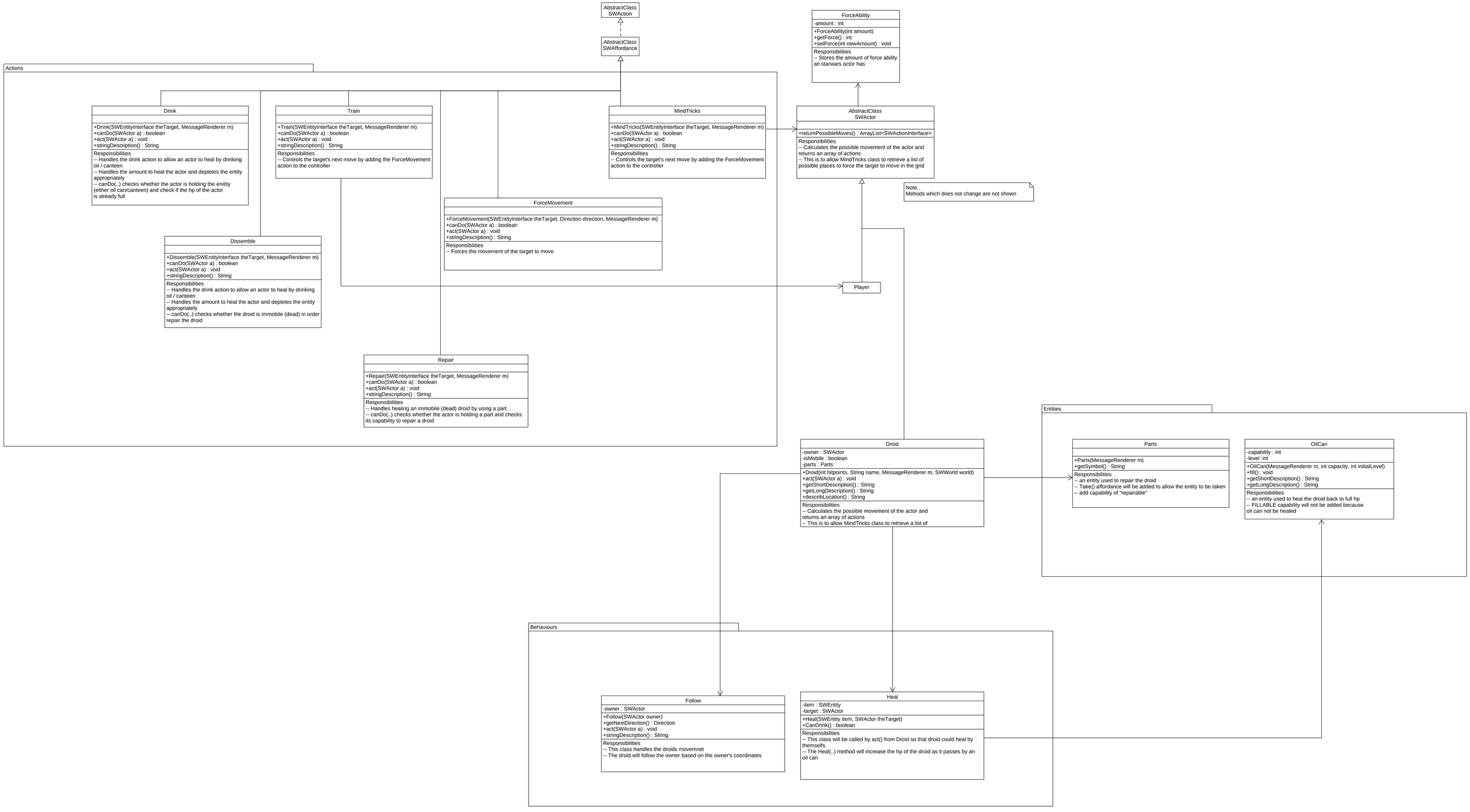
AbstractClass
SWAction
AbstractClass
SWAffordance
ForceAbility
-amount : int
+ForceAbility(int amount)
+getForce() : int
+setForce(int newAmount) : void
Responsibilities
-- Stores the amount of force ability an starwars actor has
Actions
Drink
+Drink(SWEntityInterface theTarget, MessageRenderer m)
+canDo(SWActor a) : boolean
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- Handles the drink action to allow an actor to heal by drinking oil / canteen
-- Handles the amount to heal the actor and depletes the entity appropriately
-- canDo(..) checks whether the actor is holding the enitity (either oil can/canteen) and check if the hp of the actor is already full
Train
+Train(SWEntityInterface theTarget, MessageRenderer m)
+canDo(SWActor a) : boolean
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- Controls the target's next move by adding the ForceMovement action to the controller
MindTricks
+MindTricks(SWEntityInterface theTarget, MessageRenderer m)
+canDo(SWActor a) : boolean
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- Controls the target's next move by adding the ForceMovement action to the controller
AbstractClass
SWActor
+returnPossibleMoves() : ArrayList<SWActionInterface>
Responsibilities
-- Calculates the possible movement of the actor and returns an array of actions
-- This is to allow MindTricks class to retrieve a list of possible places to force the target to move in the grid
Note..
Mehods which does not change are not shown
ForceMovement
+ForceMovement(SWEntityInterface theTarget, Direction direction, MessageRenderer m)
+canDo(SWActor a) : boolean
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- Forces the movement of the target to move
Dissemble
+Dissemble(SWEntityInterface theTarget, MessageRenderer m)
+canDo(SWActor a) : boolean
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- Handles the drink action to allow an actor to heal by drinking oil / canteen
-- Handles the amount to heal the actor and depletes the entity appropriately
-- canDo(..) checks whether the droid is immobile (dead) in order repair the droid
Player
Repair
+Repair(SWEntityInterface theTarget, MessageRenderer m)
+canDo(SWActor a) : boolean
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- Handles healing an immobile (dead) droid by using a part
-- canDo(..) checks whether the actor is holding a part and checks its capability to repair a droid
Entities
Parts
+Parts(MessageRenderer m)
+getSymbol() : String
Responsibilities
-- an entity used to repair the droid
-- Take() affordance will be added to allow the entity to be taken
-- add capability of "repairable"
OilCan
-capability : int
-level :int
+OilCan(MessageRenderer m, int capacity, int initialLevel)
+fill() : void
+getShortDescription() : String
+getLongDescription() : String
Responsibilities
-- an entity used to heal the droid back to full hp
-- FILLABLE capability will not be added because oil can not be healed
Droid
-owner : SWActor
-isMobile : boolean
-parts : Parts
+Droid(int hitpoints, String name, MessageRenderer m, SWWorld world)
+act(SWActor a) : void
+getShortDescription() : String
+getLongDescription() : String
+describLocation() : String
Responsibilities
-- Calculates the possible movement of the actor and returns an array of actions
-- This is to allow MindTricks class to retrieve a list of
Behaviours
Follow
-owner : SWActor
+Follow(SWActor owner)
+getNextDirection() : Direction
+act(SWActor a) : void
+stringDescription() : String
Responsibilities
-- This class handles the droids movemnet
-- The droid will follow the owner based on the owner's coordinates
Heal
-item : SWEntity
-target : SWActor
+Heal(SWEntity item, SWActor theTarget)
+CanDrink() : boolean
Responsibilities
-- This class will be called by act() from Droid so that droid could heal by themselfs
-- The Heal(..) method will increase the hp of the droid as it passes by an oil can