Module	ENG1
Year	20/21
Team	33 (Short Circuits)
Members	Jack Lord, Neo Metcalfe, Sam Rodgers, Mohammad Abdullah, Qi Tang
Deliverable	Software Testing Report

A.

The software testing method we used is Agile. Our team created a test plan to keep on the schedule. We listed all the items that needed to be tested and also prioritized in order of importance. The testers and developers needed to work and communicate closely. Communication and collaboration are key. *(Kruger. N, 2018)*

Agile testing can improve the quality of the product and let us finish the project in a short duration that is why we choose it cause we only have a few weeks to complete the project. Agile testing enabled us to find the problems and deal with them quickly and require a minimum amount of resources. We could fix the bugs in the middle of the project and not at the very end of the project. Agile Testing is highly adaptable to changes. When we changed some requirements in assignment 2, the testers could still keep testing continuously as new features were added in order to save much time.

We used the Behavior Driven Development(BDD) method to help team members correctly understand each feature and build a good communication with stakeholders. *(ReQtest, 2018)*

Also through using Exploratory method to find the hidden risks within a product. This method is more adaptable to changes. When the change happened, the testers tried to identify the functions and execute the test by themselves.

We used unit testing for the new features that we developed and wrote unit tests for the rest of the game wherever possible. This was bolstered by the use of manual testing on the UI, menus, and general gameplay while playing the game.

B.

We currently have 15 unit tests, which amount to 28% coverage. All tests currently pass, since the code was written with test driven development in mind. However it was quite difficult to test a lot of the system, due to the way the libgdx library was used, many parts of the system required an active game to be running which isn't possible while testing. We got around some of this using a mocking library (Mockito), but much of the system had been written in a way that made testing difficult.

Manual testing was done as well as unit testing, we ensured that the menus were all able to be navigated as expected and all buttons had correct labels. We also played the game itself to test that all gameplay features were working correctly.

We have a test coverage report available on the website through the “jacoco test coverage report” link.

REFERENCE

1. Kruger. N(2018). *Agile Testing Methodology---5 Examples for the Agile Tester*. Available at: https://www.perforce.com/blog/alm/what-agile-testing-5-examples#:~:text=Agile%20testing%20is%20software%20testing,tested%20as%20they%20are%20developed. [accessed 9th February]

2. ReQtest(2018). *Agile Testing Process – Principles, Methods & Advantages.* Available at: https://reqtest.com/testing-blog/agile-testing-principles-methods-advantages/. [accessed 9th February]