Software Engineering Methods

We decided to use agile methodology for this project, due to the timescale and type of project. The Agile methodology will allow us to focus the software development in short periods of time, allowing the team's development progress to be monitored and highlight important tasks to be focused on in the next time period. Agile methodology also allows us to make changes as the project progresses, as it encourages close collaboration with the stakeholders and therefore regular updates and demonstrations.

This methodology is suitable for this project due to the nature of the project. There may be regular changes to the requirements over the development period as it is assessed, and this is the first collaborative project for many of the members of the team.

Collaboration Tools

A number of collaborative tools were used over the course of the project to help the team communicate, collaborate and store information.

Our primary communication source was a Discord [1] server, which was chosen due to its flexibility as a platform. We created a number of channels, so that relevant discussions could be grouped together. This platform also allows easy sharing of images, such as screenshots of the project whilst under development

To aid the design of the system, the PlantUML [2] software was used for our UML documents. This tool uses textual descriptions to generate the relevant UML diagram, which is saved to the team's filestore and website.

To create the game in Java, the language requested by the customer, we decided to use the libGDX [3] game development framework. This will allow the team to create 2D or 3D games, with features such as built-in physics and management of keyboard and mouse inputs, which will help with the development process. This framework, along with the recommended development environment will help the team to focus on implementing the user's requirements, rather than the low level mechanics of game development.

The team's code and documentation were stored in a number of locations, to enable easy collaboration.
Documentation was stored on Google Drive [4], using Google sheets. This platform allows for real-time collaboration between team members, easy organisation through folders and a change history so contributions to the documentation can be tracked.
Our code was hosted on GitHub [5], as the team had previous experience with this platform. Other solutions, such as Apache Subversion [6], were considered but the team decided GitHub was the appropriate solution for us. Using GitHub allows the

team to use version control for the code, allowing features to be added on multiple branches independently, before combining into the final codebase.

For our website, GitHub Pages [7] was selected as the hosting platform. This was quickly decided due to the fact that it is a free platform, and uses markdown text to create pages.

Team Organisation

The team was divided into two main groups, to focus on implementation and documentation. It was decided this was the best way to split the team due to the short timeframe of the project, and due to the size of the deliverables.

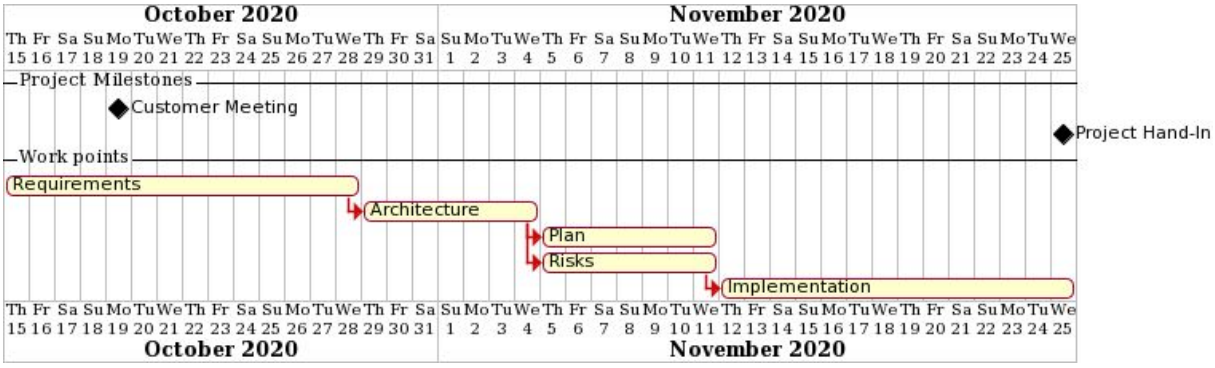
Implementation Team

- Joshua Cottrell
    - UI
- Indraj Gandham
    - General Code & Architecture
- David Kelly
    - AI

Documentation Team

- Andrew Chapman
    - Risk & Implementation
- John Cherry
    - Requirements & Plan
- Mircea Zisu
    - Project Manager

Project Plan



The Gantt chart above outlines the way the group plans to complete the project and the timescale for each sprint of the agile project. The Implementation stage starts early in the project to allow for the development group of the project to review how the selected Java framework works and develop a prototype to show to the clients. This prototype will help inform the rest of the design process and to validate the architecture designs. Once the architecture has been finalised, the final project implementation can happen.

Throughout development, the documentation group will work in parallel to the implementation group, documenting each stage as it is developed and implemented. The plan above was chosen as an optimistic timeline, and each section may take more or less time to implement. Using an Agile methodology allows the team to adapt to changes in timescale for the project.

The project plan will be reviewed on a weekly basis, and updated charts put on the updates page of the team's website - https://team-glytch.github.io/updates.html.

Bibliography

[1] Discord, “Discord”, [Online]. Available: https://discord.com/

[2] PlantUML, “PlantUML”, [Online]. Available: https://plantuml.com/

[3] badlogicgames, “libGDX”, [Online]. Available: https://libgdx.badlogicgames.com/

[4] Google, “Google Drive”, [Online]. Available: https://drive.google.com

[5] GitHub, “GitHub”, [Online]. Available: https://github.com/

[6] Apache, “Apache Subversion”, [Online]. Available: https://subversion.apache.org/

[7] GitHub, “GitHub Pages”, [Online]. Available: https://pages.github.com/