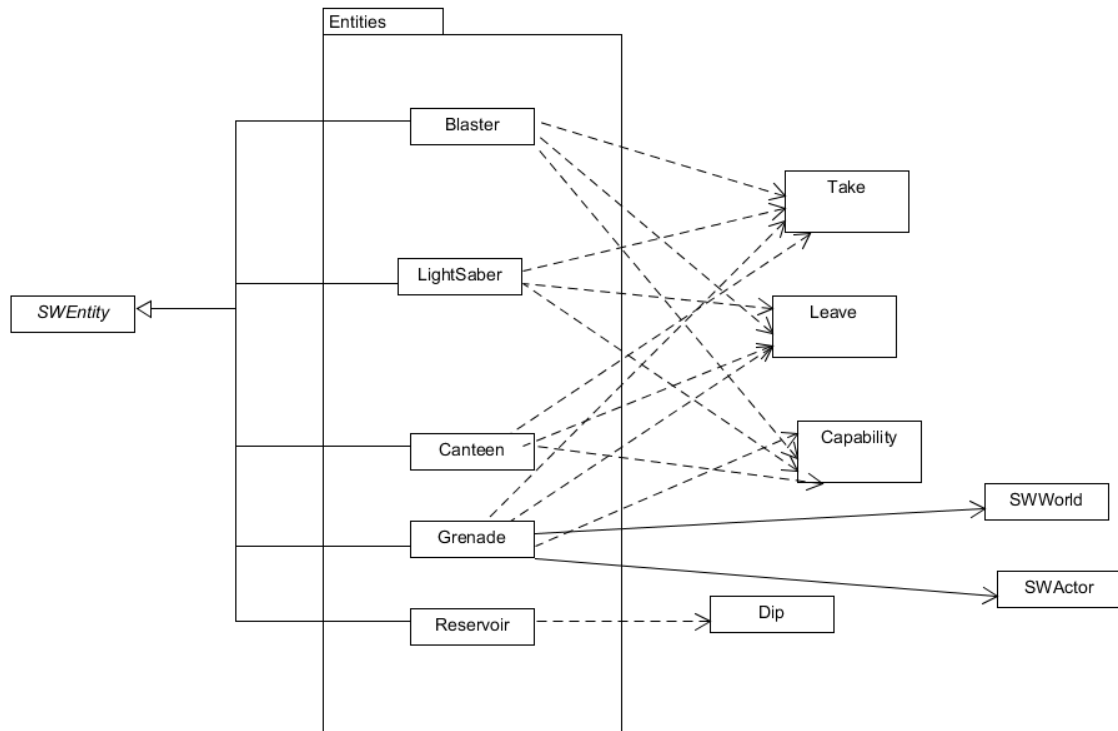
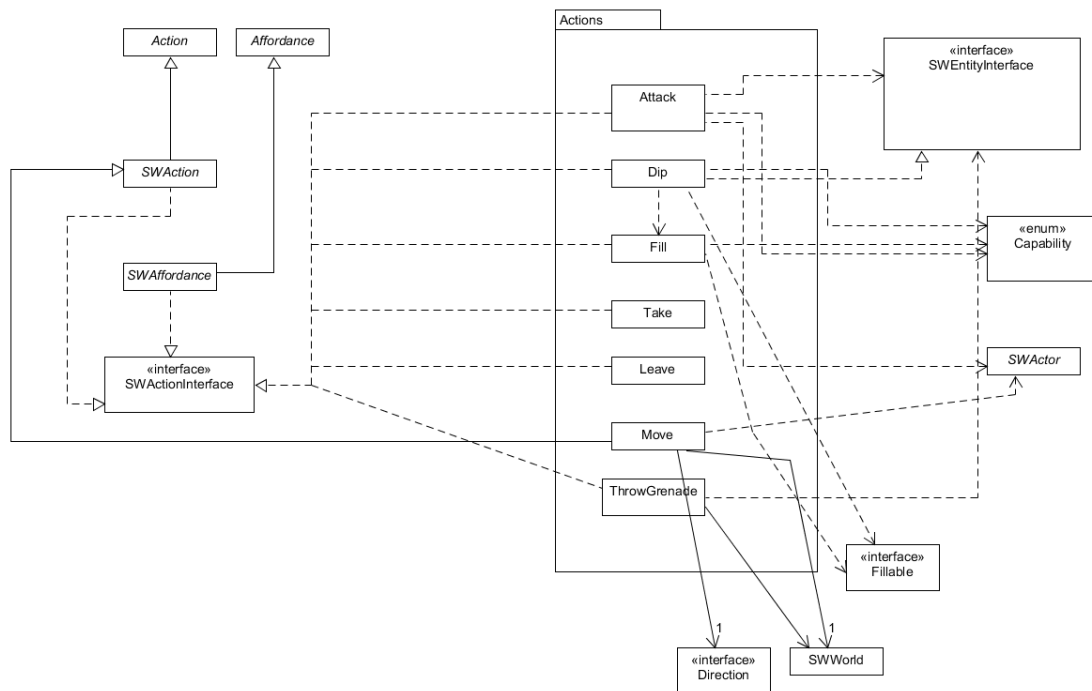


