

LO3 - Apply a wide variety of testing techniques and compute test coverage and yield according to a variety of criteria evaluate the limitations of a given testing process, using statistical methods where appropriate, and summarise

This document will set out the techniques to be used during the testing process.

The tests to be carried out:

I decided to use the JUnit framework to carry out the testing of the software. This is because it was both easy to integrate while using IntelliJ IDEA and Maven as well as being a framework I have previous experience using.

Apptest.java: Will contain the unit test code for the input validation.

OrderValidatorTest.java: This will contain the unit tests and the system test for the order validation. As well as the system tests for the order outcomes.

OrderControllerTest.java: Will contain the code for proper exception handling.

MovementContollerTest.java: Will contain the code for proper exception handling.

DroneControllerTest.java: Will contain the code for proper exception handling.

How the tests will be carried out:

I will be carrying out an array of testing techniques including functional, structural and combinatorial. These will be further detailed on an individual level below.

Functional testing: Test cases we based on both the test plan and the requirements.

Unit tests: These tests were carried out to check that functions(especially the check functions) carried out met the restraint requirements.

System tests: These tests were carried out to ensure that the expected behaviour occurred.

Structural testing: Using structural testing in conjunction with functional testing to ensure that all aspects of the functions under testing are verified.

Combinatorial Testing: This method of testing was implemented so that I could generate a vast array of inputs in a multitude of combinations to ensure coverage of unanticipated errors.

How the test will be evaluated:

1. First and foremost, the **tests passing** is the most significant. This would signify that the software operates as expected.
2. The **diversity of the tests** is also important. The quality of the testing relies on covering the software thoroughly. A range of techniques
3. Lastly, the **code coverage** is a very important metric to analyse. We can get a numerical analysis of how tests are performed.

The Results of the Testing:

In this section we will analyse the results of the testing conducted against the criterion stated in the previous section.

1. 100% of the tests passed. This was a very good result but also almost mandatory when trying to demonstrate that the test suite was working as intended.
2. As the tests cover all the potential corner cases they have demonstrated rigorous diversity.
3. However, I was only able to achieve a line coverage of 88% (an improvement from 70% in the initial round of testing). This result was disappointing as total line coverage was identified as a desirable criterion. If given more time I would first seek to improve this score. Additionally I achieved an 95% method coverage.