Method selection and planning

Team 10

Gergely Gal
Ben Leaver
Ali Tekin
Utkarsh Sing
Hux Stevenson
Faras Hbal
Chihun Park

Methodologies

Methods and Justification

Our team has used a **Hybrid Methodology** throughout the project, with a focus on meeting deadlines, adapting to changes, and producing high quality code and deliverables:

- **Waterfall-Hybrid** for the **Project Structure**. The high-level plan for our project, consisting of the six assessment deliverables (D1-D5), is a Waterfall-Hybrid, visualised in the baseline Gantt chart below. For some of the deliverables, e.g. Requirements and Architecture, the dependencies between them and clear waterfall phases mean we have logical milestones and easy tracking of progress. The project also benefits from the agile approach of the methodology, as making iterations and prototypes allows us to be flexible and hit tight deadlines.

- **Scrum-Inspired** for **Project Management**. Scrum is an agile framework that focuses on iterative progress, which provides flexibility when managing the project structure. Our one-week work cycles act as our Scrum sprints, with the Weekly Snapshots being our 'sprint reviews/planning'. This management strategy allowed us to adapt our original Waterfall plan when the Requirements deliverable was delayed, being able to acknowledge the issue quickly and switch to a more agile approach for subsequent deliverables.

- **Extreme programming** for the **Development** of the project. An agile development methodology that has a focus on high quality code is ideal for the code-dominant LibGDX engine. XP has a fast feedback loop that benefits the project because we implement mechanics and test consistently, as shown by our frequent GitHub commits.

Development/ Collaboration Tools and their Fitness with Methods

The development and collaboration tools we used to support the project and the team working were:

- **PlantUML** allows us to create Gantt charts to visually represent the schedule of the project, and show how different tasks overlap and depend on each other. This supports our **Waterfall-Hybrid** project structure, and can be exported to provide us with Weekly Snapshots. We have also used it to create various structures for the Planning and Architecture deliverables.

- **Visual Studio Code** is compatible with Java 17 and LibGDX, and has the necessary plugins for debugging and testing tools (Jacoco) needed for **XP** programming practices. Also compatible with Gradle which we are using as our build tool.

- **Git and Github** allows for Continuous Integration via commits and pull requests, which is needed to support **XP** development. Git is a version control system system that allows us to track changes in our code. Everyone in the team can access the code and view the version history to understand changes that have been made, which is also useful for **agile management**.

- **LibGDX** is a free and open source game development framework that provides the

core Java components for game making. It suits our project as it is highly modular, allowing for **agile**, parallel development and design.

- **Google Drive** and an **Instagram** group chat allow us to collaborate on materials and communicate effectively throughout the project. The group chat allowed for a simple way to discuss progress and share 'Daily Scrum'-like updates with the team, adhering to our **Agile Management Methodology.**

.

Team Organisation

Approach

Our approach to team organisation was less structured, and didn't include the classic team roles of leader, researcher, teamworker, etc. Rather, we all worked as a group and the roles evolved more naturally, with people performing the tasks that made the most sense for them to perform at any given moment. We fell into roles suited for us rather than assigning them at the start. There was not a clear leader in our group which allowed for everyone's voices to be heard equally and decision making was done as a group.

Justification - For the Team

We felt that this approach would be good for our team for a few reasons. The first of which was that we didn't automatically fall into different roles, in that there was no clear person suited for each role initially. We felt that letting people fall naturally into roles over time would allow for better, more fitting roles than if we had just assigned them. For example, people who were better at organisation gravitated to the organisational tasks, meaning that roles were more suited to the people in them. Following on from the point about having no clear leader, this allowed for everyone on the team to have an equal part of decision making, and multiple viewpoints were heard at every meeting.

