4a) Software Engineering methods and tools

The team decided from the start that the best method to use in developing the project would be any that follows an agile project management framework, and thus we settled on SCRUM. SCRUM is a method that focuses on continuous improvement with the acknowledgement that the team doesn't know everything about the project (in this case, the game) at the start. This especially matches the team's experience as it also is every member's first software engineering group project.

SCRUM was especially chosen as we had continuous weekly opportunities to get feedback on our work through meetings and asking questions at the designated practical slots with our customer. This allowed us to make the requirements clear at the start, and in case there was a change in the requirements, to adapt as best as possible. SCRUM also promotes interaction between team members (following the Agile manifesto), resulting in some great ideas generated during analysing/brainstorming processes of parts such as design of the game and the implementation of said design.

The plan-driven approach was avoided as the group felt like the lack of weekly communication/feedback from the customer could lead to some documents not meeting standards or missing some requirements, as the plan-driven project management framework relies solely on a final hand-in. Additionally, the inability to alter the product after getting feedback could've been really impactful on the project as the customer could have requested a change in some of the requirements initially presented to the team.

The main tools we used throughout this project were Git(Hub), Discord, Gannt charts and LibGDX:

1. The team heavily relied on Discord for communication, as well as for quick exchange of files members were tasked with working on. Regular meetings were held on Discord when meeting in person wasn't an option (e.g Christmas and the holidays where the team members were all away). Discord provided the perfect environment for us to communicate ideas with options that allowed for separate channels for each part of the project, the ability to upload/download media, etc. The reason Discord was chosen over other communication tools such as Slack was because all of the team members were not already familiar with Discord but most already had it and used it regularly.
2. Github was used for file distribution (files such as jsons, bitmaps, assets and pages of code) across the team allowing the members to upload/download and edit files as the project is being developed. Git is a well-tested tool used for programming projects and thus we selected it as it provides everything we need and it is well documented and surrounded with vast amounts of information and support online compared to newer/less popular tools for the same purpose.
3. A regularly updated Gannt chart was used for team management. Every week after a meeting we had a select team member update it, filling in what task/what percentage of a task has been completed, as well as assigning different members to different tasks/deliverables. Similar tools such as Trello and Asana were considered, but were later dropped as the Gannt charts could be implemented into the word processing software we were using

making it a lot easier/accessible than accessing everything on a separate website.

4. Finally, the game engine the team settled on for the implementation of the code is LibGDX. Although many other methods exist for making 2-D games with Java, LibGDX was selected primarily because of the vast amount of documentation on the library online. Additionally, it is very easy to set up and use as it can be implemented with almost any IDE which has access to java e.g VS Code, IntelliJ IDEA, etc.

4b) Team organisation

The team followed a decentralised structure, not having a single person as a leader. Tasks were agreed on and handed out to volunteering members/members who thought they were best suited for the task during each meeting session. For every task, a shadow member was assigned to make sure the work was completed even if the originally assigned member could't do it on time. This was also implemented in the creation of four major roles which were utilised to complete this project up to standards.

Since the first day, the team assigned the 4 major roles out to people with an aforementioned shadow for each. The role being the "meeting chair" which was assigned to Samuel and tracked by Alana. The Meeting chair led the meetings in the correct direction, staying on topic as well as strictly overseeing progress to make sure progress is made.

Alana personally held the "secretary position" throughout the project with Jack as their shadow. The secretary took care of noting down important decisions made during meetings as well as keeping track of what has been done/what there is to do every week.

The "librarian" role was delegated to Faran and further assisted by Galin. Faran was tasked with keeping track of and backing up documents and files created throughout the course of the project as well as organising resources and any links made to work referenced in the creation of the game.

Finally Aisyah took care of the "report editor" assignments, which was also assisted by Galin. The report editor saw that all the production of deliverables was up to date and up to standard assuring the best quality work, on time.

