# Sprint Retrospective, Iteration #1

| Task | Assigned to | Estimated effort (in hours) | Actual effort (in hours) | Done (Y/N) | Notes |
|---|---|---|---|---|---|
| Set up project structure | Nathan | 3 | 3 | Y | Created the main classes of the game logic |
| Get a Heroku backend running | Mac | 1 | 3 | N | We figured out this wasn't necessary for the project. |
| Set up SQLite database | Timea | 2 | 2 | Y | |
| Create games table (id, username, alias, timestamp, score) | Timea | 1 | 0.5 | Y | Took less time, because I found a good explanation of how this is done online |
| Create users table( username, password) | Timea | 0.5 | 1/6 | Y | This task was very similar to the previous one, so because I've done it once already, it took less time |
| Create a client side login page | Jasmine | 2 | 2 | Y | FXML file with the necessary fields to login |
| Initiate Client-server connection | Mac | 4 | 0 | N | Not started because this was not necessary. |
| Set up user authentication (using salted hash passwords) | Pepijn | 2 | 4 | Y | This took longer than expected because we had to figure out how to configure the database correctly. |
| Set up user registration | Pepijn | 2 | 2 | Y | |
| Set up JavaFX | Jasmine | 1 | 1 | Y | Added the required plugins in the build.gradle file. |
| Create a client side account page | Jasmine | 3 | 2 | Y | |
| Create a client side login page | Jasmine | 2 | 2 | Y | |
| Create a client side menu page | Jasmine | 1 | 0.5 | Y | FXML file with the buttons to switch screens |

| | | | | | |
|---|---|---|---|---|---|
| Create a client side main page | Jasmine | 1 | 0.5 | Y | FXML file where you have can either switch to the login or to the register screen |
| Create main class for the project | Jasmine | 2 | 2 | Y | |
| Define abstract class SpaceEntity | Mac | 2 | 2 | Y | |
| Create basic asteroid logic | Mac | 2 | 1 | Y | Most of this was changed later but I established a backbone |
| Create an isColliding method | Jasmine | 1 | 2 | Y | This method checks if the objects are overlapping on the screen |
| Space Entity movement | Mac & Nathan | 8 | 8 | Y | Most of this time was spent trying to think out the theory, while the implementation was actually much simpler. |
| Create game screen | Jasmine | 2 | 5 | Y | |
| Bullet movement | Pepijn | 1 | 1 | Y | The bullet moves in the direction the player is facing |
| Player ship laser | Pepijn | 2 | 2 | Y | |
| Player ship rotation | Mac | 1.5 | 1 | Y | |
| Player ship thrusting | Mac | 2 | 1 | Y | This was actually easier than expected |
| Timing mechanism | Nathan | 6 | 4 | Y | Created basic AnimationTimer |
| Make UML diagram + use case descriptions | Timea | 1 | 2 | Y | Took more time because I had to change some things after feedback. |

# Main problems encountered

## Problem 1

**Description:** Not all code was tested while it was being implemented, this lead to testing problems close to release.
**Reaction:** We tried to mock the Objects we had trouble with during the testing but this didn't work so we pushed the testing of some classes to the next sprint.

## Problem 2

**Description:** FXML files can be tricky to combine with the actual code. This happened when creating the Game Screen as we couldn't add the player, asteroids and bullets to the screen because they were generated with the code.
**Reaction:** To solve this problem we decided to create the Game Screen exclusively with the code. This made the objects show up on the screen and all the methods implemented are working properly.

## Problem 3

**Description:** We didn't plan enough work for 2 weeks, I (Timea), and I (Nathan), feel like we didn't contribute enough in this Sprint.
**Reaction:** We added some extra issues to the backlog during the sprint, and will divide the tasks better so that next week, everyone has a specific part of the code they are responsible for.

# Adjustments for the next Sprint Plan

- Testing along with the implementation so we won't run into testing issues last minute.
- Add a lot more issues to the sprint backlog so we have (more than) enough to work on during the sprint.
- Spread the work evenly and try to assign tasks that are not overlapping to everyone