

Method selection and planning
Group 12
Team 12

James Leney
Daniel Agar
Charles Thompson
Leyi Ye
Vanessa Ndiangang
Matt Smith

The initial plan was developed in consideration of the time constraints, the capabilities of the team members and the experience of the team in previous projects. Additionally, the team aimed to maximise the amount of information they could gather from the customer while also engaging in research to better understand and describe the project.

Our initial approach takes from the principles of soft systems methodology (SSM). This approach can be described as a Soft Systems Methodology [1]. Our main priorities were to create a simple, flexible structure around the seemingly complex initial project brief, identify the strengths and weaknesses of the different group members, conduct initial research, and have a broad overview of what should be done. SSM was the most suitable method as we were able to elicit requirements, understand the project from our different perspectives to see what was required and determine the risks involved early on for early mitigation.

Following the customer requirements meeting, the team completed initial research and gained an understanding of the strengths and weaknesses of the team members. The team had enough information to begin testing technologies and development tools, create a more structured plan to move forward with, develop an aesthetic design for the game, and fully list out requirements as per the needs of the customer. As such, adopting an agile development approach was appropriate, specifically taking from aspects of the scrum framework [2]. The team still needed to be flexible, to be able to distribute and redistribute work if required for reasons such as one of the team members falling ill or a misestimation of the amount of time or resources needed to cover the work required. It was also particularly important to organise work into weekly sprints due to the short time frame of the project. We still maintained most of the same roles from the initial stage of the project but expanded or restricted our roles based on the amount of work required for each task.

Development and Collaboration Tools

Collaboration tools:

Communication	Discord	<ul style="list-style-type: none">- Familiar with team members- Fast- Allows for distribution and tracking of files and code sent in the chat easily
	Whatsapp	<ul style="list-style-type: none">- Fast- Accessible to all members- Useful for quick messages and notes to be sent
File Storage	Google Drive (Docs, Sheets)	<ul style="list-style-type: none">- Accessible to all members- Allows for real-time collaboration and updates- Maintains accountability
Planning	Notion	<ul style="list-style-type: none">- Flexible
	Text Files in repo	<ul style="list-style-type: none">- Allows for the team to organise itself and develop its own structure- Allows for all updates to be seen by team members

Alternatives considered:

- Communication:
 - Email was considered but this would have been too slow and overly structured given the small size of the team. Updates were better shared asynchronously on platforms like WhatsApp and meeting minutes could be recorded to keep track of information shared.
- File Storage:
 - No other alternatives were considered for file storage. Google Drive was the most familiar and accessible option for all team members.
- Planning:
 - Jira was considered as an alternative to draw up a project plan. This is because it can be connected to other Atlassian suite tools such as Confluence to potentially use for document storage and documentation and Bitbucket to track updates to

the code. However, for the sake of simplicity, Notion was chosen as it was a simpler tool that allowed for more flexibility in changing which was better aligned with the agile principles [3] we aimed to adopt.

Development Tools

- IDE: IntelliJ
- Architecture and Design: PlantUML
- Version Control: GitHub Desktop
- Game Engine: *libGDX* [\[4\]](#) licensed under Apache 2.0
- Map Creator: *Tiled* [\[5\]](#) licensed under GPL 2.0

Alternatives considered:

- IDE:
 - VS Code was considered but given the constraint that Java was to be used, IntelliJ was better suited for that.
- Framework/Game Engine:
 - jMonkeyEngine was considered but it is better for 3D games where as LibGDX is better suited to 2D games.
- Architecture
 - Miro was considered but it would not be as easily placed within the IDE as PlantUML
- Version Control
 - GitHubDesktop was the only option considered due to its familiarity and ease of use to most of the team
- Tiled: No other options were considered due to its familiarity and ease of use to the team.

After researching and engaging with the project brief, we decided to prioritise flexibility and group ourselves according to our strengths. These priorities align with practices within the scrum methodology and adapt aspects of it to the nature of our team.

- Simplicity is key: Since we are a small team (6 people) there is no need for a large amount of documentation.
- The team should be self-organising: The team aimed to create a flexible project environment where we could shape it based on our needs.
- The total amount of work is divided into smaller “sprints” tasks in deliverables.
- Weekly meetings are arranged according to the amount of work that is meant to be done for that week (apart from the practical).
- Every meeting is used by each attending member to share their progress on the work assigned or any problems they have encountered.
- Before finishing the meeting, the team decides on and assigns the work that needs to be done before the next meeting.

