

Method Selection and **Planning**

Group 8: GeNext

Aditya Mydeo
Alex Miles-Thom
Christopher Omoraka
Faisal Nabi
Jesly Prosper
Stefan Maxwell
Zak McGuire

Methodologies

During the planning stage, we conducted research into both agile and plan-driven methodologies. The plan driven method follows a structure where each phase of the development is completed before we move onto the next. This approach is sequentially structured and can be beneficial for projects with defined requirements and large teams (over 100 members) and requires careful planning and documentation at every stage.

However, after assessing our project needs we understood that this method was not suitable. Our requirements were likely to change over the course of this project as well as we anticipated on receiving feedback from our stakeholders that would shift our priorities. Additionally, the team required flexibility and frequent communication to which the plan-driven method was not satisfactory. This led us to researching a more appropriate method such as agile.

In our team project we decided to use agile methods, specifically the Scrum framework. This worked well with our goals for flexibility and team communications. The scrum methodology allowed us to structure our work in sprints, where each sprint would last 5 days. Our weekly routine was setting sprint goals during Monday's meeting where each of us work on our assigned deliverables and tasks. On Fridays, we would meet to review our progress by identifying any challenges we had and change our approach to handling tasks. This was preferable as it allowed consistent communication for our group, whilst also providing opportunities to reflect and adapt our plan. Additionally, if a meeting was missed due to any planned absence or unexpected circumstances, this structure allows flexibility in catching up and adjusting tasks. This ultimately enhances accountability and ensures all team members are on the same page.

Some deliverables, such as architecture and implementation were completed in groups of 2-3. Using mini-groups made the increased number of meetings more useful as it allowed coordination within mini-groups, preventing members from repeating work and emphasising teamwork.

Development tools

Our team used several tools to support our development process and allow us to collaborate effectively with each other:

Jira: Jira was used to track the progress of the deliverables and who was responsible for what parts of each deliverable. It allowed us to organise sprints, assign tasks and monitor each member's progress which was essential in a Scrum based approach.

Whatsapp: We utilised Whatsapp for communication between the group for organisation of meetings, coordinate absences, and for the mini-group coordination. This served as our main communication tool and allowed us to maintain communication outside of meetings.

Github/Git: Github was used as our version control system and was used for the website and implementation. This was essential for coding within the team and ensured that all members had access to the latest version of the project. Github allowed us to review each other's codes and make

any changes we saw fit. Additionally, we could manage branches to reduce the amount of conflicts that could occur.

IntelliJ and Visual Studio Code: These integrated development environments (IDE) were used based on each team member's preference. They both supported collaborative coding and version control integration with Github. This allowed members to write and debug code.

Alternative Tools

We discussed using several other tools but decided against them for numerous reasons:

Discord: Discord is another communication tool but we decided against Discord as only some of us were familiar with it so that would result in time wasted with creating accounts and familiarising of the software ultimately minimising its utility.

Eclipse: Eclipse was considered for implementation however members of the team decided on VScode and IntelliJ as more people were familiar with this software and learning how to use a new IDE would have been inefficient.

ChatGPT: We originally planned on utilising ChatGPT for graphical and audio assets however alternatives were found that were more useful so we decided against this.

Roles and Responsibilities

For the general purpose of the project we had 3 main roles that were given out to members so that we could all know our responsibilities. These roles were: Scrum Master, Product Owner and Development Team. Whilst all members were a part of the development team, the scrum master was assigned to Alex. As Alex was in charge of the method and planning sections we found it suitable for him to also be responsible for coordinating meetings, managing sprints and ensuring that we follow the scrum framework. Faisal was assigned to be the product owner who acted as the main point of contact with our stakeholder (Tommy Yaun) and prioritised tasks to meeting customer requirements. He was chosen to do this as he was initially on the requirements section along with Stefan.

All tasks were assigned based on each member's strengths whilst also accounting for people's choice in what they want to do. We all decided to partake in eliciting requirements from our stakeholder in the initial interview so that we could all ask questions that helped our relative area we were working on. From there, we individually or in smaller groups worked on our own sections. Working in smaller groups, for example Implementation was a group of Christopher and Zak allowed them to complete deliverables that required more time and focus.

