

## Testing Strategy

Most of our unit tests test methods with a very well defined goal like the validation methods. The main control methods like `combat()` for attacking and `exchangeCards()` for exchanging, for example were tested using manual user testing. We tested that they work as intended and that the entire game happens the way its supposed to in the order its supposed to first, and then we tested what might happen if the player puts in some incorrect input. We tried to break the game in as many ways as possible in our manual tests and found it was fairly resilient to bad input. All of the UI components were also tested manually.

The unit tests thoroughly test the output of the method they test, assuring that everything the method is designed to do works correctly. Each class that we felt required unit testing has its own JUnit test class which tests all of its methods that we needed to test. Main was the exception to this as it contains a lot of functionality, so we decided to split up its tests into a few different classes like `ReinforcementTest`, `CombatTest`, etc. which cover all of the validation methods in the main class and provide a high coverage of all the code in those methods.