Method Selection and Planning

Group 1
Assessment 2

Emma Hogg
Isaac Rhodes
Isioma Offokansi
Toby Meehan
Seungyoon Lee
Ali Yildirim

Method Selection and Planning

Methodology & Tools

Upon the release of the projects on the VLE, we felt it necessary to have a meeting to test the other groups' projects and view their deliverables so we could get a good idea of which project we wanted to continue with before we saw the presentations. After reviewing all the games in our cohort, we believed that choosing team 3's game would provide us with a solid foundation. This belief comes from the Javadoc within the code being clear. We were looking out for good documentation because working with new code becomes less confusing when you have a description on what methods and variables are doing. A large flaw of the project we want continued with is that it crashed a few minutes into the game but we felt as though we could erase the issue with careful and committed changes to the code.
Although we had requirements for assessment 1 and the expected features were written in the product brief, we held a meeting with stakeholders to gain a clearer idea of how they wanted the project to evolve.
While we waited on the feedback from assessment 1 to be released, we looked at the risks from the deliverables from team 3. We developed them further so the risks reflected our teams' concerns and suggested methods to combat them to mitigate the impact.
After developing the risks, we focused on updating the design aspects of the project. The architecture deliverable had a variety of structural and behavioural diagrams as well as a set of CRC cards. We formed new CRC cards for the classes we knew we had to add and used them with the original CRC cards to map them onto a new class diagram that we based future development on.

We used GitHub to host the project and Git for version control. These applications work hand in hand with Git allowing us to push project files to the online repository. This allowed those coding the project to remotely update the project and to revert changes / look back on previous versions if needed. Furthermore, using our chosen methodology was made very simple since we could all push and merge our respective implementations and organise required tasks using the GitHub Projects page. We preferred GitHub over other code repositories, such as BitBucket, because it was more accessible and reliable due to it being free and hosting more users.
We used GitHub Pages to host our website as well as using its domain name. This seemed like a more appropriate option compared to other web development services, like Wix, since we could easily link our implementation and other documents using the GitHub repository.

Our team used a range of IDEs which included:

- ➔ IntelliJ
- ➔ VSCode

The IDE chosen by each individual member was a personal choice. Members chose the IDE they were most comfortable with and the only requirements were that it could clone a Git repository and had adequate version control. We found that using different IDEs had no negative impact on the implementation. This is due to us all using the same language and framework, so if anybody was ever having an issue with the code they could share their work regardless of the IDE they were using.
Overall, we picked LibGDX as the development framework for the game. This is due to us already being familiar with it from assessment 1 and team 3's game already using LibGDX.

For communication we used Discord, an online instant-messaging program. The wide array of features that Discord provides made it more beneficial to us compared to our initial consideration, Whatsapp. This is because Discord servers allow for several

text-based channels, each dedicated to their own subjects (meetings, code, planning, etc.), as well as channels for visual/audio calls instead of the single chat that WhatsApp has as seen on the left side of Figure 1. The text channels helped our organisation and reduced any confusion improving our team work and productivity.
To manage our deliverables and any additional documents we used Google products such as Google Drive and Google Docs. This allowed all of us to work on the documents simultaneously and at any time. We used these products over Microsoft products such as Microsoft Word or OneDrive because the Google products provided an optimal collaborative environment. An example of this would be the majority of our group being confident in using Google Drive due to past experience with university projects and so we greatly preferred it to OneDrive.

