CS136 Lab Section 1

Clayton Williams Joshua Pollock

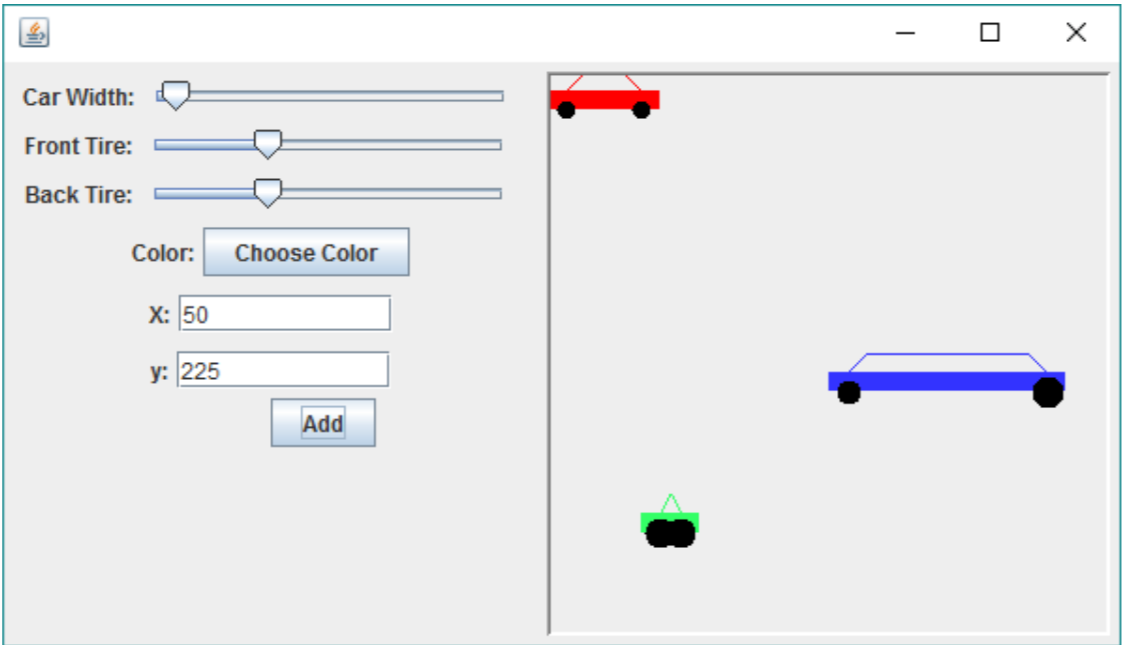
2 May 2017

**Lab 10 – Control Panel**

**Problem Statement:**

In this lab, we will be utilizing our programs from lab 3, the painting cars lab. We will be required to add on to our existing code to make a frame that contains sliders, text fields, buttons, and a color chooser to customize our car’s color, size, and location.

**Constraints:**

1. A slider ranging from 0 to 200 will control car length
2. Two sliders with ranges of 0 to 50 will control the size of the tires
3. A JColorChooser will be able to prompt the user to choose a color for the car
4. Two text fields will be created to get user’s input for X and Y coordinates
5. An example output is:
   1. 

**Assumptions:**

* That the format of the frame does not need to match the exact layout of the example

**Features:**

* createComponents()
* CarComponent cars
* JFrame frame
* JPanel sliderPanel
* JButton addCar
* JSlider frontTire
* JSlider backTire
* JSlider carWidth
* JTextField yCoor
* JTextField xCoor
* JButton color
* JLabel xLabel
* JLabel yLabel
* JLabel widthLabel
* JLabel fTireLabel
* JLabel rTireLabel
* int frontWheelSize
* int backWheelSize
* int carSize
* Color carColor
* int xcoord
* int ycoord
* CarViewer frame =new CarViewer();

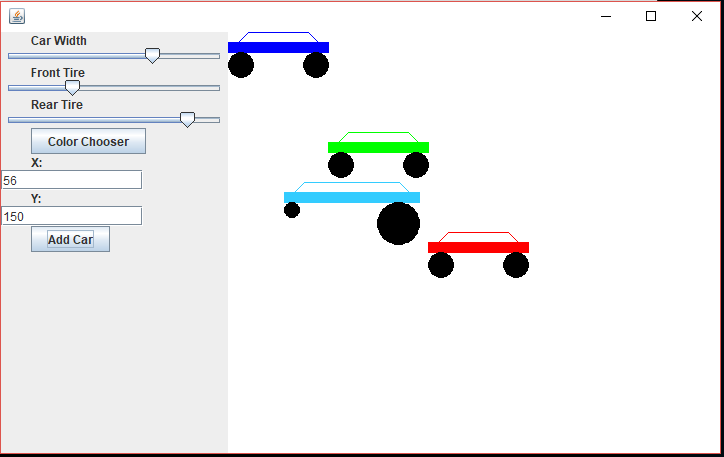
**Planning and Implementation:**

We started off this project by fixing our lab 3 code. There was just a little that needed to be fixed but our code would not function. These fixes were quite simple and were just removing dependencies on the PaintBucket and fixing some of the coordinates for the car drawings.

After this we began rewriting CarViewer so that it would contain a panel that would contain the cars themselves, and another panel that would contain the sliders/text fields. We decided that the background of the cars panel should be white and the sliders panel could remain the default color.

Once we finished these planning phases were moved on to the implementation of the program. We did this easily, but did run in to a few errors along our process. Whenever we added a car to the panel, it would also add another panel. We had no idea why this was occurring and eventually asked for help from the TAs. Their recommendation was to create an arraylist that would hold every car when it is drawn and to create a drawing method that would paint the arraylist. This solution worked and besides a few tweaks, our code was complete.

**Running Application:**



**Reflection & Refactor:**

Overall, this lab was relatively straight forward and a good opportunity to refactor the previous Car Painting lab. The hardest part of the lab might have been trying to fix a bug in which every time we hit the “Add Car” button it seemed to create a new panel, and would not display the cars properly. We are quite satisfied with our solution and do not see much room for improvement.