

Engineering 1 Group Assessment 2  
Continuous Integration  
Cohort 3 Group 23

Team Members:

Harry Draper  
Seif Hussein  
Tikhon Likhachev  
Thomas Maalderink  
Joshua McKean  
Sebastian Armstrong

## CI Method + Approach

Our approach to continuous integration ensured that our project ran smoothly and development was kept on track by automating necessary tasks such as building and testing our code. This helped us to discover and fix any issues that arose quickly, helping to make sure they did not carry over into future iterations of the project. We also worked in a single GitHub repository to make sure we did not lose track of any necessary parts of the project.

Our approach involved our CI tool automatically running the build command each time we pushed a change to the repository. Additionally, we had the CI tool run a series of automated tests as part of the build process, so that we would detect any further errors or bugs that needed to be fixed. The tool enabled us to quickly find out which tests failed by looking at the output on the job itself, where it specifies the name and line number of the test(s) that failed. We also had the build upload the generated JAR file to the repository, to make it easier for developers and users to find the latest version.

Another important part of the process was frequent commits. Anyone working on the project committed any changes they made to the repository reasonably frequently so that any issues that arose would be detected quickly in order for it to be fixed. This, along with the fact that we ensured our builds were quick, meant that we were able to guide the development process in a structured and effective way, minimising bugs and issues. It also helped to ensure that all developers were working on the latest version of the code.

This approach was suitable for our team because it enabled us to collaboratively work on the project while maintaining a high standard of quality and detecting issues rapidly. This enabled us to fix any issues quickly and continue with development of further features. It also made sure that we did not release buggy builds of the project to users by having a package of automated tests that ran on each build. Additionally, having the continuous integration framework in place early on meant that we were testing our software throughout the project and that, later down the line, builds that contained bugs were not released to users.

## CI Infrastructure

To implement our CI infrastructure, we used Github Actions with a yaml file describing the workflow we wanted to implement. Our initial implementation simply ran the Gradle build command every time there was a push or pull request on the main branch. The workflow evolved throughout the project to continuously support development and, later on, to support the creation of an executable JAR file of the game each time the workflow was run. This enabled us to ensure that the game was playable in such a state that any player could access it.

The final version of the yaml file detailing the workflow does the following:

- Sets up java version 11
- Sets up Gradle
- Runs the build command
- Packages the JAR file
- Runs our series of unit tests
- Uploads the JAR file as an artifact of the build

Additionally, for debugging purposes, the workflow lists the files in the directory of the runner.