Sprint Review for Week 6

Context Project: Computer Games
Group: 4
Based on pair programming.
deadlines for this week

Green rows are

User Story Numbers:	**Task:** Ordered on highest priority first and lowest priority last	**Task Assigned To:** [Responsible] ([Partner])	**Estimated Effort per Task:** Effort prediction in total amount of hours spent by all team members:	**Actual Effort:** in total hours by all team members:	**Done:** Yes/No	**Notes:**
-	Back button on the device should return you to the previous screen	Jurgen	1	1	yes	
	Buttons in settings and stroll have to function as described	Jurgen	1	1	yes	
-	Fix font bug when exiting and reopening the app	Jurgen	3	3	yes	
-	Decoupling GUI and Model (and testing)	Ben (Jurgen)	15	14	yes	Not completely tested.
-	Work out game concept for presentation	Ben / Jurgen	4	4	yes	
	Create presentation slides/notes		2	3	yes	
	Prepare presentation		2	2	yes	
1	Work out how rewards are given and how rarity is defined	Jean (Martijn)	12	14	yes	
	Create a data structure from integer to colours/patterns and vice versa		20		partial	Mapping needs to be done but color generation works.
-	Refactoring timer for better functionality	Ben (Jurgen)	6	8	yes	And mostly tested.
-	Create sprites for the collectibles	Martijn (Jean)	20	14	partial	Sprite contain glyphs, which will be fixed in next iteration.
	Create an architecture for collectibles for easy extension		12	4	yes	It’s only an abstract class, not much work.
2	Implementing movement checking during stroll	Nick (Jurgen)	8	5	partial	Library has been added, but doesn’t work yet with Stroll
	Create an additional event so we have 2 functioning events		25	22	partial	Not tested.
	Modify events to match the theme of the game and have visuals.		8	0	no	Decided to prioritize on other issues.
3	Create a design for the collectible screen	Jean (Martijn)	10	4	partial	Unfortunately we did not get this completely done at the end of this sprint. This will be moved to next week.

	Display rarity of collectible in the collection screen		2	0	no	The collectible screen needs to be done before we can implement this.
-	Playtesting with external people	Nick	6	3	no	No actual testing done but document and setup is complete. Has to be shared with external people.
-	Update the EAD	Nick	5	5	yes	
4	Research into networking (e.g. Bluetooth/NFC)	Ben	0+	2	yes	Server created on a new branch, but not yet merged in this iteration.
	Research into server/client (e.g. Firebase or default Android server)		0+	2	yes	
5	UI should still function after switching device orientation (e.g. landscape to portrait)	Jurgen (Martijn)	0+	8	yes	In order to fix the font bug (above) the WorldRenderer and GameSkin had to be updated. This was fixed at the same time.

Extra Task:	Task description:	Time spend	Notes
I	Checkstyle, PMD, Findbugs fixing, Javadocs	8	
II	Assets refactoring	4	Wasn't solid, had to be refactored for better functioning.
III	Initial server setup on 'server' branch	18	Since we determined this is <u>really</u> important in the next iterations and many things are dependant on it, we wanted a headstart with the server.

User Stories	
User Story Number:	**User Story:**
1	As a user, I want to be able to earn different kinds of collectibles
2	As a user, I want to be able to encounter different events, because having only one event makes the game stale and dull.
3	As a user, I want to be able to see which collectibles I have obtained.
	As a user, I want to be able to see how rare my collectibles are.
4	As a user, I want to be able to connect with other users to play the game with them or against them.
5	As a user, I want to be able to turn my phone's orientation to my own liking.

Sprint Retrospective:

Main problems encountered:

- A single bug can take a lot of time to fix, because a refactoring the code is needed. While thinking about how the application should work does not always mean that it will also work as expected.
- Checkstyle rules and testability can create conflict. In order to Mock certain objects the methods cannot be final. But Checkstyle tries to enforce this. This gave many errors that we didn't know where they originated from.
- Checkstyle, PMD, Findbugs takes a lot of time. We hadn't done this properly in the beginning and now had to fix many rules at once.
- Creating testable code for a game is not always a trivial task. Since there are many factors that change the state of the game (as well as user input) it can be hard to determine what a proper test is.
- We currently work on a branch for a long time and when the feature is finished, finally create a pull request. The problem is that this creates many merge errors and makes it hard to review.
- We deviated from the original sprint plan. Both to refactor as well as in the planning for the next week. While we still did enough work, it did mean that the sprint plan did not get completed.

Major differencing in expected time and actual time:

- For user story 1 we needed slightly less time than we expected. This was because it was easier to implement than we thought at first.
- In general the amount of hours we expected to work on the project is lower than the actual. We simply hadn't spend enough hours this week. In the previous iteration we spend more time than we initially wanted. This left us somewhat exhauster for this week. Some members also had other responsibilities during the weekend which reduced the number of hours as well.

Positive improvements:

- For the time we spend on the project this week we actually got quite a bit done. After much refactoring the code finally reaches a stable state while at the same time we understand both the application logic as well as the LibGDX framework better.

Improvements for the next sprint:

- This week we want to keep Checkstyle, PMD and Findbugs up to date whenever we commit code. By doing this while coding, we want to prevent it to feel as an annoying task.
- While it's good that some members are eager to code, it doesn't help when you sleep at 03:00 or later and have to be up at 07:00 again. After 24:00 no more work is allowed.
- We refactored our code way too much. While it is really good to have clean code, it doesn't help if we can't finish the tasks we defined in the sprint plan. We need more features at this point to make the game enjoyable.
- We need to do more testing. Sometimes testing couldn't be done because the Classes weren't properly coded, but that was not always to case. Regular testing or TDD must become more of a standard.
- In order to have less conflicts in pull requests, we need to have more frequent and smaller pull requests.

- Try to review open pull requests whenever there is a moment you have nothing to do.
- Create more issues on github so everyone is up to date with bugs and other problems in the game.