Summary
5/4-2018

Attendance:
Eirik Sognnes
Marianne Luengo Fuglestad
Paraneetharan Sabaratnam
Stian Fagerli
Jonas Triki
Jonas Trædal
Jonas Mossin Wagle

Written by: Jonas Mossin Wagle

Objective of meeting and what has been discussed:

- **Exercise 1**
    - How to become more efficient
        - Be more clear of what's done
        - Delegate tasks base on previous experience
    - Improve planning
        - Last iteration, we made sure explicitly not to duplicate work, and plan to keep it like this.
        - Once we have broken tasks into smaller tasks it is easier to estimate effort of the respected task
    - Make the git repo (even) easier to navigate in
        - Discussed this and the group felt that this is a point we could not further improve.
    - Add more functionality to the application and make it more robust
        - Expan already written code.
        - Write more tests
        - Implement new functionality
        - Refactor code
    - Delegation for updating documents
        - Paraneetharan and Marianne volunteered
- **Exercise 2**
    - - Support multiplayer, i.e. a human player must be able to play against another human.
        - Already done.
    - - Offer an advanced level AI player.
        - Jonas Trædal and Jonas Wagle volunteered to cooperate for this task.

- - Store ranking (winner statistics) in a database. Use a free SQL database engine like SQLite, or an online database service like https://mlab.com/, Heroku Postgres or Amazon RDS.
    - Jonas Triki and Stian Fagerli volunteered
- - Offer help for novice users to select best move, start the game at a random chess state, or other functionality that your targeted audience will appreciate.
    - Everyone should contribute to this
    - Decided to at least implement the following:
        - Hinting
        - AI computation on seperate thread
        - Better high score menu
        - More choices when promoting piece.
        - Better win-loss screen
        - Support more chess game types. For instance blitz, bullet, etc..

In addition, the following non-functional requirements should be satisfied:

- - Offer sound and/or animation effects that enhance the gaming experience.
    - Animation on AI move, to make it more clear for player.
    - Mark where AI moved from
    - Sound effects on move
        - Have an option to turn off sound
- - All images and other graphics, source code and build script used must have an open license.
- - The Java code shall be documented according to best practices using JavaDoc
    - Those who haven't already documented their code with javadoc, will do this after the meeting
- - The advanced level machine player must make a move within 3 seconds.
    - Jonas Trædal and Jonas Wagle volunteered

We have not yet received feedback from exercise 3, and will therefore do not have the possibility to elaborate over the feedback until the next meeting.