

## Sprint Retrospective, Iteration #3, Group: 2

User Story	Task	Task Assigned To	Estimated Effort per Task (in hours)	Actual Effort per Task (in hour)	Done (yes / no)	Notes
As a developer, I want to use software design patterns to make implementing code easier.	Pick two software design patterns (i.e., strategy, observer, decorator, Singleton) out of all the patterns described during class.	Dennis & Tim	1	2	TRUE	
	For each chosen pattern, make a class diagram of how the software design pattern is structured statically in your code	Dennis & Tim	4	4	TRUE	
	For each chosen pattern, write a natural language description of why and how the pattern is implemented in your code (and implement if not yet present)	Dennis & Tim & Rens & Erik	10	12	TRUE	Tim did not write any patterns but did pair programming with Dennis. Erik did implement a pattern which was not planned before.
	For each chosen pattern, make a sequence diagram of how the software design pattern works dynamically in your code	Dennis & Tim	4	4	TRUE	
As a developer, I want to extend my game to implement more features according to what our supervisor wants	Talk to the TA to obtain a new requirement for the game	Erik & Gijs	1	1	TRUE	
	Write a requirements document for the extension given by the TA	Erik & Gijs	1	2	TRUE	
	Send the requirements document to the TA for evaluation	Erik & Gijs	1	1	TRUE	
	Write tests for the requirements	Erik & Gijs	2	0		
	Implement the requirements	Erik & Gijs	5	7		
	Design a UML for the extension	Erik & Gijs	2	0		
As a developer, I want to improve the game	Write a requirements document for the improvement, and send it to the TA for approval	Rens	2	2	TRUE	

## Sprint Retrospective, Iteration #3, Group: 2

	In design phase use responsibility driven design, design a UML diagram and push that in 1 pdf file to github	Rens	2	2	TRUE	
	Implement improvements required	Rens	8	6	TRUE	It was a little less to effort to implement this feature than originally anticipated
As a user, I want to be able to play the game without bugs	Fix all checkstyle errors/warnings	Dennis	3	3		
	Fix all findBugs errors/warnings	Rens	4			
	Fix PMD errors/warnings	Erik	2			
	Let the build fail when the above requirements(checkstyle, findbugs and PMD) are not met	Tim	3			t
	Write JUnit tests for at least 80% coverage (excluding GUI)	Gijs	4		TRUE	
As a developer, I want to have all of the tasks in one place and to be able to mark them as completed.	Move all of the tasks described above into issues.	Erik	2		TRUE	

## Main Problems Encountered

### Problem #1: Almost all merges on last day

Description: Code was written better distributed this week, however everyone was working on different branches so on the last day a lot of merging was need to be done since a lot of features touch the same code some implementations needed a lot of rewiting to be done to function correctly.

Reaction: Better communication between people who implement different features and maybe even pull other branches into own branch so when the features will be merged into the develop branch they will not have as many merge issues and less code has to be rewritten because the features will work better together.

### Problem #2: Issues

## Sprint Retrospective, Iteration #3, Group: 2

Description: Issues should be closed as soon as it is resolved. Many issues weren't closed quick enough after completion which gave a low percentage of completion for sprint #3 which did not make any sense. At some point, the amount of issues became too overwhelming and results in too little action taken to update them.

Reaction: Something like Waffle.io can help us here. Or we can tag issues with certain labels to enable powerful filters and see only a few issues at a time.

### **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.

## 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.
- More usage of Waffle.io to get better overview of the open issues.
- Communicate more about how features are implemented and prepare code earlier for merging so less merging needs to be done on the last day and less code has to be rewritten.