## Reasoning behind my key design and architectural decisions.

First I implemented the first exercise as it was requested.

For architecture I went with my general style. I tried to point out elements where the system needs expansion for sure. I used inheritance on those points, created abstract classes, and in the children I implemented the unique behaviors. And for the balancing variables my rule of thumb was, that if it can make the game too easy AND too hard, then it has the right amount of impact on the game. I used HUD to create the HUD widget and add it to the viewport, where I tried to collect all relevant information for the player, with in mind to use tick only on gradually changing elements.

I created the spawners which were the cornerstone for level building.

When the framework was ready as requested, I played with it and collected ideas, what would change the game style. I knew my tools are the new enemies, and the new power ups.

I wanted some completely different enemy from the first one.

The first enemy is rather ranged. They aren't really letting you up close, so you have to shoot accurately for them. So I wanted to make the second enemy close combat. But I wanted them to be completely different, so I made them invincible when they were far away. To make it fair, I took their shooting ability. When they are close enough, they lower their shield, and they lock on and try to collide with you. Without the visuals it was frustrating that I couldn't see when they are vulnerable, so I added a very basic one.  
With this new enemy type you have to tactic which ones you let closer first.   
Some nice byproduct of this invincibility, that the rocket powerup (the one from the first exercise) is not that useful for single enemies. If it goes by them, does no damage. So the invincible enemies can be used as splash-damage points. You keep your distance, you can hide behind them, since they shield you from other bullets as well, and when the flock of first enemies comes near to the invincible one, I simply shoot them with my rocket, so the whole flock is gone.

For the new powerup I decided to copy the enemy toolkit. I wanted to have a dash on the player too. But I wanted some defensive power-up as well. So I mixed them up, and came up with the idea that you can heal yourself when you dash into enemies. This felt good, and even challenging with the first enemy, because they start to shoot at you when you are approaching them, and try to sidestep you.

The game Rules were implemented in the game mode, so if we want a different one, it's easy to swap. Here you can find the enemy spawning rules.

Every balancing variable is exposed. So the designer could play with them:

* The length of the session
* The base spawn rate
* The growth of spawn rate
* Enemy\character\projectile speeds
* How big area you see from the map
* Enemy trigger distances, where will they switch behavior
* How long a powerup is active
* Powerup range
* How often a powerup is generated on the field
* How much health you get back from dash-collision
* etc…