Method Selection and Planning 2

Cohort 2 Team 5

Harry Beaumont-Smith
Tom Nolan
Will Punt
Ruth Russell
Mimi Shorthouse
Lottie Silverton
Stanley Thompson

Method Selection and Planning

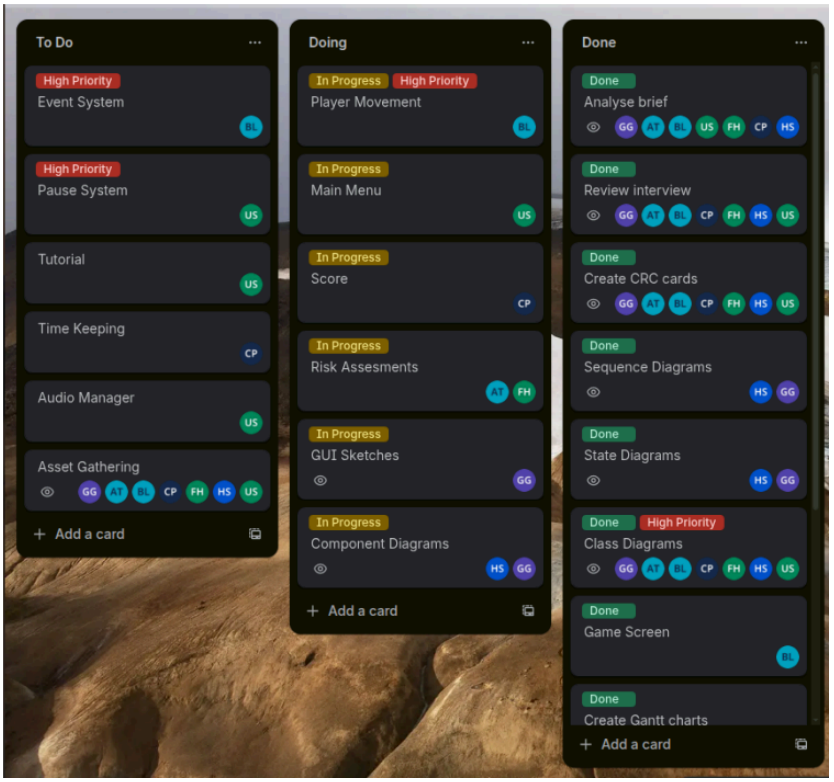
Note: development of the game was done by two separate teams over the course of the project. This document has been written to include the different methods used by both teams, and to account for the changes made to the project.

Method Selection

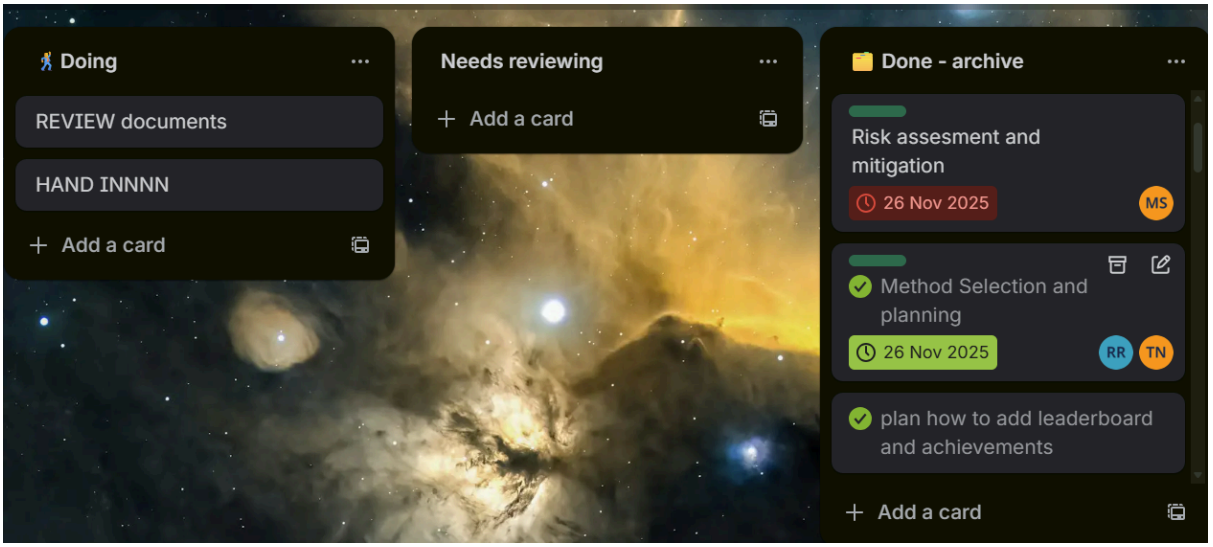
The original team decided to use a Kanban methodology due to its flexibility, iterative nature and ease of collaboration. As an agile method, Kanban supports incremental development and ease of dynamically delegating tasks. This is especially important as members of the teams have different commitments.