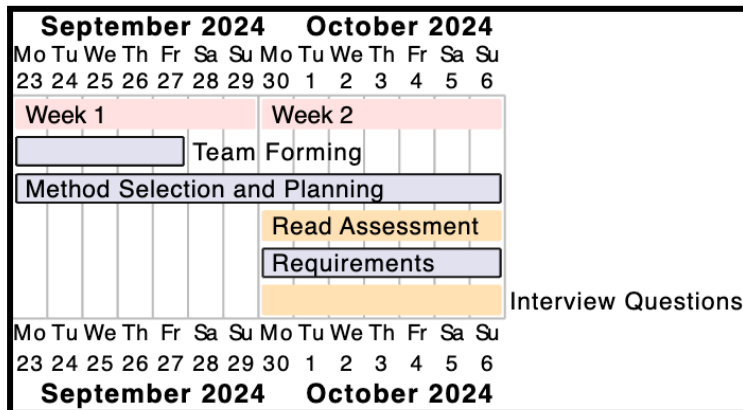
We decided to have no clear leader throughout the project as we decided that we all work better without the potential occurrence of micromanagement and hyper controlling behaviour as this would be unproductive for us. As well as this, 2 weekly meetings on a Monday and Friday were

implemented to allow everyone to know how much progress / difficulties anyone was having to limit the risk of under performing or not participating.

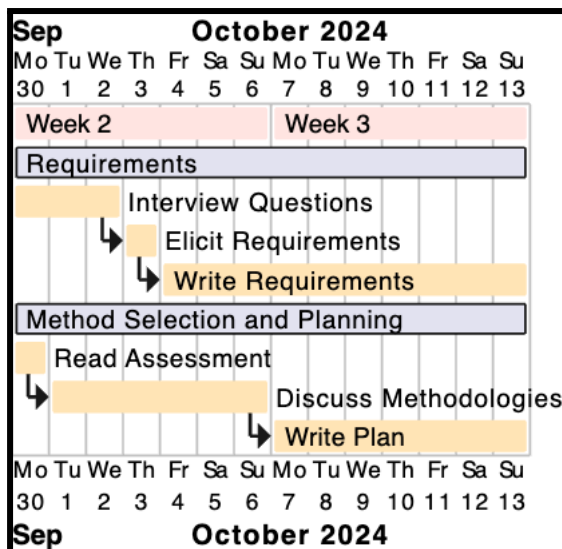
This organisation approach was well- suited to our team. By combining clear roles with collaborative task structure, we were able to remain in an adaptable environment so that all of our user requirements could be met. Using the Scrum framework meant we had flexibility in allocation and we often found that tasks for particular sections were often assigned to different members each week depending on their availability and which tasks were more of a priority. Additionally, on the Friday meeting we could use the session to reflect and provide feedback to each other in order to successfully complete the tasks the following sprint.

