Method Selection and Planning

Cohort 3 Group 6 - Carbon Goose

Members: Bailey Horsley, Owen Jones, Rory Ingram, Ken Jacob, Abishek Kingslin Raj, Louis Polwarth, Adam Johnson

For the project, the team selected the Agile framework which focuses on collaboration, adaptation and continuous improvement. We are going to be using IBM's Rational Unified Process (RUP) for our engineering method, as this method is plan-driven, and used for large projects in a stable environment. Starting with storytelling, we will outline the requirements needed for the completed project to create a plan of our Use Cases. From here, we will convert these Use Cases into a more Object-Oriented design, with Class Diagrams and modelling using UML. Model-driven development is perfect for this project, as we have a fixed product brief to follow. Therefore we are able to expand on the product brief and client meetings, and refine accurately into models, without any changes being made to the requirements.

In each meeting, we report what work members have finished, currently working on and will continue working on for the week. In addition, at the end of each sprint, we review what has been completed and what not, what needs priority and what to do differently for the next sprint. If members have any problems and delays in their work, their sprint time and deadline will be recalibrated and adjusted accordingly.

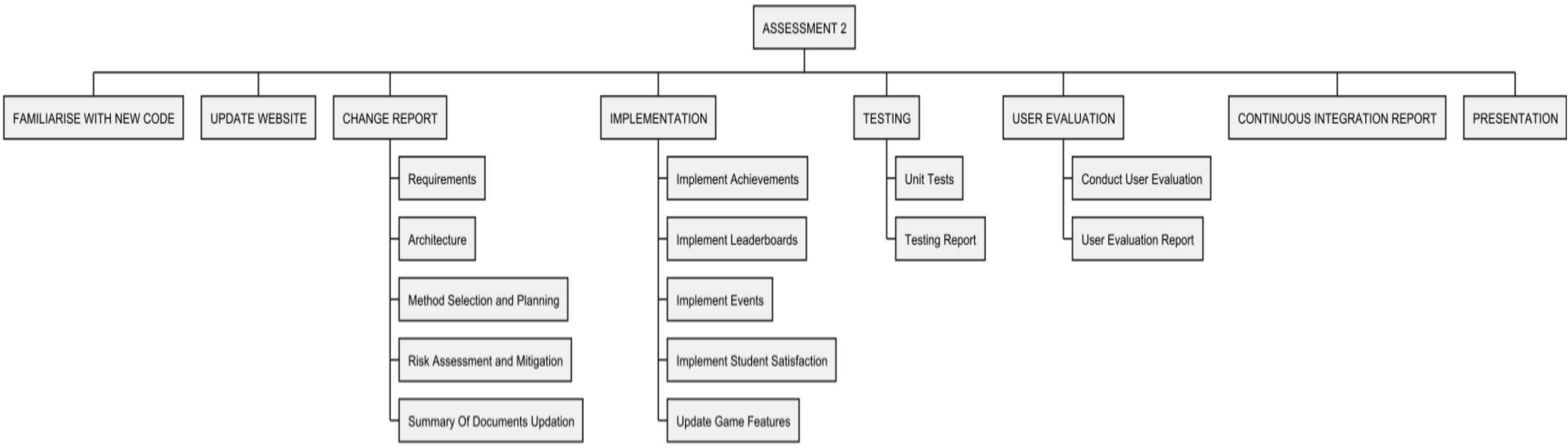
Development/collaboration tools we used for our project:

- Google Drive/Docs: used for recording meetings and practical session minutes. The minutes detailed the progress of the projects and everyone's respective tasks. The shared drive organised and gathered all of our important documents in one space, allowing for a collaborative environment which also enabled all team members to contribute to the project on a real-time basis.
- Plant UML(Tasks Management): it was used to create the Gantt chart that contains a clear view of the different tasks assigned, their start and finish dates, and each team's member contribution. The visual format made it easier to track everyone's task(s) progress and clearly outline their priorities through dependencies of different members' tasks during the project development.
- Github(Version Control): it was chosen for source code management. It is where the code is stored and the tool facilitates collaboration of all team members. It helped ensure that the code was constantly up to date.
- IntelliJ IDE: it was the IDE chosen to develop the code. Its features like code completion, debugging and refactoring tools improved productivity and allowed us to focus on building the game more efficiently.
- Java 17: we used it to develop the core components of the game.
- LibGDX: it was our chosen game development framework. It provides the necessary tools for building 2D games. It also offers high flexibility for managing graphics, input handling, and game physics, making it suitable for developing our game.
- Discord: it was our chosen platform for communication, it facilitated smooth communication between the team members during the development of the project.

