Sprint Reflection # 3

Game: Fishy

Group: 6

User story	Task	Task assigned to:	Estimated effort per task:	Actual Effort per Task (in hours)	Done (yes / no)	Notes
Exercise 1: Our own improvement will consist of refactoring and making sure each class has the responsibility it should. This also includes creating interfaces for all the classes, as a structural way of rethinking the RDD in the code.	Create requirements document for 'system' refactor.	Michiel	1 hrs 0 min	0 hrs 30 min	Yes	
	Review responsibilities of each class, some classes are responsible for multiple things.	Michiel	1 hrs 0 min	1 hrs 0 min	Yes	Some clear problems were identified (most of which have been fixed).
	Create a package with interfaces for every class, to help with the notion of RDD.	Michiel	1 hrs 0 min	3 hrs 0 min	Yes	This took way longer than expected. Making interfaces for all the classes turned out to be a lot of work.
	Make the necessary code changes to reflect the 'new' responsibilities.	Michiel	2 hrs 0 min	4 hrs 0 min	Yes	The code is a lot better structured now.

						There are still some things which can be changed, but that is for a next sprint.
	(Might not be necessary) Adjust the UML to reflect the new code.	Clinton	0 hrs 20 min	0 hrs 20 min	Yes	
	Implement Save Game feature	Dmitry	1 hrs 0 min	2 hrs 0 min	No	Due to the feature causing errors and failed builds and being unable to fix them by the deadline, the feature was scrapped.
Exercise 2: We have chosen the factory and singleton design patterns, these are not used in the code yet, so they will have to be implemented first.	Create a requirements document for the factory method design pattern.	Matthijs	1 hrs 0 min	0 hrs 40 min	Yes	
	Implement the code changes according to factory method design pattern.	Matthijs	2 hrs 0 min	3 hrs 0 min	Yes	
	Write a natural language description of why and how the factory design pattern is implemented in the code.	Clinton	1 hrs 0 min	0 hrs 40	Yes	It was shorter than expected, but this was because the task was finished by Clinton, as Matthijs didn't

						have time for this task.
	Make a class diagram of how the factory design pattern is structured statically in the code.	Clinton	1 hrs 0 min	1 hrs 0	Yes	
	Make a sequence diagram of how the factory design pattern works dynamically in the code.	Clinton	1 hrs 0 min	1 hrs 0	Yes	
	Create a requirements document for the singleton design pattern.	Sunwei	1 hrs 0 min	0 hr 40 min	Yes	Actual effort of this task was shorter than expected, since there were not a lot requirements to write.
	Implement the code changes according to singleton design pattern.	Sunwei	2 hrs 0 min	2 hrs 40 min	Yes	Actual effort of this task was longer because after implementing Singleton pattern to Logger class, there were problems with JUnit test case. It was fixed on time.
	Write a natural language description of why and how the singleton design pattern is implemented in the code.	Sunwei	1 hrs 0 min	1 hrs 0 min	Yes	

	Make a class diagram of how the singleton design pattern is structured statically in the code.	Clinton	1 hrs 0 min	1 hrs 0 min	Yes	
	Make a sequence diagram of how the singleton design pattern works dynamically in the code.	Clinton	1 hrs 0 min	1 hrs 0 min	Yes	
-	Read the Software Engineering Economics paper.	Dmitry	1 hrs 0 min	2 hrs 0 min	Yes	
	Explain how good and bad practice are recognized.	Dmitry	0 hrs 20 min	0 hrs 30 min	Yes	
	Explain why visual basic being in the good practice group is a not so interesting finding of the study.	Dmitry	0 hrs 30 min	0 hrs 45 min	Yes	
	Enumerate 3 other factors that could have been studied in the paper and why you think they would belong to good/bad practice.	Dmitry	1 hrs 0 min	1 hrs 30 min	Yes	
	Describe in detail 3 bad practice factors and why they belong to the bad practice group.	Dmitry	1 hrs 0 min	1 hrs 0 min		

Estimated work effort per member:

Clinton: 4 hrs 20 min

Dmitry: 4 hrs 50 min

Matthijs: 4 hrs 0 min

Michiel: 5 hrs 0 min

Sunwei: 4 hrs 0 min

Actual work effort per member:

Clinton: 5 hrs 0 min

Dmitry: 7 hrs 45 min

Matthijs: 3 hrs 40 min

Michiel: 8 hrs 30 min

Sunwei: 4 hrs 20 min

Main problems encountered

Problem 1: Save Game

Description: The implementation of the Save Game feature was very difficult to implement due to the fact that the save game had to keep track of not only the information such as score, but also parameters such as the sprite's dimensions, powerups carried etc, which caused a lot of errors and build failures during testing.

Reaction: The feature was scrapped, hopefully to be implemented on a later date

Problem 2:

Description: Implementation of Singleton design pattern for Logger class, it went smoothly, however, after it initially caused JUnit test case of Logger class to run into problems.

Reaction: The errors were fixed after the discussion with the teammates.

Adjustments for next sprint plan

This sprint was the only sprint where we did not finish all the tasks. The reason behind this is explained in problem 1. Also some members has spend way too much time on their tasks than what was estimated. For the next sprint, we should assign smaller tasks to member to alleviate the some work effort, since we have been underestimating actual work effort.

We will also change our priority system from numbers to letters. We will give our branches better names, as the branches we have made do not really describe what we are doing in that branch. We will also write shorter and clearer commit messages.