Johan’s diary.

9/2 - 2 hours: Wrote group rules together with Axel and Pavlos.

11/2 - 2 hours: I researched JSON-files and JSON parsing, since we will have to use these methods in our project. I looked at wikipedia and some informative youtube videos.

16/2 - 30 minutes: Me, Axel and Pavlos had a meeting with our TA, where we received some advice and general instructions.

16/2 - 1 hour: I researched how to parse JSON-files with Axel, and we managed to install the Bytestring package via Cabal.

Since it was necessary in order to parse with Aeson. We had a look at these sites:

https://www.schoolofhaskell.com/school/starting-with-haskell/libraries-and-frameworks/text-manipulation/json

https://hackage.haskell.org/package/aeson-1.5.5.1/docs/Data-Aeson.html

17/2 - 3 hours: I followed the aforementioned guide on schoolofhaskell.com. It gave me a better understanding on how to parse files from the internet. Although, I could not get the FromJSON and ToJSON instances to properly work with the Steam API.

So, I started working on the report, its introduction part.

18/2 - 2 hours: I sat in a call with Pavlos and Axel, where we discussed our progress on the project. Additionally, me and Axel assisted Pavlos getting his libraries to work, which we did after a little while.

We also discussed our findings on how to work with the Aeson package and the Generic package. We also experimented with it, in order to obtain a better understanding.

19/2 - 2 hours: I read up on how to parse JSON-files with the Aeson package. I obtained some useful information on one guide, and I applied some of that new knowledge to my Haskell file.

20/2 - 2 hours: I continued to work with Aeson. I wrote some data types for our project. The data types had to resemble Steam's JSON-file, so I made sure it was correctly done.

Three data types in total had to be written.

22/2 - 2 hours: I examined Axel's progress on the JSON instances and his successful attempt at parsing the JSON-file. After that I began working on a function that returns the common games, represented by a list of strings.

23/2 - 3 hours: I continued to work on my list function. I came up with two possible solutions, the first compares two lists at a time and the second counts how many times a game occurrs in a concated list of owned games.

I ended up getting stuck on the first solution so I moved on to the second one. I got pretty far on it, but I got stuck on the final function.

24/2 - 3 hours: I asked for advice from Pavlos and Axel with the function I was working on. Together we could write a small and efficient function that does its job great! Pavlos wrote a function that utilized the intersection function in Data.List.

It proved to be exactly what we were looking for, so it was implemented in Main.hs. Later in the day, I worked on the rapport.

25/2 - 3 hours: I examined how far Axel and Pavlos had gotten on the project. I read through their code in detail in order to fully grasp their implementation. I later began to describe the program in the report.

26/2 - 5 hours: I worked on the report and had a look at the previous student's works in order to get a good idea for the structure of it. Later, we had our meeting with the TA, where he gave some suggestions on how to make our program more comprehensive.

Directly afterwards, we held a meeting discussing which new functions we should try and implement and how to divide the tasks between us. Then I began working on my assigned task. Which is creating a text file containing the common games. This is going well so far.
I used writeFile.
27/2 - 4 hours: I looked through Axel's new contributions and decided to expand upon them. I fixed some bugs associated with retrieving Steam usernames and I began to implement a way to print out the users together with the common games. I worked on that for quite a while,
then I did some work on the report.
28/2 - 5 hours: I finished the implementation that prints the usernames of the users. Afterwards I fixed some bugs that occurred when obtaining the names of users with certain settings set on private. Then I worked on the report, I wrote on the structure section. After that
I made a flow chart that illustrates how the functions communicate with each other. Lastly, I wrote some function specifications in Main.hs.
1/3 - 7 hours: I wrote a bunch on the report, specifically the section that describes how the program works in detail. Then, I wrote function specifications and checked with Axel how he was progressing on the playtime function.
Afterwards, the group held a meeting in which we all updated each other and discussed which tasks remain. We talked about how to write test cases and the report. Axel showed us his progress and me and Pavlos tried
to give some advice. I wrote a to-do list for the group. After that I worked on the test cases and finished the function specifications. Later, I had a look at Axel's new features and wrote on the report.
2/3 - 7 hours: I began with examining Axel's new functions. Then I described the functions and how they interact in the report. After that I began writing the discussion section on the report and some additional test cases in main.hs and JSONParsing.hs. I also updated the flowchart
with the aid of Axel. Afterwards, we had our meeting with the TA, where he told us how we should hand everything in and how to prepare the oral presentation. The group held a meeting after that, where we discussed how to proceed and what had to be done. After that was done, I took a break and later
wrote some more test cases and proofread and wrote a bit on the report.
3/3 - 4 hours: The group discussed what was left to be done and we decided on who should do what. During the day, I finished the report, wrote all the sections that were missing and cleaned it a bit. After that I helped Axel with writing the test cases and they were also finished.
The HUnit tests worked. After that I fixed the footnotes with Pavlos and added references, and we fixed some headlines as well. We also included the code references.
Since we are supposed to give clear credit when we have read about code on the internet.
After that, we checked that everything was right and we handed everything in on Studium.