After taking over the project, as a team, we decided to keep the core fundamentals of the Kanban organisation method. However, we tweaked our methodology to adhere to the ENG1 guidance, which emphasises that plans may be provisional and require continuous planning. In the original project, their planning was documented as being quite fixed, despite their agile approach. The updated approach now accounts for the tentative nature of early task estimates. Which need continuous revision as task durations, workload and team availability become clearer over the course of the project.

In order to maintain this continuous replanning, we used Trello. Trello allows us to decompose large tasks into smaller manageable workloads represented by each card. Cards could then be sorted into workflow columns depending on the completion of the task (to do, in progress, needs reviewing, completed). Allowing the team to meet deadlines for the deliverables while gaining a visual indication of which stage of the project we might be at.



The old trello created in the first deliverable



A screenshot from our trello which we continued working on.

Resources

Development of the software was done using the LibGDX framework, which is an open-source Java toolkit for building cross-platform games from a single codebase. We used it because it provides unified APIs from graphics, input, audio and asset handling. Its extensive supporting documentation and examples made it an appropriate choice, allowing us to learn it effectively and rapidly.

Version control of the software was managed with Git and was hosted on GitHub. Git provided a reliable history of changes, branching for parallel work when it was needed and safe merging of contributions. GitHub added a shared remote repository with pull requests and code reviews. This was effective for the group, and its web interface made it easy for the members who have not had experience with GitHub before.

For written documentation we used google docs to write all our supporting PDFs. It was easy for all team members to use and made it easy to implement any tables or diagrams into our documents.

We also used PlantUML for graphs and diagrams. This allowed us to plan and develop our understanding of the architecture of the game, and was useful in both the planning and development phases of the project.

After initially receiving the newly altered brief and the previous teams' work we were able to add to these diagrams. This led to several iterations of the diagrams. These were constantly edited throughout the process of coding the game, when we felt that there was something else that needed to be altered or added.

Using these resources enabled us to create the game according to the customer's constraints on the hardware and software of the game. Using LibGDX enabled us to create a game according to the customer's original hardware specifications. This was because the software is designed around helping us create a game compatible with specific hardware. We also ensured to create the game using the correct version of Java - this further met the

customer's requirements. Using LibGDX also gave us a platform to make a game that was within our coding scope and capabilities. This is due to the fact that the library is relatively easy to use and similar to aspects of things we have learnt before. This meant that it was also compatible with our time restraints, which meant that we didn't waste too much time learning new code/libraries.

