Sprint Retrospective	Iteration	3

Project Group	90

User Story	Task	Task Assigned to	Estimated Effort (in hours per person)	Actual Effort (in hours per person)	Done	Notes
Game rules	Create match state and refactor Matchscreen	Tim	5	5	Yes	
-	Do design patterns & software architecture tasks for assignment 3	Tim, Eddie	3	5	Yes	
Game rules	Implement turns	Eddie	3	4	Yes	
Game rules	Implement fouls	Eddie	3	4	Yes	
Postgame	Show leaderboard	Victor	5	5	Yes	
Postgame	Ask user to save his score	Jesse	2	2	Yes	
Postgame	Show Win/Lose message	Jesse	2	2	Yes	
Game rules	Assign ball type to players	Maikel	3	3	Yes	
-	Implement mutation testing	Victor	8	3	No	Other things had more priority
UX	Show extra elements during a match	Jesse	5	1	No	Didnt have enough time because of other tasks, especially sounds took more time than expected
Game rules	End game when black ball is potted(win/lose)	Maikel -> Tim	5	5	Yes	In the end Tim did this, because it matched better with his other tasks. Maikel and Tim swapped this one and 'Show power meter of hit'
UX	Implement sound effects	Jesse	3	7	Yes	
Game rules	Time limit per turn	Eddie	2	0	No	No time because of busy schedule and assignment
Menu	Implement in-game menu	Eddie	2	0	No	No time because of busy schedule and assignment
Game rules	Allow user to place the white ball	Maikel	5	10	Yes	Not merged yet
	Show power meter of hit	Tim -> Maikel	3	0	No	Maikel and Tim swapped this one and 'End game when black ball is potted', because it turned out to match better with their other tasks. Unfortunately, Maikel did not have enough time to work on this
Login	Logout button		1	0.5	Yes	We didn't plan this one in, but it was really easy to do (<10 lines of code)

Main Problems Encountered

Problem 1 — Merges to master

Description — We accidentally merged a feature branch to the master branch 2 times. However, we don't know if this is our own fault, because we remembered checking this and selecting dev when creating the request. Gitlab was down for a short time, and it could be resetted/didnt save this setting. However, of course it is still our own fault. We fixed it by reverting the merge and creating a new merge requests for dev. Even though 1 was only 2 lines of code, we just wanted to adhere the pull-based development model, as we did during the whole project

Reaction — Check the target branch when creating the merge requests, reviewing one of others and just before merging. We probably won't have a lot of extra merges for this project, but this is definitely something to remember for future projects. Also if we were the owner of the repo we could have made dev the default branch.

Problem 2 — Little practice for presentation

Description — We started on Thursday with dividing the tasks and defining the structure. Then everyone did his own part, practised and time it. In the end our total presentation was 19,5 minutes, which is actualy perfect (required = 20 +- 1 min.). However, because we didn't practice that much and not together at all, the transitions were a bit weird and the practice presentation was not as good as it could have been.

Reaction — Practice more for the final presentation, and also a few times together. We already agreed upon 2 times to practice, and will probably add more depending on how it went.

Adjustments for next Sprint

Description

Task 1 — Triple-check target branch of merge requests (see problem reaction for more comprehensive description)

Task 2 — Practice more for final presenation, together (see problem reaction for more comprehensive description)

Reflection: last week

	Description	Reflection
Task 1	Write down time spent during sprint	Some people did it on gitlab, some did this for themselves on paper. Worked well, everyone remebered how long they spent
Task 2	Double check assignment before handing in	We triple-checked the assignment, even found some mistakes, but finally uploaded the final version. There shouldn't be mistakes in there (fingers crossed ;)