Lab11 Team3
Group: Seow Zheng Hao, Muhammad Abdullah Akif, Danesh Mariapan

Design Rationale

- REQ 1
  The 3 Sprout, Sapling, Mature classes will extend the tree class to get similar base attributes, since the 3 types of trees have different rates of growth and different functions, extending them to a base class will reduce redundancy.
  Sprout class: a small tree with different growth rate extending the tree class
  Sapling class: a medium tree with different growth rate extending the tree class
  Mature class: a big tree with different growth rate extending the tree class

- REQ 2
  Jump action is only going to be called by mario as enemies cannot jump to higher ground. Walking off from high ground to low ground is guaranteed and can be done by mario and other enemies. Needs to perform a check for player status for super mushroom before calculating percentage. then check what object Mario is trying to jump to, then call the item class and prompt them for a jump. The jump function in the tree classes will save the chances and output a boolean for success or failure.
  JumpAction class: will depend on the ground class to check for jumps, only player can jump, enemies cannot
  Walk down class: all actors can walk down from high ground

- REQ 3:
  Goomba and Koopa will be created on the map in the Application class, at the start of the game. Both Goomba and Koopa will extend the abstract class, Actor, in the engine because both these will be using the methods present in the Actor abstract class. This will reduce redundancy. As the ground changes the state of the floor, it will be used to remove Goomba from the map and also to change Koopa to a dormant state (D), when it is defeated. When a Koopa's shell is destroyed it will drop a Super Mushroom, hence it would have a dependency with a new class called SuperMushroom. (We will be talking about the SuperMushroom class in REQ4)
  Koopa class: an enemy class that extends the Actor class.

- REQ 4:
  The two classes, SuperMushroom and PowerStar will have an association with the Player class as an object of each of the above two classes will be made when the Player has them in his/her inventory or when the Player is using it. Player class will store each Magical item in its inventory using the Item class already present in the engine.
  SuperMushroom class: a class that facilitates SuperMushroom, it is called by the Player class.
  PowerStar class: a class that facilitates PowerStar, it is called by the Player class.

Lab11 Team3
Group: Seow Zheng Hao, Muhammad Abdullah Akif, Danesh Mariapan

- REQ 5:
  Toad Class: Will have a Dependency linking all the other three usable Item Classes (Wrench, SuperMushroom & PowerStar), since the Toad is supplying the items to the Player, and they can only be obtained through the Toad.
  Coin Class: Has an Association from the Sapling Class to itself as the Saplings produce/spawn the Coins on the game map. Each coin object will have an integer / currency value which adds to the Player’s wealth when picked up
  CoinWallet Class: is added as a wallet system to the Player, and is an Aggregation relationship to the Coin Class. This is where the currency value of the Player’s total wealth will be stored.
  CollectCoinAction: This class is an added action in which the player uses when he/she is next to a Coin on the game map, the Coin is then removed from the map and the integer value (currency value) of the Coin is added to the player’s CoinWallet (which stores the integer value of the Player’s total wealth)
  BuyItemAction: This class is an added action used when the Player is trading with the Toad. When used, the Player “buys” an item from the Toad a currency transaction is made - the cost of the item is deducted from the Player’s CoinWallet’s total wealth, and the item is added to the Player’s Inventory.

- REQ 6:
  SpeakAction: Is an added Action that has an Association from the Player Class and extends the Abstract Action Class, as it is a type of Action in which the player can use when he/she is near the Toad and wants to interact with it. Only after this Action is used will the Player see the Toad’s monologue and shop items to buy.
  ToadSpeakBehaviour: Is an added Behaviour that has an Association from the Toad Class and extends the Interface Behaviour, as it is an NPC behaviour in which the Toad has to follow when being interacted with by the Player. The behaviours then target the Player using the FollowBehaviour Class.

- REQ 7:
  A new class reset game is created and calls the different classes that need to be removed once the game is reset.
  Rest Game class: calls the classes that will be affected by resetting the game