Code Coverage Analysis

# Instructions

1. Enter your team letter and name: J3M
2. Create screenshots of each tier-level page of the JaCoCo coverage report. Crop these screenshot as necessary so that the pertinent information is readable.
3. In your analysis, provide suggestions for improvements in the unit tests for each tier which are more insightful than the obvious "Write more tests" suggestion. There will be significant gaps in coverage due to the fact that you might not have unit tests for all of the project’s components. That’s expected.
4. In the last section of this file, identify two specific components, i.e. classes, that were unit tested. You will identify one with good code coverage and one with poor coverage.
5. Upload the final Word file to the *Code Coverage* dropbox in the **Team** **Exercises** category.

These statements in orange are additional instructions for the content for this analysis document. Perform the action then remove this text.

# UI Tier

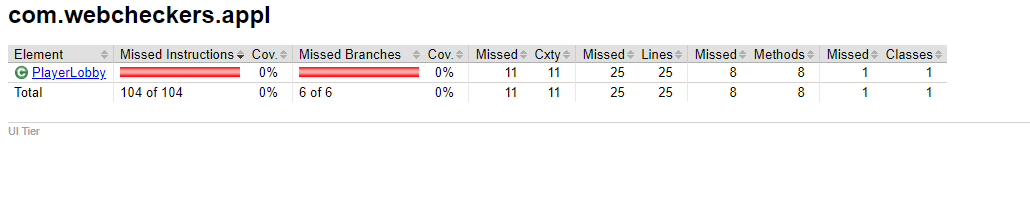
This is our analysis at the UI Tier code for the project.

## Analysis

We had a difficult time creating working tests for the ui elements. The only two we did anything on were GetHomeRoute and PostSigninRoute. Thus we need to create more tests to get better coverage

# Application Tier

This is our analysis at the UI Tier code for the project.

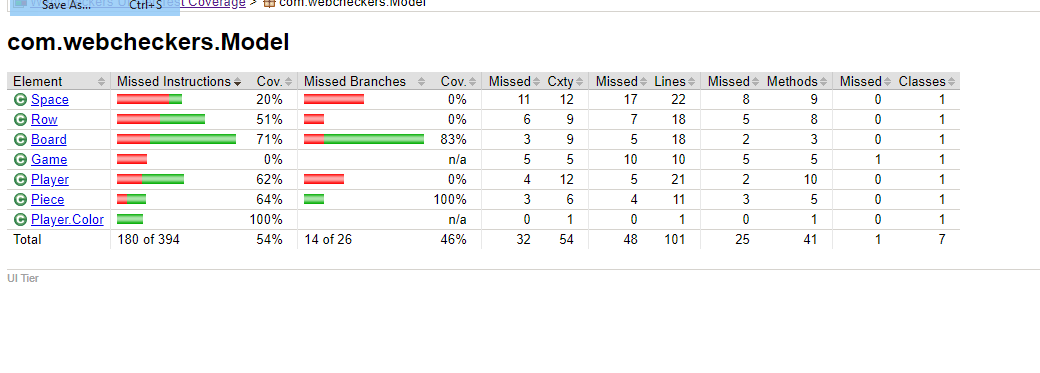


## Analysis

Since we did not create a unit test for Player lobby we did not get any coverage here. We should create one in the future to increase our coverage.

# Model Tier

This is our analysis at the Model Tier code for the project.



## Analysis

We have about half of our code covered here. This is ok for the moment, as much of this is still not fully implemented. However as we implement more of the project, it would be a good idea to create more tests.

# Well-tested Component

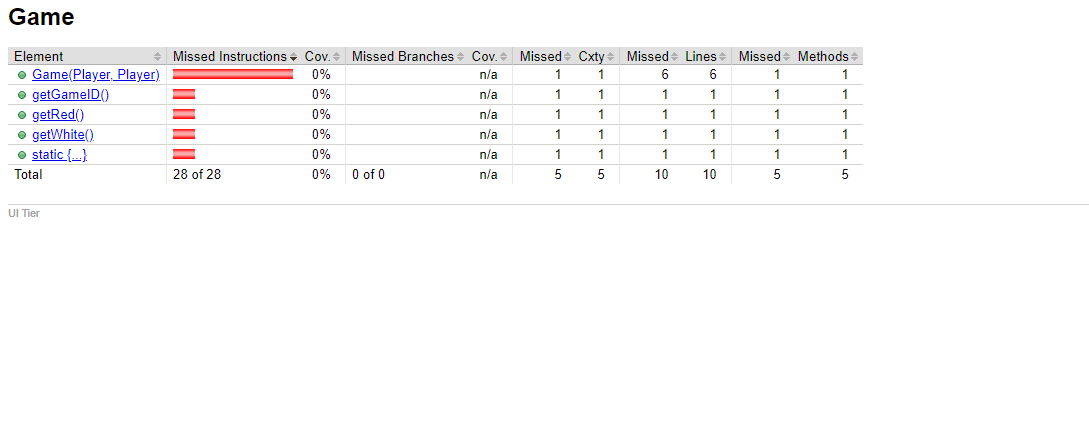
This is our analysis of a well-tested component.

## Analysis

All of our created function are shown to work here, the only things not tested are functions that are not yet implemented.

# Poorly-tested Component

This is our analysis of a poorly-tested component.



## Analysis

We have not done a unit test for this class yet, due to it not being fully implemented. Thus once we implement it, we need to create a proper unit test for this class.