Meeting Agenda

Group: Super duper Omega gruppen
Date: 2020-09-07
Chair: Oskar
Participants: Samuel, Oskar, Erik, Behroz, Sebastian

Objectives (5 min)

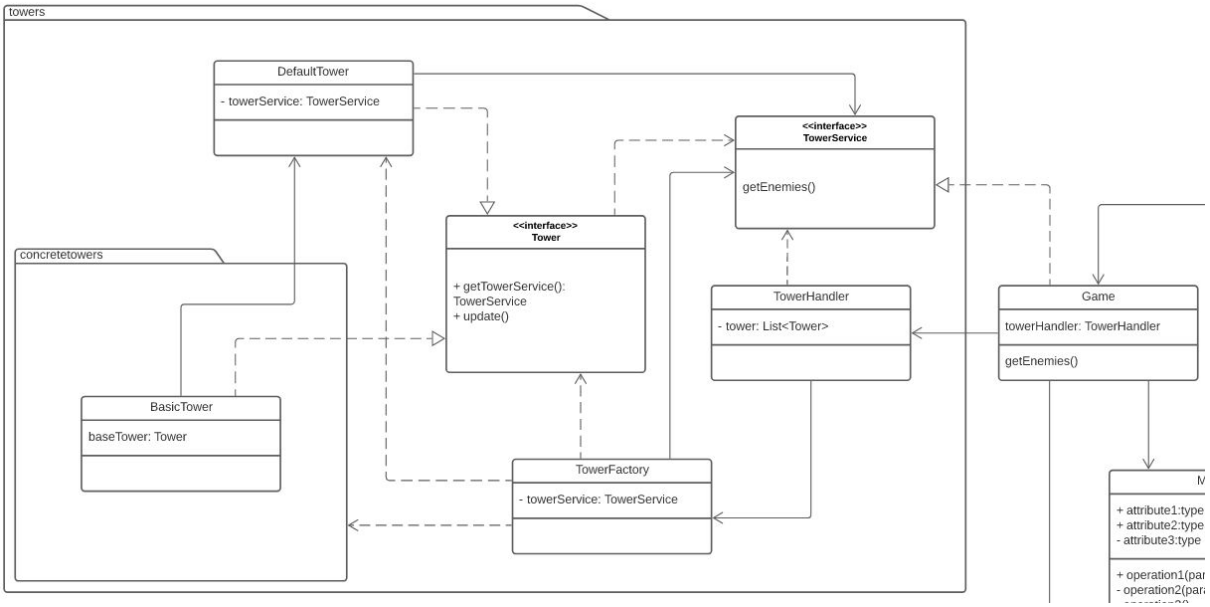
- Create new tasks and assign them
- Create DoD for new user story and change DoD for orange user story
- Create Domain model, according to today's lecture
- Draw basic GUI

Reports (15 min) from previous meeting

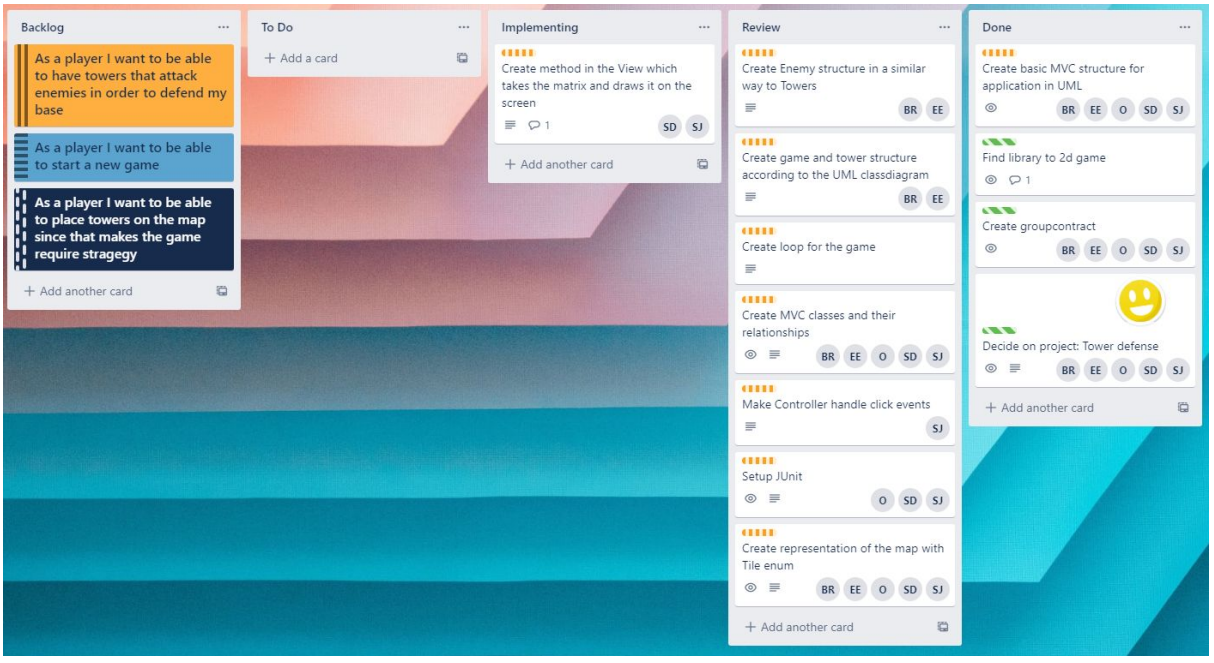
After talking to Pelle last friday we got feedback that one of our user stories was too large so we broke it up. Instead of having “place towers” be in the same story we moved that to a separate story so that the first one we are working on will only have towers that are hard coded into the code.

