

Team Plan

Team NASA

Team skills and interests

All team members have knowledge in Java. We also have some members that have skills in other programming languages, such as JavaScript or Python, but we all agreed to stick to Java as the primary language.

Roles and responsibilities

For now, we've chosen the following roles in our team:

Team leader: Jonas Triki

Git repository leader: Jonas Mossin Wagle

Generalized roles:

- Jonas Triki: libGDX, database
- Jonas Wagle: AI
- Jonas Trædal: AI
- Marianne: Piece logic, documentation
- Stian: Database
- Sofia: libGDX
- Paraneetharan: Documentation
- Eirin: Piece logic, bug fixes, tests
- Elise: PowerPoint

Organizing the Git repository

We have decided to organize the Git repository in two branches: One for development (dev) and one for production (master). Each folder in the respective branch is going to represent a section of the application as a whole. For instance, we have one folder for documents called “docs” and one folder called “summaries” to keep track of all the meeting minutes. We have agreed to keep commit messages simple and clean, describing what changes has been made.

The organization of the repository was mainly planned in the first iteration of the programming project (obligatory compulsory 2). We decided to use the “docs” folder to keep track of all the .pdf/.tex files. Tasks have been distributed over all the team members to make sure we do not work on the same thing, and during the weekly meeting we discuss what we have done and if we need to make any changes. Additionally, we

agreed to take an agile approach as to how to organize the Git repo. We planned on using libGDX and a separate folder for the Java projects.

As of the current iteration (obligatory compulsory 4), this has been accomplished. We feel that there is no room for improvement regarding the repository's folder structure at the moment; none of the members have any problem with the current structure.

Risk analysis

What could go wrong?

- Team members can “fall off”
- Team members may be inconsistent with delegated work tasks
- Internal conflicts
- Imprecise time management
- Bad planning

Addressing issues

How should the team react if it goes wrong, and if so, how to reduce the effect it has on the project?

- Addressing issues as soon as possible. Do not let things lie.
- Make sure everyone is on the same line and that everyone does what they are supposed to do.
- Things usually take more time than expected; don't set unrealistic deadlines.

Ensuring efficiency

How can the team become more effective, and utilize the human resources as well as possible?

- Break down big tasks before delegating them. This will give us a better overview of what we need to do, and what challenges may arise.
- Be more clear of what has been done, as well as what issues we have encountered. For instance, tell members in Slack if there are any bugs in the last commits.
- Delegate tasks based on previous experience; for instance, if someone is really good with SQL, they should be handling the database.