Next, apart from following the SCRUM method, the team divided the entire group into smaller groups of two. Each small group could work on different aspects of the project simultaneously. At the same time, any individual team member could refer to someone in the group if they encounter any type of problem. The role assignment of the project is shown in the following table:

Deliverable	Responsible	Collaborator	Informed
Website	Matt		Charles, Daniel, Jame, Leyi, Vanessa
Requirements	Charles, Matt	Daniel, James, Leyi, Vanessa	
Architecture	Daniel, James, Vanessa, Charles, Matt, Leyi		
Method Selection and Planning	Leyi, Vanessa		Charles, Daniel, Matt, James
Risk assessment and mitigation	Charles, Matt	Vanessa, Leyi	Daniel, James
Implementation	James, Daniel		Matt, Charles, Leyi, Vanessa

“Responsible” means that for the task, this member does the main portion of work in terms of that deliverable. “Collaborator” means the member needs to stay aware of what needs to be done and provide help if and when it is needed. “Informed”, as the name indicates, they only need to be informed of the progress and outcome of the work.

Following the first meeting as a team in the first week of the project, the team assessed each member's strengths and preferences to assign their roles and responsibilities. The main aim was to determine the requirements from the product brief, the risks and their mitigation strategies, as well as researching game engines, assets, and technologies to be used. Using that information, we created an initial plan for the whole project [[Gantt Chart1](#)], which was flexible and allowed for changes over time.

Initial Systematic Project Plan

Stage 1: Research and Requirements Elicitation

15th February – 29th February

- Assess team capabilities and strengths
- Assign roles
 - Technology
 - Planning And Organisation
 - Risk Assessment
- Watch all available lecture videos
- Conduct initial research to mutually understand the vision of the project
- Determine the requirements of the product
- Determine risks involved

Stage 2: Development

1st March – 13th March

- Set up technology
- Design architecture of the game
- Build the product based on the initial architecture design
- Review the product and test how it works

Stage 3: Final Product

14th March – 20th March

- Refine the code behind the product based on the challenges presented
- Refine architecture design to reflect changes in the game
- Add documentation to accompany the code
- Write up deliverables to reflect the progression of the project

After the customer meeting on the 29th of February, the structure of the project became more focused on the main tasks set out in the project brief. The structure could be divided into 2 components, programming-related and non-programming-related

portions of the project. This allowed for a more organised workflow and for team members to each take ownership of various tasks for which they were responsible.

General Systematic Project Plan

WP 1: Brainstorming and research: 19th February - 8 March	High Priority Dependencies: Blocking Game Design and Requirements
Work Package 2: Write up for the assessment: 15th February - 21st March	Medium Priority Dependencies: Blocked by Brainstorming and Research
Work Package 3: Requirements and Risks: 21st February - 6th March	High Priority Dependencies: Blocked by Brainstorming and Research; Blocking Architecture
Work Package 4: Architecture: 1st March - 13th March	High Priority Dependencies: Blocked by Requirements and Risks; Blocking Implementation
Work Package 5: Game Design: 15th February - 6th March	Low Priority Dependencies: Blocked by Brainstorming and Research, Blocking Implementation
Work Package 6: Implementation: 29th February - 21st March	High Priority Dependencies: Blocked by Brainstorming

After week 3 [\[Gantt Chart 2\]](#), the main difference compared to the previous week's planning is that we started working on different areas simultaneously, and advanced some of the work that was previously planned to be done after the customer meeting. Since most research had been completed, requirements and risks could be finalised and architecture could be drawn up, all relatively independent of one another.

In week 4 [\[Gantt Chart 3 and to-do list\]](#), the plan for the project started to stabilise more. There were no big modifications in comparison to the previous weeks. The team members responsible were able to begin coding and building the game.

Finally, from week 5 [\[Gantt Chart 4\]](#) to the final deadline in week 6 [\[Gantt Chart 5\]](#), the main objective was to finish the implementation of the game, write up documentation and complete the deliverables for submission.

References:

- [1] C. Reaiche and S. Papavasiliou, "Soft Systems Methodology," *jcu.pressbooks.pub*, Aug. 2022, Accessed: Mar. 07, 2024. [Online]. Available: [https://jcu.pressbooks.pub/pmmethods/chapter/soft-systems-methodology/#:~:text=Soft%20Systems%20Methodology%20\(SSM\)%20is](https://jcu.pressbooks.pub/pmmethods/chapter/soft-systems-methodology/#:~:text=Soft%20Systems%20Methodology%20(SSM)%20is)
- [2] B. S, "What Is Scrum Methodology? & Scrum Project Management," *Nimble*, Dec. 23, 2022. <https://www.nimblework.com/agile/scrum-methodology/>
- [3] Agile Alliance, "12 Principles Behind the Agile Manifesto," *Agile Alliance*, Nov. 04, 2015. <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>
- *libGDX* [4]: a cross-platform Java game development framework licensed under Apache 2.0
- *Tiled*: a 2D level editor licensed under GPL 2.0, BSD 2-clause licence and Apache 2.0 [5]