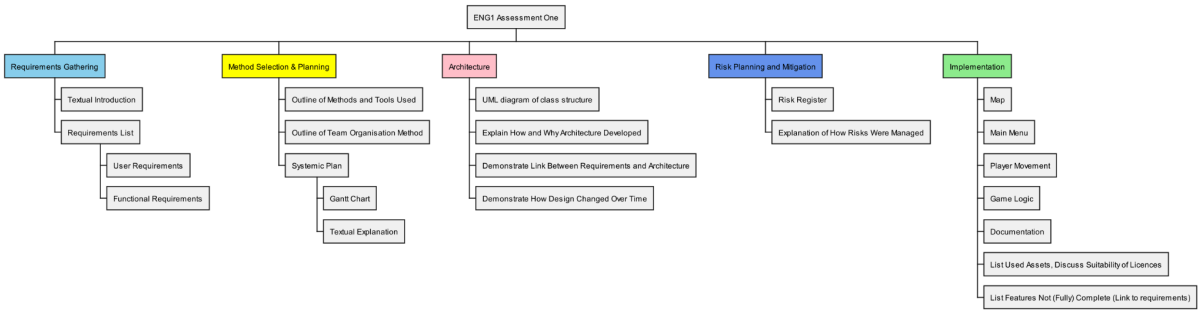
Project Planning

Planning breakdown

To understand the problem fully we created a breakdown of all the work that needs to be done. This was based primarily on the assessment brief and was guided by the customer meeting.



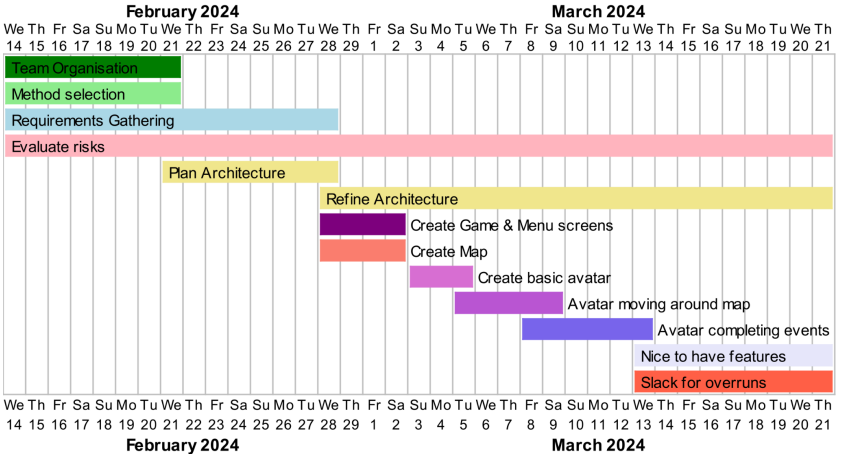
Project Timeline

At first we created a gantt chart to plan how we would work on the project over the phase of the first assessment. We predicted the time it would take to complete each task, as we embarked on the project we found that reality often didn't align with our predictions. We have created Gantt charts to show how our predicted timelines changed over the course of the project.

Original timeline

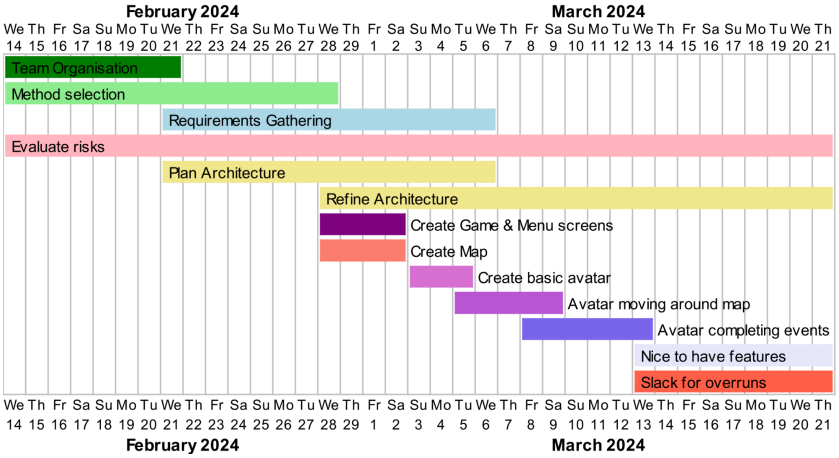
The planned timeline when we started the project is below. We provision time for each stage of the project such that each stage builds on the last. Whilst the bulk of the requirements gathering and method selection were be planned to take place in the below time allocations they will be treated as ongoing concerns for the duration of the project.The "Refine architecture" and "Evaluate risks" portions of the project will be worked on from their start date till the completion of the project.

