Sprint Plan #3

Bejeweled - Group 20

User Story	Task	Responsible	Assigned to	Estimated Effort	Actual Effort	Done	Notes
	Update old UML	Hugo	Hugo	0.5	1 hour	Yes	
Jewel swaps and the processing of matches between corresponding colours are made more visible by animations.	Make Animation UML	Hugo	Hugo	3	2.5 hours	Yes	
	Make matches clear simultaneously	Daniël	Daniël	3	2 hours	Yes	
	Implement swap animations	Daniël	Daniël	4	6 hours	Yes	Finding a way to implement animations was the hardest part. After that, the same method could be used for all types of animations.
	Implement fall animations	Daniël	Daniël	5	30 minutes	Yes	
	Implement fill animations	Daniël	Daniël	3	10 minutes	Yes	
	Testing	Björn	Björn	3	3 hours	Yes	Junit testing was not possible so had to use manual testing.
Composite Pattern implementation (Match class)	Implement pattern	Rob	Rob	4	4.5 hours	Yes	Also needed to refactor clearMatches() method of Board
	Natural language description	Rob	Rob	2	1 hour	Yes	
	Class Diagram	Rob	Rob	2	45 minutes	Yes	
	Sequence Diagram	Rob	Rob	4	45 minutes	Yes	
Observer design pattern implementation	Implement pattern	Evan	Evan	2	3.5 hours	Yes	
	Natural language description	Evan	Evan	2	30 minutes	Yes	
	Class Diagram	Evan	Evan	2	2 hours	Yes	
	Sequence Diagram	Evan	Evan	3	45 minutes	Yes	
Exercise 3	Exercise 3.1	Björn	Björn	2	1.5 hours	Yes	
	Exercise 3.2	Björn	Björn	1	10 minutes	Yes	
	Exercise 3.3	Hugo	Hugo	2	1 hour	Yes	
	Exercise 3.4	Hugo	Hugo	2	1.5 hour	Yes	
When the user wants to start a new game. He can press the new game button.		Björn	Björn	2	2 hours	Yes	

Problems/Remarks

Implementing Composite Pattern

At first, implementing the composite pattern did not seem a lot of work. However, we did not consider that the clearMatch() method needed to be refactored as well.

Next time, we should look at all the parts of the code that need to be changed to implement a feature to avoid surprises

Implementing Observer Patter

The implementation of the observer pattern was harder to do than was originally expected because we thought that the pattern was essentially already in the project due to the listeners that we had, but the listeners only fulfilled part of the observer pattern and the other part still needed to be implemented.

To fix this i just had to work a bit harder which wasn't that bad. but to avoid this from happening again we need to make sure that we fully understand what a certain implementation entails so that we can avoid underestimating it,

Testing

Junit testing was impossible for animations. So we did have have a problem at that moment since we mentioned testing in the sprint plan and of course we do not want software to be untested.In the end we did a good job by using manual testing. We thought of user stories and scenarios to test it.

Discussing bad practices

It took me a little longer than expected to fully understand the paper about good and bad practices of software engineering and make up additional factors to consider for the study, since it already considered over 50 factors

Next time I think it would be better if I read the paper multiple times spread over a few days, so that when I make the assignment, I will be able to make it faster.

Also I won't use automatic UML generation tools anymore, since they are often inaccurate.