Sprint Retrospective, Iteration # 1

User Story	Task	Task Assigned To	Estimated Effort per Task (in hours)	Actual Effort per Task *(in hours)*	Done *(yes / no)*	Notes
As a user I want to be able to log into an account I've already created and know that my password is securely stored.	Set up Database connection	Filip	10	10	Yes	
	Authentication using a username and password via database	Alin, Filip	9	7	Yes	
	Hash Password	Filip, Alin	3	0	No	
As a user, I want to be able to store my score along with my name in a database.	The score of each player should be recorded in the database	Alin, Filip	2	2	Yes	
	At the end of each play, the user should be able to enter his/her name together with the recorded score.	Alin, Filip	3	3	Yes	Database updates the specific user's information with the entry he makes at the end of the game and saves those updated values for future retrieval. After inputting the new score, user can see his progress later on.
As a user, I would want to see the top 5 scores on the leaderboard	At the end of each play, the game should show the top 5 scores that have ever been recorded.	Filip	3	3	Yes	

The player wants the game to be over when pacman collides with a ghost. There should be several ghosts that move around the board according to their own behavior. The player wants the ghosts to move following the same rules as the ones Pacman abides by, this includes not being able to go through walls, and wrapping around the board.	There are four ghosts, each with their own unique AI (according to the original PacMan)	Simran , Paul	3	14	No	One ghost (Blinky) is being implemented for now. A framework is present to simplify creation of more ghosts. Took way longer than expected as we first implemented the ghosts using javafx and then switched to another method. Additionally we had to make other methods that would help us access the other entities,etc
As a user, I want to be able to: move pacman up, down, left and right using the keyboard, wrap around the board when I 'move off' the board, and be stopped by walls	PacMan movement behavior implemented	Alin, Paul	2	3	Yes	KeyListeners check for arrow keys press and then change pacman's direction of movement to that of the arrow key. PacMan continuously moves on the board, unless game is paused or won.
As a player I want to be able to play a level, when I press play. As a player I want to be able to move to a pellet and eat it, so I can earn points. As a player I want to be able to	Setting up a level with walls and pellets	Alin, Ruben, Paul	3	3	Yes	Level includes a List of entities, represented by pellets, walls, pacman and ghosts.Pellets can be collected, while walls do not allow passing through them.

move towards a wall and not go through it.						
As a user, I want to be able to play the game on board displayed on the screen, on which there are walls, pellets, ghosts, different other sprites and my PacMan.	Creating the board	Alin, Paul, Ruben	3	4	Yes	There are 2 tightly coupled classes, Board (which is the functional Board) and BoardCanvas (the graphical one). They are made by MapParser from a .txt file
As a user, I want to be able start the game and see a new window popping up.	Creating basic window	Simran	1	1	Yes	Creating a basic window was very simple as not many components we needed.
As a user, I want to be able to keep track of my score progress and my information, as well as being able to access other player's information.	Connect Database with the game	Filip	6	6	Yes	
As a user, I want to be able enter my username and password and click a button to log in. As a user, I want to be able to make an account by adding my username and password.	Implement register method	Filip, Alin	2	2	Yes	
	Creating a register/login screen	Simran	1	3	Yes	Took a while to get used to the different components in javafx.
	Connecting the database with the GUI	Filip, Simran	5	5	yes	
As a user, I want to be able to keep track of my score and the progress I made. As a user, I want to be able to create a new	Implement interfaces to access the database through GUI	Alin, Filip	3	3	No	A few more controllers have to be implemented.

account or to delete an existing one						
As a user, I want to be able to start the game and be prompted by a menu screen. As a user, I want to be able to choose different options from the menu screen. As a user, I want to be able to access the leaderboard, my profile or start a new game	Creating a menu screen	Simran	2	2	yes	
As a player I want to stop when I hit a wall and die when I hit a ghost and collect a pellet when I hit a pellet	Entity Collision	Ruben	4	4	Yes	
As a player I want to have pacman move when I press the appropriate key. As a player I want to get points when pacman eats a pellet. As a player I want to move to the next level / win the game when I eat all the pellets.	Finish Game Logic	Alin	2	2	Yes	Score is tracked throughout the level, username is displayed in the game window. Level is won when all the pellets have been eaten.
UML class and use case diagrams	Make UML use case diagram for the game	Paul	3	5	Yes	The UML diagram for assignment 1 is a very simplified version of a UML diagram

						i also made, This includes Use Case Definitions.
	Make UML Class diagram for the game	Ruben	1	1	Yes	
As a user, I want to be able to play the game and see 2D animations and different graphic styles of the board and sprites in the game..	2D graphics using JavaFX	Ruben, Simran	6	8	Yes	
As a user, I want to know that the game is written properly and future updates or fixes can be just as easily approached by a new team of developers as the ones that implemented the original version, in case I want other developers to tackle my project. As a user, I want to be able to have new features implemented in my project with as less effort and costs as possible.	Basic code structure of the project, with factory classes and proper launching layout	Alin	3	3	Yes	The game loop happens inside of GameController,where the jfxView is loaded by the BoardCanvas and LevelFactory sets up the environment for the specific level.

Main Problems Encountered

Problem 1

Description:

Had a few conflicts on meeting times. Many of us were available for different time which would lead to someone missing out on the TA meeting.

Reaction:

Led to a discussion and finally came to a conclusion and chose an hour where everyone could be present. It was also stated that everyone should keep themselves available for during the entire shared lab. The TA has open slots for us to choose from, so we have a week to choose a slot where everyone will be available.

Problem 2

Description:

While working on different tasks, we sometimes ended up doing the same one without having any knowledge that our code will become duplicate and lose its significance. Different implementations of the same task were given at times, without having the specific need for doing so, which caused small troubles in communication inside our team further on.

Reaction:

We have set up our meetings to include afterwards a clear note of who is doing what, note based on which we created the backlog and following issues on gitlab. We are still working on distributing tasks in such a clear and balanced manner, so that everyone has its separate list of tasks. Communication inside of group has greatly improved.

Adjustments for the next Sprint Plan

Made a group contract to avoid confusion and set in stone some basic rules for improving inter-teammate communication and diminish the occurrences of overlapping work (whenever it is not specifically required to have 2 versions of the same thing).

Task distribution will be improved at its management level. We will take more care of splitting tasks evenly and assigning them properly, with clear boundaries and requirements.