

Sprint Plan #1

Game: Fishy
Group: 11

User Story	Task	Task Assigned To	Estimated Effort per Task
Exercise 1			
	1.1: Following Responsibility Driven Design, derive classes, responsibilities and collaborations using CRC cards. Describe each step and compare the result with the actual implementation.	Ricardo	2
	1.2: Following responsibility Driven Design, describe the main classes that are implemented in the project in terms of responsibilities and collaborations	Ricardo	2
	1.3: Why are these other classes less important? Reflect if some of those non-main classes have similar/little responsibility and could be changed, merged or removed. If so perform the code changes, if not, explain why you need them.	Martijn	1
	1.4: Draw the class diagram of the aforementioned main elements of your game	Martijn	2
	1.5: Draw the sequence diagram to describe how the main elements of your game interact.	Martijn	1
Exercise 2			
	2.1: Describe the difference between aggregation and composition. Where are composition and aggregation used in your project? Describe the classes and explain how these associations work	Leon	2
	2.2: Are there any parameterized classes in your source code? If so, describe which classes, why they are parameterized, and the benefits of parameterization. If not, describe when and why you should use parameterized classes in your UML diagrams	Leon	2

	2.3: Draw the class diagrams for all the hierarchies in your source code. Explain why you created these hierarchies and classify their type. Considering the lectures, are there hierarchies that should be removed? Explain and implement any necessary change	Millen	3
Exercise 3			
	3.2: During the analysis and design phases of all the tasks below (exercise 3.1) use Responsibility Driven Design and UML. (Push to the repository in a SINGLE pdf file including all the documents produced)	All	2
3.1: Scenario 1: Button clicked Given the user has launched the Fishy GUI; When [button] is clicked; Then the logger will show “[button] was clicked at ([X],[Y])”;	Logger working for “Play” button	Danique	2
	Logger working for “Exit” button	Danique	0.5
3.1: Scenario 2: Collision smaller enemy Given the user has launched the Fishy GUI; and the user has entered the “play” state of the game; When the player collides with an opponent with a smaller size than the player size; Then the logger output will be “player collided with smaller enemy fish of size [enemy size], player size increases with [size], new player size is [size]”;	Logger working for collisions with a smaller enemy fish	Danique	0.5
3.1: Scenario 3: Collision larger enemy Given the user has launched the Fishy GUI; and the user has entered the “play” state of the game; When the player collides with an opponent with a larger size than the player size; Then the logger output will be “player collided with larger enemy fish of size [enemy size], game lose condition triggered”	Logger working for collisions with a bigger enemy fish	Danique	0.5
3.1: Scenario 4: Winning game	Logger working for winning game	Danique	0.5

<p>Given the user has launched the Fishy GUI; and the user has entered the “play” state of the game; When the player’s score is equal or larger than 500; Then the logger output will be “game win condition triggered”;</p>			
<p>3.1: Scenario 5: State change Given the user has launched the Fishy GUI; When the user enters a new state; Then the logger will show “user entered state [state]”;</p>	Logger working for “Menu” state	Danique	0.5
	Logger working for “Play” state	Danique	0.5
	Logger working for “GameEnd” state	Danique	0.5