Sprint Retrospective, Iteration # 1

Group: 2

User Story	Task	Task Assigned To	Estimated Effort per Task *(in hours)*	Actual Effort per Task *(in hours)*	Done *(yes / no)*	Notes
As a developer, I can see an overview of the interactions between the main elements of the game	Make CRC (Classes, Responsibilities, Collaborations) cards for our classes	Erik & Gijs	3,5	6	Yes	
	Describe main classes	Tim	3	2	Yes	
	Decide if you responsibilities for classes need to be changed, merge or remove classes accordingly	Tim	3	1	Yes	
	Create class diagram of main elements	Gijs	2	2	Yes	
	Create sequence diagram	Erik	3	2	Yes	
As a developer, I can see an overview of the classes using a Class Diagram	Describe main elements	Tim	3	1	Yes	
	Describe if we use parameterized classes and the benefits	Gijs	2	0.5	Yes	
	Draw class diagrams for all hierarchies	Erik	3	1	Yes	
As a developer, I can log the application	Log all interactions between objects that may happen during the game	Rens & Dennis	14	12	Yes	Pull request Logging functionality #94
	Use responsibility design and UML when implementing logging	Rens & Dennis	6	N/A	Yes	Effort for this is divided over the other 2 tasks, as this task is not really a task you can do on its own. Next sprint, we should make all tasks completable.
	Define the requirements for the logging implementation	Rens & Dennis	4,5	4	Yes	

Sprint Retrospective, Iteration # 1

As a user, I want to be able to play the game without bugs	Integrate checkstyle/findbugs/maven/PMD into Travis CI	Tim	1	2	Yes	
	Fix all checkstyle errors/warnings	Erik & Tim	2	6	Yes	See “Problem #1”
	Fix all findBugs errors/warnings	Dennis	2	2	No	See issue #107
	Write JUnit tests for at least 80% branch coverage (excluding GUI)	Rens & Tim (& Gijs & Erik)	6	14	No	See “Problem #2”. Gijs and Erik started helping on this task later on as it was too much work for two people.
	Fix PMD errors/warnings	Erik	2	2	Yes	

Sprint Retrospective, Iteration # 1

Main Problems Encountered

Problem #1: Underestimation of Checkstyle work

Description: Managed to fix all of the Checkstyle issues, but 2 hours was an underestimation. There were many more checkstyle issues (over 600) than we expected to see.

Reaction: In future projects, we should integrate static analysis tools at the start of the project, on day 1. In conjunction, we should have checked the amount of checkstyle issues to fix before estimating hours to put into the task.

Problem #2: Underestimation of testing work

Description: At the start of this sprint, we had too few tests. As a result of this, reaching 80% branch coverage in a week was close to impossible unless we treated this as a full-time job. The goal “reach 80% branch coverage excluding GUI” was simply too ambitious. At the end of the week, we ended with 45% branch coverage excluding GUI tests.

Reaction: Next sprint, we will set a new target for reaching test coverage. This target will be more thoroughly discussed and more realistic.

Problem #3: Coordination and communication

Description: Certain checkstyle issues, such as fixing magic numbers, were done by two team members at the same time. As a result of which work was duplicated. This was also the case when writing tests. There was a point at which a group member A said to another group member B “I’m testing class X”, after which B replied with “Oh, I actually just started testing X”.

Reaction: Communication should be improved to minimize the probability of doing duplicate work.

Problem #4: Branch naming conventions

Description: We all have our own preferences of naming branches. Branch names now contain a wide variety of schemas, which results in an unorganized-looking project.

Reaction: As stated by our TA, a good option would be to use something like Git Flow for this.

Sprint Retrospective, Iteration # 1

Adjustments for the next Sprint

- As stated earlier, discuss more thoroughly about a test coverage target before setting one.
- Before starting fixing Checkstyle/PMD/Findbugs issues, a new issue should be created and assigned to the person doing the task. Same goes for creating tests.
- We will be using Git Flow to make our new branch names more organized.