Meeting Agenda

Group: Super duper Omega gruppen
Date: 2020-10-05
Chair: Erik
Participants: Samuel, Oskar, Erik, Behroz, Sebastian

Objectives (5 min)

- Discuss where to place WaveHandler
- Discuss how to refactor selected tower
- Create new tasks / assign them
- Decide when to do peer review

Reports (15 min) from previous meeting

Everybody helped adding author and purpose to all of our java files.

Erik renamed MageBear to BearryPotter.

Samuel: Added graphics for sniperBear and BearryPotter as well as particles and projectiles for them and made them display.

Oskar refactored Wave to be an Iterator instead, as per Pelles recommendation. Implemented spawnMultipleWithDelay method in EnemySequence. Created WaveHandler which manages active waves and WaveData which defines all the waves for the game. Merged all the Wave code into master.

Behroz created sniperbear the first hitscan tower.

Sebastian: Made so view takes in consideration that screen width and height actually are not these sizes. Added a next wave button and connected it to the model.

Sebasian and Samuel: Added a tower panel where you can select towers to place. Also added a ghost tower that is displayed when you select a tower in the TowerPanel that follows the mouse when inside the map. Also added range to the ghost tower so you see what range the tower will have when placed.

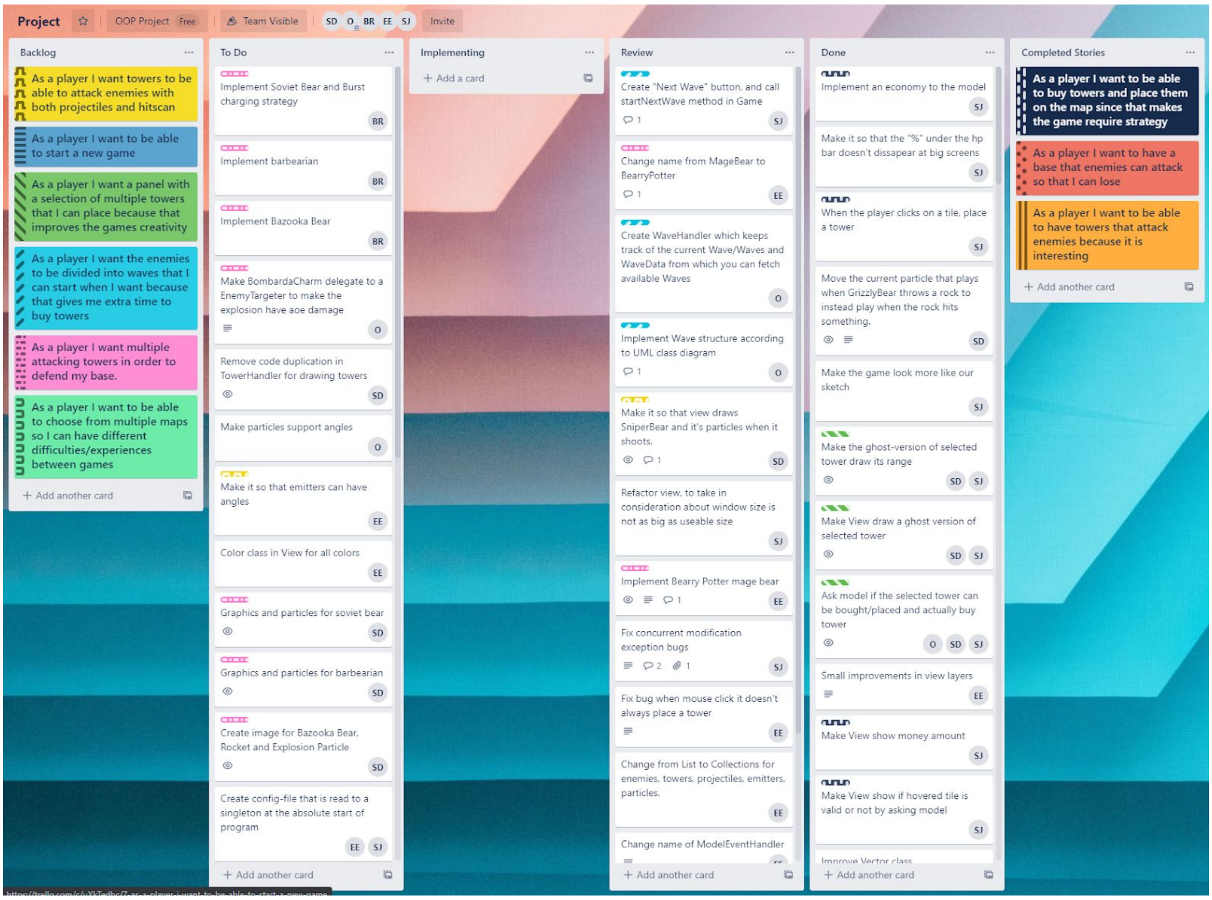
Discussion items (135 min)

Right now, WaveHandler is inside EnemyHandler which is inside Game. The only part of the code creating Enemies is WaveHandler, which should mean that only WaveHandler has access to EnemyFactory (which it now doesn't). We briefly discussed where WaveHandler and EnemyHandler should be, or if we should remove one of the classes, but realized this is harder than what we thought and made a task on Trello to discuss it further (to avoid taking up meeting time).