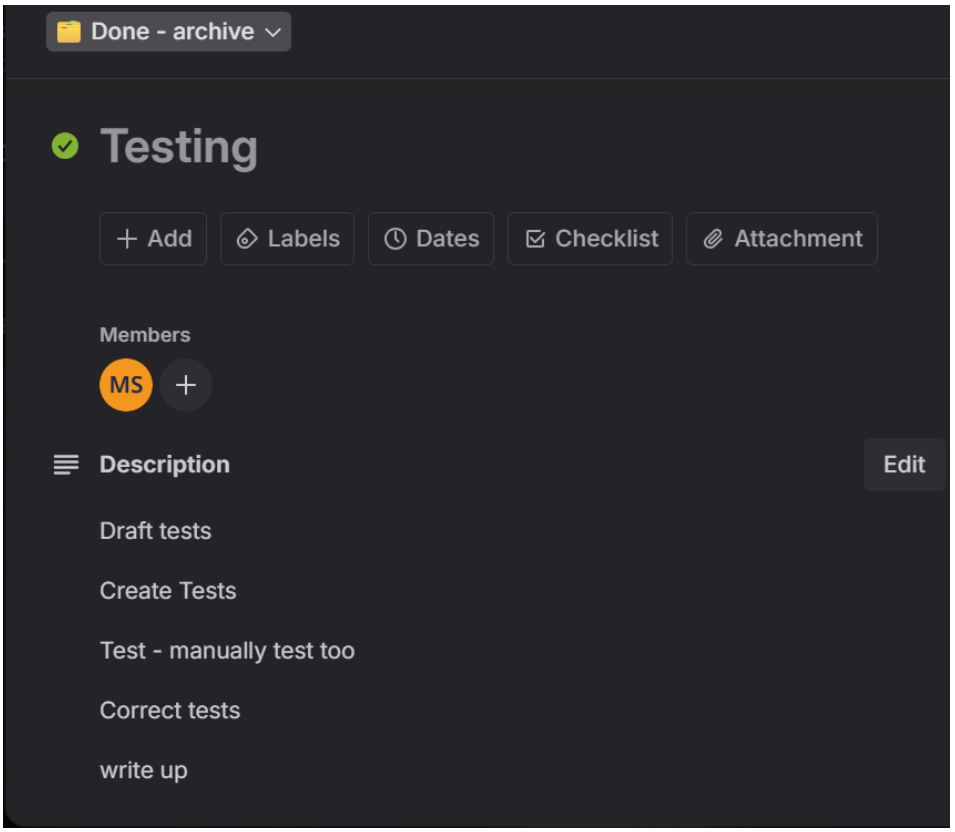
To conform to legal constraints for each resource we ensured that we acquired the legal rights to use the resource, or confirmed that it was open-source. So when using assets that we didn't create - which we tried to do as much as possible - that we had the legal licensing right to use those assets within our own game.

Project Organisation

Due to the mostly remote working environment the team had decided on using an Instagram group chat to maintain a point of contact between all members of the team so any questions or concerns could be addressed. All members were already familiar with Instagram so using it was easy and accessible. Although most work was done independently, when collaboration was required outside of our usual meeting times we did so over a voice call.

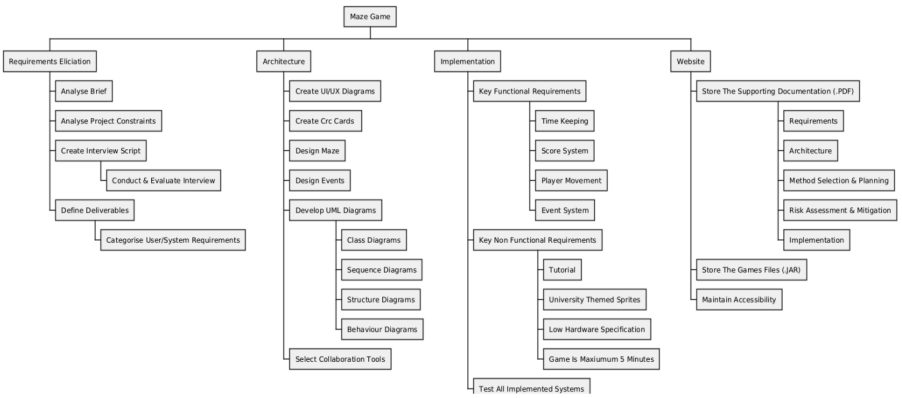
The group has a scheduled weekly meeting every Wednesday, these meetings were used to discuss the work allocation between members of the team and complete tasks assigned to them. Most of the plans had to be adjusted to meet deadlines as the amount of time the group would have to complete each package was more significantly hindered than anticipated by other university work. This meant the team had to use an iterative approach to our tasks. Additionally once development had begun group members would have to do risk monitoring at the end of their meetings. This was effective as all members were present in these meetings so the discussions could cover the status of all risks identified. Requirements for the upcoming week were elicited in meetings, and then tasks were broken down and assigned to team members based on personal preference. Tasks were usually assigned to individuals or small groups for efficiency.

