

Method Selection and Planning

Cohort 1 Team 9 - Team 'ARRRGH'

Dominic Hall
Firas Marzouk
Ben Morrison
Harry Whittaker
Amelia Wigglesworth
Zehang Li

4. (a)

After thorough research, we decided upon using 'libGDX' as our library, as it seemed like an easy to learn Java Development Framework. LibGDX has been used to produce some popular indie games such as 'Slay the Spire'. We considered some alternatives such as Lightweight Java Game Library, but thanks to its simplicity and well rounded options to better implement our game with regards to the brief, we decided upon using libGDX.

Using Github, we took our individual tasks and created our own branches separate from the main branch to implement our individual classes and sections of code. Once we completed individual tasks, we were able to push the branches together in order to create the final product. Github desktop allowed us to open each branch in IntelliJ, and view each change within each member of the teams' branch. This meant we could peer assess each branch to ensure there would be no conflict once the pull requests were made.

With Ben's background in graphic design, all the assets in the game have been created by Ben using Adobe Photoshop, allowing for original designs that are tailored to our specific needs.

We initially attempted to split the software engineering methods between all members of the team, with 2 members overlooking all other members' work. In the end, with Harry's extensive research on libGDX, he led the software development with the group's support.

IntelliJ was utilised for software development, as it makes code writing much easier. IntelliJ makes suggestions during use, assists in code completion and has smooth debugging capabilities.

We considered using Visual Studio Code for the software development but upon consideration of the tools available, and its integration with Github, we settled with IntelliJ.

Using tutorials for libGDX, we managed to develop an understanding of how to implement many of the classes we wished to implement. Tutorials allowed us to figure out how to implement sprites for the game, and how to make sprites interact with each other in the way we wished them to. These tutorials also meant we could produce a simple AI for enemy ships, as well as how to create a title screen and an end screen. Producing text for the UI in order to show score and time was also a skill learnt through these tutorials, and we manually had to produce hitboxes on the ships and environments to ensure there was a collision effect where the ship doesn't just float over land.

Using Discord, we also implemented a Github bot that would send a notification to all members of the group when a pull request was made or if any conflicts yet to be resolved were created. This allowed the group to view complete branches and ensure that committing the branch would not create an issue.

Similarly, Github was used for the website, where Dom, Harry and Ben used branches to work on certain implementations within the site. Once all parts of the website were created as we wished, all branches were pulled together to create the final product.

4. (b)

To organise the team, we initially discussed everyone's weaknesses and strengths. In doing so, we discovered talents such as Ben's skill in graphic design, and that Dom knew how to create websites and use the project planner Jira, amongst other skills from all members of the team.

This gave us the opportunity to assign leading roles early on to the members we believed best suited to the tasks, to which they would head a requirement with the assistance of the other members. For a project as big as this, ensuring the right members were leading the requirements they were most talented at meant we could efficiently handle tasks to meet requirements.

We created a shared google drive, under which we could store all documents and resources we would use for the project. With the google drives connected to our university emails, we are able to access the drive from anywhere. We also created a group email, which when used would allow all members to receive emails.

Using Jira, we created a project plan, which allowed us to view a roadmap of our project and view what tasks everyone had to complete. We planned weekly meetings to keep everyone caught up on what we should be accomplishing by the end of the week. For each meeting, we created a rotating schedule for who would chair meetings and who would make notes on the meetings, ensuring everyone had a fair chance to talk on any issues they had to raise. Using the logbook, we recorded all member's attendance for meetings and any issues raised during meetings.

A shared Github was also created for use during the project. This allowed everyone to view changes to the website, as well as make changes to the website themselves. Once the project is completed, the github will allow anyone to view the code used to produce the website.

We use Discord to communicate, as it was the easiest way to get in contact with everyone. The server is split into different text channels for different aspects of the project development. This allows us to raise issues for certain topics in certain channels, making it easier to differentiate.

