Method Selection and Planning

"Mathochist Studios" Cohort 4, Team 11

Euan Cottam
Charlie Thoo-Tinsley
Harri Thorman
Will King
Zach Moussallati
Aiden Turner
Marcus Williamson
Joshua Zacek

Methodology

As a team, it was decided early on that we would use an agile software development methodology, specifically Scrum. This was due to the scope of the assigned project and the minimal time allotted. Based on this decision, we initially assigned our Monday meetings as our Scrum, the weekdays as our Sprint, and Fridays as our Review Day. Any changes demanded would be made over the weekend. This loop ensured we made progress at the required pace.

This methodology evolved slightly as we began holding architecture and implementation discussions during meetings.. As the Implementation team was dependent on Architectural decisions, they began to be involved in communications, thus the workload was able to be spread, and Friday meetings allowed for discussions on the next sprint. This increased the pace of development.

Tools/Software Used

In most cases, when choosing tools, if a member were familiar with the tool, that tool would be prioritised over other alternatives that had similar compatibility with the project. This decision was made to minimise the time spent familiarising with the software, allowing for a faster pace in line with the demands of the agile methodology. However, if an alternative tool were a better fit for the project, it would be prioritised; thus, team members were expected to learn such tools. A list of the tools that fall into this camp would include LibGDX, GitHub (with compatibility for their preferred IDE), and PlantUML.

For communications, the team initially used Instagram Messenger. This choice was made in the first group meeting with little consideration, as we needed a communication method that was quick and easy to set up at that stage. Instagram became redundant relatively early in the project, as team roles were established and it became clear that partitioned, organised communication was required for the remainder of the project. At this stage, both WhatsApp groups and Discord were considered as possible replacements, both due to their ability to create multiple organised text channels; WhatsApp has groups, and Discord has servers. WhatsApp was chosen as members were most familiar with it over Discord.

The game engine chosen for this project was LibGDX. This is a Java-specific engine with cross-platform capabilities for Windows, Linux and macOS. It has a proven track record of developing 2D games. It's open source with an Apache 2.0 licence, meaning it allows the client to distribute commercially if they so choose. Due to its popularity, as well as the plentiful documentation, the implementation team could utilise patterns from the community if they so chose, allowing for a quicker turnaround. Alternatives such as Unity and jMonkeyEngine were considered. In jMonkeyEngine's case, the infrequency of updates and the lesser notoriety of games using the software worked against it. As for Unity, recent licensing controversies and its not being specifically designed for Java development have called into question whether it would have been a suitable fit for software that may be maintained in the future.

Trello was used as an informal planning tool at the start of each meeting to arrange task priorities and their dependencies quickly. Being web-based meant individuals could stay updated on tasks they and others were working on. Due to the nature of its use, alternatives were not considered, as its use did not underpin the majority of the project progress. For a more formal representation of tasks, PlantUML was used to create Gantt charts, as well as other UML diagrams, in the documentation. It was chosen over other alternatives due to its ability to be accessed offline and create non-UML diagrams, such as Gantt charts. However,

due to a feature discovered later in the project in IntelliJ, many UML diagrams could be created automatically.

GitHub was used as our code repository. Being online facilitated remote collaboration, which also aided agile development, as we could push and pull code during sprints. Git could have been an alternative; however, it lacked the same web compatibility, which was the reason it was not used. IntelliJ and Visual Studio were both IDEs used by members of the Implementation team. These were chosen because they were the preferred IDEs of both members, and their GitHub compatibility meant that each member could use their preferred IDE.

G-Suite was used to store and create all non-product documents. G Suite, most notably, included Google Docs, Google Drive, and, to a lesser extent, Slides and Sheets. Google Drive offers a convenient, multi-platform file storage solution, allowing users to share access to documents across multiple accounts with 15GB of free cloud storage. Google Docs allowed team members to create documents and export them as PDFs, as demanded by the project. Some alternatives, such as OneDrive, were considered. OneDrive, as a service, is very similar to G-Suite, and the differences tend to be trivial, so familiarity was prioritised.

Photoshop was chosen for sprite design due to our game artist's prior experience and familiarity, as well as the efficiencies provided by its tool range, e.g., the Magic Wand, which allows for quick recolouring and adjustment of assets. It also allowed for clean exporting into a format that could be rapidly prepared for use in the game engine.

Organisational Structure

Team organisation within the first two weeks of the project was informal, primarily as team members got familiar with each other and the team brief. This approach made sense during the early stages, as the team became familiar with one another and the project. However, some roles emerged within these first two weeks: Joshua, who created a mockup for the website, which he would also do for several other GUIs throughout the project, and Aiden, who implemented the website and continued to maintain it throughout the project. Also, over the weekend, Will created custom assets acting as team branding.
By week three, once the client meeting had passed, a decision was made to assign formal roles. Roles were created for each deliverable, along with art and design roles, as the decision to use custom assets demanded this, and a scrum leader, as the team chose to use an agile methodology. The number of team members assigned to each role was weighted relative to the number of marks for each deliverable, with each team member contributing approximately 11 marks to distribute the workload; however, no team consisted of just one person, in a conscious effort to reduce risk. Role allocation prioritised experience, as it meant less time was needed learning skills in a time when progress could be made, and then gaps in the teams were filled with members volunteering for them. Most roles remained unchanged for most of the project; however, in later weeks, several roles were modified due to unforeseen absences and the need to adjust workloads accordingly. All of this was recorded on a Google sheet for later reference. The roles for each team member are listed in the following table. Roles marked with * were lost later in the project, and those with ** were gained.