Work packages were put as a ticket on a shared Kanban board. Each ticket was labelled with its priority and which team member has been assigned to the task. A larger Gantt chart was made which details the more holistic view of the work that needed to be done in the project that was used to estimate when tasks needed to be started by and when they should be finished along with detailing the dependencies of some of the systems needed to be developed for. This made it clear when tasks needed to be completed by and gave a clear linear development path for different sub systems of the game.



A close up of one of the work packages - within which we have split up into tickets of work to complete.

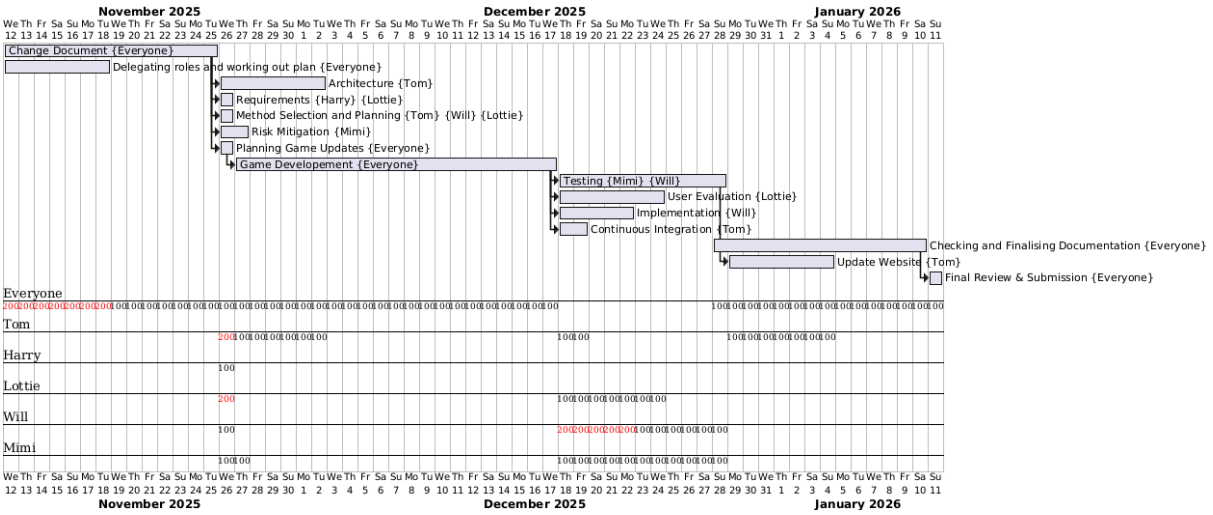
Below is the work breakdown diagram made using UML. It shows the necessary work packages needed to complete the game to the clients requirements. The architecture subtree decomposes the architecture of the game into simpler tasks like creating crc cards and developing UML diagrams, furthermore the develop UML diagrams task is split into sub tasks to create Class, Sequence, Structure and Behavior diagrams. The breakdown also splits the implementation of the game into the functional and non functional deliverables needed to meet the requirements.



Previous groups project plan.

Our groups' updated project plan, involving testing as a section, added functionalities as well as changing the documentation required at the end as deliverables.