Justification - For the Project

As mentioned in the justification for the team, our method allowed our team to work in roles more suited to their skillset. WIth everyone working in areas that they are best suited for, it meant that the project could proceed more efficiently. With the point of everyone having an equal voice, and everyone getting the chance to put forward ideas, this allowed us to have a constant stream of different points and views, allowing us to make valuable changes to the project based on a big picture made up of everyone's contributions. This way of team organisation was suitable for the chosen software engineering methods as well, particularly for the scrum-inspired part of our organisation, which features flexibility in the project structure.

Planning for Assessment 2

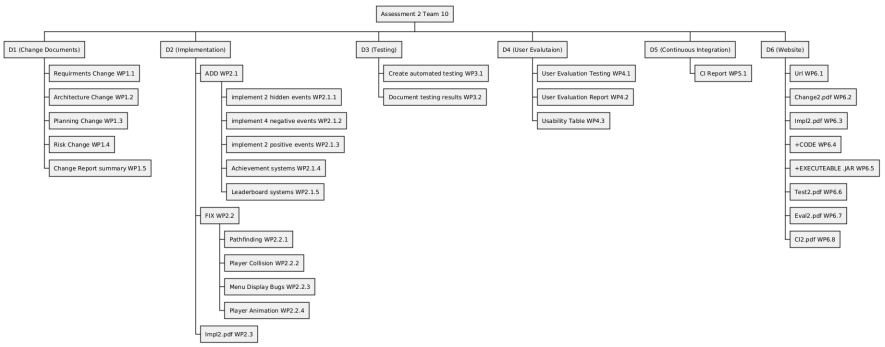
New Key Tasks

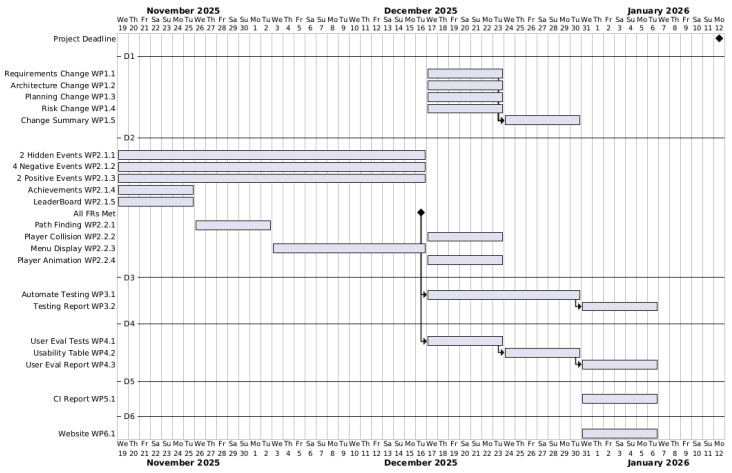
The table below summarises our 5 key deliverables. It outlines the timelines, Initial priorities, and dependencies decided after the selection of the new base game and the new requirements needed to add to this new project.

Deliverable	Start on	End by	Initial Priority	Dependencies
D1 Implementation	Week 1	Week 4	High	
D2 Change Reports	Week 6	Week 8	Low	
D3 Testing	Week 4	Week 7	High	D1 Implementation
D4 User Evaluation	Week 4	Week 7	Medium	D1 Implementation
D5 Continuous Integration	Week 2	Week 7	Low	

Project Schedule Change

A new project schedule had to be devised after analysing what deliverables the current version of the project has completed and what requirements it still does not have, the work breakdown structure covers all the necessary work packages needed to create a project that meets all the deliverables and Gantt chart which is used to create an initial waterfall suitable schedule for the development of the game.





https://thedebugthugsuoy.github.io/

References

Kolovos, D. (2025). Project Planning and Risk Management. Department of Computer Science, University of YORK

IEEE (2018) ISO/IEC/IEEE International Standard - System and software engineering life cycle processes - requirement engineering
https://standards.ieee.org/ieee/29148/6937/

The Debug Thugs (2025) Escape the Maze - ENG1 Assessment 1 Project Website
https://thedebugthugs.github.io/

Team 10 (2026) Escape the Maze - ENG1 Assessment 2 Project Website
https://thedebugthugsuoy.github.io/