 Class `DrawingPanel`

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. Some remarks:

- 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. The eraser should behave similarly.

- The airbrush does not have to leave a continuous trace.

- 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.

- Do something similar for the circle. For example, when you first press the mouse, you fix the center of the circle and the second time you fix a point that is on the circle and draw.

6 Submission

Please fill in the time spent on A5 at the top of class PaintGui and also tell us what you thought of the assignment. 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.*