CS 1632 - DELIVERABLE 2: Unit Testing Ruby Rush

Sean Mizerski

Github Username: stm107

https://github.com/stm107/D2

Overview and Challenges

The main challenges that I faced when writing this program and the associated tests was that this was really a crash course on Ruby for me. I had never written anything in Ruby before besides simple exercises, so writing an entire game (though it is a simple game) and all of the associated tests was a learning experience.

A lot of my time was spent thinking, “I know how to do this in Java, so how would I do it in Ruby?” Ruby is a user-friendly language in that you can write code that is pretty similar in style to java (except for those pesky semicolons!) and it will basically work, with some exceptions. The real annoying issues that I had centered around using rubocop. Initially, I wrote some large chunks of code and ran rubocop at the end, and ended up having more errors than lines of code. Rubocop doesn’t like java-style if-blocks or the use of the tab button in Sublime Text, and I had a lot of that. I also had a ton of trailing white space. I must hit the space bar when I’m thinking through what to write next. As a result, I ended up rewriting most of the code, doing small segments at a time, and running rubocop after I completed each method. I still have 2 rubocop violations, but those can be taken care of with the three get-out-of-jail-free cards we get on this assignment.

Other than that, I found using Ruby and Minitest to be fairly simple and intuitive. I didn’t have any major struggles there.

Screenshot of Unit Test Run