1. Grenade:

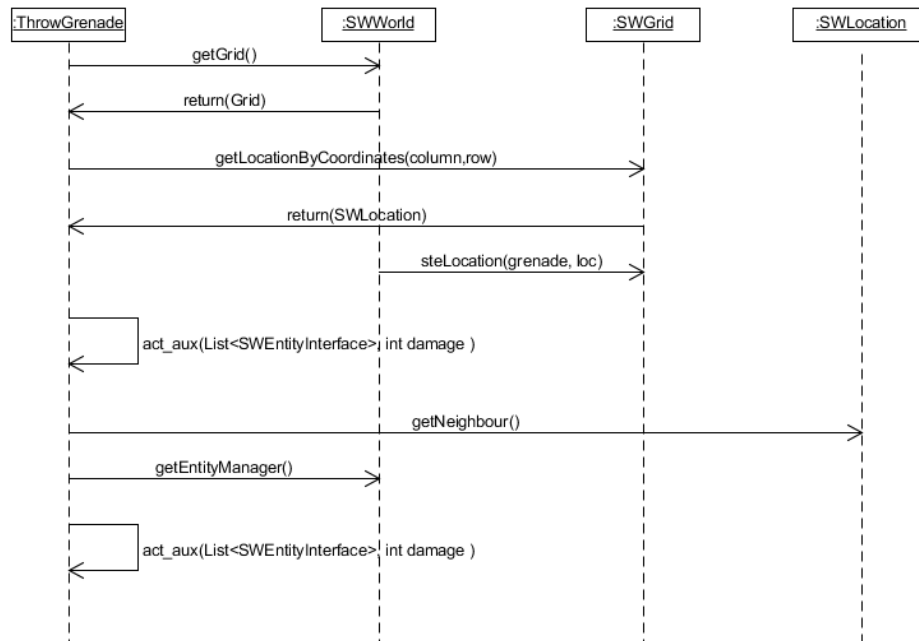


We decided to make a grenade a part of the entities package, this is because a grenade is an object that can be picked up or put down by any actor, just like the other entities we have (blaster, lightsaber, canteen and reservoir). It has associations with **SWWorld**, because we need to know which world the grenade is in so we can access its grid and inflict damage on the neighbouring entities as required, and **SWActor**, because we need to know which actor owns the grenade to prevent them from taking damage from it.

