SENG201 Project Report
Students: Hugo Marshall, Jeesung Park
ID Numbers: 66460203 (hma173), 97994835 (jpa239)

The project is split up into 5 packages monsterfighter, monsterfighter.core, monsterfighter.ui, monsterfighter.ui.gui and monsterfighter.test. The game is run from the monsterfighter package. The monsterfighter.core package contains all the classes that provide the game's functionality. The monsterfighter.ui package has the manager which controls all the screens, with the monsterfighter.ui.gui package having all the screen classes.

Inheritance is used with an Item abstract class with the items. All 4 of the subclasses, weapon, shield, heath heal and level up, have unique use methods that improve the input monster statistics differently. We knew that all the subclasses would have a use method, but it would be different. So we decided to use an abstract class.

Our overall test coverage of the entire game is 53.1%. With monsterfighter having 95.5%, monsterfighter.core having 95.5%, monsterfighter.ui having 32.6%, monsterfighter.ui.gui having 12.0% and monsterfighter.test having 100% coverage. The test package has 100% coverage as all our tests run successfully.

The monsterfighter.core and monsterfighter test coverage is very high as nearly all the methods in the classes are tested with only a few exceptions. The test strategy was to call a function with different values that would likely occur in the game and test that the correct values were updated. This often meant having one value on either side of an if statement, one where the if statement was true and another where it was not. While loops were tested with values that would cause the loop to break in different spots to ensure the correct result was obtained. For loops had multiple inputs checked to see that the correct values were updated for all inputs, not just the first or last.

Some methods that could not be tested were the all random events method, as this set the seeds for all the random events methods randomly, so the outcome could not be predicted and therefore could not be tested. The main method in the game environment could not also be tested as it started the screen popup.

Both monsterfighter.ui and monsterfighter.ui.gui have very poor coverage, resulting from some methods being called from the testing game environment. This poor coverage is because it is complicated to test screen popping up and closing with j unit tests. Instead, we manually tested the game throughout development to ensure that the screenmanager and screen classes were behaviouring as expected.

Jeesung statement:
The project helped me understand java a lot better and was a fun way to learn. I mainly worked on the GUI part of the programming, so I learned a lot about how one class can communicate with another class. The project took more time and effort than I thought, so I wish I had started on the project a lot earlier.

I thought it would be good to use Github because it had a git desktop app used as an interface to commit and push code. However, Github ended up not working properly due to some commit issues with Eclipse. So we wasted a lot of time trying to fix this issue but ended up using Gitlab.

I think the communication between my partner and me was pretty good, and we were able to spend time in the labs discussing ideas. Having Gitlab was very useful, and I found the built-in Eclipse commit, push, and pull buttons quite handy.

In total, I spent 60 hours on the project. I agree that Hugo contributed 50% of the final project.

Hugo statement:
Coding a big project like this takes considerably longer than expected. Until this project, I had just been coding smaller classes with a few methods and did not realize how quickly it could become so complex. Although sometimes it was frustrating when things were not going well, it was gratifying to see the game's progress and see what your work was going towards, like displaying the first monsters and doing the first fights.

Things that did go well were breaking the classes into clearly defined methods with straightforward functionality. For example, this meant that the item/monster purchasing methods could be made and not have to be changed for the gui. We had some issues with using git. Although it is helpful to share our work instantly, we had some merging problems, which I think resulted due to our lack of practice using the software. One improvement in the future would be to work on which monster attacks first. With our implementation, we have to either have the system or player monster always attack first, but I would like to change this to some random event with the chance changing depending on the difficulty. For project management, in the future, I would do more work to find out how long things actually take and then use this to better plan milestones so that I would be on track from the start.

In total, I spent 47 hours on the project. I agree that Jeesung contributed 50% of the final project.