Reflection on Sprint #1

Game Bubble Trouble
Group 24

User Story	Task	Assigned to	Effort	Actual hrs	Done	Notes
Exercise 1	Design classes from requirements	Martin	High	2hr	Y	
	Compare to our game	Hung	Medium	3hr	Y	
	Draw class and sequence diagram	Hung	Medium	1hr	Y	
Exercise 2	Answer questions about UML	Ruben	Low	2hr	Y	
	Create UML class diagrams	Ruben	High	1hr	Y	
Exercise 3	Set up requirements for logger	Christiaan	Medium	1hr	Y	
	Design logger from requirements	Chris	Medium	1hr	Y	
	Implement logger	Chris	High	6hr	Y	Also implemented events
The user is able to receive power-ups from bubbles	Implement power-ups	Ruben	High	0hr	N	Useful function for next sprint
The user is able to enjoy music while gaming and receives audible feedback from game events	Implement audio in game	Christiaan	Medium	0hr	N	Another sprint
The user can play the game without errors or failures	Write extra tests	Martin	High	2hr	Y	Around 40-50% tested, better TDD next time
	Fix bubbles (stuck beneath floor, wrong splitting)	Hung	Low	4-5hr	Y	Also fixed collisions, higher effort than estimated

Main Problems Encountered

Problem: Testing
Description: We still have a reactionary testing strategy; classes are written quickly and tests are written by one person who first has to understand the code before being able to test well.

Reaction: The testing coverage is still relatively low (40-50%) and there's still one person creating tests.

Problem: Time estimation
Description: Some tasks were given a 'low effort' estimation, but turned out to take longer than expected. This is because it was decided the smaller problem could best be fixed by a bigger overhaul of the system.

Reaction: More time was spent on the lower effort tasks than was planned.

Adjustments for the next Sprint

To fix the testing problem, we want to focus on using a Test Driven Development style, where we first write the test and then write our code. This will also help improve our code on other areas.

To fix the time estimation problem, we want to look out for problems that might need bigger changes in the overall code. Discuss how a person sees his fix working to estimate this bigger change before estimating time.