# Design Rationale

## Force abilities

* Some characters can use the Force, others cannot.
* Actors which can use the Force indicate this capability by an attribute, canUseForce
* They also have a certain level of Force ability indicated by “Force points”
* At different levels, the characters have different Force capabilities, ie. they can perform different actions or can resist other actors’ actions on them
* The capabilities are indicated by classes which implement the SWForceActionInterface
* The “Force actions” created currently are:
  + Train – increases Force points of target
  + ForceMove
  + LightsabreAttack (requires Lightsabre to be carried)
* This design improves extensibility of Force abilities and characters which can use the Force.
* If further Force-capable actors or actions are to be added in future, this design maximises the reusability of the code.

## Droids

* Droids are considered another type of actor in the game – but with certain unique attributes
* They cannot use the Force whereas there is a possibility that other actors can.
* Droids have owners which can be any other actor, but it cannot own other characters itself – and follow their owners. Each droid has one owner.
* Therefore, Droid should not inherit from SWActor but from Actor directly.
* Droids can also be considered as composite objects – they are made up of DroidPart ‘s
* Droid actions
  + There are some actions only droids can perform
  + These classes must implement SWDroidActionInterface
  + Once more, this allows us to reuse code if we need to add further droid actions in future
  + Actions which relate to droids but can be performed by other actors do not need to implement this interface
* Regaining health with oil cans – this necessitates a new entity, OilCan, which droids can pick up like any other.

## Healing

* Healing is implemented as an action which can be carried out if actors encounter a canteen, or droids an oil can
* Two SWAction subclasses will perform this role – Drink for SWActor’s, UseOilCan for Droid’s