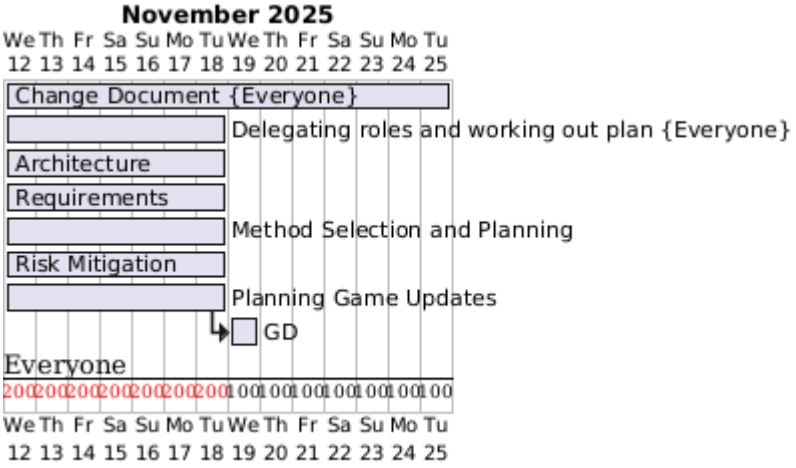
Below is the gantt chart used to display the dependencies of each task and their necessary start and end date counted in days from the start of the project. The Gantt chart shows 5 main sections of the project that need to be completed in order to complete all the required deliverables requirements, planning, architecture, implementation and risk management. Planning is broken down into creating the work breakdown structure and using that to develop a gantt chart to clearly visualize the dependencies between the designs that needed to be made and the implementation of the systems to create the game. Additionally we used a kanban board to clearly display the status of each work package and to assign task priorities using coloured labelling.