4. (c)

Initially, we had to plan who would set up all the tools we would use. At the beginning, we had to decide upon the team name, which would be at the lowest priority for all tasks, and what software we would use. We at first decided to focus on the website development, which Dom led, producing the framework and moving on to publishing the final look of the website.

Each deliverable was given a leader for the work who would be responsible for most of the work on it. We began deliverable production from when we began the project to allow for a maximum amount of time to work on this. Amelia led requirements planning, and produced most of the questions for the customer meeting.

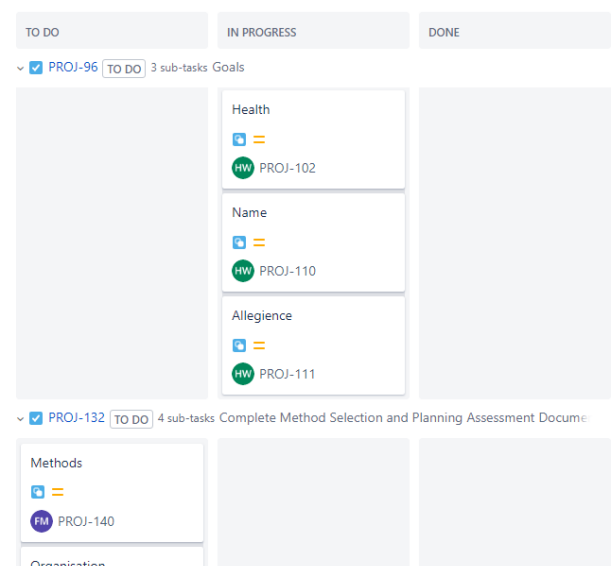
Architecture was spearheaded by Harry, who began research on all the libraries for software development. This involved creating representation of the architecture, including all classes to be used in the product.

Method selection had Firas as lead, and this involved coordinating with Harry on which libraries were researched and for reasoning on why libraries were chosen over others. Method selection also allowed Firas to produce some additional questions for the customer meeting. This task depended on implementation and method selection, as without these being in progress or complete, many things to be written on couldn't be understood.

The risk assessment had to be completed weekly, meaning Harry had to work on this from the beginning of the project in November 2021. With it being a small task, Ben looked over all documents to ensure it was all detailed and up to date with any relevant risks that may have been discovered as the project progressed and appropriate contingency plans.

Implementation of the game itself was the largest task at hand, with the highest priority of all tasks available. As it was such a large task to undertake, small tasks were given to each member of the group, with Harry leading the implementation. We had to ensure there was a basic understanding of coding to use libGDX, and following this have a base code for all to work upon. With a simple ship and lake coded in, all members were able to work on their subtasks independently within their own branches. With this being worked on from the end of winter break, and a finishing aim of 31st January for finalisation, this task was completed in ample time.

Before all of the implementation began, Ben designed the logo for the team, as well as pirate logos for the website and the basic assets for the game itself. This involved creating the ships, land and water design as well and piers for areas of land within the lake. Implementation depended on this, as without basic sprites for the game we wouldn't have anything to look at when producing the code.



Attached below is the Gantt chart for the project, where tasks were started upon our second meeting in Week 9. We allowed all members of the team to have plenty of time to complete all the deliverables, to ensure any questions could be answered if there was a need.

Whilst we initially planned to start implementing at the beginning of winter break, we thought it best to allow time for research and a break from university work. At first, Firas and Logan were put at lead for software development, but upon Harry's thorough research into libraries and tutorials, he was able to lead the group in regards to the development of the game.

To the right is an example of how our weekly plan would look like, with all members being assigned a task to complete when the sprint ends. All members of the team are able to move tasks from to do, to in progress, to done, ensuring that all members are aware of what has been accomplished.

Such a system allows members to see what is of high importance for completion and what is of low importance, in turn letting the group prioritise some tasks over others.

The team worked with agile software development in mind, ensuring that the team worked together flexibly and played upon each other's strengths to complete the task efficiently.

