Method Selection and Planning

Team 16

CatepillaDevelopment

Yousif Al-Rufaye
Joe Fuller
William Gracie-Langrick
Jack Hardy
Bailey Uniacke
Ben Young

a)

As a group, we collectively decided we would be using an Agile method to implement the game. In particular, we have used the scrum framework, in order to deliver code and get tasks done in short sprints. Either everyday or every two days, the implementation team will meet to discuss what tasks have been completed and what to work on next. They would also discuss any risks that they may have encountered or identified, hence one of them doing a risk review.

We chose an agile methodology because it helps us to work more efficiently and effectively. Delivering code in small intervals and sub tasks between the team, instead of being in one, large task. We also chose this approach because it fits in nicely with our UML modelling and our entity component diagram.

We collaborated primarily on Discord, scrum meetings, documentation meetings and meetings in general took place here. This took place via voice chat and text chat. Voice chat being used to discuss larger tasks at hand which may require many, if not all members of the team, as it is much easier to communicate through voice chat and will avoid any text ambiguities. Text chat was used to discuss smaller tasks at hand, for example, when to organise voice chat meetings between team members and generally asking for help. The text chat is useful as small points can be brought across to the team in a swift manner, instead of having to organise a meeting and taking time. Discord in particular is useful to us as it allows us to be in line with our agile methodology. Helping us to organise daily scrums on the voice chat.

We have also used GitHub for our collaboration needs. GitHub being a website that runs git behind the scenes. Git is a tool that is used to manage version control for various files. This is extremely useful to us as a collaboration tool as it allows us to host our local git repository, allowing the implementation team to work collaboratively on various parts of the project.

b)

Given the agile method that the team decided upon, our team had a very unstructured approach to team organisation. Tasks were assigned as required, and we have relied on responsibility and trust to ensure that we had shared the workload fairly and evenly, with people speaking up if they were struggling or felt they were doing too little.

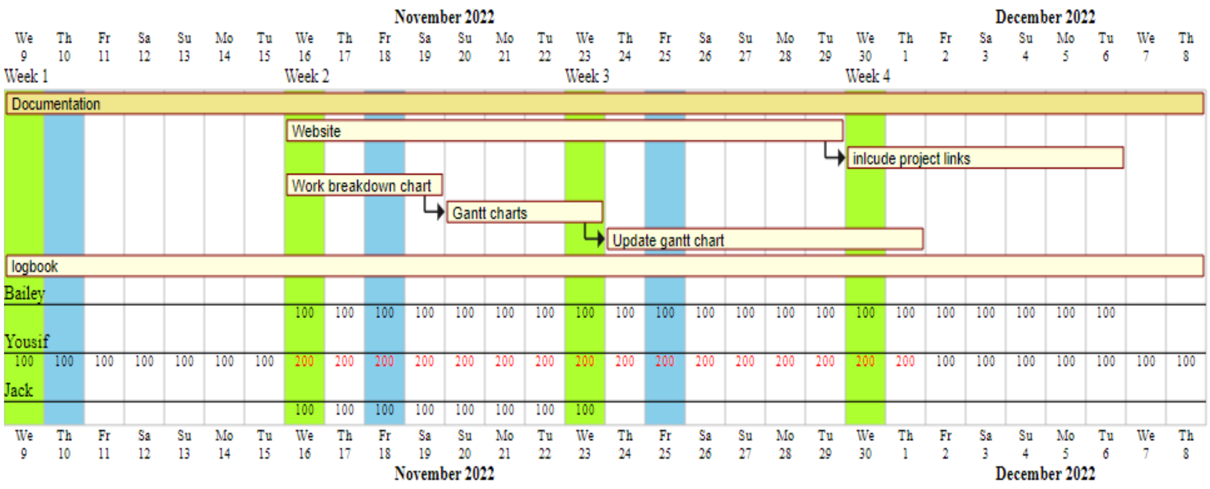
Where members of the team had particular strengths; we, of course, assigned them to roles they believed they would be strong in. However, this was our first time completing a large project like this and all members have similar past experience. So, often, no member was sure what a task would entail and whether they would be strong at it. This was the cause of uncertainty and it gave planning the feeling of opaqueness where organisation felt like speculation rather than decision making.

