User stories are only necessary for large extensions of the game (e.g., exercise 3.1, assignment 1). In all the other cases (e.g., exercise 1 and 2, assignment 1), user stories can be omitted (but task splitting, assignment, and estimated effort are to be done).

	User Story	#	Task	Subtasks	Task Assigned To	Estimated Effort per Task (hours)	Done		Kleurcode
1	-		**1.1** - Following the Responsibility Driven Design, start from your requirements (without considering your implementation) and derive classes, responsibilities, and collaborations (use CRC cards). Describe each step you make. Compare the result with your actual implementation and discuss any difference (e.g., additional and missing classes).	Derive classes from requierements. Describe each step you make.	Lilian	0,5	2		Rogier
				Derive responsibilities from requirements. Describe each step you make.	Lilian & Karin	0,5	2		Christian
				Derive collaborations (use CRC cards). Describe each step you make.	Lilian & Karin	0,5	1		Karin
				Compare the above result with our implementation and discuss differences.	Lilian & Karin	0,5	2		Lilian
2	-		**1.2** - Following the Responsibility Driven Design, describe the main classes you implemented in your project in terms of responsibilities and collaborations		Karin & Lilian	4	2		Bas
3	-		**1.3** - Why do you consider the other classes as less important? Following the Responsibility Driven Design, 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	Explain why the other (non-main) classes are considered less important.	Karin & Lilian	1	2		Fieke
				Reflect if some of the non-main classes have similar/little responsibility and could be changed, merged or removed.	Karin & Lilian	1	2		All
				Perform the code change or explain why you don't need them.	Not necessary	2	2		
4	-		**1.4** - Draw the class diagram of the aforementioned main elements of your game (do not forget to use elements such as parametrized classes or association constrains, if necessary)		Karin	5	2		
5	-		**1.5** - Draw the sequence diagram to describe how the main elements of your game interact (consider asynchrony and constraints, if necessary)		Karin	4	2		Not started yet
6	-		**2.1** - What is the difference between aggregation and composition? Where are composition and aggregation used in your project? Describe the classes and explain how these associations work	Describe the difference between aggregation and composition	Fieke	0,5	2		Working on it
				Describe where we use composition & aggregation in our project	Fieke	0,5	2		Done!
				Describe the composition & aggregation classes and explain how these associations work.	Fieke	1	2		
7	-		**2.2** - Is there any parametrized class in your source code? If so, describe which classes, why they are parametrized, and the benefits of the parametrization. If not, describe when and why you should use parametrized classes in your UML diagrams		Christian	8	2		
8	-		**2.3** - Draw the class diagrams for all the hierarchies in your source code. Explain why you created these hierarchies and classify their type (e.g., "Is-a" and "Polymorphism"). Considering the lectures, are there hierarchies that should be removed? Explain and implement any necessary change	Draw the class diagram for all hierarchies in the source code.	Fieke	5	2		
				Explain why the hierarchies were created and classify the type.	Fieke	1	0		
				Are there hierarchies that should be removed and why (look at lectures)	Christian	1	1		
				Implement any necessary change and explain it.	Christian	1	2		
9	As a developer, I want a file to be created during gameplay that keeps track of all actions performed on/by the player, so that I will be able to find bugs easier in the software, because this file will report not only which potential errors occurred, but also with which steps the software reached it.		**3.1** - Extend your implementation of the game to support logging. The game has to log all the actions happened during the game (e.g., player moved Tetris piece from position X to position Y). The logging has to be implemented from scratch without using any existing logging library. Define your requirements and get them approved by your teaching assistant.	Define requirements extention	Rogier	1	2		
				Create classes, requirements and collaborations for this extension	Rogier	2	2		
				Create a class diagram for this extention	Christian	2	2		
				Create a sequence diagram for this extention	Christian	2	2		
				Implement the extention according to these diagrams.	Rogier	6	2		
10	-		**3.2** - During the analysis and design phases of this extension use responsibility driven design and UML (push to the repository a single PDF file including all the documents produced)	/ see above.		/	/		
11			**4.1 -** Use plugins correctly	Make sure that when you have edited a class, that maven reports no checkstyle, findbugs or pmd errors in that class.	All	/	/		
12			**5.1 -** Changing classes based on Responsibility Driven Design	Update methods should be splitted into different update methods that have their own responsibilities	Bas	2	1.5		
				Level1State contains methods that are equal to each level state, so this should be divided under other classes.	Bas	2	1.5		
13			**5.2 -** Player should be able to jump onto bubbles		Bas	2	5		
14			**5.3 -** The game should have multiple levels.		Bas	2	2		

