Sprint Retrospective

Iteration #3 TEMPLATE

Project	Laser reflection
Group	62

User Story	Task	Task Assigned to	Estimated Effort (in hours per person)	Actual Effort	Done	Notes
As a player, I would like my progress to be saved.	Create methods to create read and update user score in the database. Implement score updating when the level is won.	Liudas	1h	1h	Yes	The first implementation won't work with replaying levels. If we choose to add the functionality of replaying already won levels. The implementation will have to be changed to only increment the score when an unplayed level is won
As a player, when I hit the goal I would like to win the level	Change play screen code to redirect the player to the menu screen when the level is won. Increment the player's score in the database.	Liudas	1h	1h	Yes	This can be later changed to redirect player to the next level automatically, depends on what we decide
As a player, I would like to be able to choose which level I want to play.	Implement a level select screen.	Selim	3h	1h	Yes	The menu is a simple menu with buttons displayed on the menu. These buttons represent different levels which, when pressed upon, a player can play. At the moment, all levels are available to all players. For the future we might be able to grey out certain levels depending on the score of the player.
As a player, I would like to be able to see a scoreboard with top 10 players with highest scores	Create a leader board button in the menu. When the button is pressed the top 10 players and their scores should be retrieved and displayed in a dialog with leader board gui.	Liudas	2h	2h	Yes	There were problems with the first version of the implementation not showing the players in descending order, but was fixed in the second version.
As a player, I want to be able to suspend the game and resume it.	Implement a settings menu.	Selim	2h	2h	No	The settings menu is a menu which, while in play, shows up on top of the playscreen when pressing the ESCAPE button. It has three buttons: "Resume", "Settings" and "Exit". " Resume" and "Exit" are self explanatory, the "Settings" is a placeholder for features that might be added in the future. The current implementation varies with our master branch, so it is not in use at the moment. We will have to slightly refactor it to make it fit.
As a player, I would like a pleasant visual experience while playing the game.	Improve design of laser and mirrors	Selim	2h	30m	No	This sprint we focused a bit more on the requirements which we agreed upon for our assignment and on implementing design patterns. We checked the design out for a little while, and still have to figure out a theme which fits our GUI and style of game (futuristic-ish)
As a player, I would like to have a varied gameplay experience.	Implement several goals to be hit in the level.	Paul	2h	2h	Yes	Implementing several goals has been achieved by keeping track of the amount of goals being in the stage. Whenever a goal gets hit, the stage will decrement the counter and keep track of it. Because of our observer pattern, we can easily send out a WIN event to our level who will then know that all goals are hit.
As a player, I would like to have a varied gameplay experience.	Implement bombs , which should be avoided, in the level.	Paul	2h	30m	Yes	As above, we can send out a LOSE event to our level, in case a bomb gets hit. Once the observer pattern was fully implemented, implementing and testing a bomb tile was trivial.
As a player, I would like to play more than just one level.	Implement multiple levels for the game.	Kasper	4h	6h	Yes	Implementing multiple levels took a bit of time and required some refactoring of our current code. We chose to implement the levels with the observer pattern. Our PlayScreen has an array of levels which all have their own texturemap and laser. Switching to a new level equates to the PlayScreen switching between these levels in this array.
not applicable	Implement design patterns.	Kasper & Paul	5h	10h	Yes	While not an item for an user story, it did take up a lot of time during this sprint therefore it's worth mentioning. After consulting with our TA during our meeting, we saw that some of our initially thought out design patterns weren't quite how we thought they were. After some reconsidering, we could implement some new features and existing features together with these design patterns. More info on these design patterns can be found in the documentation.

Main Problems Encountered

Problem 1	Outdated implementation of features
Description	Some features which were implemented during the begin of the sprint and put up for review, will after some time, become outdated when other new features come in which refactor existing code. This sprint we also had the special case of having to implement design patterns on a short notice, which caused a lot of this phenomena as well.
Reaction	As a result, the new features who worked from the latest main branch will have to afterwards try to fit in these new features
Problem 2	Pipeline failure
Description	Due to the late reimplementation of our design patterns, we had to rewrite a lot of tests. In order to pass the deadline, we decided to leave one modified class untested, which caused the pipeline to fail in the master.
Reactions	We will make sure that the final version of our code, which will be delivered next week, has all tests necessary.

Adjustments for next Sprint Plan

Adjustment 1	As said in problem 1, the problem was caused mostly by the deadline for our assignment. The fact that this deadline fell together with our sprint planning, made this sprint quite an unusual one. Therefore we couldn't really find any glaring fault in our work process to be improved for the coming week.