

Class Descriptions for Requirement 1: Let It Grow!

Class: <<abstract>> Tree

Description: The original tree class is modified to be abstract, and it will contain all of the common attributes and methods that all sub-classes of tree will need.

Class: <<interface>> GrowCapable

Description: This interface will contain the method(s) that enable certain trees (sprouts and saplings to grow into larger trees).

Class: Sprout

Description: The smallest variant of tree. 10% chance to spawn a Goomba each turn, unless an actor stands on it.

Takes 10 turns to grow into a sapling tree.

Class: Sapling

Description: The medium tree variant. 10% chance to drop a coin each turn. Takes 10 turns to grow into

a Mature tree.

Class: Mature

Description: The largest tree variant. 15% chance to spawn a Koopa each turn, unless an actor stands on it.

20% chance to wither and die, being turned into dirt each turn. Can grow a new sprout nearby every 5 turns.

Class Descriptions for Requirement 2:

Class: LowGround (abstract)

This class is responsible for ensuring Floor and Dirt have similar characteristics that define them as 'low ground' for the game. This could include that they can be moved down to from higher grounds.

Class: HighGround (abstract)

This class is responsible for ensuring Wall and Tree have similar characteristics that define them as 'high ground' for the game. This may mean that they can be jumped to and sit higher than the lower grounds.

Class: JumpActorAction (abstract)

This class is responsible for the 'jump' movement that the player can do to move from low ground to high ground. it changes the location of the player and is a unique action to the player only. Enemies cannot access this action.

Class description for Requirement 3: Enemies

1. <<abstract>> Enemy:

Description: The abstract enemy class has all the attributes and methods that are common for all types of enemies (i.e Goomba and Koopa)

2. Koopa:

Description: The Class Koopa represents the enemy of type Koopa. It contains attributes/methods that are inherited from the class Enemy as well as methods/attributes specific to its class

Class description for Requirement 4:

Class: MagicalItem (abstract)

This is the parent class for all magical items. It is inherited from Item and is responsible for defining methods and attributes that are shared by both magical items SuperMushroom and PowerStar. It also has a capability set.

Class: SuperMushroom

This class is responsible for the item SuperMushroom that the actor can pick up and consume. This defines unique characteristics and dependencies with other classes that are affected by the super mushroom.

Class: PowerStar

This class is responsible for the item PowerStar that the actor can pick up and consume. This defines unique characteristics and dependencies with other classes that are affected by the power star.

Class Descriptions for Requirement 5: Trading

Class: Toad

Description: Toad is a friendly NPC actor that mario can interact with. This class will contain Toad's core information and functionality.

Class: <<abstract>> ToadAction

Description: The ToadAction class will contain the various interactions Toad can have with Mario, namely

speaking with him, or trading with him.

Class: Trade

Description: The trade class will cause a text-based interaction with Mario, upon which mario can purchase items from Toad.

Class: Coin

Description: To purchase items from Toad, Mario needs to collect and trade coins for the items. These coins

are objects that can be found while playing the game, and when picked up by Mario the object will be removed,

and the value of the coin will be added to Mario's wallet.

Class Description for Requirement 6: Monologue

1. Class Toad:

Description: The class Toad represents an actor of type Toad. The class Toad contains all the attributes/methods specific to the class Toad and can inherit methods/attributes from the abstract class Actor to ensure all requirements of Toad Speaking is fulfilled.

2. Class Speak:

Description: The class Speak deals with all the dialogues that the instance of class Toad needs to say depending on specific conditions.

3. Trade:

Description: The class Trade deals with the trading that the user (Mario) does.

4. <<abstract>> ToadAction

Description: The class contains all the attributes/methods that all instances of type Toad should have. The class ToadAction inherits from the abstract class Action as there are methods that every actors can do.

Class Descriptions for Requirement 7: Reset Game

Class: ResetManager

Description: This class handles the resetting of the game, which the player can do once per game. It will

reset enemies, coins, player health and status, and 50% of trees.

Class: <<interface>> Resettable

Description: This class will check whether or not the game can be reset, which is determined by whether the

game has already been reset once in this playthrough.