

# **Continuous Integration**

## **Group 3**

Liam Martin

Aaliya Williams

Lucy Crabtree

Kai Nichol

Sammy Hori

Tim Gorst

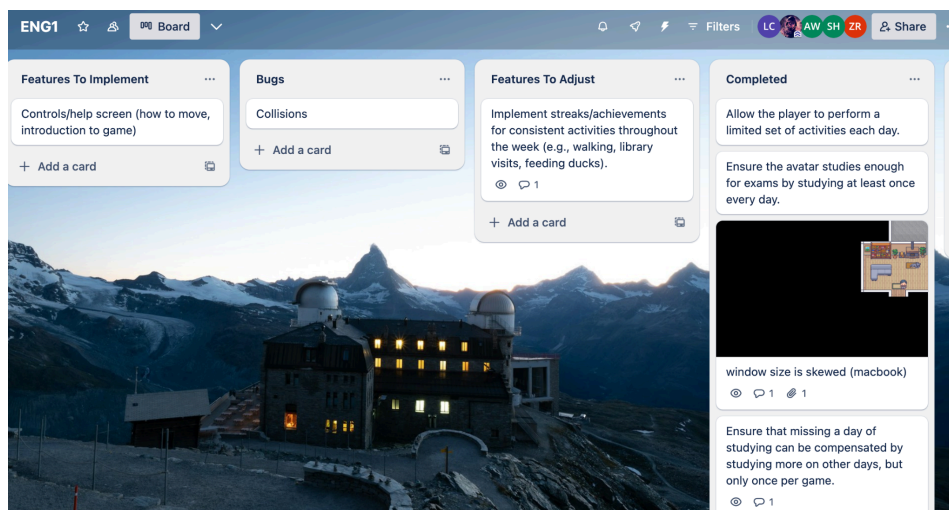
Zac Ribbins

## Continuous Integration Methods:

Our project employs robust Continuous Integration (CI) methods to ensure smooth collaboration and efficient development. Firstly, we utilise Git as our version control system, enabling seamless code management and facilitating the CI process. This allows us to track changes, collaborate effectively, and maintain a history of our codebase.

Central to our CI strategy is our GitHub repository, where all our code resides. This repository is meticulously organised, ensuring clarity and ease of navigation for all team members. We integrate project management tools such as Trello, which aids in task tracking and issue management. This integration streamlines our development workflow, allowing us to prioritise tasks and address issues promptly.

In line with CI best practices, we adhere to a frequent integration schedule. Code is integrated into the main branch after the completion of each task from our Trello board. This ensures that our main branch remains up-to-date and reflective of our current progress.



*Lists from our Trello Board*

Effective communication is essential for successful collaboration, which is why we use collaborative tools like Discord and WhatsApp. These platforms provide real-time updates and facilitate communication among team members, fostering a collaborative environment conducive to productivity.

To maintain a healthy main branch, we follow best practices such as avoiding large commits and utilising feature branches for development. These practices promote code cleanliness and minimise conflicts, ensuring smooth integration and collaboration. Throughout the project, we've logged approximately 400 commits, reflecting our team's active engagement and progress.

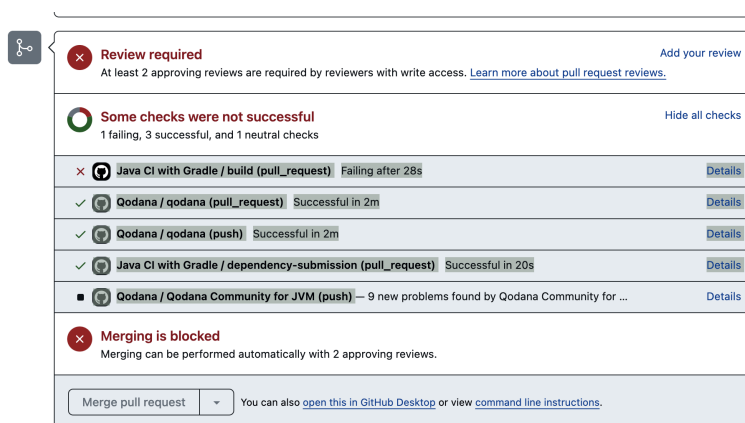
## Continuous Integration Infrastructure

Our Continuous Integration (CI) infrastructure is built on a foundation of automation and quality assurance tools, ensuring the reliability and quality of our codebase. At the heart of our CI process is GitHub Actions, which automates our build process and executes various checks on each commit.

Using GitHub Actions, we've configured workflows that automate the build process using Gradle, our chosen build automation tool. This streamlines our development workflow, allowing developers to focus on coding while ensuring consistent and reliable builds.

In addition to build automation, we employ Qodana for quality assurance. Qodana assesses our code for functionality and readability, ensuring that it meets our minimum standards and integrates smoothly into the project. Reports generated by Qodana are automatically shared with the team via Discord, enabling us to address any issues promptly and maintain code quality.

Our CI pipeline architecture consists of several stages, including build, test, and deploy. Each stage is meticulously configured to execute specific tasks, such as compiling code, running tests, and deploying to staging environments. Monitoring and notification mechanisms, such as email alerts and Discord bots, keep the team informed about build and test results in real-time.



*Automated checks when PRs are created*

In summary, our CI infrastructure is designed to automate repetitive tasks, ensure code quality, and facilitate efficient collaboration among team members. By leveraging automation and quality assurance tools, we've created a streamlined development workflow that promotes productivity and reliability throughout the project lifecycle.