Suoyi Yang
ID: 304449872
Email: suoyiy@ucla.edu
Pic 20A Disc 1B
6/2/2016

Extra Credit Assignment Report

For the extra credit assignment, I chose to create the painting app, which I named Painter. The project function is extremely similar to Microsoft Paint so I based much of the design and layout of my project on Paint. Thus I created a JPanel to the left section of the JFrame, which consisted of two columns of buttons. From the buttons, the user has the option to draw rectangles, circles, lines, draw freely, erase, or draw colored shapes. I chose to add ImageIcons to the buttons instead of words and button titles because I thought it would both be more aesthetically pleasing and allow users to see exactly what they would be drawing. I did have to resize the images to allow the buttons to be more appropriately sized.

When it came to adding colors to the shapes and lines, I initially planned to create several color buttons for the colors offered by the Color class, but I found there were very limited amount of colors available. I searched around on the java documentation website and found JColorChooser to be the ideal class to use. It neatly pops up a dialog with a large variety of colors to choose from and create. In addition, there is no need to create separate ActionListeners for each color button.

When the program is first launched, I set the frame to size to 1000x800 pixels. For each additional new painters created by the user, it will ask for the desired size of the new painter. When the user exits or quits from the new painters, it will not affect painters previously opened. Instead it will simply dispose the frame without stop the program. Only when the first original painter is closed will the entire program stop. Before exit and quit can be executed, the user must first confirm their choice to quit.

I decided to create a separate class called Canvas that extended JPanel. This would be added to the center of the JFrame and would be where all the drawing takes place. I set the background color to white to more closely mimic a paper. I also had to use Graphic2D shapes and objects because when simply using graphics, the shape would immediately appear on click without allowing users to drag out the mouse to set the size. I used MouseListeners to allow shapes to be created based on user's mouse clicks and drags. I designed the Painter so that the shape would be drawn when the mouse is released. However, to allow users to see what they are drawing before mouse is released and image is drawn, outline of the shape the user is drawing will appear when the mouse is still being dragged. I created methods to draw shapes and lines, which returns shapes and can be added into an ArrayList. The ArrayList will then be looped in the overridden paintComponent method to allow previously drawn shapes to remain on the JPanel when new shapes are made.

Circles, rectangles, and lines are created using Ellipse2D, Rectangle2D, and Line2D classes. The free draw is made by storing all the points the mouse dragged through into an array and using the drawLine method to connect all of them. Eraser is made similarly but the color is set to white instead.

I created another class called ShapeInformation to store the color the user is currently using along with the shape being created. It also contains the stroke size since I made the stroke size of the eraser larger. If I simply used the static Color variable of the Painters class or use an if-statement to set the size of stoke in the for-loop, each time the user choses a new color or the stroke size is changed due the eraser, all the drawings previously drawn are changed to match the color and stroke of the current shape’s setting. To prevent this, I created a ShapeInformation object to store a shape along with its stroke and color. In the for-loop, it will call methods of ShapeInformation to set the color and stroke size of the current shape.

When it comes to saving the image, users can chooser from saving it as a PNG or JPG. It was shown in class how users could choose a file to open in an open dialog. I used a similar method to save the image and allow user to choose the save location using a save dialog.