We also got feedback on our UML class diagram for towers. He thought that having an adapter between the interface that Game implements and the towers might be overengineering it and that the towers should have a reference to the interface directly.

Therefore we removed that adapter and renamed the interface TowerService instead. See diagram below:



We formalized our development process to have these 5 steps:



Backlog: where we create stories that might be implemented some time in the future.

Todo: tasks that are created from those stories. The tasks have a tag with the same color as the story that it corresponds to. Here the tasks also get assigned to people.

Implementing: writing tests and implementing the feature in that task

Review: Someone other than the person who wrote the code looks at the requirements of the feature and makes sure that they are all met. They should also look over the code to see if there are possible improvements.

Done: the task is finished.

Discussion items (35 min)

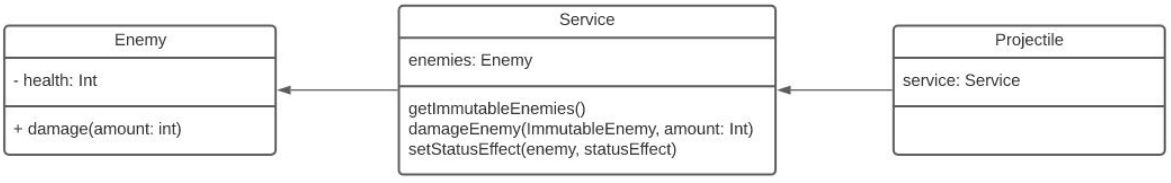
When thinking of tasks to our current story we decided that we could further divide that story. We removed the requirement of a base that the enemies attack and made that a separate story.

We created a new story for having multiple towers: As a player I want a panel with a selection of multiple towers that I can place because that improves the game's creativity.

We also defined DoD for all our current stories.

We discussed how to best implement a Vector. We want the position of towers, projectiles and enemies to be represented by a 2d vector with an x and a y component. Projectiles and enemies can be on decimal positions but a tower can only be on whole number positions. Therefore we need one IntVec and one FloatVec. We could implement this with a generic type Vector<T extends Number> but that would mean that every time we need to use it we have to refer to it with Vector<Integer> which seems more cumbersome than just writing IntVec. The other option is to write two different classes but that means that we duplicate a lot of the code. We investigated if java has any typealias functionality but we didn't find anything. Despite this, we are still leaning towards using a generic type at this time.

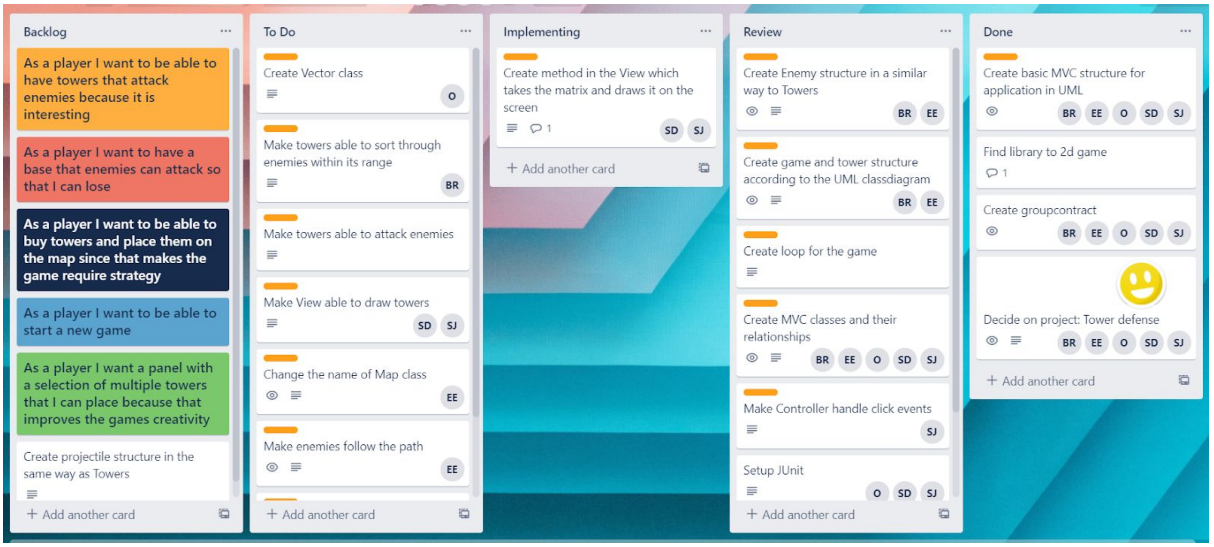
We discussed how enemies should be damaged. Currently we want to keep external references to enemies immutable. Projectiles should be able to know the positions of enemies but not be able to call .update() on them. This leads to a problem since projectiles might want to call .damage() on an enemy. In order to do so their reference would no longer be able to be immutable. We discussed if it might be possible to have a service in between which can damage enemies.



The problem with this is that the service will have to convert the ImmutableEnemy reference to an actual enemy. This can be done by either casting it (which is bad) or storing the enemies in some kind of Map which converts from ImmutableEnemies to Enemies (which is also a bad solution). At the moment we are undecided on how to implement this and decided to ask a professor.

Outcomes and assignments (5 min)

We created new tasks and stories. Our current board looks like this:



This meeting took 2h50m, mostly because we discussed how to implement things. Therefore we didn't have time to do a domain model or a GUI. We decided that certain topics about the code design should be done as tasks and not during the meeting with everyone present. Then we get more time to do more important things during the meeting and people can decide for themselves if they would rather partake in the discussion or work on a separate task.

Wrap up

Still don't know how to deal with projectiles and how to damage enemies. We decided not to do all Objectives on the agenda and postpone them to the next meeting.

Next meeting is on 2020-09-10 13:00