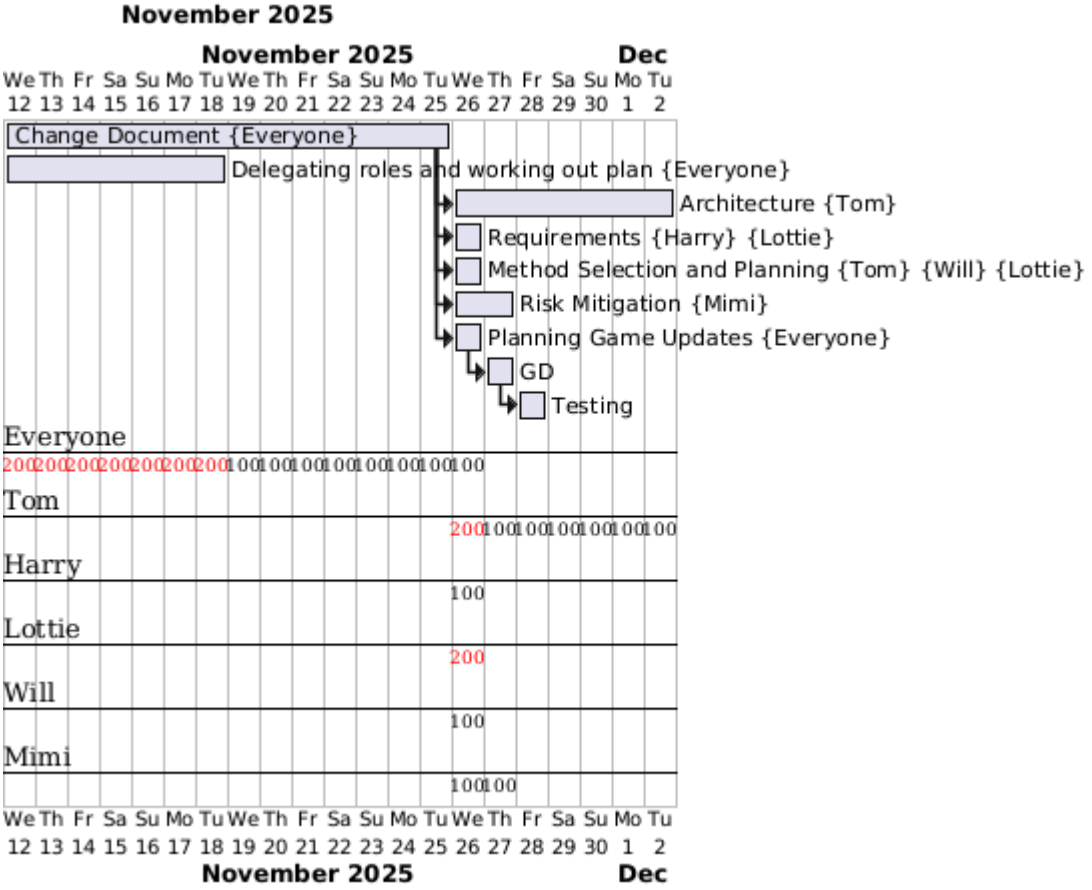


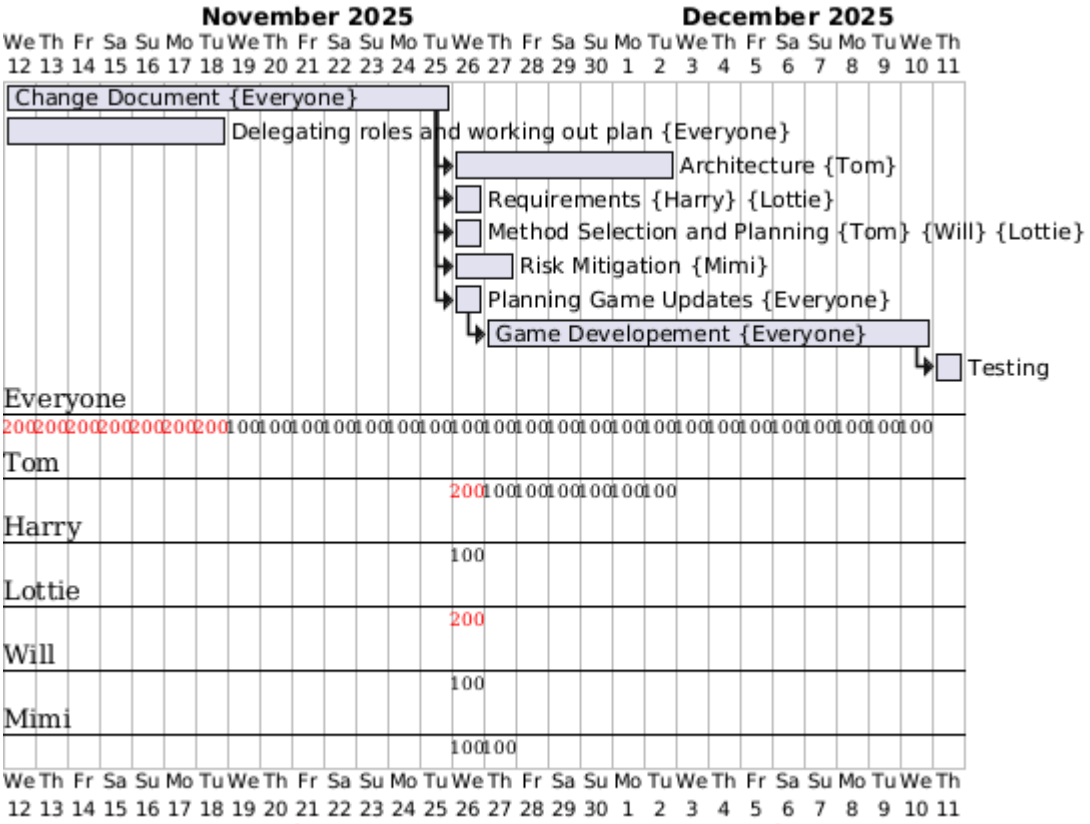
Our groups' final gantt chart that we used to structure our work and manage the progress of our project. Simultaneously holding us accountable for the work that we are supposed to be doing.

