**Design changes**

Group abc decided that we would use the sample solutions that Spike Spike has provided to continue on with assignment 3. This was due to the design flaws that we had done in assignment 1 and 2 which would require a lot more work being done to adjust to the new functionality in assignment 3.

**Darth Vader and Force Choke**

* Darth Vader would inherit from the SWOrganicActor class.
* A wander behaviour was given to Darth Vader to allow him to walk.
* A new capability will have to be created to allow him to force choke other users.
* ForceChoke
  + A FORCECHOKER capability was created to satisfy the canDo method of the ForceChoke class.
  + DarthVader would also be given a forceChoke behaviour
  + The ForceChoke behaviour that Darth Vader will use checks the current location if there are any other entities in the location and passes on that Target and MessageRenderer to the ForceChoke action which will be where the force choke occurs. The forceChoke action will check if the target is Luke and randomize some numbers to check if they will either be force choked/attacked with a lightsaber or changed to the dark side. If they aren’t luke then there will be a 50% chance that they will be force choked.

**DeathStar and YavinIV creation**

* InitializeDeathStar and InitalizeYavin methods were created in SWWorld to create the grids of the new maps
* The same SWGridController for Tatooine was used as the parameter for both DeathStar and Yavin
* The SWGrid constructor needed to be changed to allow different size grids to be created
  + Previously it needed just factory as a parameter but we changed it to require the factory and also two integer values for the rows and columns of the needed map
* Other methods in SWWorld had to be modified to get the correct map height and width so we made the height/width methods required a string parameter to know which grid height and width to return.

**Stormtrooper and Call Backup**

* Stormtroopers would inherit from the SWOrganicActor class
* Stormtroopers were given the following capabilities:
  + Weapon - Allow them to attack with the blaster
  + CallBackup - Allow them to create a new stormtrooper
* Stormtroopers were given the following behaviours:
  + AttackNeighboursBehaviour - allows them to attack other actors
  + CallBackupBehaviour - allows them to create a new stormtrooper
  + WanderBehaviour - allows them to wander the map in their available directions
* CallBackup
  + Checks if the current location that the stormtrooper is on is empty
  + If it is empty, a number will be randomized, it it is less than 0.05 then it will schedule the CallBackup action for the stormtrooper
  + The CallBackup action retrieves the current x and y coordinates of the stormtrooper and sends it to the createStormTrooper method in SWWorld which will then create a stormtrooper in the same location as the previous stormtrooper.

**Winning and Losing**

* Boolean variables (default set to false) were created in SWWorld to hold the current gameState
* A gameState variable was created in Application to check whether the game will still go on, this will be checked on every turn to see if any of the conditions have been met
* When certain conditions are met, the boolean values in SWWorld were changed to true to match the event that occurred
* For example, if Darth Vader dies, the boolean setWinGame is changed to true - this will also change the gamestate to false in the Application driver class.

**Interstellar Travel**

* Added capabilities
  + Pilot - allows the actor to fly
  + Fly to Tatooine - allows the actor to travel to Tatooine
  + Fly to Death Star - allows the actor to travel to the Death star
  + Fly to Yavin - allows the actor to travel to Yavin
* The player starts off with the Pilot, Fly to Death Star, and Fly to Yavin capabilities and as the player flies to a different map the capability to fly there is removed and the previous map capability is added back.
* Created the Millennium Falcon like other Entities and gave it the 3 affordances which are the 3 actions that was created to fly to each individual map.
* Travelling was done by firstly creating a SWGRID variable to hold current grid and the current grid is the grid that would be displayed which is the grid that the player is on. The player’s location is later set to a specific map depending on the choice of the user.
* The companions that are following will make a check to see if the current map is the map that they are on, if the map has changed means they need to be teleported to the correct map. So Follow Behaviour is changed to automatically change the locations of these companions to the map which the player is on.

**Mon Mothma**

* A class was made to interact with Mon, this action checked if the 3 required actors are together when they interact with her which will end the game.
* Mon was given the affordance of MonMothmaCheck to allow the player to interact with her.
* The player was given the Capability to talk to her.
* If the 3 required actors aren’t there she will say her quote

**Admiral Ackbar**

* Admiral Ackbar was given the Talking Behaviour which every turn he has a 10% chance of saying a quote