SWActionInterface - unmodified
MindControl- new *remove spacebar from Mind Control
Force - new
SWActor - modified
SWEntitifyInterface – unmodified
SWLegend - unmodified
BenKenobi - modified
MCInformation - new
MCNeighbours – new
Affordance - unmodified

Design Doc

ForceAbility
<u>Implementation</u>

The force ability will be added as an attribute in SWActor. All Actors will have force levels assigned to them similar to hitpoints.

Force
<u>Implementation</u>
Force is a new abstract class. Sub-classes from the Force class include MindControl and other actions that have a force requirement. The Force class would contain the level of force required as well as some other information that are not finalized.

MindControl
<u>Implementation</u>

Prep
The mind control ability will be added as an action. For the Player to use mind control, the player would have to achieve a certain force level. Affordance 'mindcontrol' will be added to SWActor such that all actors that mind control can be attempted on have this affordance. A boolean method will be added to SWActor to indicate True if actor is being mind controlled and False otherwise. It is set to False by default. The actor's sequence of actions (act) is then modified to always check the mindcontrol boolean. If the boolean is True, the actor will skip other behaviors and just perform a movement and reset the boolean to False for the next turn.

Usage
The option to use Mind Control will appear when the Player has a high enough force level and there is an entity that has the mindcontrol affordance in the same space. Using mind control on an actor would obtain the target the same way the Attack action gets the target. The mind control action will then prompt a request for the target's force level. The force level of the user and the target is compared to check if the user has higher force than the target. If the user has a higher force level than the target, the boolean mind control method for the actor will be set to True and the direction indicated by the user will be given to the Actor. This way, the actor will reset the boolean and perform the move in the direction indicated during the next turn. If the user does not have a higher force level than the target, the game will print a message indicating failure to use mind control.

For NPCs to use mind control, similar to the AttackInformation and AttackNeighbours behavior, mindcontrol will be added as MCInformation and MCNeighbours, to determine whether there are Actors around that are suitable to be mind controlled. The direction that NPCs choose will be random. The same condition checks as when the player uses mind control will be performed. Choosing between Attacking and using MindControl will be randomized.

Design Rationale

Having force level as an attribute will allow for easy accessing and allow for the player to keep track of his force level easily. OnlyOkayDesigns decided that we would like to have a library of abilities that can be unlocked the higher the force level of the player. This would perhaps create a more dynamic and interesting game. This is why we chose to add the abstract Force class which all actions that require force would extend from. The decision to add affordance to Actors that can be mind controlled (almost all) and performing a second comparison between force levels allows the player to make more decisions rather than the game deciding what decisions a player can make. The sequence in which we chose to mind control NPCs seemed to us as the most effective way whereby we are able to reuse most methods that are already implemented. Using a randomzier to choose between using mind control and attacking makes the NPC more dynamic therefore we chose this. Further changes such as using mind control if NPC's hitpoints are below a certain threshold may be made in future implementations.
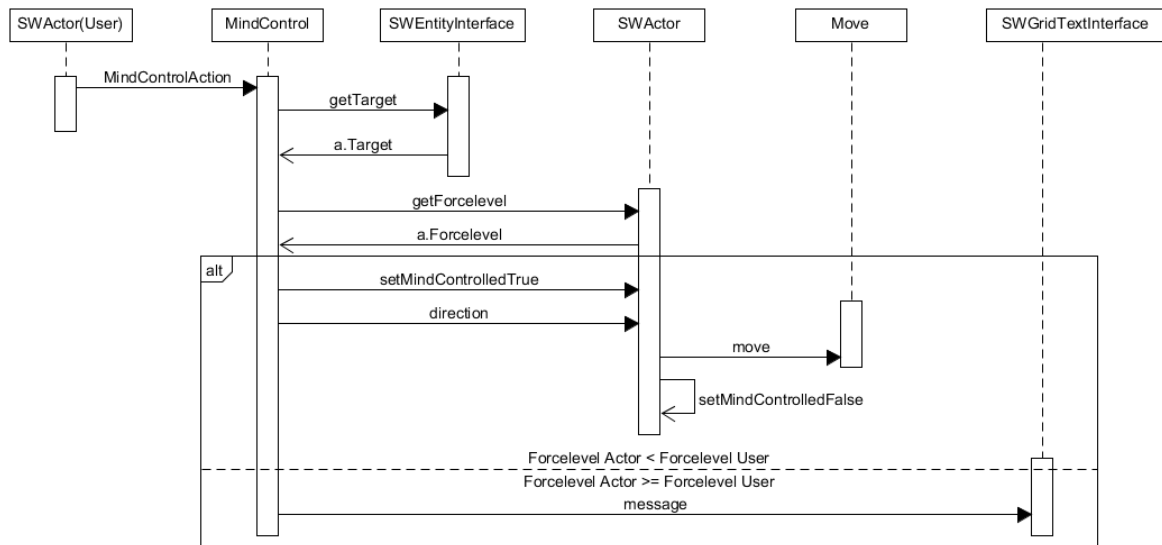
Sequence Diagram for MindControl