Cokie Parker
Marcelo Morales
Andrew Suh

Project Proposal

Background

https://github.com/ahmetcandiroglu/Super-Mario-Bros

The subject application that we plan to test is the popular super mario bros game. This particular application was developed by a group of students for a class project. This game is used for leisure and for one's own enjoyment. This game can be run on any computer that has JDK installed. A typical user for this software application are casual gamers. This application contains around 2192 lines of code. There are 33 classes, 1 interface, and 92 methods.

Testing Plan

We plan to use many different testing techniques to ensure a comprehensive testing environment.

1. Blackbox testing using the .jar file of the game to ensure the specification requirements are followed.

2. Whitebox unit & integration testing to ensure correctness of each individual class and methods.
3. Mutation based testing on game classes
4. Interaction based testing between all the entities involved in playing the game
5. Property-based testing to ensure the correctness of methods using inherent properties, possibly more effective than traditional unit testing due to the large testing input domain.

We will aim to get full statement coverage and as close to full branch coverage as possible. After implementing the four different testing techniques above, we may choose to additionally use mock testing if time allows.

Testing Tools and Frameworks

We will be leveraging different tools and frameworks to maximize our testing efficiency.

- JUnit - unit testing for Java
- Selenium - GUI testing
- PIT - mutation testing tool for Java
- Interction based testing

- Unit testing for all of the classes
-

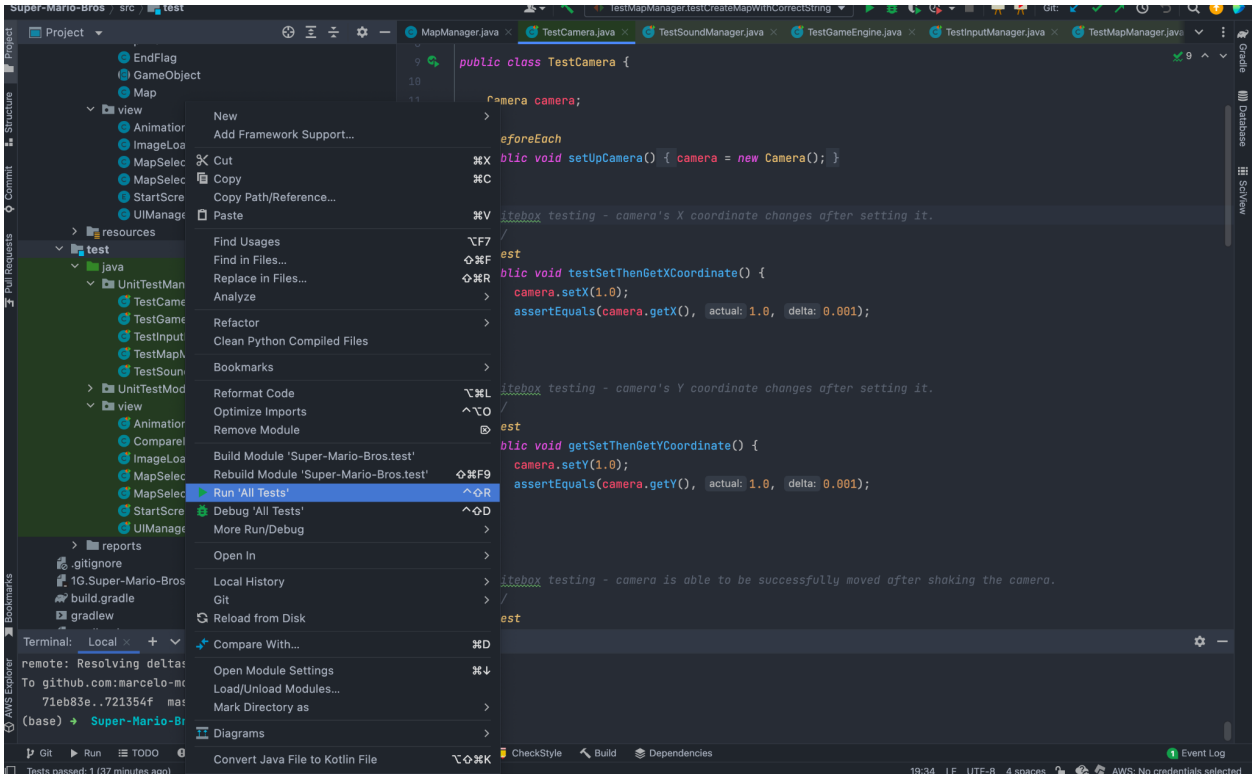
Tasks

To ensure that all members work on their assigned tasks within the timeline, we have listed all the necessary tasks and assigned a member and a completion date.

Task	Assignee	Completion Date
Set up GitHub Repo where tests will be located + corresponding project board	Marcelo	4/8/22
Unit Tests/Whitebox test of all entity classes - brick, enemy, hero, prize	Marcelo	4/13/22
Integration Test of user playing game from start to end	Cokie	4/13/22
Blackbox testing of game classes with each other - brick, enemy, hero, prize	Andrew	4/13/22
End user GUI interaction testing using Selenium	Marcelo	4/20/22
Property-based testing of classes/methods using PIT	Cokie	4/20/22
Randomness testing to ensure the random functions of the game has even distribution of outcomes	Andrew	4/20/22
Clean up code, expand coverage, fix faults, and possibly add additional testing techniques	All members	4/25/22
Presentation preparation	All members	4/29/22

Running Tests against suite

- Download zip file submitted, or can also go to public forked repo https://github.com/marcelo-morales/Super-Mario-Bros and git clone
- Run ./gradlew jacocoTestRepo to produce the jacoco report that produces our given statement and branch coverage
- ./gradlew pitest to produce our respective mutation scores
- Recommend to open up project in IntelliJ so that can all unit tests at once
    - From three testing packages we have under the test package, can look at the individual unit tests, whitebox and blackbox tests as well



- The different types of testing that we completed are systematic blackbox and whitebox unit and integration testings and mutation testing to achieve as high branch and statement coverage possible.