

The implementation of the Achievement System will work with an Observer design pattern. This means that there will be a few classes defined as “subjects”. These are the game classes where actions happen that are interesting for achievements. The subjects will have a “notify” method which will notify every class that wants to be notified of a specific change or state that happens in the class.

Example: when the player fish dies, a notification will be sent out.

The observer will read out the specific notifications a subject sends out. For each property that can be tracked, there will be an observer. The property classes will track how often a certain property has happened.

Example: the death property class will increase a counter with 1 every time it gets a notification that the player fish dies.

If the property reaches a certain state, it will send out a notification (and will therefore also be a subject) to the main achievement class.

Example: when the death counter has reached 5 (which means 5 deaths), it will send out a notification.

The main achievement class will then check if the certain state of the property also means that an achievement is unlocked by taking all “unlocked” properties and comparing them to the achievement requirements which are stored in an achievement status class.

The achievement status class will store all the requirements to obtain an achievement while the achievement class will have the responsibility of changing the status of the achievement and granting an achievement to the player when the rules are met.

This set-up will also make it easier to implement achievements that depend on unlocking more than 1 property.

The achievement class will then send out an order to the achievement screen, which will make a colour adjustment to turn a greyed out achievement picture into a colour-version to indicate the achievement has been obtained.

The biggest advantage of this design is that we do not actually have to change the state of the classes where the actual changes are made. Therefore the game can run normally and will not get the responsibility of also creating and