

Sprint Retrospective

Iteration1

Project Group

90

For this sprint, we did not distinguish between user stories and tasks. This is because we did not have the system requirements yet and also since we did not have a good idea how libgdx is structured/works, or how exactly to make the game and split it up into very specific tasks. We just made a small product backlog on the gitlab issueboard, ordered on priority and took the most important ones. Although we think this was fine for the first sprint, for the next sprint we will split up user stories in smaller tasks, as described in the adjustments

User Story/Task	Task Assigned to	Estimated Effort (in hours)	Actual Effort	Done	Notes
Agree upon first libraries to use	Everyone	1	1	Yes	
Set up libgdx	Victor, Maikel	2	5	Yes	
Custom CI/CD	Victor	3	3	Yes	
System requirements	Eduard, Tim	5	5	Yes	
Implement ball physics	Maikel	8	7	As good as	Not fully merged since needs to be reviewed
Let the user interact with game elements	Jesse, Tim	8	2	No	Not possible since game elements not displayed/created yet , and needed to spend much time on assignment 1
Show game elements	Jesse (.Victor, Eduard)	5	7	No	Only walls done, took a long time to understand and learn libgdx

Main Problems Encountered

**Problem 1** Unfinished tasks  
Too many tasks, the assignment and project organization took a lot of time, but still we can't forget really coding the game.  
More focus/time on coding needs to be spend. Everyone should spend more time in general and we decided to have 'weekly coding meeting' on Tuesday so everyone works on the project

**Problem 2** Large tasks  
Some tasks were quite large/did not have clear definition of done  
Although it was part of a conscious choice as explained above, for the next sprint we should have smaller tasks, with a clear definition of done. We should first pick some user stories of the product backlog, then split them up into smaller tasks and finally divide him. Something which will probably make this easier is first doing the planning on paper/in document and only then transform it to issues. This way we will have a better overview of the whole pictures. To display the user stories on gitlab, we can use labels/epics.

**Problem 3** Last-minute merges  
In the last few hours before the deadline we had to do a lot of merges to have some new code for the assignment release  
Next time create and handle merge requests a lot earlier. When someone has some small feature working, he should immediately make a merge request and say this in the group so 2 people can review it. Retrospective is okay to do on Friday afternoon, but if a lot of problems/merges to be done it might be an idea to meet in the Thursday lab instead of just Friday afternoon

Adjustments for next Sprint Plan

	Description
Task 1	Spend more time on coding and have 'weekly coding meeting' on Tuesday
Task 2	Make smaller tasks, first make sprint planning on paper/document, use labels/epics for user stories on gitlab
Task 3	Create and handle merge request way earlier than last day, possibly meet in Thursday lab

Reflection: last week adjustments

NOT APPLICABLE, FIRST SPRINT SO NO ADJUSTMENTS FROM LAST WEEK