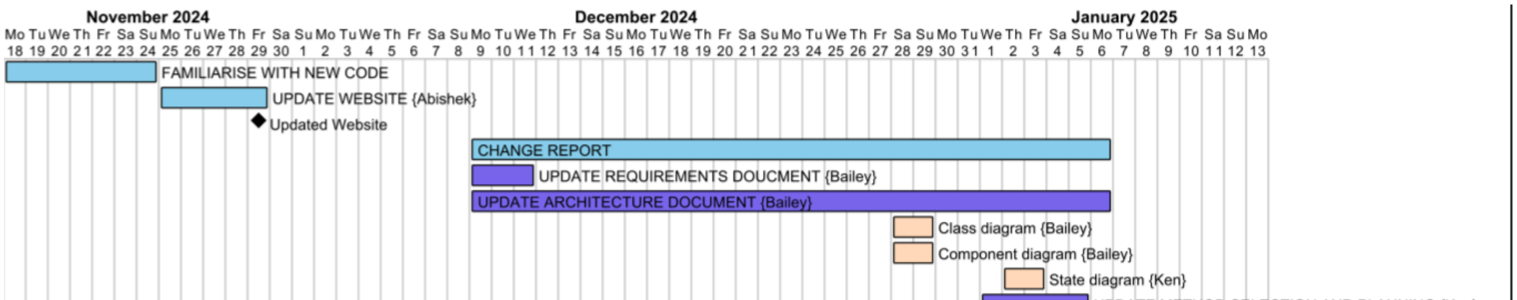
To effectively manage our project, we established a structured approach to team organisation, focusing on regular group meetings among members, clear communication, and use of collaborative tools. We held two in-person group meetings per week, during which we researched and worked on project-related tasks, and assigned tasks to each team member. We kept a meeting log that recorded the previous week's meeting agenda to ensure that the work was completed and the next week's meeting objectives, which we reviewed at the start of each meeting. Before our

Christmas break we allocated tasks among members to be done over the break and also scheduled a few meetings to discuss the work progress.

Work Breakdown



Gantt Chart



Project Plan Evolution

Week 1 (November 18 - November 24): Familiarization with the acquired deliverables

After selecting the team whose project we wanted to take over, during our weekly meeting we decided that our main goal should be to get familiar with the code and documentation acquired by us, so we decided to give a week for all members to go through the new code and deliverables.

Week 2 (November 25 - December 1): Updating the website and initial allocation of work

During week 2, we started working on updating the website we inherited from the other team. However, due to an issue with the transferred documents we could not work on the new documentation and we deferred the documentation work for the next week. We also allocated few other work between members

Week 3 (December 2 - December 8): Start implementation of new requirements

We had still not resolved the issue with the documents of the other team, but we started working on implementation of the new requirements for assessment 2 and updating some of the game features

Week 4 (December 9 - December 15) : Started work on documentation and allocated work to each member for the christmas break

The issue with the documents was resolved by the end of week 3 and we started working on updating the documentation and continued working on implementing the new requirements on week 4. By the end of week 4 we completed updating the risk assessment and mitigation document. Since this was the last teaching week of this semester we allocated work to do over the Christmas break during our meeting and scheduled a few online meetings to discuss the progress made by each member on their work.

Revision week (January 6 - January 12): Implementation, documentation, user evaluation and testing completed

By the start of revision week, we had finished updating our documents, updated the game to include the new requirements and also updated some game features. We also conducted our user evaluations over the Christmas break. Unit tests were also finished before this week. By the end of the week we had completed all our work and were ready to submit our project. During our meeting we also started discussion on the presentation.