Module: ENG1/ASSESSMENT2

Title: Continuous Integration Report

1. Charles Stubbs
2. Annabelle Partis
3. Kieran Ashton
4. Yu Li
5. George Tassou
6. Alex Shore

a).

We use IntelliJ IDEA for development and compilation. IntelliJ IDEA is a very practical Java integrated development tool. It has great compatibility with our chosen game engine libGDX, which is very suitable for developing our Java projects.

Then we use GitHub as the version management and hosting of the source code of our project. GitHub is currently the most popular and practical source code hosting platform, and it is also very easy to use. It can automatically clone the latest version from the project code base on the network to the local. After the modification is completed, the local new code is compared with the code saved in the server, and the new code is updated to the project code library on the network to ensure that the project code library on the network is the latest code.

We use JUnit to test our project. JUnit is a Java unit testing framework. Which is the most popular and most convenient in testing Java projects. Unit testing can help us quickly locate bugs after we modify part of the code. It can improve our work efficiency.

For some complex scenarios, we used Mockito with JUnit to test our project. Mockito is another testing framework for Java, which allows the creation of test mock objects in unit tests for the purpose.

b).

Continuous Integration is a kind of software engineering workflow. It is a measure to continuously integrate each member's working copy of the product into the mainline. It can resolve and optimize code conflicts between different members during development to improve work efficiency during multi-member development.

In group meetings, we divide the project into different functions and arrange them for group members. Distribute tasks to everyone according to the functions that need to be completed, to minimize the possibility of code interference during development. And in subsequent meetings, we will regularly discuss our own modifications and development with each other to ensure that members can understand other people's code more quickly.

First of all, we use IntelliJ IDEA to develop the project. Before uploading the source code, we will now perform a preliminary compilation test locally for the newly developed functions.

We use GitHub for source code management. Members will update to our team’s project code base through GitHub after completing a function.

Because it is a simple project and cost reasons, we did not set up a server for automated continuous integration. After discussion, we decided to manually compile and test the source code every day, and share it in our discussion group.

We obtain the latest source code through GitHub at a fixed time every day and build it with IntelliJ IDEA.

After that, we run our unit tests package for every function, which is written by ourselves. We use unit tests to find out if there is any bug quickly after we modify part of the code. If all unit tests passed, we moved to manual tests for complex scenarios.

Finally, we will share the test report to our discussion group and report the problems encountered during the test (if any).