

# Method Selection and Planning

Cohort 1 Group 2

Yusuf Almahmeed  
Jane Boag  
Charmaine Chamapiwa  
Charlie Coyne  
Anupam Cunden  
Bertie Kerry  
Leo Xu-Latti

This page does not count toward the total page number

## **Software Engineering Method**

Our team mainly followed an agile software engineering method, as it best suited the needs, and creative nature of our project. This method focuses on short development iterations, light but sufficient documentation, and adaptive planning methods, all which aligned well with the design process of our game. It also encouraged an inclusive relationship with the customer, allowing for continuous feedback where we could clarify requirements and adjust features in response to client comments, to meet the required deliverables.

Throughout the project, we worked in short development cycles as it helped us focus on one feature at a time. By focusing on and implementing a specific feature, such as interactive game events (U2.1, U2.2, U2.3), at a time, the team was able to identify issues early and make sure each feature was fully functional before starting on the next one and continuing development. After each iteration, we held team meetings where we discussed progress and challenges, and reviewed what worked well and what needed improvement. Since we frequently reflected and adapted, we always had strong and frequent communication about what needs to be done and who was responsible for each task. Also, maintaining some documentation supported our flexibility allowing different members to easily continue work where others left off.

Although our main software engineering approach was agile, we still had some aspects of the plan-driven method to give more of a clearer structure. Before starting development on our project, we created a simple roadmap outlining the tasks we aimed to finish each week along with each member's preferred role based on their strengths and interests. The roadmap gave us a clear sense of direction, while still being flexible enough to adjust to the project's progress and any challenges that arose along the way. The combination of both of these approaches ensured that the project remained well organised throughout whilst still being adapted in response to changes.

## **Collaboration Tools**

To complement our chosen software engineering method, we used tools that allowed us to contribute seamlessly such as WhatsApp, Google Drive, Google Docs, and GitHub.

### **WhatsApp**

Outside of in-person meetings, our main mode of communication was WhatsApp. It allowed for quick and efficient communication whenever we worked remotely. Since we all have WhatsApp on our mobile devices, it was the more convenient option as it was more accessible compared to other communication tools that required a computer. All the chats are saved in history, which also helped when looking back on previous discussions, questions and responses.

### **Google Drive**

Google Drive was used as our main storage system for all our files and documents. To help stay organized, all documents were kept in a single shared folder. This allowed easy accessibility to all the latest versions of all files, for team members without having to send each file manually. This prevented files being duplicated and made sure everyone was

working on the most recent version of each document. Google Drive's cloud based storage system meant that files could be accessed from anywhere on any device, so team members could contribute from their own devices or from the computers in the university lab seamlessly.

### **Google Docs**

Google Docs was our main tool for collaborative writing and documentation. Multiple team members could edit, and comment simultaneously, which made co-writing, collaboration much easier and efficient. Additionally, Google Docs saves automatically, preventing file loss and version conflicts. It also has version history, which allows the team to look back over previous drafts or contribution history.

### **GitHub**

We used GitHub as our main platform for all our code and documentation. It allowed every team member to work on their own separate aspects of the code and then share it to the main repository. Here other members could view, edit and refine each other's work. This ensured a structured approach to the game code, and encouraged collaboration. GitHub also made it easy for all the team members to access the most recent version of both the game and website.

### **Fitness of Tools and Alternatives**

All of the selected collaboration tools above complemented our software engineering approach. Together, they supported seamless collaboration, efficient communication, and easy access to shared resources. Google Drive and GitHub maintained up-to-date versions of documents and code, while WhatsApp and Google Docs promoted frequent interaction and automatically saved all previous versions and messages. Overall, the combination of these tools proved especially valuable when our plans adapted or required adjustments.

Initially, during the planning stage, the team considered alternative options, such as using Discord for communication. However, some members were not familiar with the platform, so we decided to use WhatsApp instead. For our documentation, we agreed that Google Docs was a better alternative to Microsoft Word, as most members were already comfortable with. Google Docs also has better collaboration features and it integrates easily with Google Drive. Finally, Bitbucket was considered as an alternative to GitHub, since it also provides similar collaboration features, but GitHub was preferred and chosen for its simplicity and because several members already had prior experience using it.

