**Adam Darr**

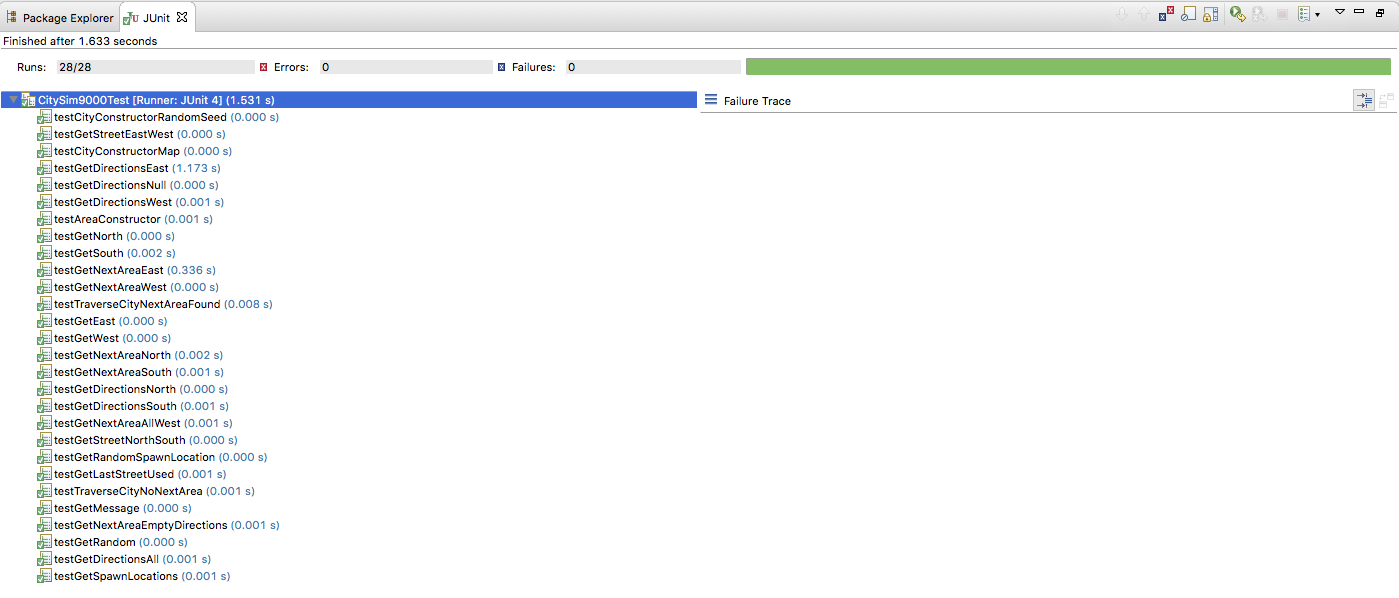
**https://www.github.com/adamdarr/city-sim-9000**

**CS 1632 - DELIVERABLE 2: Unit Testing and Code Coverage**

**Summary and Testing Concerns**

On a whole, creating both the CitySim9000 program and writing unit tests for it went well. Since I have used JUnit previously, the only points of struggle were setting up the environment, figuring out the Mockito syntax, and coming up with the proper equivalence classes to test for. Setting up my environment took a decent amount of time because it was difficult to get the Mockito libraries properly added to my build path in Eclipse. However, once I got the environment set up, learning Mockito wasn’t terribly difficult since I have used EasyMock in the past and the syntax is similar. While both testing frameworks are comparable, it still took a bit of time to get my head around some of the subtle differences between EasyMock and Mockito. Finally, finding equivalence classes to test and writing them as unit tests was likely the hardest part of the assignment for me. For example, my traverseCity() method has two equivalence classes, either the current area has some other area or areas associated with it that the driver can traverse to or the current area has no so such places to traverse to, meaning the driver has left the city. Testing for these equivalence classes proved difficult since my traverseCity() method is void. In order to get around this, I had to record the output stream and compare it to what the proper output should be (eg. the output should contain something like “Driver X has left the city!” if the driver leaves the city). Struggling through these unit tests, however, gave me a deeper understanding of both doubling and stubbing since both were necessary to test the method described above. In the end, I was able to have all my test cases pass (see below).

**Screenshot of Executed Unit Tests**



**Unit Test Code Coverage Report**

