My Findings

Where code was added

Missing code 01: ConnectionBasicsTest.java, Line 44

Missing code 02: ConnectionBasicsTest.java, Line 52

Missing code 03: ConnectionBasicsTest.java, Line 63

Missing code 04: ConnectionBasicsTest.java, Line 78

Missing code 05: ConnectionBasicsTest.java, Line 86

Missing code 06: ConnectionBasicsTest.java, Line 97

Missing code 07: ConnectionBasicsTest.java, Line 100

Missing code 08: ConnectionBasicsTest.java, Line 126

Missing code 09: GameTest.java, 35

Missing code 10: GameTest.java, 46

Missing code 11: GameTest.java, 61

Missing code 12: GameTest.java, 77

Missing code 13: RankingTest.java, 79

Missing code 14: RankingTest.java, 82

Missing code 15: RankingTest.java, 86

Missing code 16: RankingTest.java, 94

Missing code 17a: RankingTest.java, 100

Missing code 17b: RankingTest.java, 113

Missing code 18: RankingTest.java, 123

Missing code 19: RankingTest.java, 126

Missing code 20: RankingTest.java, 130

Missing code 21: RankingTest.java, 134

Missing code 22: RankingTest.java, 184

Missing code 23: RankingTest.java, 196

Missing code 24: RankingTest.java, 204

Missing code 25: RankingTest.java, 255

Missing code 26: RankingTest.java, 265

Missing code 27: RankingTest.java, 275

Missing code 28: RankingTest.java, 281

Missing code 29: RankingTest.java, 301

What you can test with Selenium

ConnectionBasicsTest.java

GameTest.java

RankingTests.java

Strategy1.java

Strategy2.java

Level of test: System test

Our project - Selenium is used to do the human interaction components of a web project. This can be used to insert text into text boxes or click button on the page to make sure that the program goes through each paths oor works how it is intended for it to go.

General - “Selenium is a framework for the automated testing of web applications. Using Selenium, you can basically automate every task in your browser as if a real person were to execute the task. The interface used to send commands to the different browsers is called Selenium WebDriver.” - <https://www.smashingmagazine.com/2018/04/feature-testing-selenium-webdriver/>

What you can test with Selenium and Junit

With JUnit we are making sure that the proper text is presented on screen and that the program works the way we expect it too. With Junit we were able to test to see the number of human players that are in a game and the text that supposed to appear on screen.

What you cannot test with Selenium combined with Junit

Selenium

Can not test an all AI game and if the proper number of human players are connected. Also, you cannot check that that the right output is displayed on the page based on the decision that was made. Also in general you can’t use selenium if the program is not web based application.

Strategy 1

For Strategy 1, You are able to get the AI to test, if they have a straight or higher but, in reality having the AI test if they can get a full house based on if they have a pair, three of a kind, or two pair is not possible. This can be tested if we rig the game in the back end but, not by AI, because cards are given out randomly, we cannot assure that they will end up with a full house in the end.

All AI Game

The possibility to have an all AI game is not possible to test with Selenium and the current project. The way that it is currently code of the project requires at least one player to be able to function properly. If there is no human player then there would be no admin of the game therefore you can not ever hit the start button.

Strategy 2

Strategy two can not be done through selenium and using Junit. The reason why this wouldn't work is because it requires that player 1 always users strategy 1. The issue with this is we don't actually know which card the player has in their hand or how to find out what their current hand is without having to write new code. For the else statement we aren't really able to tell which cards are shown so we won't be able to tell if there are 3 of more cards of the same suit shown. The fact that we do not know these information then it makes it all very difficult to be able to test this strategy. In the assignment I tested strategy 2 by rigging all players to follow either strategy 1 if/ else and by also checking the strategy 2 else. For testing strategy 1 we have a hand of straight or higher and for the else we have a hand that is one pair which gets “Lucky” and ends up getting a full house. For the last AI player he see that the second AI now has 3 cards of the same card face up and keeps all his cards that are a pair or more, which in my case is a three of a kind. Once this method is done, it should have covered all the possible steps that is required for strategy 2.

how you’d go about using Selenium and other testing tools to test your game’s interface AND your game’s logic in a systematic way

The way you can test you game’s interface you can use selenium. This allows the programmer to interact with with the web page and interact with the buttons and textboxes that are located on the page. In addition to the interface we have to test the logic of the game. One way we can test the logic is by adding print statements as we test and constantly test as you code, but in a more professional and automated way you should use testing applications like cucumber or selenium. In this course we used junit to test different methods and functionality of our programs. For example, Junit can be used to make sure all the console output matches up with the desired result we want from each test. Another thing that we did was use cucumber to test the different user stories that a project would have.