In this timeline care is given to give significant time and consideration to planning and architecture before jumping into the project. The cost of this within the project is the reduced time to implement the game, therefore the turnaround time of implementation tasks is planned to be relatively short.



First revision

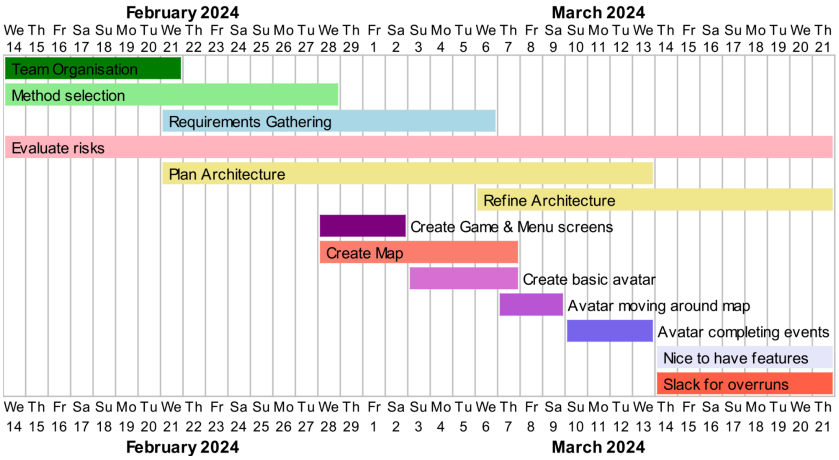
We ran into our first road block following the plan as our client meeting took place on the 29th February, this meant our requirements could not be finalised until after this date. We made use of this time to focus on selecting out methods and tooling, team organisation and risk management. We allowed some extra time to plan architecture after the meeting once we have a more complete picture of the problem.



Second Revision

We ran into a second road block early in our implementation of the project. We found that creating and rendering the map was more difficult than initially anticipated, in addition we ran into some bugs that took time to resolve. For example we had an issue where transparent tiles appeared with a black background. Overall this resulted in the map phase of the project taking longer than planned.

In addition we found that the architecture diagrams took longer to produce than we had anticipated, we had planned out the diagrams enough to proceed with the implementation but we allowed another week for the diagrams to be fully developed.



Justification of Approach to Team Organisation

To make best use of the expertise of team members, we communicated and identified each member's strengths and weaknesses. Once that was done, we assigned team member leadership over different aspects of the project (Based on our previous conversation). We then had a general coordinator responsible for ensuring the health of the project as a whole. All team members will work on all aspects of the project as is needed in line with the work plan we laid out (as can be seen in the *Project Planning* document).

We selected this method of team organisation as it allows all team member’s to take ownership over aspects of the project, we felt as a group this would be important to boost project buy-in. It also ensures all aspects of the project are advocated for throughout the project and no tasks fall through the cracks.

To ensure the project is resilient we will ensure there is ample communication between members, each aspect of the project will have a channel on the group Discord in which all members will be kept informed. This means if by some means a member of the group becomes unavailable (for example due to illness) someone else can easily pick up from where they left off.

This method of organisation fits well with how we will coordinate the project as laid out in our *Outline and Justification of Selected Methods* document.In each practical as we assign tasks there is a member to ensure that sufficient progress is being made on each aspect, this allows us to be more flexible and know more easily when we need to adapt the plan.

.

