Documentation Assignment 3

SW World class update:

Added in Darth Vader, 5 storm troopers and Leia.

VaderPower class :: starwars actions

This class focuses on Vader's actions, using the act class to determine if Luke is trained by Ben and if so affects the chance where Luke will be turned to the Dark side. With this implementation, I chose to set Luke's health to 0 if Luke is turned to the Dark side as, it would also indicate that the game is over.

StormT class :: starwars.entities.actors

This class is used to create the 5 storm troopers that are initialised on the map. In it's constructor, it is equipped with a blaster. The storm troopers have 3 interactions when using the act() function. It would either attack (attacks can be missed), move around normally like any other actor, or spawn another Storm object in its location. Although the spawn interaction was not implemented.

Vader class :: starwars.entities.actors

Like other actor classes, has the default act function and in the constructor, equipped with a lightsabre. Vaderpower is also added to its affordance so Vader can interact and force Luke to the Dark side.

Player class :: starwars.entities.actors

Changes made to this class include setting validation statements so Luke can die, and therefore end the program.

Lightsaber class :: starwars.entities

Nerfed the damage on the lightsabre so it does not one hit kill Luke.

Changes to the Design

Before, the 'take' action can be used to pick up droids, such as R2D2, and if so will not be able to pick up a lightsabre. We have added classes takeDroid and takeHumanoid in order to protect the outcome's design. This will allow Luke to be able to take R2 and Leia to the Millennium Falcon all at once.

The map switch needs to be further refactored in the future as currently it is only sustainable in this brief design. If other maps were to be created, it is encouraged to allow different maps to be created in the SWWorld initially and then switch to different maps when an act allows it to.

Design and Future Implementation

When implementing ideas into the star wars code, when implementing characters that are able to be interacted/killed, they should be created in the 'actors package' as each turn they may be interacted by multiple actors. The act() function will allow the actor to be able to perform an action, such as train, move, take, dissemble, which are shown in the actions package. To allow the actors to perform the actions, it is required for the specific affordance to be added to the constructor of the actor class.

To initialise objects created in the entities or the actors package, the SWWorld class is where every object is initialised and placed onto the 10x10 map. Looking at the patterns, it is apparent that you create the object, set the symbol and set the location via a location finder which is already implemented in the grid. You can also control the movements of a certain actor by setting a patrol route in the SWWorld section, or setting the movements in its actor class, demonstrated in the Droids class.

Symbols are used mostly in most validation statements, and are primarily comparing itself and a target in the same location. By doing so, it allows Luke and R2D2 to be able to only repair droids or dissemble them into parts.

Furthermore, there is an implementation where you can validate an actor based on its' or the target's capability. Knowing whether an item is a weapon or fillable, allows actors to know what it can interact with, and not always relying on symbols.