Team plan

Roles and responsibility

Leader: Eivind Dagsland Halderaker

Scrum-master. Responsible for delegating tasks and making sure that the development team learn to self-organize.

Git repo: Simon Riple and Helge Mikael Landro

Responsible for merging master and develop branches each monday. Be able to help with possible git problems or answer questions the other team members might have.

Tools: Håkon Ettestøl Osland

Try to have a total overview over the different tools we are using and making sure that all other team members understands how to use the different tools.

Skills and interest

Front end

Helge Mikael Landro

Simon Telle Riple

Amund Lindstøl

Back end

Eivind Dagsland Halderaker

Agnete Gridseth Røstad

Alise Haukenes

Håkon Ettestøl Osland

Magnus Flatheim Jensen

House rules

- All team members can have preferences to what they want to be working on, but are also flexible to helping other team members with other tasks if necessary.
- If one of the team members are struggling with a task, he or she should be able to ask the other team members for help, so that the task in hand will be finished on time.
- All team members are required to meet on time. If prevented from meeting on scheduled time, you must inform at least one other team member who is going to be at the meeting.
  - Valid reasons for not showing will be accepted.

Repo structure

- The project will consist of a Master branch, and a DEV branch. Every week a functional DEV branch will be merged into the Master branch.
- The file structure will separate the source files for the project and the Document files. The libGDX framework consists of the folders 'core', 'desktop', 'gradle/wrapper' and 'html'.
  - core/assets: This folder contains the assets for the game, including the files for storing information about the game and players, and the graphics files
  - core/src: The folder for all the source files
  - core/out: The output files for the project

Risk Analysis

- Uneven workloads
  - Different work schedules, skill level or involvement in the project can lead to uneven workloads. This is a big challenge in any team project. If some of the group members does not show the same effort as the rest, that will lower the team spirit.
  - Measure: Delegating the work-tasks when everybody is present will balance the workload. The group members also have to be engaged when the tasks are handed out. Since we will be using Scrum as our project process we can also play scrum-poker to agree on the size of the task, and then delegate them evenly.
- Double work
  - Because we have a tight schedule for finishing the deadlines we cannot waste time with having multiple people doing the same thing. If the tasks are not organized well enough this will be a problem.
  - Measure: Communication and clearly defined tasks are key to not having two persons doing the same thing. We will organize our tasks with git-issues. In a git-issue we will clearly define what should be done in the

task and who is responsible of the tasks. We should also agree on when the task should be finished. When the task in finished, it will be placed in a column for finished tasks.

- Disease, withdrawal from course
  - There is always a risk that some team members will be sick or withdraw from the project. If a person has more knowledge about the tools or code that we produce than the other team members, this could be a big problem for the team.
  - Measure: To keep the project from being too reliable on a few persons we will try to exchange knowledge about the tools we use, and what our code does. We will also write readable code and documentation such that it is easy to read.
- Bad time estimates
  - One of our biggest challenges is to estimate the time of tasks. This is because we do not have so much experience with a big project. Bad time estimates can be critical in projects, because the workload can be bigger closer to the deadlines.
  - Measure: We think that the key to finishing our project is to start the work as early in the process as possible.
- Bad project structure
  - Bad project structure can be a mess. This can be a problem because the team will spend much time on searching for the documents they need.
  - Measure: The group do not have much experience with using git in a project. Therefore, our git-responsible will keep a git-course so that everyone will get an introduction to the basics of working with git. We will also keep a good structure in the repository, by actively discussing how we want it to be.
- Personal conflicts
  - Personal conflict can cause team members to have low motivation, and the team could lose their team spirit.
  - Measure: To avoid personal conflicts, we will be nice persons and act with common sense. If, however, a problem occurs we have agreed that we will try to solve the problem with the persons involved first.