Taks	Subtask	Task assigned t	Estimated Effort per Task	Actual Effort per Task	Done	Notes	
1.1 - Following the Responsibility Driven Design, start from your requirements (without considering your implementation) and derive classes, responsibilities, and collaborations (use CRC cards). Describe each step you make. Compare the result with your actual implementation and discuss any difference (e.g., additional and missing classes).	Derive classes from requierements. Describe each step you make.	Lilian	0,5	2	Yes		
	Derive responsibilities from requirements. Describe each step you make.	Lilian & Karin	0,5	1	Yes		
	Derive collaborations (use CRC cards). Describe each step you make.	Lilian & Karin	0,5	0,5	Half		
	Compare the above result with our implementation and discuss differences.	Lilian & Karin	0,5	0,5	Yes		
1.2 - Following the Responsibility Driven Design, describe the main classes you implemented in your project in terms of responsibilities and collaborations		Karin & Lilian	4	1	Yes		
1.3 - Why do you consider the other classes as less important? Following the Responsibility Driven Design, 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	Explain why the other (non-main) classes are considered less important.	Karin & Lilian	1	0,5	Yes		
	Reflect if some of the non-main classes have similar/little responsibility and could be changed, merged or removed.	Karin & Lilian	1	0,5	Yes		
	Perform the code change or explain why you don't need them.	Not necessary	2	0	Yes		
1.4 - Draw the class diagram of the aforementioned main elements of your game (do not forget to use elements such as parametrized classes or association constrains, if necessary)		Karin	5	1,5	Yes		
1.5 - Draw the sequence diagram to describe how the main elements of your game interact (consider asynchrony and constraints, if necessary)		Karin	4	2	Yes		
2.1 - What is the difference between aggregation and composition? Where are composition and aggregation used in your project? Describe the classes and explain how these associations work	Describe the difference between aggregation and composition	Fieke	0,5	0,5	Yes		
	Describe where we use composition & aggregation in our project	Fieke	0,5	0,5	Yes		
	Describe the composition & aggregation classes and explain how these associations work.	Fieke	1	0,5	Yes		
2.2 - Is there any parametrized class in your source code? If so, describe which classes, why they are parametrized, and the benefits of the parametrization. If not, describe when and why you should use parametrized classes in your UML diagrams		Christian	1	,5	Yes		
2.3 - Draw the class diagrams for all the hierarchies in your source code. Explain why you created these hierarchies and classify their type (e.g., "Is-a" and "Polymorphism"). Considering the lectures, are there hierarchies that should be removed? Explain and implement any necessary change	Draw the class diagram for all hierarchies in the source code.	Fieke	5	4	Yes		
	Explain why the hierarchies were created and classify the type.	Fieke	1	0,5	Half		
	Are there hierarchies that should be removed and why (look at lectures)	Christian	1	,5	Yes		
	Implement any necessary change and explain it.	Christian	1	,5	Yes		
3.1 - Extend your implementation of the game to support logging. The game has to log all the actions happened during the game (e.g., player moved Tetris piece from position X to position Y). The logging has to be implemented from scratch without using any existing logging library. Define your requirements and get them approved by your teaching assistant.	Define requirements extention	Rogier	1	1	Yes		
	Create classes, requirements and collaborations for this extension	Rogier	2	1	Yes		
	Create a class diagram for this extention	Christian	2	1,5	Yes		
	Create a sequence diagram for this extention	Christian	2	,5	Yes		
	Implement the extention according to these diagrams.	Rogier	6	3	Yes		
3.2 - During the analysis and design phases of this extension use responsibility driven design and UML (push to the repository a single PDF file including all the documents produced)	/ see above.		/		/		
4.1 - Use plugins correctly	Make sure that when you have edited a class, that maven reports no checkstyle, findbugs or pmd errors in that class.	All	/		/		
5.1 - Changing classes based on Responsibility Driven Design	Update methods should be splitted into different update methods that have their own responsibilities	Bas	2	1.5	Yes		
	Level1State contains methods that are equal to each level state, so this should be divided under other classes.	Bas	2	1.5	Half	Will be refactored in wk 4	
5.2 - Player should be able to jump onto bubbles		Bas	2	5	Yes		
5.3 - The game should have multiple levels.		Bas	2	2	Yes		

Problem #	Description	Reaction	
1	Game was only runnable on Bas' machine	Updated libraries on all other machines	
2	File structure wasn't correct.	We created a whole new project	
3	A lot of failed builds	Check Maven + lots of debugging	
4	Group communication	Communicate more before and during working when not working together	
5	Test cases where omitted	We got 5/10 points, so we need to improve that for upcoming weeks.	

Adjustment	**Motivation**
Not too many building failures after each other	
Tagging in every commit	
Write tests	
Communicate more before and during working when not working together	