Lessons Learned

During the development of the project there were some minor and major hurdles we had to conquer in order to arrive at a working project. In the following are the most important lessons we have learned.

Management of GitHub project

It is no question that using GitHub for the development of the project was the correct decision. However, the management of different components was not an easy task at first.
Since everybody has their own coding preferences we first had to come up with a standard on how to implement new features/additions for the game. We set up a README.md file that explained how one should go about creating a new feature for the game. We used the possibility of creating new branches, which would not affect the master branch. After the successful implementation of the feature and adequate documentation the branch could be merged back into the master branch. However, in some cases this might cause merge conflicts if somebody else modified the same block of code. These conflicts had then to be resolved after merging. Therefore we also adopted a coding philosophy of trying to extend the code at all times, rather than editing existing code.
Furthermore, after the successful implementation of a feature a pull request had to be done for merging the branch into the master branch. This pull request then required the successful review of at least two other group members. When accepting the pull request the reviewers had to look through the made changes and pay attention to sufficient documentation of the code.

Importance of scrum meetings

During the development of the project one of the most important aspects of the project were the daily scrums the team had. In this way everybody could be up-to-date with the status of the different game features.

Use of proper diagram formats

It sometimes occurred that a team member used an own way of displaying a certain diagram even though it could be presented by the means of an UML diagram.

Proper time planning

Quite often we underestimated the time required to implement a feature. This was partially due to merging the branches and resolving the conflicts.

Coherent task distribution/Importance of communication

At some points in time when implementing a new feature not enough guidelines existed on how the feature should exactly be/act. This would then cause problems in case of successful implementation of the feature only to find out later by reviewing it with other team members that the feature should have worked differently.