Sprint Retrospective, Iteration #2

Task	Assigned to	Estimated effort (in hours)	Actual effort (in hours)	Done (Y/N)	Notes
Default database content	Pepijn	4	4	Y	
End of game (basics)	Pepijn	3	1	N	Partly finished, Other tasks took a bit more time than expected and were finished close to the deadline so there was no time left to finish this.
Implement Game timer	Jasmine	5		Y	
Create client-side Pause screen	Jasmine	3	4	Y	
Populate LeaderBoard table	Jasmine	1	2	N	
Implementing pointing system	Jasmine	3	3	Y	
Implement Asteroid types	Nathan	3	2	Y	
Good class structure	Nathan	4	10	Y	
Display different Asteroids	Nathan	5	4	Y	
Apply design patterns to the class structure	Timea	5	3	N	Did not finish it because I spent the remaining time on testing instead, as that was more important for this sprint.
The player can wrap around the screen	Timea	3	2	Y	I thought that it will take more time because I'd have to spend time figuring out the structure and inner workings of the Player, but by the time I got to this task, I figured it out.
KeyListener for extra commands	Timea	3	2.5	Y	
Write function to get top 5 highscores	Timea	0.5	1	Y	
Implement player's lives	Timea	5.5	6.5	Y	

Modelling class diagrams	Jasmine	3	4	Y	
Modelling sequence diagrams	Jasmine	2	3	Y	
Implement tests for all testable classes	Marijn	10	11	Y	
Asteroid movement	Pepijn, Nathan	4	4	Y	
Bullet improvements	Pepijn	3	4	Y	
Test database	Timea	3	3	Y	
Implement Jacoco on the pipeline	Marijn	4	3	Y	

Reflection adjustments made for this sprint

- Testing along with the implementation so we won't run into testing issues last minute.
    - We tested (almost) everything during the implementation so our test coverage is way better than it was.
- Add a lot more issues to the sprint backlog so we have (more than) enough to work on during the sprint.
    - Everyone had enough issues to work on this sprint.
- Spread the work evenly and try to assign tasks that are not overlapping to everyone
    - We had almost no issues with overlapping work.

Main problems encountered

We couldn't test the JavaFX view objects, so we restructured the main object structure to remove the view from the constructor, so that testing was possible.
Asteroid class has a random generator, so the Unit tests are not 100% fool proof.

Adjustments for the next Sprint Plan

More focus on testing, especially Mutation and Integration testing.
We need to pay extra attention to test coverage before pushing because the pipeline will fail if the coverage is too low.
Start implementing key issues earlier so bugs will be discovered sooner (and won't end up in the release).