With all these roles, the team made sure to have 2 meetings a week, one of which involved the customer, Tommy Yuan, keeping him updated on work completed as well as discussing the details of the direction their product was taking. When face-to-face meetings/discussions with the customer weren't possible, the team kept up the communication using emails, and when the members couldn't gather face-to-face for meetings, Discord was used, allowing the members to connect and work on the project from their own homes during the holiday periods.

In conclusion, the implementation of a decentralised structure, with 4 specialist roles and the shadows system worked perfectly by allocating equal amounts of work per person as well as ensuring the game progressed precisely as needed every week at a steady pace and ensuring the customer was kept happy and involved in the process through the use of agile methods.

4C) Plan

To assist with our project we have diverged the project up into some key tasks. These tasks include, reviewing, requirements elicitation, risk evaluation, architecture establishment, implementation and website development.

The first task we need to complete before we can proceed with our project further is a review. A review includes investigating previous systems similar to the one we are aiming to create, meeting with our customer to establish all user goals that they have from the project and reviewing potential software/libraries available that might be useful to help us with our project. As this is the only task we can undertake at the beginning of our project this is the task with the highest priority and must be completed first as the other tasks are dependent on the completion of this in order for them to commence. This task is commencing right at the very start of our project (10/11/2022) and we aim to have reviewed all previous systems and met with our customer within 2 weeks so that we can start eliciting requirements, as this task is dependent on the outcome of the customer meeting. We are aiming to have this task completed by 23/11/2022.

Once we have this knowledge as well as a set of user goals from our customer we can begin to develop our system requirements that we will aim to fulfil with our end product. Most other key tasks such as architecture establishment, Implementation and testing rely on the system requirements being established before they can begin hence developing these requirements is of a high priority. Therefore, we will aim to have this completed within 4 days after our customer meetings, starting on 24/11/2022 and completed by 27/11/2022.

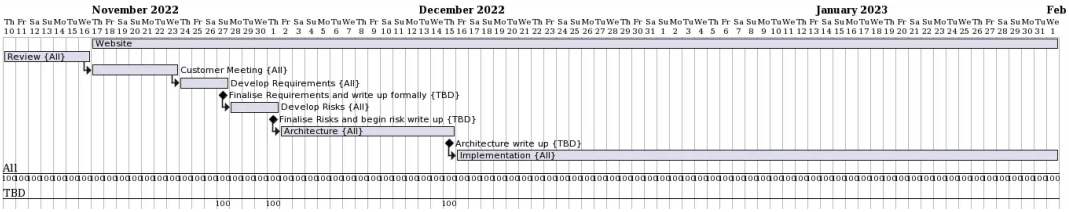
However, a task that does not depend on any other task is the development of the website. Therefore, this can be started before the customer meeting and worked alongside all of the other tasks. As the website can be developed concurrently throughout the whole project this task is not of the utmost priority as no other task is dependent on the completion of this. Hence this can be started on 17/11/2022 and as long as it is completed before 01/02/2023.

Once the requirements have been established, the risks of the project can be assessed. This task cannot be completed until the requirements have been established to ensure that all possible risks are assessed. This task is vital to the success of our project and therefore holds a high priority to be completed. This will begin when all requirements have been established, beginning on 28/11/12 and completed by 1/12/2022.

The task of designing the architecture of the system is dependent on the completion of the risk analysis and requirements elicitation. Therefore, once these two tasks have been completed the design of our game can be decided upon and UML diagrams can be drawn up. This task is off the highest priority as without the correct design of the program the implementation of the game will not be possible hence this is a task that has to be completed before implementation can begin. The architecture design can start to be undertaken when the risk analysis is complete and therefore commence on 2/12/2022 and end on 15/12/2022.

Finally the last key task of the project is the implementation of the actual game. This task is dependent on all of the other tasks (except the website) being complete before this can begin to proceed. However, as this is the actual system that will be given to our customer this is arguably the task that carries the highest priority. However, due to all of the previous tasks that need to be completed in order for this to begin, the implementation will not be able to begin until 16/12/2022 and it will be completed by 01/02/2023.