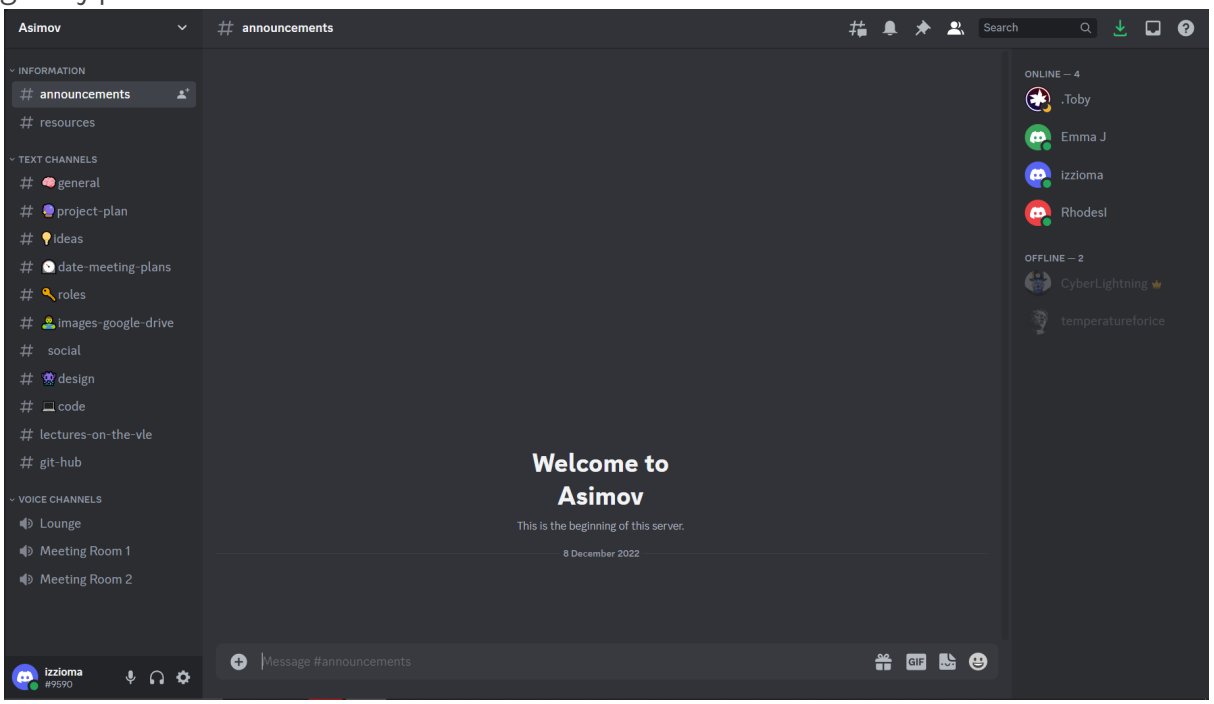

Figure 1 *- Screenshot of our Discord server*

Team Organisation

For this project, we used the SCRUM methodology. The reason we chose this methodology was because we had a strict time frame with a lot of work to do. We believed it was most appropriate to choose a methodology that required frequent feedback cycles and worked optimally even with small groups. The key is the customer's needs and SCRUM enables us to adapt quickly to changes in requirements, and updating the customer on our current progress to provide a more satisfactory end product. Scrum is an agile methodology based on iterative sprints (short periods of time where we'd work on a set number of features). Each week, we would have a sprint and during the weekly meetings, we discussed: our implementation, gave feedback on work, suggested solutions to problems, and added new goals for the next sprint. This can be evidenced by the logbook of our project which can be found on our website.

We felt it crucial to assign roles within our group that aligned with the SCRUM methodology. Isaac was assigned the role of SCRUM Master. They were responsible for: organising the tri-weekly SCRUM meetings (adapted from the daily standup to allow for members to attend to their other commitments) where all team members discussed what had been done since the previous meeting, and ensuring each member knew what task they were responsible for and reviewing tasks from the previous sprint to see what could be improved before each new sprint began. Emma was assigned the role of Product Owner. They were responsible for: guiding members to reflect the customer's requirements in their work and regularly communicating with the customer to clarify and update requirements. The entire team was part of the Development Team. To varying degrees we were responsible for the implementation and deliverables. Each member chose tasks that they felt they were most proficient in.
The role of SCRUM Master and Product Owner were decided by members nominating themselves for the role and, through group discussion, we settled any clashes that arose in nominations. This was done by distributing minor role responsibilities throughout the group so the workload was more balanced.
Using SCRUM made sure we kept on track of the core features and understood what features each member (those working on the implementation) would implement along with whether there were any problems during the previous sprint.

Our SCRUM Master arranged weekly meetings on Wednesdays. They made sure to confirm this day was good for the weekly meetup before we started the holidays. During these meetings, we would check the progress of each member's assigned task to make sure they were on track, and then when the task is completed, assign new tasks for the next meeting.

If members weren't available for the weekly sprint meeting, we would delay it momentarily to accommodate for the highest number of team members before starting the next sprint. If a team member was still not available in those delayed meetings, we would relay information through our Discord group chat and would meet again when they were free if they were struggling on a task. We would also upload a brief summary of the meeting and any new tasks were assigned through Trello. Trello is a kanban-style list-making website. A screenshot of one of our trello boards can be found in figure 2. While Trello didn't directly match with our management style, our team felt it was preferred over similar products that related more directly to our overall management style like Jira. This because it had a very low skill entry level while fully meeting our needs as a task manager. When we were looking into Jira we found it was more technical than Trello and, because we wanted to keep work outside of the implementation simple, we opted for Trello.

