# Sprint Reflection 4

Game: Bubble Bobble

Group: 7

| User Story | Task | Task Assigned To | Estimated Effort per Task (hours) | Actual Effort per Task (hours) | Done (yes/no) | Notes |
|---|---|---|---|---|---|---|
| As a developer, I want to have clear documentation for the features of this week. | Create a requirements document for Exercise 1.1. | Joost | 1h | 1h15 | yes | |
| | Create an assignment 4 document for all the answers of the questions. | Lili | 1h | 45min | yes | |
| As a developer, I want to use inCode to help me find errors and flaws in the code | Use inCode to compute software metrics on your project. | Lili | 1h | 30min | yes | I put extra time, in case there would be installation issues, but it all ran very smoothly. |
| | For a first fatal flaw, explain the design choices or errors leading to the detected design flaw. | Jeffrey | 1.5h | 2h | yes | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | For a first fatal flaw, fix the design flaw or extensively and precisely explain why this detected flaw is not an error and, thus, should not be fixed | Jeffrey | 4h | 5h | yes | We had two god classes in our code,'Player'and 'LevelController'. |
| | For a second fatal flaw, explain the design choices or errors leading to the detected design flaw. | Lili | 2h | 2h | yes | |
| | For a second fatal flaw, fix the design flaw or extensively and precisely explain why this detected flaw is not an error and, thus, should not be fixed | Lili | 3h | 3h | yes | |
| | For a third fatal flaw, explain the design choices or errors that have lead to the third design design flaw to be avoided. | Toine | 2h | 30min | yes | This was just as simple as moving one method from SpriteBase to Coordinates. |
| | For a third fatal flaw, explain where this design flaw could have probably affected your system and how you managed to avoid it. | Toine | 1h | 30min | yes | |
| As a developer, I want the code to | Refactor the SpriteBase so that it doesn't have too many | Toine | 4h | | | See Additional Tasks Done. |

| not have too many responsibilities | responsibilities. | | | | | |
|---|---|---|---|---|---|---|
| As a player, I want the score and lives that I have achieved in a previous level to be taken to the next level. | Carry over scores to the next level | Matthijs | 3h | 2h | yes | It took a long to find where it went wrong. In the end it was actually one line which made the error occur. |
| | Carry over lives to the next level | Matthijs | 3h | 2h | yes | Same problem as above. |
| As a player, I want to play with my friends | Add second character to game, with score, controls, and regular properties. | Jim | 3h30 | 4h | yes | |
| | Create the option in the menu for a multi player game | Jim | 30min | 30min | yes | |
| | Create consequences when one player dies. | Jim | 1h | 3h | yes | See Main Problems Encountered. |
| | Create sprite for second player. | Jim | 30min | 30min | yes | |
| | Create extra information in the help-screen. | Joost | 2h | 2h30 | yes | |
| | Create consequences for | Joost | 1h30 | 1h | no | See Main |

| | | | | | | |
|---|---|---|---|---|---|---|
| | winning a level. | | | | | Problems Encountered. |
| | Create consequences for game over | Joost | 1h | 1h | no | See Main Problems Encountered. |
| | Fix displaying lives and score | Joost | 1h30 | 2h | yes | |
| As a player, I want the game to be able to play on Linux. | Application exception for Linux needs to be resolved. | Matthijs | 2h | 7h | yes | Had problems getting everything to work with relative paths for usability with a jar. |
| As a developer, I want to remove unnecessary code. | Remove all JavaDoc for @Override methods | Toine | 30min | 15min | yes | |
| | Remove comments from Instance Variables | Toine | 30min | 1h 30min | yes | Also fixed a lot of other style issues after this |
| | Fix that all model classes have JavaDoc | Lili | 1h | 1h | yes | |

# Additional Tasks Done

### Refactor SpriteBase

All objects in Model that needed a sprite were extending SpriteBase. Since this wasn't a correct responsibility division, we refactored the code so that the objects in Model contain a sprite instead of extending one.

Done by: Toine, 3h and Jim, 6h.

### Bubble Bug Fix, Purifying The Observer Pattern

The branch started out as a bug fix, but the when solving this the real cause was how the observer pattern was implemented. The problem was that the update function of the Observables was called in a Timer in an Observer. Of course this should not be the case. The refactor resulted in a way purer Observer Design Pattern, which is a lot more stable because the old version caused a lot of ConcurrentModificationExceptions and NullPointerExceptions.

Done by: Jim, 4h.

# Main Problems Encountered

### Tasks were accidentally done double.

Description:

When defining the sprint plan Jim and Joost coincidentally planned 3 'consequence' tasks which all took account for:

- what happens when a player dies?
- when is it game over and what happens afterwards?
- when is a game won and what happens afterwards?

When handling the first issue, Jim also coincidentally fixed the second and third option when doing this, because the task wasn't strictly enough defined.

Reaction:

All tasks should be defined well and should be divided strictly. When multiple tasks are overlapping, it is wiser to assign all of those to one person, to prevent work being done twice.

## A not-working version was handed in.

Description:

When the working version has handed in, it was a version that crashed on startup. The cause was a combination of different OS and merge conflicts.

Reaction:

When the grading told us that the software did not start, it was fixed as soon as possible. Shortly after, it was discussed within the group how this could happen, that it never should happen and how it is never going to happen again.

# Adjustments for the next Sprint Plan

- Don't push things that don't fully work.
- Start earlier in the week with our work.
- Add more comments to larger pull requests.
- Pay more attention to testing after merging.