## Organisation Approach

Our team had weekly in person meetings where we discussed progress made towards the project and planned what needed to be done next. During these meetings everyone had the chance to share ideas and work out how to move forward. To maintain communication between these meetings, we used a WhatsApp group chat to share progress updates and discuss any issues or challenges that arose. This allowed these issues to be solved quickly, and everyone was able to stay up to date with the project, even if they couldn't attend a meeting.

After reading the project brief, we had many unanswered questions that needed clarification. We compiled these questions into 3 main topics which we then asked at our client meeting to help us gain a better understanding of the requirements and expectations. Following this client meeting, we discussed as a group what needed to be done, creating a loose to-do list, allowing each member to choose their tasks based on their strengths or interests. This worked for the team and the project as it ensured tasks were allocated fairly, giving everyone an equal amount of work to complete.

Collaboration was a main part of our organisation approach. So that nobody had too much work, or were made to do tasks they weren't comfortable doing, our group had a list of odd jobs which could be picked up by anyone who finished their work early. If team members encountered difficulties or a task was taking particularly long, other members stepped in to share the workload.

This approach easily supported collaboration, as it encouraged communication, teamwork and adaptability. Regular in person meetings and maintaining contact through WhatsApp helped every team member to stay informed and involved with the game's progress. It also allowed ideas as well as challenges to be discussed and solved simply and collectively. This structure strengthened the group dynamic and helped maintain steady progress through the project.

## Plan

Key Task	Start Date	Finish Date	Priority	Dependencies	Explanation
Finalise design decisions.	14/10/25	21/10/25	High	None	Confirm the overall concept of the game, events and the theme based on the client interview.
Implement movement and basic character model.	14/10/25	21/10/25	High	Design Decisions	Create the player's basic movement and on screen representation for testing in game.
Implement a camera that follows the player around the map.	14/10/25	21/10/25	High	Movement	This makes sure that as the player moves through the game, the camera automatically follows their position, keeping the character visible on screen.
Implement a timer, with the game ending after 5 minutes. (U6.2)	14/10/25	21/10/25	Medium	Movement	Add a timer in the game which ends the game after 5 minutes.
Implement a pause, and unpause function with basic UI. (U3.1)	14/10/25	21/10/25	Medium	Timer	Allows users to pause and resume gameplay, which freezes all movement and the timer during this time.
Change architecture to MVC from OOP.	21/10/25	26/10/25	High	Initial implementation needs to be complete.	Restructure the code for easier stability and maintenance.
Finalise and implement the map design, with hitboxes. (U1, S1)	21/10/25	26/10/25	High	Movement	Design and implement the maze layout.

Implement art assets. (U5, U6)	21/10/25	26/10/25	Medium	Map Design	Add the visual elements to the map e.g. people.
Catch-up and refinement tasks.	27/10/25	01/11/25	Medium	All of the above	Consolidation week allowed for the team to catch up on tasks that may have taken longer than expected to complete.
Implement score feature. (U4, U4.1, U4.2, S4)	01/11/25	09/11/25	High	Timer Events	Implement scoring based on event interactions and time taken to complete the game.
Implement code for power ups and obstacles. (U2, U2.1, U2.2, U2.3, U4.2, S4)	01/11/25	09/11/25	High	Map Events	Add events to hinder and benefit the user as they progress through the game.
Implement sound. (U9)	01/11/25	09/11/25	Low	Main features complete.	Add background music to make the game more fun to play.
Bug fix and polish.	01/11/25	09/11/25	High	All previous tasks.	Test and refine the code for gameplay and make sure all documents are complete.

The project plan evolved throughout time spent working on it as we adapted to problems and challenges. Initially, the majority of the team worked on programming and developing the game as setting up the main functionality of the game (maze design, movement and game events) was the most demanding.

As the project progressed and more of the game implementation was completed, focus shifted to documentation and reporting which made up a significant portion of the project. Tasks were reallocated to ensure all areas of the project were covered, with work distributed evenly. This adaptive approach helped with team organisation and the nature of the project as it made sure every member contributed equally throughout and encouraged collaboration.