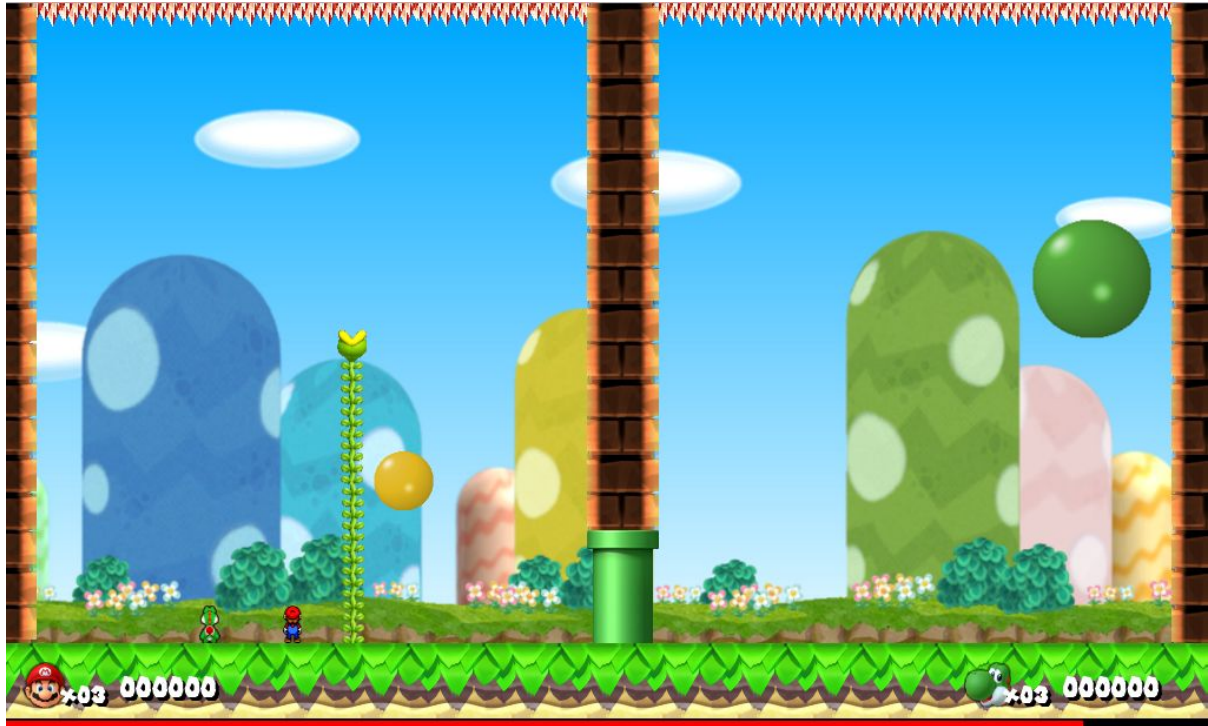# Software Engineering Methods

Reflection on the project

Friday November 4th, 2016

By:

Dana van Hassel   4441494

Matthijs Kok   4437659

Sterre Noorthoek   4375769

Timo van Asten   4485270

Wouter Raateland   4384938

Review

Our code changed significantly during the sprints. In the first few sprints we did not use design patterns as much as we should have. The goal of the first assignment was to put us under pressure, and not care a lot for code quality. This made our technical debt quite large after the first few weeks. We spent the next assignments improving our code, implementing design patterns, paying technical debt, and adding features requested by our TA. We implemented a lot of design patterns, namely: Singleton, Factory, State Machine, Observer, Builder, Strategy and Decorator. In the first few weeks when we were forced to implement these design patterns, it seemed really devious and unnecessary. However, weeks later it became clear why and how this design patterns helped the program to be more readable and more extensible.

Our process

During the Software Engineering Methods lab we have learned from the different aspects of software development. Firstly, we had to apply what we learned during the lectures. This was a good way to make sure we really understood the material. Secondly, working in a team created some challenges. We experienced how important, difficult and time consuming communication can be. Especially during the first two weeks there was no overview about who was working on what topic. After the first sprint we decided to work on this. Every Monday we looked at our schedule and tried to pick two mornings or afternoons that most of the group members were available. Then we rented meeting rooms and came together to work on the assignment. Also, we made a clear backlog and agreed to stick to it. This way we learned about working in a group and improved our communication skills. Still, the teamwork was not perfect and there remain points to be improved. After a while, some members of the group ran ahead of others. This is natural as there will always be a skill difference. We did document our code, but not the overall flow of information. In the future we need to write more about the purpose of our code, instead of what logic it performs. We should do this in the class JavaDoc. In addition, there should have been more documentation around the pull requests. All together, we will definitely use what we have learned in this course in the future to design and implement software systems.

The course

After having reflected ourselves, we also reflect on the course. First of all, it was a fun idea to make a game. We really enjoyed it. However, concerning the organization of the course we have some points that should be improved. And we have brainstormed to think of how this could be done.

To begin with the sprints: in our opinion, and the opinion of all other groups we spoke to, the sprints were way too short. We had less than five days to work on the assignment. Taking into consideration that we were supposed to attend all lectures while we all had different variant courses (which have lectures on different times), hand in deadlines and weekly assignments from other courses, do self study and other activities like work and social activities, this was quite a difficult task. All together, there was barely any time to discuss and work on the assignment together. It would be nice to also have the weekend to work on the assignment. We think it would be better to have sprints from Monday to Monday the week after (seven days). Now we did not

do a lot of work on the assignment during the weekends, since we had already handed in the previous assignment and had to wait for the meeting with our TA to work on the next assignment.

Apart from the fact that sprints were too short, we also think the assignments were too large. For us it seemed like this course is 100% a project instead of a course with weekly assignments. The practical parts cost a lot of time and are only 50% of the grade of this course. We also had to put time and effort in the other part of the Software Engineering Methods course and of course in the other two courses. We think a lot of students were forced to give less priority to their other two courses in order to pass the Software Engineering Methods lab. A solution would be to make the assignments smaller. That way we would still apply the techniques learned in the course and learn about working in a group, but it would not affect other courses negatively.

Conclusion

Apart from that we thought it was a fun idea to create a game for a course, we can not think of a single positive thing to say about this course. The wrong choices were made regarding the work pressure distribution, making the course feel like a project, while it is not. This sound echoes through all other groups we have spoken to. We are sorry to say that we did not like this course.