Sprint Reflection #3

Game: Fishy
Group: 11

User Story	Task	Task Assigned To	Estimated Effort per Task (in hours)	Priority (A-E)	Done (yes/no)	Notes
Exercise 1: 20-Time, Reloaded						
	1.1: Making requirements for new extension/improvement	Leon	0.5	A	Yes	
	1.2: Sending requirements to TA	Leon	0.1	A	Yes	
	1.3: Powerup Factory	Millen	2.5	A	Yes	
	1.4: Powerup 1: Adding shield powerup from turtle.	Martijn	3	B	Yes	
	1.5: Powerup 2: 1 up	Millen	3	B	Yes	
	1.6: Powerdown 1: Poison fish which inverts the controls	Martijn	3	B	Yes	
	1.7: Powerdown 2: Adding powerdown when fish speed gets a lot higher	Millen	3	B	Yes	
	1.8 UML of the newly added features	Martijn/Millen	1.5	B	Yes	
Exercise 2: Design Patterns						
	Pattern 1: State pattern					
	2.1: Write a natural language description of why and how the pattern is implemented in your code.	Ricardo	0.5	A	Yes	
	2.2: Make a class diagram of how the pattern is structured statically in your code	Ricardo	0.5	B	Yes	
	2.3: Make a sequence diagram of how the pattern works dynamically in your code	Ricardo	0.5	B	Yes	
	Pattern 2: Singleton pattern					
	2.4: Write a natural language description	Leon	0.5	A	Yes	

	of why and how the pattern is implemented in your code.					
	2.5: Make a class diagram of how the pattern is structured statically in your code	Leon	0.5	B	Yes	
	2.6: Make a sequence diagram of how the pattern works dynamically in your code	Leon	0.5	B	Yes	
Exercise 3: Software Engineering Economics						
	3.1: Explain how good and bad practice are recognized	Danique	0.5	B	Yes	
	3.2: Explain why Visual Basic being good in the good practice group is a not so interesting finding of the study	Danique	0.5	B	Yes	
	3.3: Enumerate 3 other factors that could have been studied in the paper and why you think they would belong to good/bad practice	Danique	0.5	B	Yes	
	3.4: Describe in detail 3 bad practice factors and why they belong to the bad practice group	Danique	0.5	B	Yes	
Own assignments:						
	4.1: Refactoring code	Leon	5	C	Yes	Was already done before the sprint plan was made
	4.2: Testing					
	4.2.a: Writing tests for new features	Martijn, Millen	3	D	Yes	
	4.2.b: Writing tests for models	Danique	3	D	Yes	
	4.2.c: Writing tests for opponents	Ricardo	3	D	No	Two of the three opponent classes were

						already tested
	4.2.d: Writing tests for position	Leon	3	D	Yes	

We chose not to test the classes LevelState, LoseState, WinState and MenuState because this is for the most part OpenGL context which can not be tested really well, also because Slick2D is a bit annoying with this as well. Thus we decided that we would focus on all the other files and test those really good. We also found that certain methods in certain classes also directly use OpenGL context, thus these methods were also hard to test.

There was as little bit of miscommunication about the UML for Exercise 1. We thought no UML had been made, and made one last minute. But we later found out that the UML in fact was already made.