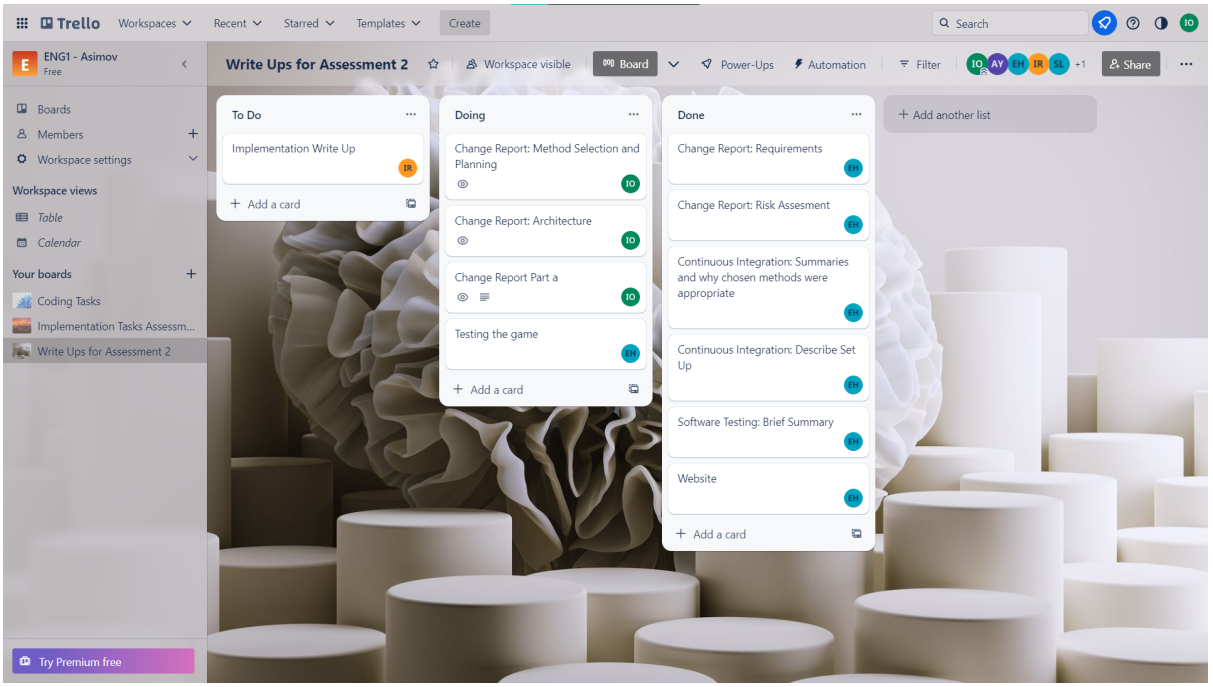

Figure 2 *- Screenshot of our Trello board for the deliverables that needed the write ups*

Planning

The plan evolved as people grasped the requirements of the project and as the project went on. Firstly, we had to confirm that we were comfortable in our assigned roles that are described above when we had to start working on assessment 2, 3rd of March. We split up and assigned tasks. As we started to proceed with testing while waiting for assessment 1 feedback, one of our group members unfortunately had to take a leave of absence.

We already had a course of action ready if such an event were to occur because team 3 had already considered the problem. We re-assigned their work across the remaining five members of our group and communicated with the CUSTOMER/PROFESSOR to let them know what was happening. Following that event we reviewed and altered the risks in the deliverable to tailor to our team's experience on the 8th of March.

An early task that needed to be completed was the meeting with our customer. This would help in narrowing down what was needed for the project. This took place on the 14th of March. The entire team attended the interview. Two members asked the customer the questions we had prepared and the remaining members wrote down the responses. After the interview, we compiled the responses into a concise list of requirements and made our CRC cards.

During the holidays we had regular weekly meetings on Discord. Exact dates can be found in the logbook but meetings were consistently on a Wednesday or Thursday to accommodate team members.

After the break we held meetings throughout the remainder of the term between the 17th and 2nd of April, exact dates can be found in the logbook. During these meetings we smoothed out a number of issues that had cropped up during development, such as the data leak, and tied up the ends of many other pieces of work such as the deliverables and the testing. During this time we also planned, made, and practised the presentation that would be happening on the 3rd of May.

All Gantt charts can be found on our website.