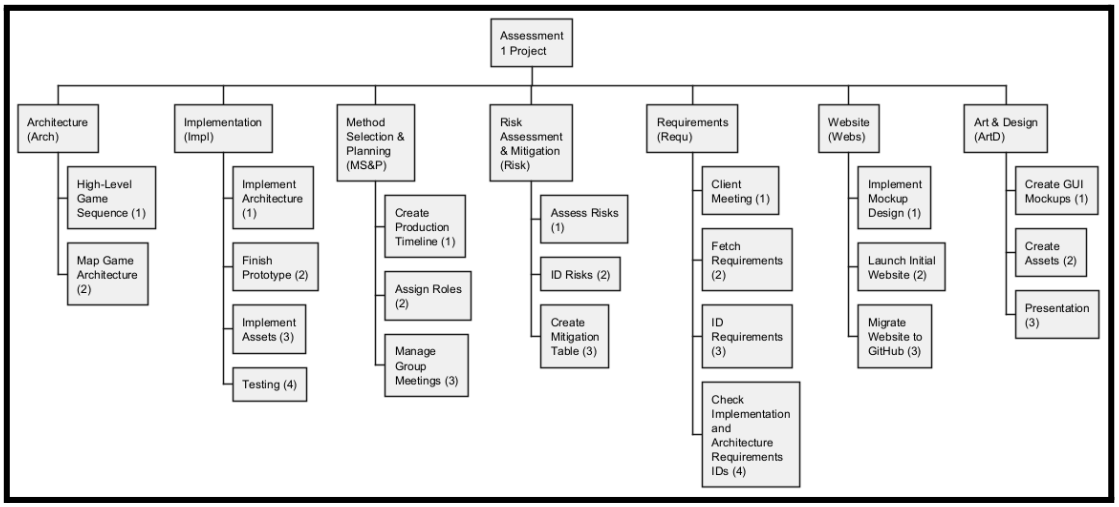
Name	Roles	Name	Roles
Will King	-Art and Design (Game Assets) -Implementation	Hari Thorman	-Risk Assessment and Mitigation -Requirements

Charlie Thoo-Tinsley	-Risk Assessment and Mitigation -Requirements	Aiden Turner	-Website -Implementation
Euan Cottam	-Method Selection and Planning -Architecture	Marcus Williamson	-Architecture -Implementation
Zach Moussallati	-Method Selection and Planning* -Architecture	Joshua Zacek	-Scrum Leader -Art and Design (GUI) -Method Selection and Planning** -Architecture** -Requirements*

Systematic Plan

Key Tasks

The diagram below illustrates all primary responsibilities for the project and the teams' respective scopes of responsibility. Each task is IDed by its team code and designated number, e.g. the task “ID Risks” would be “Risk_2” and the task “Testing” would be “Impl_4”.



Week 1 - Team Forming

Within this week, ArtD_1, Webs_1, and Webs_2 were all completed due to limited dependencies. Requ_1 and ArtD_2 were started; most other tasks, especially those in Arch and Impl, depend on this. Rooms booked for the following group meetings fall under MS&P_3.

Week 2 - Client Meeting Preparations

Although the task was unclassified, shared Google Drive and GitHub repositories were created to aid with the workflow. The remaining tasks are still dependent on Requ_1, which is scheduled to be completed after the client meeting on Monday of week 3.

Week 3 - Project Organising & Requirements Retrieval

Requ_1 has now been completed following the client meeting. Requ_2 and Requ_3 could now be started, and their priority is high, as once completed, Arch_1, Arch_2 and then Impl_1 can be underway. MS&P_2 has been completed to maintain an organised workflow. MS&P_3 . Risk_1, Risk_2, and Risk_3 are all dependent on the previous one; progress is to be made on them over the next two weeks.

Week 4 - Architecture/Implementation Pt 1

MS&P_1, completed at the start of the week, would be gradually amended over the following weeks. ArtD_1, create a mockup for the start screen so that Arch_1, Arch_2, and Impl_1 have been started with the opening screen, as most requirements don't affect this portion. Requ_3 completed, allowing for Arch_1, Arch_2 and Impl_1 to continue on time. ArtD_2 is making progress parallel to Impl_1. Risk_1, Risk_2, and Risk_3 are partly finished.

Week 5 - Architecture/Implementation Pt 2

Risk_1, Risk_2, and Risk_3 are complete. Arch_1, Arch_2, and Impl_1 are progressing and now working on the central gameplay portion. Mostly preparing ahead of the Consolidation Week.

Consolidation Week - Architecture/Implementation Pt 3 and Writeup Drafting

Arch_1 and Arch_2 are now mostly complete, and therefore, final write-ups for Architecture are set to begin, along with MS&P requirements and risk assessments. Impl_1, Impl_2 and ArtD_2 to be finished by the end of the week, so that work can be made on Impl_3 and Impl_4 in the final week.

Week 6 - Beautification, Finalisation and Testing

Continued progress was made on all write-ups, which were expected to be finished by either Saturday or Sunday. This also involved Requ_4 beforehand. Webs_3 was a new addition to the plans this week to facilitate a smoother handover for those who may take over the project. The main body of ArtD_2 is complete; however, Impl_3 and Impl_4 require minor adjustments and, in some cases, new assets. ArtD_3 is set to be completed after submission.

Gantt charts and Weekly Devlogs can be found at https://mathochiststudios.com/log.html