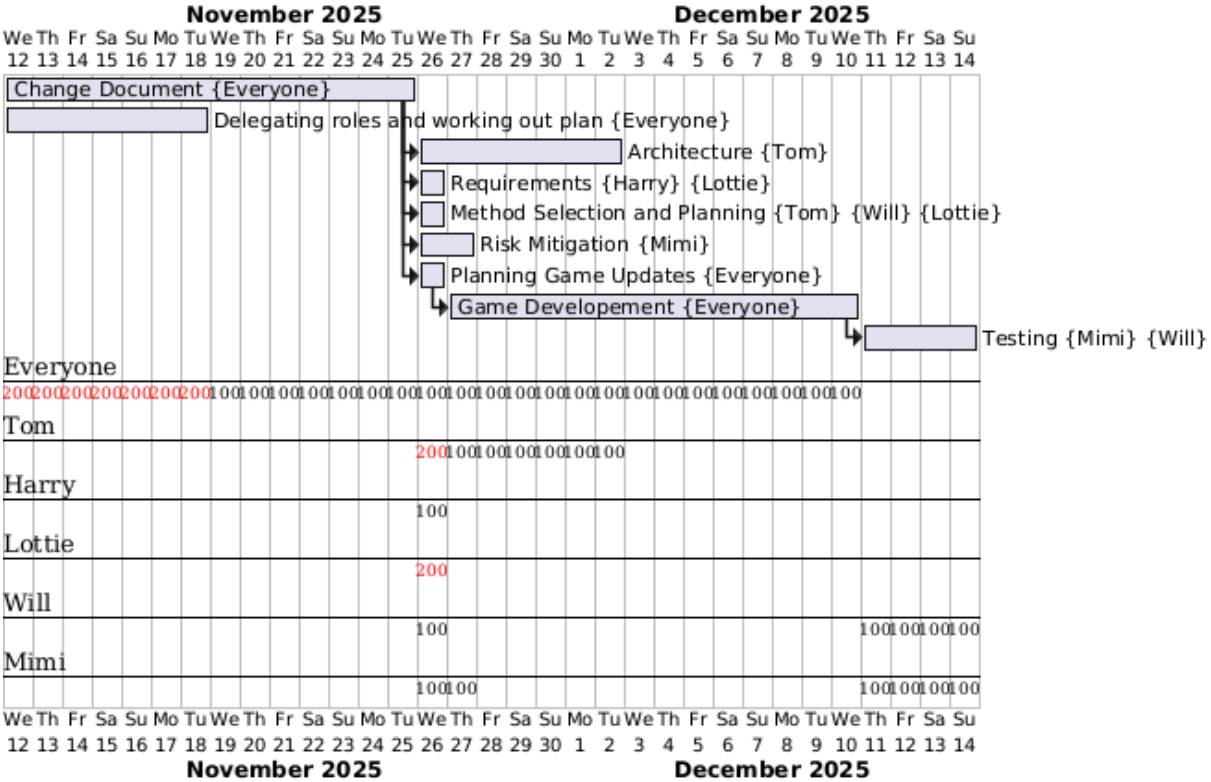
Overall, the plan helped guide our project and keep us up to date. As the previous group kept their work quite organised we were able to quickly adapt their method to suit our needs. The process, although we didn't stick to it too strictly, helped us. The weekly updates and group meetings helped us track our progress and work out if we needed to delegate more or less time to certain tasks. Throughout the course of the project, by having iterations of our plan, and generating iterations of our Gantt chart we were able to keep on top of our project and stick to our deadlines.