Weekly Snapshots of Plan

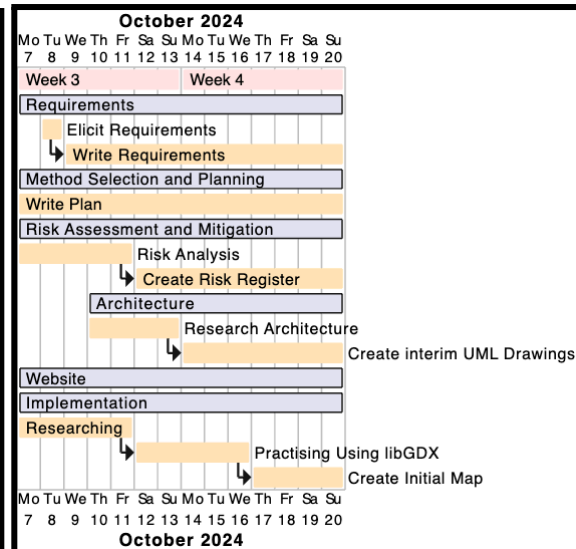
Week 1



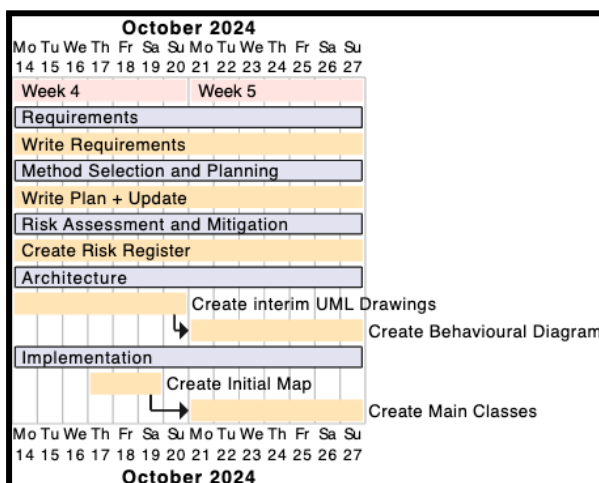
Week 2



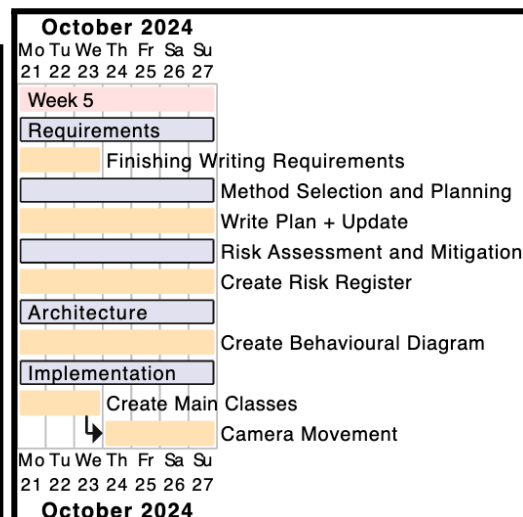
Week 3



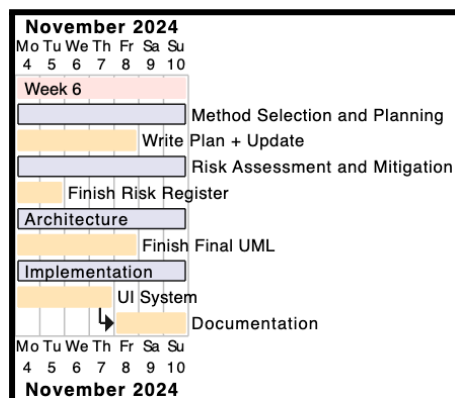
Week 4



Week 5



Week 6



Initially, the plan consisted in week 1 of all of us working on the team forming activities, creating our team logo and coming up with the name GeNext. We discussed briefly about what possible ideas we would use to complete our product brief however nothing was set in stone as the assessment document was not released.

In week 2, we read the assessment as a group to decide what tasks people would be best at doing as well as what people most preferred to do. We also worked on interview questions for our stakeholder to elicit user requirements from them. Our intention was to do this on week 2 and start the write up for requirements. Furthermore, we discussed possible methodologies to use, concluding Agile- Scrum was the best.

During week 3, we conducted interviews as we were unable to secure a meeting in time for week 2. The write up for requirements began towards the end of the week and a plan for each week was devised based on the tasks we needed to complete. Additionally, we started to analyse any potential risks that could occur. Research was also conducted to discuss how the architecture would be handled and an interim version of UML designs was started. On the implementation side, the website was made using HTML and CSS and research into what game engine and libraries was conducted. LibGDX was confirmed to be our game engine and the implementation team started to practise using this.

In week 4, the architecture team finished creating the interim, UML designs and began working on creating behavioural diagrams. For requirements, they were still getting finished, often getting updated to fit the user's needs. Additionally, throughout week 4 a risk register was made to combat any potential problems that may occur. The implementation team began to create the initial map mid-week and on Friday the discussion of what main classes should be included began.

During week 5, the write up for requirements was completed and the majority of the risks within the project were identified. The architecture team finished behavioural diagrams allowing the implementation team to carry on with their progress. The implementation team created the classes used in the game and began working on camera movement.

In week 6, the risk register was completed and so was the final UML diagram. The final plan write up was finished on Friday as well. On the implementation side, the UI system was designed and made. Finally, the documentation for the code was completed and all formatting of PDFs was done.