Brandon Kindrick

Jose Mireles

Daniel Williamson

CS 136L-1

Due: April 12, 2016

**Lab 7: Control Panel**

**Understanding the Problem:**

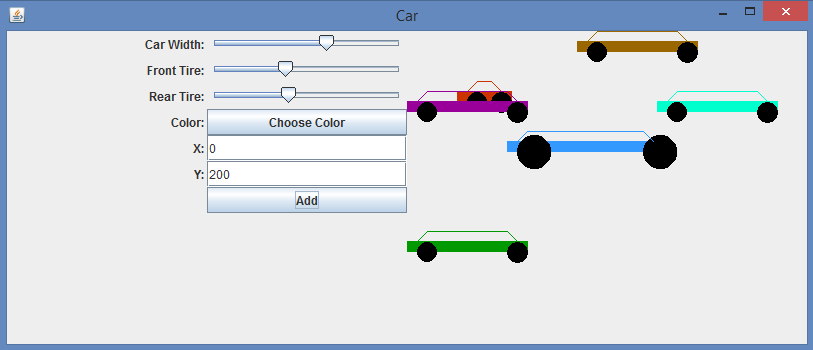
In this lab, we were tasked with building a GUI window that contains buttons, sliders and text fields that control the length, both tire widths, color and x-y coordinate of Car objects. These cars are then printed to the window using a button. We had to use three JSliders, two JButtons, and two Text boxes. The three JSliders were used to control the Car Width from 0 to 200, the rear tire width from 0 to 50, and the front tire width from 0 to 50. One button was used for changing the color, and the other to add the car to the screen. The two text boxes were used to change the car’s x and y coordinates.

**Planning:**

We began by taking our CarComponent and Car class from a previous lab. We altered the code so it still compiled without the PaintBucket class. We then made a new class called Customizer that took the place of the previous class CarViewer. This initialized our JFrame and allowed us to print our cars. Our constructor for the new class included the buttons, sliders, labels, and test fields needed for the GUI. In addition, we added a mouseClicked method that would perform a certain action depending on what button was clicked. For the first button (Adding a car), it took the values from the x and y coordinates, car width, back tire width, and front tire width sliders. Also, we created a private method to create a color choosing panel for the user. These values are the parameters for the Car class. This added the car onto the frame with the color, car measures, and coordinates that the user specified.

**Implementation:**

To test this program, we built several cars of varying length and with different colors. Our program detects whether a color was entered and if none is selected; it is defaulted to Black. The program can handle cars printed to the screen, only within the Jframe. The cars will not print if they are printed off the screen. Also, the program will give an error if you do not type any coordinates into the x and y text fields. Entering a double, float, char, or string into the text fields will also cause errors in the program.



**Conclusion:**

This lab was not terribly difficult but became tedious at times. The first major issue we encountered was getting the labels to print next to their respective button, slider or text box. We figured out how by using the “JLabel.RIGHT” keyword as a parameter. The next major issue we ran into was getting our Add button to draw a car to the screen. This took A LOT of trial and error but we accomplished it by changing our if statement within our Mouse Clicked event. The last major issue we ran into was the color choosing widget. We had no issue making a car with a hard coded color print to the screen, but when it came to coming up with a color chooser to pick the color, we ran into issues. We fixed that adding a dialog that take sin the action of the button with a default color of black.

If we went back and refactored our code, we would either throw an exception or default our x - y text fields to a certain number. That way it would not crash if they were left blank. If we threw an exception, it would also prevent the program from crashing if a char or string was entered to the text fields.