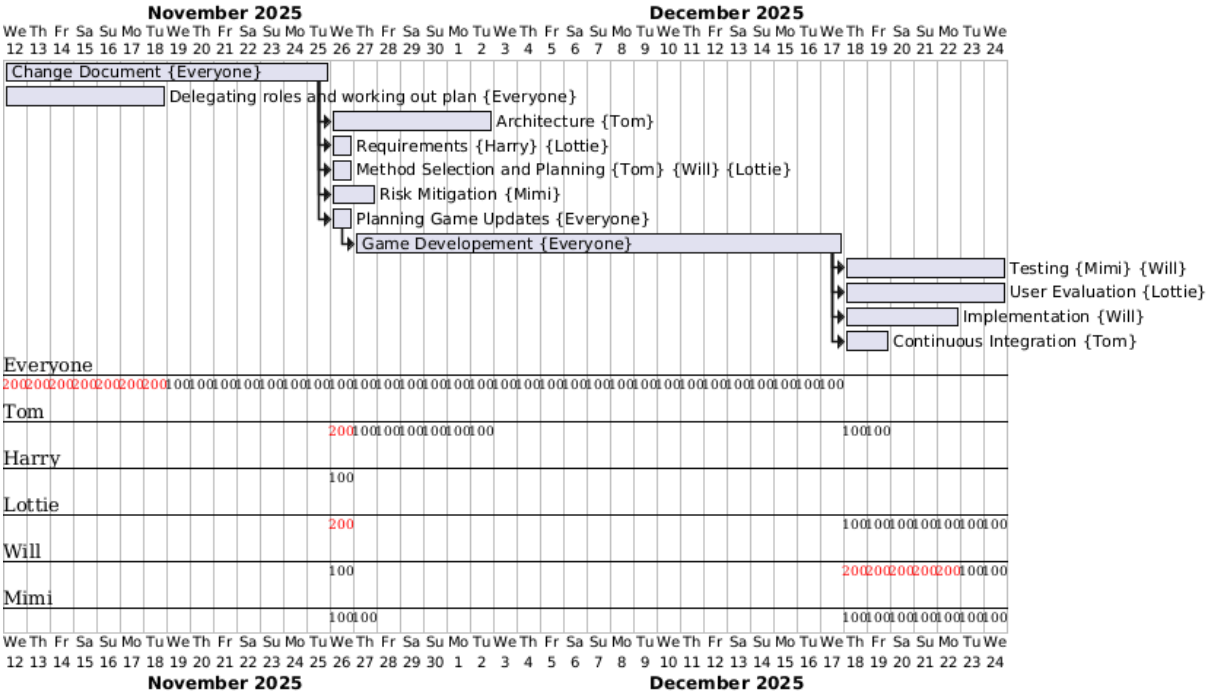
Iterations of gantt charts:


November 2025
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu
12 13 14 15 16 17 18 19 20 21 22 23 24 25
Change Document {Everyone}
Delegating roles and working out plan {Everyone}
Architecture
Requirements
Method Selection and Planning
Risk Mitigation
Planning Game Updates
GD
Everyone
200 200 200 200 200 200 200 100 100 100 100 100 100 100
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu
12 13 14 15 16 17 18 19 20 21 22 23 24 25


November 2025
November 2025
Dec
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2
Change Document {Everyone}
Delegating roles and working out plan {Everyone}
Architecture {Tom}
Requirements {Harry} {Lottie}
Method Selection and Planning {Tom} {Will} {Lottie}
Risk Mitigation {Mimi}
Planning Game Updates {Everyone}
GD
Testing
Everyone
200 200 200 200 200 200 200 100 100 100 100 100 100 100 100
Tom
200 100 100 100 100 100 100 100
Harry
100
Lottie
200
Will
100
Mimi
100 100
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2
November 2025
Dec


November 2025
December 2025
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11
Change Document {Everyone}
Delegating roles and working out plan {Everyone}
Architecture {Tom}
Requirements {Harry} {Lottie}
Method Selection and Planning {Tom} {Will} {Lottie}
Risk Mitigation {Mimi}
Planning Game Updates {Everyone}
Game Developement {Everyone}
Testing
Everyone
Tom
Harry
Lottie
Will
Mimi
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11
November 2025
December 2025


November 2025
December 2025
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Change Document {Everyone}
Delegating roles and working out plan {Everyone}
Architecture {Tom}
Requirements {Harry} {Lottie}
Method Selection and Planning {Tom} {Will} {Lottie}
Risk Mitigation {Mimi}
Planning Game Updates {Everyone}
Game Developement {Everyone}
Testing {Mimi} {Will}
Everyone
Tom
Harry
Lottie
Will
Mimi
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14
November 2025
December 2025


November 2025
December 2025
We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Change Document {Everyone}
Delegating roles and working out plan {Everyone}
Architecture {Tom}
Requirements {Harry} {Lottie}
Method Selection and Planning {Tom} {Will} {Lottie}
Risk Mitigation {Mimi}
Planning Game Updates {Everyone}
Game Developement {Everyone}
Testing {Mimi} {Will}
User Evaluation {Lottie}
Implementation {Will}
Continuous Integration {Tom}
Everyone
Tom
Harry
Lottie
Will
Mimi
November 2025
December 2025