**FIT 2099 Assignment 1,2,3 Game Engine design changes proposition**

**Game ending mechanics**

**Problem**

Currently, before processing each actors’ turn, the World class check if the game is still running by seeing if the player is still in the game.

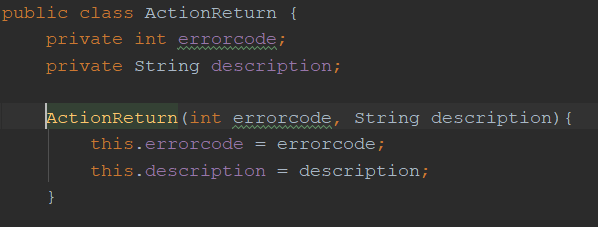
This is problematic for a few reasons:

* Ending the game requires the map to have a method to delete all players.
* Player can’t be removed from all maps for a turn which hinders future features possibility

**Solution**

Currently, when an Action is executed it returns a description string:

Instead, It would be better if Actions’ execute() method returned an Action Return class, which would contain the string description of the action, as well as an error code.

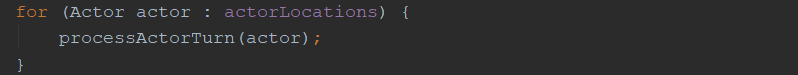
This could be 0 if game continues or 1 if game is over, the World class could then check the error code before executing the following actors turn.

This would also allow for better exception handling, as the World could handle multiple fringe cases using this method (e.g. error code 3 = player is stunned and can’t move)

**Processing Actor turn mechanics:**

**Problem**

Currently, if the game is still running, the World will process each actor’s turn regardless of what map they are on.

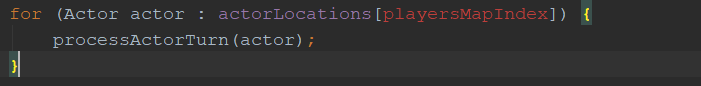


This causes a few problems:

* Firstly, the GUI is cluttered with descriptions of enemies’ actions from a different map which is both confusing and cumbersome. If the game had many more maps it would be downright unplayable
* Secondly, this leads to overcomplication in the actor classes to ensure no bugs are caused from an enemy’s turn being processed when the Player is not on their map (checking range for example)

**Solution**

The world should store an actorLocations object for each map and a pointer to the player’s current map so that it only processes the actors on the player’s map:



**Item getAllowableActions mechanics**

**Problem**

Currently, an item’s getAllowableActions methods do not take an actor or map as arguments. This means that the items use can’t be different from actor to actor or dependant on which map it is on as it is with Ground and Actor objects. This means for example, that an item can’t be prevented from being picked up only by players.



**Solution**

Item’s getAllowableActions method should accept map and other actor parameters, like the Ground or Actor classes.



**Skills should have allowable actions**

**Problem**

Currently, when processing an Actor’s turn, the World find the Actor’s options from the following sources:

* Items in Actor’s inventory allowable action
* Adjacent Actors’ allowable actions
* Adjacent ground’s allowable actions
* Items on current ground’s allowable actions
* Skip turn action

However this does not allow for skills to give a player certain move options.

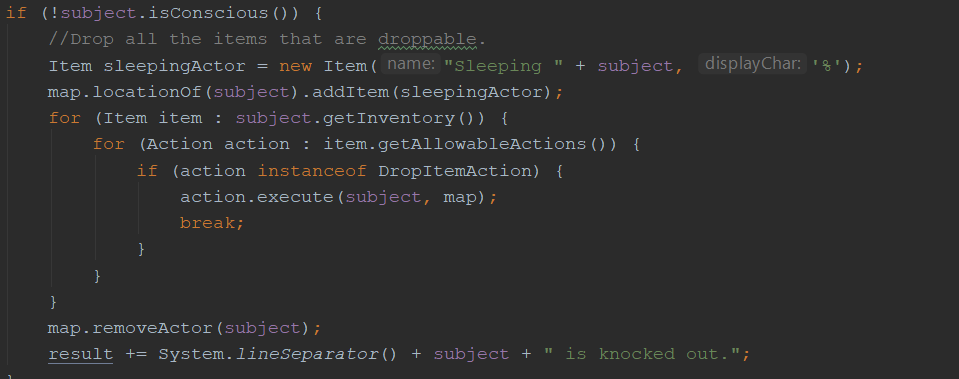
**Solution**

Have the world also check all the Actors’ skills and make skills a class instead of enum with a getAllowableActions() method. This would allow for numerous future implementations and would make detecting certain conditions far cleaner and easier (e.g. if player needs to have done certain moves, then it gets a skill that allows it to perform a special action, and world only needs to check for 1 skill instead of numerous dummy items to represent actions taken which is less clean and makes less semantic sense.)

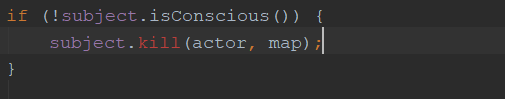
**Player death mechanic**

**Problem**

Currently, the attack action is responsible for detecting if the attacked Actor is unconscious and performs the steps after their death. This prohibits having different things occur when different Actors are killed.



**Solution**

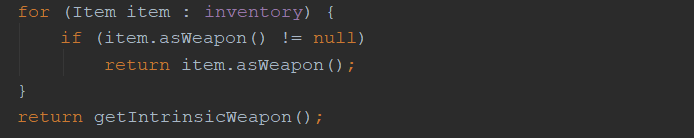
Actor deaths should be controlled from the Actor class. There should be a killActor() method that is called when the attack action detects that the attacked Actor is unconscious. This method would receive the attacking actor and game map as arguments.

If the engine worked this way, processing Yugo Maxx’s death would be a lot easier, and it could allow for numerous future expansions such as having an enemy cover the player is harmful goo upon death.

**Choosing Actor weapons**

**Problem**

Currently, the world does not give the Actors a choice about which weapons can be used to attack. The Actor’s getWeapon method simply selects the first weapon it finds in the inventory or returns the intrinsic weapon.



This prevents an actor from choosing weapons and having weapons have advantages and downsides which would make the game mechanics more complex and interesting. (e.g choose to attack Goon with sword for max damage but ninja with nun chucks as there is less chance of missing).

**Solution**

The getWeapon method should return a collection of weapons, and an Actors getAllowableActions method should get this collection from the actor in question and return an Actions class with numerous attackActions. This would also require attackActions to be constructed with a dedicated Weapon in the constructor.