

# Continuous Integration Report

Cohort 1 Group 7

## **Group Members:**

Kyle Clifton (wnp512)

Bailey Evers (rkh552)

Matt Hollyhead (fpk506)

William Martin (zjc524)

Max Pither (rkm538)

Enkhtuguldur Sarantsetseg (wbj512)

Ronny Watt (dvn513)

### Methods and Approaches:

For our project, we made use of Continuous Integration (CI) tools to help quickly identify errors in the build process, and make it easier to quickly access the latest builds for testing. We made use of tools that allowed our automated software tests to be executed after every new commit made to the project. We also configured the project to save a full build of the game's .JAR file for every commit made, ensuring all contributors and team members have easy access to the latest versions. In addition to this, we also set up a system for automatically publishing release builds, allowing us to easily share the latest stable version on our project repository. We found these approaches to be suitable for our project, as we have some members of the group not working on the implementation, so incorporating a way for them to see easily the current status of the project and download the latest build without needing to setup the project repository on their own machine helped to keep the whole group up to date on the status and progress of the project.

### Infrastructure:

We made use of the "Java CI with Gradle" workflow available through Github Actions in order to implement our continuous integration infrastructure. Based on the YAML template provided by Github Actions, we configured the workflow to automatically execute the Gradle build process after every push to the main branch (as well as on any pull requests). Part of this process also involves running the automated software tests built for our project, which allows Github to clearly mark each commit to show whether the tests are passing or not. Furthermore, we also configured an action that will upload the generated build of the game as an artifact, meaning that every push to the repository will result in a new build being generated that can be easily accessed and tested, without needing to setup Gradle or an IDE on a local machine. On top of this, we used a community-made action called "action-gh-release" which allows us to automatically publish new release builds to the repository simply by creating and pushing a new tag from within our IDE.