**Class Responsibilities**

**RocketPad: (Inherits Ground)**

Class Responsibility: (Updated)

This class will simply create a Ground called RocketPad which has the ability to build a new Rocket Instance. It does this by returning a new action “BuildingRocketAction”

This class has two parameters:

moonBase: This is the location where the player gets transported to.

Enemy: This is the enemy we give the skill to walk on LunarSoil only once player enters moon.

**BuildingRocketAction: (Inherits Action)**

Class Responsibility: (Updated)

This class holds the ability to build a rocket on the current location of the RocketPad. Several parameters are passed into this class in order to effectively build a rocket(which has a parameter to know where to go to, i.e the MoonBase, and the enemy, who will be given the skill to move on the Moon only once the player gets there)

Steps:

We first go through the actors inventory and check if he has the RocketEngine and the RocketBody, and for each part found we increment the count of items and remove it from the inventory(As there can only be one instance of RocketEngine and RocketBody)

Finally, if the conditions are met, we will add a new Rocket instance.

The class will then return the appropriate message

**Rocket: (Inherits Ground)**

Class Responsibility:

This class will simply create a Ground called Rocket which has the ability to transport the player to the moon base certain requirements are met(mentioned below). It does so by returning a new action “FlyingRocketAction” as allowable actions to an actor.

This class has two parameters:

finalDestination: This is the location where the player gets transported to

Enemy: This is the enemy we give the skill to walk on lunarSoil only once player enters moon.

**FlyingRocketAction: (Inherits Action)**

Class Responsibility:

This class holds the ability to move an actor from earth to the moon base location(Which is held within the parameter passed into this class named “otherLocation”). Before executing, the action, this class needs to check if the player has at least one oxygen tank and a spacesuit. Once the player used the rocket, we add a skill called “OXYGEN\_SKILL” to the player when he is in on LunarSoil and we remove the skill if the player is on Floor(Earth Soil). Finally we also give the enemy(namely YugoMaxx) the ability to start moving on the move by giving him the skill “CYBERNECTIC\_IMPLANTS”, this way he will be 10 squares away from the rocket when the player is on the moon.

The class will then return the appropriate messages.

**SafetyTransportationAction: (Inherits Action)**

Class Responsibility:

This class holds the ability to move an actor to Rocket Pad on Earth when he runs out of oxygen, this action simply removes the “OXYGEN\_SKILL” from the actor and moves the actor to the passed in location.

The class will then return the appropriate messages.

**OxygenDispenser: (Inherits Ground)**

Class Responsibility:

This class will create a ground type that allows a player to dispense oxygen by returning an allowable action called “DispenseOxygenAction” for the player to execute the aforementioned action. However this action will only be allowed if the player has a “SpaceSuit” item in his inventory(to prevent other actors from dispensing the oxygen), otherwise this class will simply return the default list of actions.

This Ground will only allow the actor to enter if he has a spacesuit.

**DispenseOxygenAction: (Inherits Action)**

Class Responsibility:

This class holds the ability to dispense an instance of “OxygenTank” onto the location that is passed into the class(Usually on the location of the OxygenDispenser).

The class will then return the appropriate messages.

**OxygenTank: (Inherits Item)**

Class Responsibility:

This class is used to create an instance of OxygenTank item. Each instance of OxygenTank has its own count of the Oxygen level, this count can be incremented and the value of count can also be return to check if the tank is empty or has remaining moves\* for the player.

The function “empty()” is used to return boolean to indicate if the this instance of the tank is empty or not.

\*This is the balance number of moves that the player has.

**OxygenSkill: (Enum Type)**

Class Responsibility:

This class simply holds the skill that will be given to the player when the player flies to Moon. This skill is used so that on every turn of the player the count of the oxygen level is incremented only if the player has OXYGEN\_SKILL. Since the Oxygen level should only increment only on Moon, the skill is only added when player flies to Moon and removed when the player reaches Earth.

