Sprint Retrospective, Iteration #3

Task	Assigned to	Estimated effort (in hours)	Actual effort (in hours)	Done (Y/N)	Notes
Design Patterns (assignment) Template Pattern	Timea Pepijn	8 2	8 2	Y	Initially wanted to do the Iterator pattern, but then realized it wasn't suitable for us.
Software Architecture	Jasmine	8	8	Y	Refactored class structure to match the chosen architecture
Populate LeaderBoard Screen Table	Jasmine	2	2	Y	
Implement levels	Timea	8		N	Decided that we had to prioritize other issues.
The game shall play sound effects when certain events happen	Pepijn	5	4	Y	Done by Mac instead of Pepijn
The game shall play music while in progress	Mac	5	6	Y	
Implement enemy spaceships functionality	Nathan	12	11	N	Didn't finish because this was not tested yet.
End of the game (basic)	Pepijn Jasmine	3 1	3 1	Y	Jasmine also helped a lot with this
Fix game table standard data bug	Pepijn	2	2	Y	
Implement max amount of login attempts	Pepijn	5	4	Y	
The game shall have a moving background	Mac	5	2	N	Decided we had to prioritize other issues.
Fix pause screen bug	Jasmine Pepijn	2 2	2 2	Y	Took longer than expected due to issues with FXML not being able to perform certain actions on screen creation.
Teleport time-out	Mac Pepijn	2	2	Y	
Add tests based on PIT mutation score Bullet	Pepijn	2	2	Y	The setup of PIT took more than expected due to a

Database Asteroids (childclasses) Hostiles Asteroid	Timea Mac Nathan Jasmine	2 4 4			problem with the mapping of the classes.
Kill Asteroids when they fly off the screen and when the player dies.	Timea	1	1	Y	We ended up not removing the asteroids when they fly off the screen because after trying it out we decided it didn't improve the gameplay. We decided to also remove them when the player loses a life.

Reflection adjustments made for this sprint

- More focus on testing, especially Mutation and Integration testing.
    - Setting up the mutation testing tool took a lot longer than expected. This delayed the focus on mutation testing a bit but each of us took a class with a relatively low mutation score and added tests to improve the score. More documentation on the mutation tests can be found in the doc/mutation folder on our repo
- We need to pay extra attention to test coverage before pushing because the pipeline will fail if the coverage is too low.
    - We configured the pipeline to fail if test coverage (line) for a class was below 50%. The pipeline only failed twice because of this and, after that, we added tests relatively quickly.
- Start implementing key issues earlier so bugs will be discovered sooner (and won't end up in the release).
    - We tried to do this by assigning correct weight to all issues but this was still difficult because the key issues tend to make more time so they were finished later than expected, and thus some bugs were still discovered quite late.

Main problems encountered

There were issues in the backlog that had low priority and there were issues not present that had higher priority in the MoSCoW.
Mutation testing revealed to be tricky to implement.

Adjustments for the next Sprint Plan

- Next time check better the missing issues based on priority
- Refactor and clean up the code

- Debug the code so that bugs don’t get discovered before release