Weekly Meeting Sprint 3 - Week 17 (lv. 5)

Scrum Master: William Nordgren
Secretary: Jakob Persson and Erik Andersson

Customer Value and Scope / Application of Scrum

- the chosen scope of the application under development including the priority of features and for whom you are creating value / the sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)

Our main goal this sprint has been to deliver a MVP for the customer and work on features that add value for the product owner by avoiding gold plating. We revised our scope with the product owner early on in the sprint to make sure the development team and product owner are on the same page.

- your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders

We have revised our definition of done, since the previous version stated that in order for something to be done, the product owner has to give their approval, but we learned that we don't want to ship anything to the product owner that isn't done according to our definition of done, and therefore we had to change our D.O.D to avoid the paradox that we can't ship anything not done to the product owner, but for something to be done it has to be shipped to the product owner. This iteration of our D.O.D requires the development team to give its approval for a feature for it to be considered done.

Social Contract and Effort

- the time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)

In this sprint, we delivered less even though we spent roughly the same amount of time. This is the case since in this sprint a lot of time has been spent on fixing bugs, merging and making sure all the parts fit together. In the first sprint less time had to be spent on this and thus could be spent on developing new features. A potential fix is to lower our velocity when we know we have bugs to deal with to give ourselves a margin of error for when bugs appear, since it's impossible to find all of them at the start of a sprint.

Design decisions and product structure

- which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)

We have decided that our project has a simple enough structure that is divided into sub-parts, and these sub-parts do not require diagrams or models to be understood. We mainly use comments as documentation for our code and to make it more understandable/readable for others.

Application of Scrum

- best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)

In order for us to develop as a team in an agile way we make sure to continuously teach and learn from each other, since we all have different knowledge about different topics. We use different online sources for us to learn about how Unity works, since this is a software we had not worked with earlier. The same goes for using C# as the programming language.

We noticed that a few of our user stories were larger than velocity 5 and it was difficult to finish these during the sprint. Our reflection of this is that user stories larger than about velocity 5 should be divided into two or more user stories.

Side note till thesis:
Since we did not have any bugs in the first couple of weeks, we did not plan for the number of bugs that needed to be fixed when heading into the next sprint. The time it took to fix the bugs was more than we thought and therefore we used time to fix bugs that we had planned to use on working with other user stories. The lesson here is to give ourselves some margin to spend on fixing bugs and that extra time can be spent on smaller user stories if we are lucky and don't need to fix bugs.

Sprint 4 planning

At the moment we have some bugs and therefore we need to spend some time fixing those. We will lower our velocity next week because of this. The planning for sprint 3 will be made on monday instead of today (friday), since we have discussed and decided that it is better to start clean each week.