Therefore, our original plan with all key tasks, starting and finishing dates as well as the dependencies can be shown here:



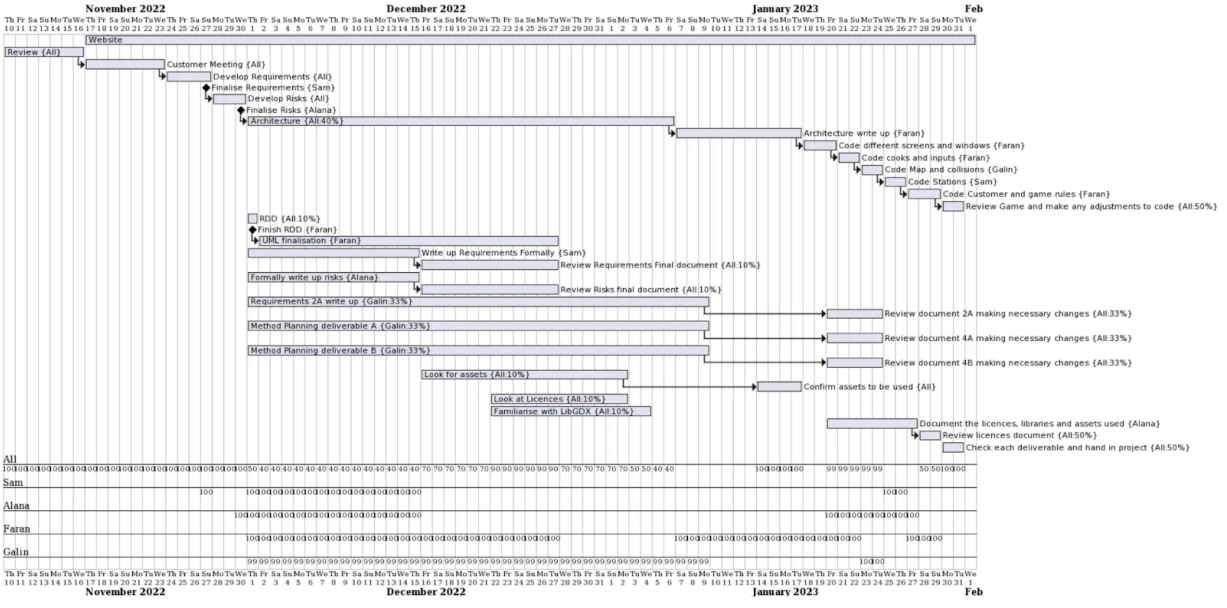
However, as we started to work through our plan we realised some tasks needed to be further divided as well as the expected dates changed throughout the whole project.

We soon realised that designing the architecture of our system was going to take longer than originally planned. This was further delayed by the Christmas period which meant we could not work as efficiently as a team without in person meetings to discuss our different approaches. We also realised that the design of our architecture needed dividing into RDD, UML etc hence we had to change our plan to follow these steps in order to design the overall architecture of the game before we could implement. Overall the initial architecture of our system took the whole of the Christmas break to complete and was only completed on 6/1/2023 as opposed to 15/12/2022 as originally planned.

One thing we also did not consider in our original plan was when we would individually work on the deliverables of the project, this is soon something that was implemented into our plan with individuals given 2 week periods to complete these documents allowing time for review and change as we are implementing an agile method of development.

Once our initial architecture was complete we had a good idea of how our game needed to be implemented. This meant that the task of implementing could be further split up into smaller subtasks that together would make up the whole system. These subtasks naturally had dependencies, meaning that one part of the game could not be implemented without the previous task being complete etc. These subtasks, dependencies and dates can be seen on our final plan. Furthermore, throughout the implementation of our game, we further updated our architecture based on what we were discovering while coding.

Our final plan of the project can be seen here, with the final starting and finishing dates of each of the key tasks as well as any tasks that were broken down into subtasks.



The full project plan can be seen here showing the original plan and how the plan evolved over the entire project changing weekly. https://eng-25.github.io/#gantt