Method Selection and Planning

Cohort 3
Group 11
Team Aubergine
Joshua Wainwright, Piotr Koziol, Sarvesh Sridhar, Harrison Barrans, Daniel Thwaites, Callum Newton, Arnav Jamidar, Harry Turner

Production Methods

We used the Rapid Application Development model during this project, which is a model which prioritises speed and is highly suitable for small projects. The RAD model splits the development into multiple simultaneous tasks, which in our case included finding/creation of assets, documentation and coding. This model involves small iterations made to a base prototype, with new features added or changed based on customer feedback, this close development with the customer allows for quicker development due to less overall changes being made.

We picked RAD due to its compatibility with small projects and the potential development speed advantage compared to other models. The RAD model requires a consistent source of feedback for further iterations and changes to be made, which was possible due to the availability of the customer, our feedback received was limited, however it enabled us to make changes to better fit the requirements we received from our interview.

During development, the team utilised many tools to improve project flow. These included Github for code development, google drive for collaborative project writing, whatsapp for quick communication and google calendar to schedule meetings. We also used a Miro board to quickly present and share ideas.

For project management we discussed using task management tools such as Trello and Github projects. Trello works like a forum for sharing ideas where you can make posts to different categories. GitHub projects are synced directly to the group repository and allows issues to be added to a list which could then be delegated to group members. Both have their positives with us favouring Github projects as it was in our software ecosystem, but ultimately we decided to use neither. Due to the small nature of the group and time span of the project it was easier to keep a dynamic and opportunistic view to the delegation of tasks with Trello providing a solution to where there wasn’t a problem and needlessly increasing bureaucracy. Github projects don't provide much benefit when used by such a small development team and would more likely overcomplicate the process.

Through the use of github, we were able to push and pull changes and collaborate and share our code. Several repositories were created for the game, website and architecture diagrams, each with version histories providing easy access to changes made. Most of the team wasn’t proficient with the methodology of github and how it works. Those who were proficient were able to set up an organisation and the initial set up and explain to the rest of the group how everything worked. The game code was uploaded to a repository where the whole team was able to see all the changes being made.

Google Drive was used as the main method of storing and sharing documents, it has a lot of similar features to github in terms of file sharing but allows for multiple people working on the same document and editing in real time, this provided a lot of utility to the team as multiple sections of a document could be progressed simultaneously.

Google Drive shares a similar purpose to GitHub in the file sharing aspect; however, the ability to work on the same document at the same time provided a significant advantage in its use case. Google Drive also provided a rigorous file and contribution history.

Whatsapp was utilised by our team to quickly convey small bits of information like meeting attendance, task completion, deadlines and general discussion. This improved the workflow of meetings as the whole group was up to date with the completion of tasks and who to expect for extra discussion. The main alternatives discussed were discord and instagram, however not every member had discord and instagram not being a dedicated messaging app meant that whatsapp was the easiest to set up and use for the group.

Task Delegation

In task delegation, we separated the team in half. One side would focus more on the technical side of the architecture and the other side focused more on more creative elements and documentation. The architecture team included Harrison Barrans, Joshua Wainwright and Daniel Thwaites. The creative team included Arnav Jamidar, Sarvesh Sridhar, Callum Newton and Piotr Koziol with Harry Turner managing the progress with both teams, facilitating interteam communication and setting up group meetings. We chose this arrangement as it played to people's strengths and allowed for specialisation in production. For the architecture team the members were more confident with java programming with Joshua having made games before, Daniel having strong experience in other areas of coding, Harrison with commercial coding history. With the creative team, members had greater proficiency with writing and communication. Sarvesh with a history of tutoring in maths was great with articulating project requirements and interlacing them in group discussion seamlessly. Callum played volleyball in the past and utilised these skills to be a team player in writing the joint documents. Arnav with vital experience in web development played a crucial role in developing the group's website, along with crucial contributions in the project's requirements and risk assessment. Piotr with prior market research and product development experience helping create effective interview questions to make the customers wants as clear as possible.

The whole group organised 2-3 weekly group meeting meeting sessions, in these sessions the direction of the project was discussed and new tasks were given out to teams, which then split the work amongst themselves. The 2 teams worked together on some sections, such as those where the technical team would need to provide their input, such as on engine choice, and the creative team providing objectives given in the interview with the customers.
Tasks were delegated and given a time frame in accordance with the deadline and current Gantt chart, this helped the project move forward at a steady pace and stay on course and keep the teams mostly in sync in order to avoid wasted time.

Gantt Chart

To ensure the team was working at a comfortable pace to meet the deadlines of the project, we created a gantt chart which shows key tasks start and end dates and who should be working on them. To keep the team organised and avoid any confusion around task designation, the chart was colour coded into the two teams discussed above.

The group finalised on this Gantt chart as it gave us all sufficient time to finish our designated tasks [LINK]. By week 2 we had met our Milestone of conducting our interviews and choosing our game engine . By week 3 we had met another Milestone of finalising requirements in order to start the Design phase and write more documentation. By week 4 we had met another Milestone of completing our Architecture documentation which enabled us to move onto the Development phase. During week 5 we continued to work on our assigned tasks. During week 6 we again continued on our designated tasks whilst also going through previous work making corrections if necessary. We then shortly met our Milestones of getting both the code and documentation completed. Throughout the project we encountered tasks that took both less and more time than expected during certain moments. In order to keep on track with the Gantt chart, we had to adjust the rate at which we were working and/or help others if we needed to.