Sprint #5 Reflection

User Story	Task	Subtask	Task Assigned To	Priortiy (A-E)	Estimated Effort Per Task (Hours)	Actual Effort (Hours)	Done (Yes/No)	Notes
As a player, I would like the ability to consume a power-up that increases the movement speed for a limited time.	Extension: Power-Up	Requirements	Adriaan	A	1	1	Yes	
		Design and Analysis	Adriaan	B	3	2	Partly	Missing UML
		Implementation	Adriaan	A	5	5	Yes	
As developers we would like well structured code.	Design Patterns	Design Pattern I - Factory						
		Implementation	Thomas	A	5	1	Yes	We could implement the same for Player.
		Description	Thomas	A	3	2	Yes	
		Class Diagram	Thomas	A	3	0.5	Yes	
		Sequence Diagram	Thomas	B	3	2	Yes	
		Design Pattern II - State						
		Implementation	Thomas	A	0.5	0.5	Yes	
		Description	Thomas	A	3	2	Yes	
		Class Diagram	Thomas	A	1	0.5	Yes	
		Sequence Diagram	Thomas	B	5	2	Yes	
As developers, we would like to take time to reflect on our progress, successes, and failures.	Reflection	Project Reflection (200 words)	Thomas	A	2	1	Yes	
		Project Reflection (200 words)	Lars	A	2	2:45	Yes	wrote ~760 words instead of 200
		Project Reflection (200 words)	Boris	A	2	0	No	
		Project Reflection (200 words)	Adriaan	A	2	0:30	Yes	
		Project Reflection (200 words)	Dylan	A	2	2	Yes	
As developers, we would like to deliver a product that has 75% meaningful test coverage.	Improve test coverage	Sprites Package	Lars	C	2	0	No	focused on States Package
		StateManagers Package	Lars	B	5	0	No	focused on States Package
		States Package	Lars	A	5	6:45	No	Discovered limitations of Mockito, which required a switch to JMockit
		EntitySpawner Package	Dylan	C	2	0	No	Out of time
		EntityTools Package	Lars	D	2	0	No	focused on States Package
		GameStates Package	Dylan	B	2	1	No	Out of time
		Players Package	Boris	A	3	3	Yes	New way of generating enemies. Needs to be updated
		ResourceToolsPackage (Sound and Music excluded)	Lars	E	2	0	No	focused on States Package
		ValueSettings	Boris	E	2	0	No	
		Animation Package	Dylan	D	2	2	Yes	
		Bubbles Package	Dylan	E	2	2	No	Could not be done because of travis
		Enemies Package	Dylan	A	3	3	Yes	
		HighScores Package	Adriaan	B	2	0:30	No	Focussed on higher priority tasks
		SinglePlayerGames package	Adriaan	C	3	0	No	Focussed on higher priority tasks
		SinglePlayerGameStates package	Adriaan	D	2	0	No	Focussed on higher priority tasks

Main Problems Encountered

Problem 1: Limitations of Mockito

Description:

Mockito is limited in its mocking capabilities. For example, it cannot mock an instance of a singleton. This is a problem because our software makes extensive use of singletons. These limitations have made it difficult to improve test coverage as quickly as planned.

Reaction:

We will be using JMockit for mock-ups, since JMockit is able to mock things which Mockito cannot.

Problem 2: Build Failures

Description:

Unfortunately there were build failures in our release, due to an error where the branches were shown to be mergeable without test failures; but were not.

Reaction:
We have learned that the merge-request User Interface on Github is not trustable when you want to merge. To make sure our branches are mergeable without failures, we will run the application and the tests using Maven locally, as well as rebuild the project on Travis at each milestone.

Problem 3: Low Test Coverage

Description:
Our project still has a low test coverage, due to some complications which resulted in a few tests which could not be completed this sprint.

Reaction:
We believe that we can focus a lot more on completing tests this sprint, as we do not have too many additional high priority tasks to complete. During our final sprint, our main focus will be on improving the test coverage.

Problem 4: Lack of Documentation

Description:
Our project also still suffers from incomplete Javadoc, which was also a result of time conflicts.

Reaction:
We hope to focus more time on filling in the missing Javadoc.

Adjustments for the next Sprint Plan

This sprint we intend using JMockit when it is beneficial. As well as focusing on improving our test coverage and documentation. We will also focus on making sure our next release does not contain build failures by building locally.