

SIMON FRASER UNIVERSITY  
Department of Computing Science  
CMPT 276: Project

## Project, Phase 3: Testing

### 1 Considerations

You will be working on your group repository. Each team member should contribute continually to all phases of the project.

Upload the report under the same “Documents” directory in your team’s repository. The report should be named “Phase3Report.pdf.” We will mark the report along with the history and final version of your code on GitLab on the specified deadline. See the Schedule page on Canvas for deadlines.

### 2 Testing Your Game

The third phase of the project is about testing your game. You will write unit and integration tests for your game, focusing on both the coverage and quality of the tests. You may modify, improve, debug, and refactor your production code as you make progress in testing your program.

#### 2.1 Unit and Integration Tests

First, you will write unit tests for your game to test single features of your game in isolation. Identify the features that need to be unit tested, and briefly explain these features in your report. Create JUnit<sup>1</sup> tests for these features. Cover as many features as possible. Next, identify important interactions between different components of your system (e.g., the logic and the user interface/file system/etc.). Explain such interactions in your report, and create integration tests for them. **Commit and push your tests often, and merge your branch and the master regularly.** Document which test case/class covers which feature/interaction in your report.

##### 2.1.1 Test Automation

You will create all the tests using the JUnit framework. Anyone should be able to automatically compile and run your test suite with Maven so make sure to keep following Maven principles. Place your tests in `src/test/java` and the required resources in `src/test/resources`, if applicable.

Complete the README file of your project with complete instructions on how to build, run, and test your game.

##### 2.1.2 Test Quality and Coverage

The quality of your tests and assertions will be assessed as well so pay attention to the quality of your test code. Discuss the measures you took for ensuring the quality of your test cases in your report.

Code coverage has been traditionally used as a popular test adequacy criterion. Measure, report, and discuss line and branch coverage of your tests. Document and explain the results in your report. Discuss whether there are any features or code segments that are not covered and why.

##### 2.1.3 Findings

Briefly discuss what you have learned from writing and running your tests. Did you make any changes to the production code during the testing phase? Were you able to reveal and fix any bugs and/or improve the quality of your code in general? Discuss your more important findings in the report.

---

<sup>1</sup><https://junit.org/>

### 3 Deliverables

- **The code:** test code under `${basedir}/src/test/java`, test resources under `${basedir}/src/test/resources` if applicable, and any potential changes to your existing implementation under `${basedir}/src/main/java`.
- **The report:** should include everything required above and should be at most 5-6 pages. Name your report file **Phase3Report.pdf** and place it in the “Documents” directory on your repository.
- **README:** should be updated with complete and clear instructions for users to build, run, and test your game.