Sprint 4 Discussion Topics

In the beginning of the sprint we felt we had to deal with our somewhat decreasing package structure quality. The amount of circular dependencies we meant to fix in previous sprints had grown too large. Therefor we dedicated the first day in the sprint to go through our system structure and get rid of all circular dependencies.

The approach was that every package should have it's own environment which handles all the traffic in and out of the package. Previously we had a Game class that contained almost all the components of the game, which meant the there were traffic both from and to the Game class, which in turns created circular dependencies.

By looking at the call hierarchy of methods/classes we moved our classes into more appropriate sub-packages and created an environment class to handle the components of it's package. By having all traffic go through these environments instead of just one Game-class we were able to remove our circular dependencies. This also makes testing easier, since environments can be swapped with do-nothing implementations or spy-derivatives of existing game logic (local dependency injection).

The cleanup also involved renaming classes according the convention set up for this project and getting rid of unnecessary old code.

The model for generic listeners was created in order for controllers to pass a way for other models to update functionality that also should be reflected on the view. This lets the models be unaware of the view since the controller has already defined the behaviour of the listener. Ultimately giving a good package structure without the above mentioned circular dependencies.

During the sprint we found that some parts of the game behaved unexpectedly on collisions. A helpful feature, found after some time spent trying to figure out the problem, was the Box2d debug renderer. This could draw all the hitboxes so we could see exactly where they were. This was also a helpful feature for testing the obstacles in the maps.