**WaterPistol: (Inherits WeaponItem)**

Class Responsibility:

This class is used to create an instance of WaterPistol. This is a weapon item that is used by the player to remove the exoskeleton from YugoMaxx. The class has a private boolean attribute “isFull” to allow it to be refilled. If the Pistol is refilled, it will be able to shoot only once, and when it is fired, we will change the boolean back to false in order to be refilled again.

**Water: (Inherits Ground)**

Class Responsibility:

This class will create a ground type that allows a player to refill his WaterPistol WeaponItem by returning an allowable action called “FillWaterPistolAction” for the player to execute the aforementioned action. However this action will only be allowed if the player has an instance of the WaterPistol WeaponItem in his inventory and will only fill that particular instance of WaterPistol, otherwise it will simply return the default Actions.

This Ground will not allow any actor to enter.

**FillWaterPistolAction: (Inherits Action)**

Class Responsibility:

This class holds the ability to fill a particular instance(which is passed into the class) of WaterPistol WeaponItem. It will simply set the WaterPistol to full (using setFull), and thus reloading the WaterPistol.

The class will then return the appropriate messages.

**ShootAction: (Inherits Action)**

Class Responsibility:

This class holds the ability to fire the pistol at another player only depending on ShootChance(a private attribute that holds a float that randomly generates a number from 0-1)

if is less than 0.7 (as there is a 70% chance of destroying YugoMaxx’s exoskeleton), we set the instance of YugoMaxx’s exoskeleton to false.

If the shootChance is greater than 0.7, we set the instance of the player’s WaterPistol’s capacity(isFull) to false, and this forces the player to reload the waterPistol back on Earth.

**LunarSoil: (Inherits Ground)**

Class Responsibility:

The class is used to create a ground which is used in the moon. It checks if the Actor has either the skill CYBERNETIC\_IMPLANTS or SPACE\_SKILL to allow them to pass through.

**newPlayer: (Inherits Player)**

Class Responsibility:

This class is for the player. Since we cannot change engine code we decided to make a new class for player which inherits from the player class in the engine. We have an boolean attribute (isStun) to check if the player is stunned by Ninja and and we have a boolean attribute (remainingOxygen) for remaining oxygen and a Location attribute (RocketPadLocation) to store the location of where the player should be transported to if the oxygen is depleted.

On every playturn of the player we check if player has OXYGEN\_SKILL . If true, it means player is on the Moon and we increment the count for the Oxygen Tank. If the count is 10 we remove the tank. We check if there are remaining oxygen tanks and set the boolean remainingOxygen. If there is no remaining oxygen a SafetyTransportAction is returned. If stun isStun true a SkipTurnAction is returned. Else it returns default actions.

**NewWorld: (Inherits World)**

Class Responsibility:

This class is used the same way as the World class. We override the run() method so that in every round of the game we check if the player is on earth and if the “Sleeping YugoMaxx” is present in the player’s inventory.If True, we display a message saying player won and quit the game. We override the processActorTurn method to add a QuitGameAction to the players list of Actions. We also check if the player selected the QuitGameAction and if true we quit the game.

**QuitGameAction: (Inherits Action)**

Class Responsibility:

This class inherits from Action class.The hotkey() method of this class is overridden to return “q” as the hotkey to quit the game.

**SpaceSkill: (Enum Type)**

Class Responsibility:

This class holds two Skills, CYBERNETIC\_IMPLANTS and SPACE\_SKILL. The SPACE\_SKILL is given as a Skill to the SpaceSuit so when player obtains SpaceSuit he will have the SPACE\_SKILL which will allow him walk on the Moon. The CYBERNETIC\_IMPLANTS are given so enemies can walk on moon.

**SpaceSuit: (Inherits Item)**

Class Responsibility:

This class is user to create an instance of a SpaceSuit with SPACE\_SKILL given to the instance by default. SpaceSuit is needed to dispense oxygen and for the player to walk on the Moon.