

Design Rationale

Newly created/modified classes with new relationships

ConsumableItems(new)

An abstract class that extends Item class. Used to represent items that can be consumed.

SuperMushroom

A class that extends the ConsumableItems class. Responsible for representing a SuperMushroom object

PowerStar

A class that extends the ConsumableItems class. Responsible for representing a PowerStar object

ConsumeAction(new)

A class that extends Action class. Responsible for handling consumable item's effects when consumed. This class was created to avoid breaking the **Don't Repeat Yourself** principle, so there's no need for separate action for each consumable item.

HighGround

This is changed because of some changes that need to be done to its allowableActions method. And, it can be destroyed now after having the effects of consuming a power star.

Effects of consuming a super mushroom

"Increase max HP by 50"

We can utilise actor's increaseMaxHp method to heal players by 50 hitpoints in SuperMushroom.consume()

"The display character evolves to the uppercase letter"

This is already done for us in original source code for Player class, in getDisplayChar method

"It can jump freely with a 100% success rate and no fall damage"

Players will be given the status of TALL. Jumping freely with a 100% success rate can be done in HighGround's allowableActions method, where we can implement a check to see if an actor doing the jump has TALL status or not, if it is then JumpAction can be executed without even checking for percentages.

After all this is done, the SuperMushroom that was consumed will be removed from the player's inventory, in SuperMushroom's consume method using removeItemFromInventory method.

"The effect will last until it receives any damage"

We can override the hurt method in Player class, where if a player does have TALL status, then remove it.

Properties of Power Star

"It will fade away and be removed from the game within 10 turns (regardless it is on the ground or in the actor's inventory)"

We can override both tick methods from Item class, the first tick method is the one with only location as parameter. This means that it is on the ground, so in every tick the ticker will decrease by 1. If the counter reaches 0, we can utilise location's removeItem method to

clear the PowerStar from the map. The second tick method is the one with location and actor as parameters, this means that it is in the player's inventory. Ticker will be decreased by 1. If ticker reaches 0, we check if actor has capability of invincible, if it does, then remove it and remove the item from actor's inventory. If not, just remove the item from the actor's inventory.

Effects of consuming a Power Star

"Anyone that consumes a Power Star will be healed by 200 hit points"

We can utilise actor's heal method to heal player by 200 hit points

" Will become invincible"

We can use addCapabilities to give the player Status.INVINCIBLE

"The actor does not need to jump to higher level ground (can walk normally)"

We can override the canActorEnter method in HighGround, checking that if an actor has the capability of invincible, it will just return true, while normally it returns false.

"If the actor steps on high ground, it will automatically destroy (convert) ground to Dirt. For every destroyed ground, it drops a Coin (\$5)."

In HighGround's tick method, we will first check if the location contains an actor using Location's containsAnActor method. If so, we will get the actor in that current location, using the getActor method from Location, and check whether the actor has invincible capability or not. Therefore there's a dependency between HighGround and Actor. If so, we can utilise Location's setGround method to set current location to dirt and add a coin of value \$5 using addItem method.

"All enemy attacks become useless (0 damage)."

Override player's hurt method, if player has invincibility status, hitPoints -= 0

"When active, a successful attack will instantly kill enemies."

Referenced in REQ3's AttackAction sequence diagram, we can check if actor has capability of invincibility. If so, reset the target's max hp to 0 using resetMaxHp method.