Outline and Justification of Selected Methods

Development Or Collaboration Tools

Google Docs

We will use google docs to create text documents such as this one. We chose google docs as it allows for easy collaboration between members. All members of the group are already familiar with the software making it an attractive option.

Alternatively we considered using markdown files stored in a git repository hosted on Github, we would use Pandoc to create PDFs. This would have also allowed us to collaborate on files and produce the necessary files for hand-in. We ultimately decided that the simplicity and familiarity of Google Docs made it a more appealing option as the learning curve would have posed an opportunity cost. Additionally google docs allows you to see in real time changes being made by others, something that will come in handy when working together in project meetings.

Git

We will use git version control to manage the game codebase, Plantuml diagrams (when used) and files for our website. We chose to use a version control system for multiple reasons. One reason is that it makes it easier to collaborate, multiple users can make changes to the files and they can be merged together. Another advantage is protection, if a catastrophic mistake happens you are able to revert to a previous version reducing the risk. Furthermore it allows us to develop separate features on different branches, meaning if a new feature is broken and introduces a bug it is still possible to test and run other features simplifying development.

We chose git in particular as it is widely used in the industry, allows team members to work offline, was well used by team members prior to the project and is integrated with GitHub allowing us to host our repository for free..

GitHub

We will use GitHub to host the git repository for the game as well as a github pages repository for the project website. Cost was a large consideration for this as GitHub provides free hosting both for the code and for the website with GitHub pages.

In addition GitHub provides github actions which we will use to perform actions when code is committed and when pull requests are made. For example we may be able to have a suite of tests that run to ensure that new changes are not breaking.

Draw.io

Some group members are familiar and confident with draw.io and therefore it will be used for some of the diagrams. It allows for the creation of simple diagrams using a WYSIWYG editor. We will keep diagrams on a shared Google Drive folder to allow for easy collaboration between members.

PlantUML

For the remaining diagrams we will use PlantUML, this allows for creation of diagrams using a markdown-like language. PlantUML diagrams can be kept in version control along with the

game code which makes it an attractive option for diagrams that may change over time and be edited by multiple users.

Intellij Idea

We have selected Intellij as our IDE of choice. We like that it is extensible, fully featured and Java first. We also appreciated the strong cross platform support, the experience is identical for team members on macOs and Windows. We have made use of project specific settings stored in the game's git repository, meaning the program will appear and act consistently for all group members. One project setting we have set is the code style which is set to the Google Java style.

We considered alternative IDEs and text editors. We strongly considered using VS code. It was attractive as it was extensible, had java support, was familiar to the team and had similar levels of cross platform support. We decided against it as java features such as code formatting and support for the Gradle build system required external extensions whereas they are built into Intellij. We felt that this meant in practice the Intellij environment was more consistent, easy to learn and intuitive.

Tiled

We chose to use Tiled for our tilemap creation. We selected it due to the abundant availability of documentation and tutorials for its use along with its ease of use compared to other options. We found that the support for macOs was less than anticipated so if we were to start the project again we might choose a different tilemap tool.

Discord

We will use Discord as our communication platform of choice. Discord has features that make it uniquely suitable. Discord has support for channels which allow for ideas to be segregated, this may prevent notification fatigue for important messages.

Software Engineering Methods

At the start of each project meeting each member of the team will give a brief description of the work they have completed over the course of the last week. We will work through any issues that have been run into as a group. We will then use the remainder of the session to determine what work still needs to be done referring to the work plan, work that's already been completed and the assessment brief. We will then adjust the work plan accordingly and assign tasks for the following week. During the week people can ask any questions or queries on the group Discord, discussion on specific tasks and areas can be discussed in the relevant sub channel to prevent the main chat from becoming overrun with messages.

This method ensures that all team members are abreast of what all other team members are up to, this means that we can help each to complete tasks and therefore ensure the quality of the final product. Additionally it helps us to mitigate the risk of a team member somehow becoming uninvolved in the project, whether by illness, dropping out or otherwise. We are inexperienced developers and will inevitably run into roadblocks and issues along the way. This method allows for the group to work through problems encountered and ultimately upskill each other in the process, which we will all benefit from as the module goes on and into the future.