ENG1 Assessment 1: Method Selection and Planning

Greenfield Development

Group 6

Members:

Mark Faizi

Ben Faulkner

Christian Foister

Lewis Hanson

Ziyad Rashad Hassan

James Haworth

Team’s Methodology brief

Agile and scrum methodology

The team chose to use the Waterfall methodology for this project due to its linear and structured approach, which suited the defined nature of the game development tasks. Waterfall provided a clear framework, where each phase—such as requirements gathering, design, implementation, and testing—could be completed sequentially before moving on to the next. This method allowed the team to thoroughly define and document requirements upfront, which was ideal for a project with a relatively fixed scope. The lack of frequent iteration and changing requirements made Waterfall an appropriate choice, allowing the team to focus on completing each phase before progressing. The structured process ensured that each task was well-organised and delivered on schedule, providing a systematic way to handle development

Discord

Discord played a key role in communication and task management. Its "react" feature allowed team members to quickly assign themselves to tasks, and the flexibility of Scrum was well supported by Discord's messaging system. Changes to sprints could be communicated instantly via dedicated channels, keeping everyone aligned with deadlines and goals. Most team members were frequently active on Discord, making it an effective tool for quickly addressing urgent issues and fostering collaboration in real time.

LibGDX

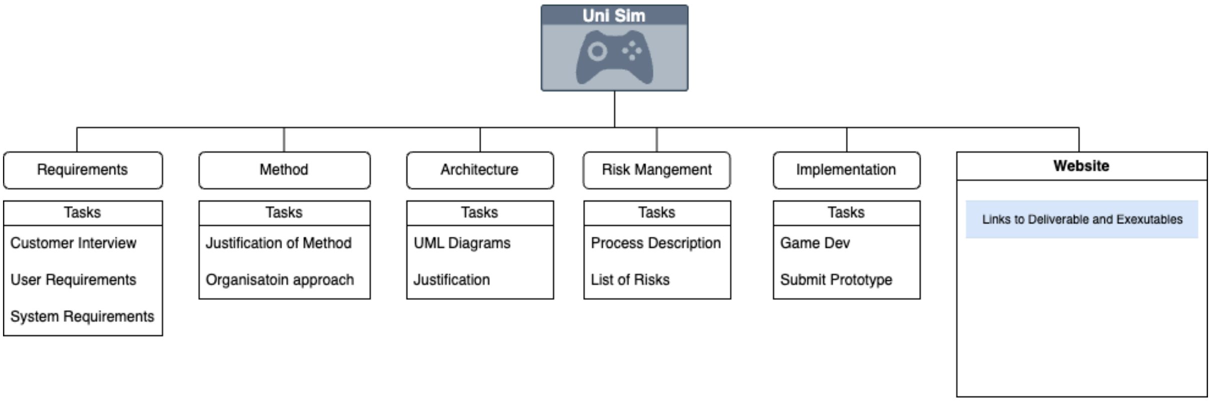
The team chose LibGDX as the game engine due to its popularity in Java-based game development and the wealth of online resources available. Since the team had no prior experience with Java game engines, the support and tutorials for LibGDX helped them quickly overcome initial learning hurdles. Its widespread use also made it easier for other teams familiar with the engine to pick up the project in the future. Additionally, LibGDX abstracts much of the complexity of OpenGL, which was particularly useful given the team's limited experience with graphics programming. Its integration with Gradle for build management further streamlined the development process.

IntelliJ

Although most team members were familiar with Visual Studio Code, the majority opted to use **IntelliJ** for this project, particularly for working with LibGDX and managing the build process through Gradle. The decision to use IntelliJ was influenced by the fact that most instructional materials for LibGDX and Gradle recommended it as the preferred IDE.

GitHub

GitHub was selected to manage the codebase due to its user-friendly interface and familiarity among some team members. It offered flexibility, allowing access through both the web interface and the command line, preventing workflow conflicts. GitHub’s tracking of branches, forks, and code updates helped the team stay informed about project progress and understand parts of the code they weren't directly involved with. However, the team noted some challenges, such as difficulties in renaming or deleting files, which required additional coordination.



Phase

ID	**Title**	**Brief**	**Due Date**
P1	Programming	Complete deliverables 2-5	15/10/2024
P2	Implementation	Finish the implementation for assessment 1 with a working executable.	25/10/2024
P3	Finish Assessment 1	Finish everything for assessment one so that anyone can access the information	11/11/2024