In between we discussed that it would be nice if the enemy that has reached the farthest is drawn last for the player to see its HP bar. We also discussed giving SniperBear a +/- 2 updates randomness in its charge time to decrease lag when many SniperBears (> 20 or so) fires at the same time.

Right now we have one switch statement for getting a tower's range and one for creating the tower. One possibility is using a HashMap with Classes as keys and factory method lambdas as values.
But we realized that combining everything - a lambda for creating the tower, its range, the path to its image - as a single object might be more beneficial.
Another idea is that View creates a tower instance to read its range/class.
We still don't know what to do here at the moment, but one thing we decided is to group related values for every tower in a data object.

We created more tasks and assigned them as well as added a new user story, now our scrum board looks like this:



Project ☆ | OOP Project Free | Team Visible | SD O BR EE SJ | Invite

Backlog

- As a player I want towers to be able to attack enemies with both projectiles and hitscan
- As a player I want to be able to start a new game
- As a player I want a panel with a selection of multiple towers that I can place because that improves the games creativity
- As a player I want the enemies to be divided into waves that I can start when I want because that gives me extra time to buy towers
- As a player I want multiple attacking towers in order to defend my base.
- As a player I want to be able to choose from multiple maps so I can have different difficulties/experiences between games

+ Add another card

To Do

- Implement Soviet Bear and Burst charging strategy — BR
- Implement barbearian — BR
- Implement Bazooka Bear — BR
- Make BombardaCharm delegate to a EnemyTargeter to make the explosion have aoe damage — O
- Remove code duplication in TowerHandler for drawing towers — SD
- Make particles support angles — O
- Make it so that emitters can have angles — EE
- Color class in View for all colors — EE
- Graphics and particles for soviet bear — SD
- Graphics and particles for barbearian — SD
- Create image for Bazooka Bear, Rocket and Explosion Particle — SD
- Create config-file that is read to a singleton at the absolute start of program — EE SJ

+ Add another card

Implementing

+ Add a card

Review

- Create "Next Wave" button, and call startNextWave method in Game — 1 — SJ
- Change name from MageBear to BearryPotter — 1 — EE
- Create WaveHandler which keeps track of the current Wave/Waves and WaveData from which you can fetch available Waves — O
- Implement Wave structure according to UML class diagram — 1 — O
- Make it so that view draws SniperBear and it's particles when it shoots. — 1 — SD
- Refactor view, to take in consideration about window size is not as big as useable size — SJ
- Implement Bearry Potter mage bear — 1 — EE
- Fix concurrent modification exception bugs — 2 — 1 — SJ
- Fix bug when mouse click it doesn't always place a tower — EE
- Change from List to Collections for enemies, towers, projectiles, emitters, particles. — EE
- Change name of ModelEventHandler

+ Add another card

Done

- Implement an economy to the model — SJ
- Make it so that the "%" under the hp bar doesn't dissapear at big screens — SJ
- When the player clicks on a tile, place a tower — SJ
- Move the current particle that plays when GrizzlyBear throws a rock to instead play when the rock hits something. — SD
- Make the game look more like our sketch — SJ
- Make the ghost-version of selected tower draw its range — SD SJ
- Make View draw a ghost version of selected tower — SD SJ
- Ask model if the selected tower can be bought/placed and actually buy tower — O SD SJ
- Small improvements in view layers — EE
- Make View show money amount — SJ
- Make View show if hovered tile is valid or not by asking model — SJ
- Improve Vector class

+ Add another card

Completed Stories

- As a player I want to be able to buy towers and place them on the map since that makes the game require strategy
- As a player I want to have a base that enemies can attack so that I can lose
- As a player I want to be able to have towers that attack enemies because it is interesting

+ Add another card

We decided that we should work on our review of other projects tomorrow.

Outcomes and assignments (5 min)

We don’t know where to place WaveHandler. (Will discuss later)

Implement TowerInfo to get rid of multiple switch statements.

Tasks created and assigned for every story except 2.

Wrap up

Next meeting: Thursday at 13:00 2020-10-08

Peer reviewing: Tomorrow at 15:30