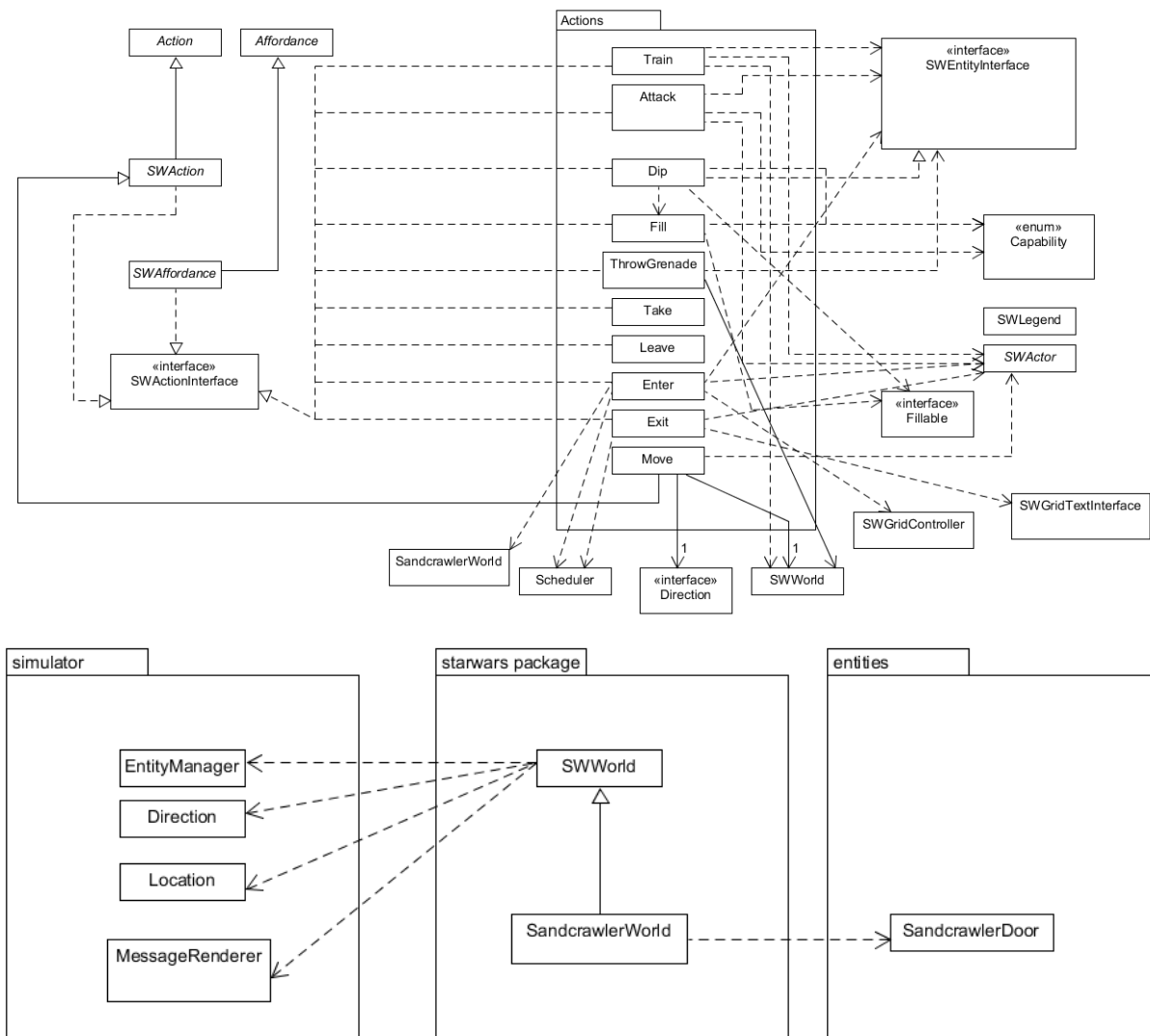
2. ThrowGrenade:

