Testing Report

Testing Methods and Approaches

Throughout the implementation process, the group will be testing the game using unit tests to ensure that the new code being written runs as expected. This test-driven strategy also means that if more code is added, the earlier tests can be quickly run to confirm that the new addition does not break previously working sections. Also, the unit tests being used will be white-box tests, meaning that the tests have access to the code and are not just concerned with the 'front-end' inputs and outputs. This white-box method is the most appropriate for our implementation phase because it allows the developers to more easily find the source of an error and make amendments immediately after discovery.

Once the features required for the implementation have been completed, a user requirements testing phase will begin, which aims to test higher level aspects of the software. The black-box method will be the most suitable for this stage of testing because it allows the group to see the game in the same way as the users. During this phase members of the group will not change the code so this testing can be performed on the same version of the game. This will ensure that a fix for a failed test doesn't cause a previously passed test to fail without the group noticing. This also allows the group to look at all failed tests as a whole and identify failed tests that could be causing other tests to fail by domino effect.

The tests will be displayed in two sets of tables; one table to display unit tests and one table to display requirements tests. Using these tables to display the tests means that it is easy to see which tests were carried out, as well as the results and evidence for each.

The tests will be split between all the group members and the results will be recorded alongside the test, stating whether the test passed or failed. If a test fails, additional notes will be entered into the table to record what the issue was. Then during the amendment phase the notes will be used to help fix the issue correctly.

Once the general testing phase has been completed, any amendments necessary will be made. After the amendments to the project have been made group members will retest their sections of the table to make sure the changes haven't had unintended consequences. The process of testing and correcting will be repeated until all tests are passed.

Report on Tests:

Test Statistics:

Test Type	# Tests	# Tests Failed	% Passed
Unit	53	0	100.0
Requirement	19	0	100.0
Total	72	0	100.0

There are 22 requirements but only 19 tests were performed. This is due to the abstract nature of some of the requirements meaning they would be difficult to test in any quantifiable manner. For example, requirement N1 states “All code and documentation produced must be consistent, readable, and maintainable.” This would be difficult to create a test for which could be passed and failed.

Testing Material:

To see the tests run during the testing phases and the outcome of each test, please use the following links.

Unit Tests: https://github.com/as2378/unlucky/raw/master/docs/files/Assessment3/Assessment3-UnitTests.pdf

Requirements Testing: https://github.com/as2378/unlucky/raw/master/docs/files/Assessment3/Assessment3-RequirementTests.pdf