

# Plan1

## Cohort 3 Group 5

### Members:

Amity Van Rooyen  
Cassian Kanhukamwe  
Dhruv Madan  
Gilda Grimes  
Jerry Anish  
Matt Ritchie  
Oakley Fiddler  
Ruby Brown

# Method Selection and Planning

## Section 1: Software Engineering Methods and Tools

### 1.i. Implemented Development Methods & Tools

The central organisation of the project revolved around the use of GitHub, which allowed for development to be continuous and allow the primary programmers to make progress on the game. GitHub was a very efficient tool that enabled developers to track progress as well as see any changes that had been made by others. As well as GitHub, Google Drive was used to track the progress of documentation, reports, and software architecture, allowing for group members to be filled in on where the project was in terms of progress.

The group's method of collaboration depended on individuals being responsible for reasonable workloads, allowing for the correct amount of team members contributing to the project's deliverables depending on the required resources, time and workload. This meant that no member of the group was entirely responsible for critical and large scale components (i.e. implementation).

### 1.ii. Justifications

The decision to use GitHub as the primary development platform was largely due to its accessibility and integration with popular development tools. It enabled version control, which was essential for maintaining stability throughout multiple iterations of the game. Each branch represented different features or bug fixes, making it easier to merge changes efficiently without conflicts.

Google Drive was chosen for documentation because of its collaborative features, which made it easy for multiple contributors to update, review, and comment on sections. This proved particularly useful during the creation of technical documentation and reports that required input from various roles.

The selection of a collaborative approach rather than a hierarchical structure ensured that decision-making could occur more fluidly, allowing for an environment that encouraged creativity and problem-solving. This was crucial given the project's emphasis on both technical implementation and aesthetic design.

## Section 2: Team Management and Organisation

### 2.i. Team Structure and Role Assignment

The team structure was designated depending on the skills that each member was most comfortable and experienced with, to allow for the best final product.

Role	Member(s)	Responsibilities
Requirements Gathering	Gilda, Amity	Understanding and conveying the customer's requirements for how the game should be.
Organisation / Planning	Oakley	Providing organisation and structure to the project.
Risk Assessment and Mitigation	Ruby	Risk management and processes.
Software Architecture	Cassian, Matthew	Structuring the development of the software of the game.
Implementation	Amity, Oakley, Cassian, Dhruv	Development of software, use of libraries and assets.

### 2.ii. Organisation and Collaboration

The team adopted a semi-agile workflow to maintain adaptability throughout the project. Weekly meetings were scheduled but at times were not feasible for various reasons. Communication was handled primarily through group chats and discussing between team members in practical sessions.

Members were encouraged to provide feedback on each other's work through comments and reviews, which helped maintain consistency in coding standards and documentation quality.

## Section 3: Project Plan and Scheduling

### 3.i. Plan Outline

The plan for development was as follows:

- 1) Obtain requirements from the customer to understand how the game should look, behave and what the goal should be.
- 2) Establish a plan for how to develop the game, obtain relevant assets for graphics, important libraries and find a suitable game engine.
- 3) Begin implementation and design the game's architecture concurrently with the production.
- 4) Bug-testing, finishing touches and finalising.
- 5) Submit project.

### 3.ii. Schedule

The schedule for development followed the plan outline as shown above, followed week-by-week up until the submission date.

Date	Week	Description
22/09/2025	Week 1	Introduction, team building.
29/09/2025	Week 2	Requirements gathering, familiarising with tools (GitHub, game engine, etc.).
06/10/2025	Week 3	Begin development, plan and document customer's requirements.
13/10/2025	Week 4	Continue development and documentation.
20/10/2025	Week 5	Early testing, refinement of core features.
27/10/2025	Week 6	Integration of assets and additional functionality.
03/11/2025	Week 7	Debugging, optimisation, and polish.
10/11/2025	Week 8	Final testing and preparation for submission.

### 3.iii. Evolution and Adjustments

As the project progressed, several changes were made to accommodate the team's new ideas and technical challenges. Initial designs for the game's core mechanics (such as more than one game map) were simplified to ensure a stable and functional final product. Some

features, initially deemed essential, were deprioritised after time constraints and workload assessments indicated they might compromise overall quality as well as the fact that this initial stage of the game was meant to be very basic.

The schedule was also adjusted to allow for more comprehensive testing mid-development. This involved extending the testing phase by an additional week and redistributing responsibilities among the team to manage the increased workload.