

Part A

Software Engineering Methods

The chosen software engineering method for the takeover was still the same as the first team; plan-based rather than agile. We also anticipate no requirement adjustments during this assessment (other than the addition of the new requirements which we were informed of at the start of assessment 2).

We had frequent team meetings at least once a week in order to:

- Discuss and agree upon key development decisions; not only the game design but now more importantly, testing/continuous integration
- Allowing each team member to individually communicate their progress on their assigned Issues, and for new ones to be assigned when necessary

When creating the initial plan/roadmap, everyone had the opportunity to say which game feature they would prefer or feel more comfortable doing within the limits of their skills, so overall we aimed to keep the workload even between everyone, and since we have been down a team member from the start, everyone had a little bit more to do.

Tools Used

Communication and Collaboration

- Team meetings were done in our Discord server due to the ease of use since everyone was familiar with it
- This was the same server used during assessment 1, so the text channels were updated to better reflect assessment 2
 - Everyone was easily able to ask questions to each other and share any progress outside of the weekly meetings
- GitHub Issues were used to keep track of the progress of each task, which provided a useful visual representation

Website

- For assessment 1 we used Notion, a simple website builder that required no coding
- However, the team we took over from used GitHub Pages for theirs so this meant we had to spend a bit of time adapting to HTML
 - The main body of the website had already been formed, so we mainly just had to add our new documents or plans to it which meant the website didn't become a burden of time

Architecture

- The original team's architecture diagrams were done with Draw.io as well as PlantUML, and that was what we used for our own project, so the takeover of this was quite simplistic
- Due to the addition of new requirements, these had to be added to the originals
 - Where necessary, some parts may have been removed however at the start of the takeover we agreed that we weren't really anticipating having to remove much of the original classes/code

Implementation

- IntelliJ was still used as our IDE as there was no reason to switch, and this is what the original team used anyway
- The LibGDX game library was also used by the original team, which is what we used for our own game so it was a framework we were already more familiar with

Part B

Team Roles

During our initial team meeting we discussed assigning the following team roles:

- **Meeting Chair:** Ensuring organised and efficient team meetings that covered all necessary updates and decisions
- **Report Editor:** Handles document production, in particular ensuring that documentation progress was largely in line with the initial working plan
- **Repository Supervisor:** Oversees the GitHub repository to ensure things like branches and commits are being used efficiently

During this discussion we decided to combine the roles of librarian and report editor as they seemed like largely interdependent tasks. We then discussed who would be most appropriate for each role and agreed on the following assignments:

- **Meeting Chair** – Kieron
- **Report Editors** – Camran and Kieron
- **Repository Supervisor** - Will

Assigning these roles enabled a smooth and efficient team working process and helped to keep track of team progress.

Task Assignments

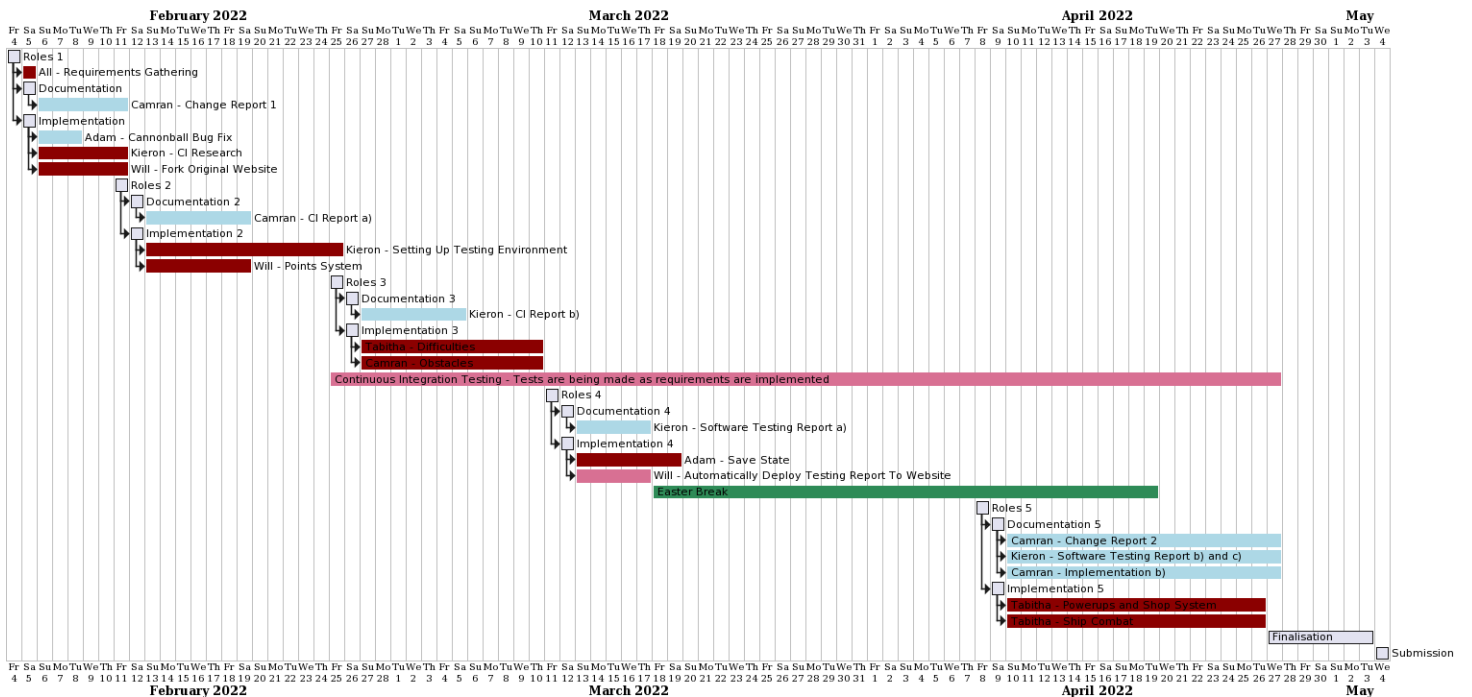
Task assignment took place throughout the assessment process as detailed in part a of this document. When the time came to assign new tasks, we tried to keep these assignments in line with each person's particular skill set in order to ensure efficiency as such throughout the project members mainly focussed on the particular aspects which they felt most comfortable with, usually due to previous experience.

Part C

Plan Snapshots

- In each team meeting, the Gantt chart would be updated to plan ahead for the next 1-2 weeks
- This helped everyone visualise the time schedules they needed to stick to in order to ensure the project was carried out smoothly

Oasis Assessment 2 Gantt Chart



Key:

- **Red:** high priority task that should be completed by the date the bar ends
- **Blue:** lower priority tasks like documentation that just need small bits added to them throughout the project