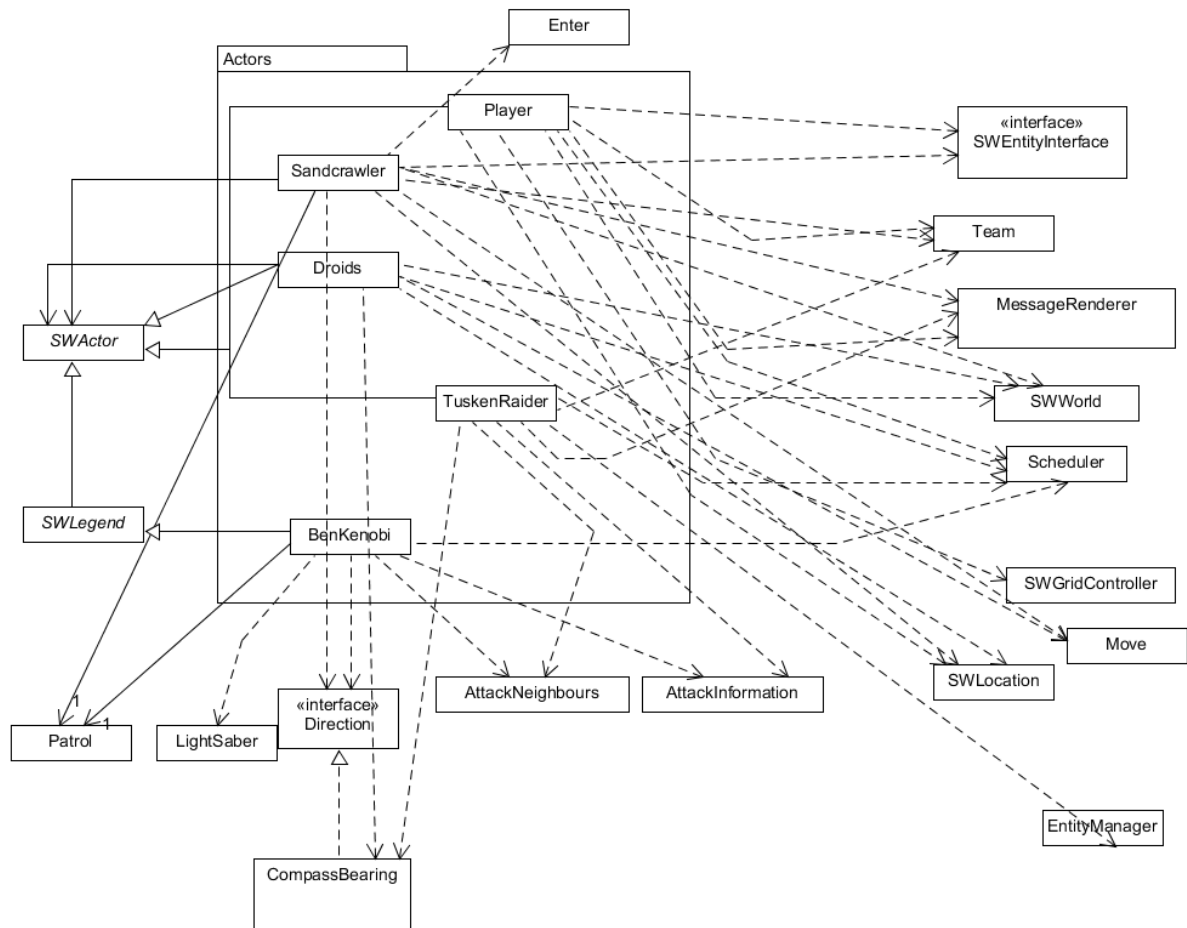


We decided to make a class `ThrowGrenade`, which we put in the `Actions` package. We did this because throwing a grenade seemed more like an action than anything else. The class needs to know about the `SWWorld` that the action is being done in so that it's able to inflict damage accordingly. We need to know about the `entity` interface so we can get the entities that exist in a given location and reduce their hitpoints.



This is a sequence diagram for an actor throwing a grenade and damage being inflicted on the location it landed and the location one step away from it. First you get the grid the action is happening in, then the user enters a location to throw the grenade. After they enter a valid location, the grenades location is reset. After it is reset, damage is inflicted on all entities in that location. After that, the location of the neighbor is found and we get all the entities in that location. We repeat the damaging of the entities in that location, this process will continue for all neighboring locations, as well as locations 2 steps away.





We created a new class called “Sandcrawler” in the actor package. This actor object will take droid objects that it encounters in the map and store them in a temporary ArrayList. These will be later stored in the interior grid of the Sandcrawler, when it is initialized.

We also created two new actions; Enter and Exit. These actions allow any SWActor to enter and exit using the SandcrawlerDoor, provided they have a high enough force level.

The interior world of the Sandcrawler is initialized when an actor enters using the Enter action. The droids are then placed onto the grid.

When the actor is exiting the SandcrawlerWorld, the droids are put back into an array list and the actor is returned to the SWWorld.