**Draft/ Ideas on how to implement REQs 3-4**

package entities/enemies?

GENERAL FOR ENEMIES

After u attack or they attack, follows player. Dead removed from map, cannot go in Floor

- update floor with canActorEnter() logic to not allow enemies

- possibly enemy status?

- check status for enemy then player cannot enter\*

- new class koopa

- 100 hp

- not removed, goes dormant

- wrench to destroy shell

- destroyed shell drops mushroom

To do :

- extends Actor

- 15% spawn chance per tree per turn, connect with REQ1

- new behaviour class for dormant

- change koopa display character to D

- doesn't let koopa move

- new action class, destroy shell either that or new code in AttackAction class

- allows player to destroy shell if player has wrench in inventory

- destroying shell will drop SuperMushroom instance

- cannot attack D, only with wrench

- D, cannot follow, attack or wander

- goomba class

- 10% suicide chance

- 10dmg, 50% chance

- 20 hp

To do :

- extends Actor

- intrinsic weapon kick, 10 dmg 50% chance

- new suicide action class

- or goomba class has new method toSuicide()?

- called on every goomba at end of every turn

- returns true or false based of random int generator between 1 and 100, <= 10 suicide by removing actor from map, > 10 not

- some goombas spawn from sprout 10%, works with Zubin's Sprout Implementations

- cannot have entire map as goomba, maybe have an int attribute representing goomba count to stop spawns

- could also have arraylist attribute containing goomba objects, size() cannot pass a limit (similar as above)

for magical items, either in-built method() for consuming item or consumingItem() action

isConsumed() method, check whether to remove from inventory. Remove from inventory at end of consume code

- new class superMushroom

- extends items

- use in-built player method to change max hp, increaseMaxHp()

- in-built player method to change displayChar, setDisplayChar()

- attribute to flag if consumed??

- consume() method to make changes to maxhealth and displayChar when called, removes itself from player inventory after by removing from player arrayList or dropping item? dropped items could also have isConsumed boolean for a cleaner to check all items on floor to destroy/remove

- set player to have SUPERMUSHROOM status or reuse TALL for implementation or player.addcapability

- while this status is active, jumpaction() has 100% success rate (jump action should have if statement to check if TALL/SUPERMUSHROOM)

- attackAction() to have new if statement to check whether target actor.hascapability(Status.TALL/SUPERMUSHROOM)

- changes display character is true, continues with action if false

- new jumpAction class

- checks superMushroom status, for jumping, normal jump with random number generator if no status

- new class powerStar

- extends items

- uses in-built methods from items, tick() and some form of counter to check if up to 10 turns have passed

- one tick() method for when star is on floor

- one tick() method for when star in inventory

- another new tick() method to start count once star consumed

- needs to remove Status.STARBUFF once consumed, either via tick or another method to keep track of the PowerStar

- code a new display() or description method to display how many ticks passed or how many turns left until star buff runs out, can have another one for Mario is INVINCIBLE or use same method to do two things

- attackAction also might need new if statement to see if STARBUFF status is present in target Actor.

- no dmg dealt if so

- possibly new action class instaKill() that only called when status STARBUFF is present

- another possible implementation is to add another if statement for player actor to see if STARBUFF status active

- new action class destroyGround(), possible to reuse old code canActorEnter, to see if false, then call destroyGround() action

- can also refactor canActorEnter to see if player has STARBUFF status, lets destroyGround() action occur after

- use on Trees and Walls classes

- can also use ActionList with Status and more types of Actions like NoEffectAttackAction when Status.STARBUFF

- add to actions list

- stored in player Inventory (under actor class)

NOTES AND THOUGHTS

Give specific capabilities of each Item based on Status buff from Enum.

canPlayerDestroy(), return player.hasCapability( enum.STATUS )

Add new actions

Add new behaviours?

How are actions added? How are they read?