In the end the team split into two groups: one on implementation and one on documentation of requirements, architecture etc. This was due to some members of the team making a headway in the programming and having a better understanding of LibGDX, they could continue programming and the remaining 3 could work on the documentation. Due to this split, the interconnectedness of the team needed to be strengthened. As the documentation team would need a deep understanding of what the programming team were up to. We were able to make this structure successful.

c)

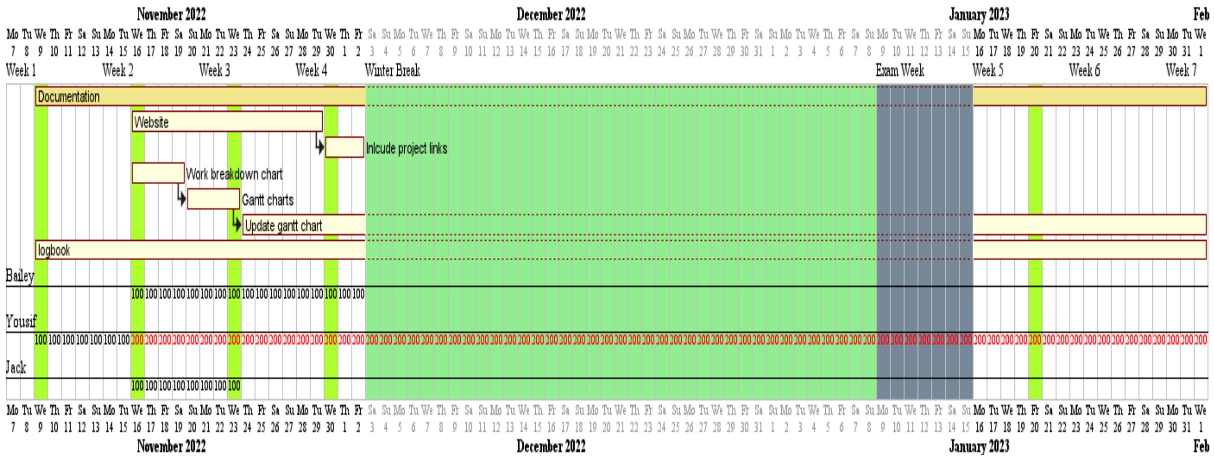
The systematic plan we created for this project is displayed through the Gantt charts and work breakdown diagrams on our website. We started creating our initial plans during the second week of the module and created 4 different versions showcasing the changes in our plans. For each version, we created a total of 5 Gantt charts for our 5 main tasks in the project, being: Design, Documentation, Implementation, Requirements and Testing. Initially, the plans we had created were very basic, not very thorough and were pretty naive as we were all unfamiliar with the software engineering process and unsure of the work/tasks we will be doing in the future. The Gantt charts had some of the days highlighted in green, red and blue with the green corresponding to our in-person meetings/practical session, red being our future practical sessions and blue being our meetings outside of the practical session. Although this was not enforced very well in the initial Gantt charts, it was done for us to keep track of meetings and how far along the project we were.

The second version of our Gantt charts was made in week 3 of the project. One of the major changes was that team members have now been allocated to tasks. This was done according to each team member's previous work in the past weeks and with how confident/comfortable each team member is with undertaking a specific task. For the tasks that were later down the project, every team member was assigned to each of the tasks temporarily. There were some minor changes to some of the tasks, such as increasing/decreasing the length of a task by a few days or by shifting the start time of some tasks. This was done due to the time allocation of tasks from version one of our Gantt charts being inaccurate regarding the actual time needed for each task.

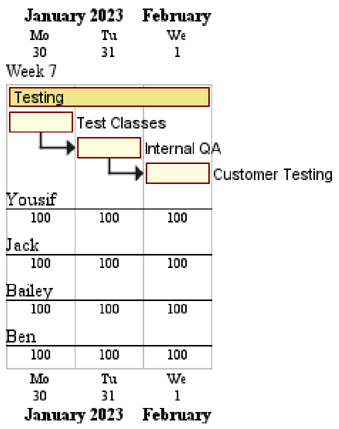


Version 2 of the Documentation Gantt chart

In version 3 of the Gantt charts, a lot of major changes were made. This was due to us agreeing that work will continue over the winter break, but during the holidays we met up and discussed what and how we should work and all agreed that it is better for us to start working after the holidays and exam week are over. Due to this, our plans remained the same throughout the winter break and exam week and were next updated when we met back up in week 2 of spring term. The Gantt charts were updated to showcase this down period in our plan. The weeks in our Gantt charts were changed to start on Mondays instead of Wednesdays, this change helped with the upkeep of our plans and lowereing confusion. We also decided to split up into two teams, 4 of us work on the implementation of the game, and 2 work on the documentation of the deliverables. Testing has been reduced to only 3 days due to time constraints. The blue and red highlighted days have been removed and only the green highlighted days remain, this was to reduce clutter on the charts.



Version 3 of the Documentation chart showcasing the Winter break and Exam week gap



Version 3 of the Testing chart

The fourth and final version of our Gantt charts showcases our decision to completely remove the testing tasks and use that time to slightly increase our time for Implementation. This was done due to time constraints during the implementation, and our team agreed on increasing the implementation time to get as many features of the game implemented as possible.

For our planning, work breakdown diagrams have also been used and are on the team's website. The work breakdowns, similar to the Gantt charts, had four versions and were used as a basis for our initial Gantt charts. The tasks outlined in the work breakdown diagrams follow similar change to the tasks on our Gantt charts, such as the Testing being completely emitted from the fourth and final version.