Sandcrawlers rationale

Classes to add:

* Sandcrawler
* Door
* Exit

Sandcrawler:

* Will extend SWActor
* Internal locations will be implemented as a SWGrid
* One of the internal locations will have a Door entity
* Movement delay implemented via protected attributes in the SWActor class:
  + movementCooldown
  + idleCounter
* Following methods will be added to the SWActor
  + canMove()
  + setMovementCooldown()
  + resetMovementCounter()
  + patrol() – This protected method will execute the patrol behaviour for all SWActors, this will allow us to keep all our checking within the same class

Door:

* Will extend SWEntity
* Will have the Exit affordance

Exit:

* Will extend SWAffordance
* Override canDo() method to check that the SWActor is a force user in order to satisfy the design requirements. We will likely do this by calling the method SWActor.canUseForce()
* If canDo() returns true then the SWActor will be placed at the location of the Sandcrawler. We will do this by calling the SWActor.setLocation(Sandcrawler.getLocation())