Comments/Documentation
Group 1: Ted Weber, Michael Zazula, and Zellie Macabata

Actor.java: The Actor class is the class that creates instances of the actors i.e. how many players are playing the game. It has all the attributes an Actor has, and stores the values when manipulated. There are only getters and setter methods in the Actor class.

BoardController.java: The BoardController initializes the number of days (which is dependent on the number of players), sets up the game-board layout by instantiating SceneCards in each SceneRoom, and then starts the game.
playerTurn(): Checks the current player's position. There are three states that a player can be in…

1. Not in a role or not in the CastingOffice -- If the player is in a SceneRoom and if there are movies that "are active/in production", BoardController gets all the active roles based off of the current player's rank.
2. The current player has a role -- If the player is in a role, the player can check their status and the movie status however many times they want until they choose to act on their role or rehearse. If they act or rehearse, they can only check statuses up to that point until they end their turn/
3. In the CastingOffice -- The UI will pass in an input that deals with the desired rank that the current player wants and how they want to pay to upgrade their rank (dollars or credits). If the player decides not to upgrade, it will ask the player if they want to move or check their statuses.

closeRoom(): The room is a "Dead room" if all the shotCounters are gone, but the players can still move through the room. It resets the current player's role statuses to be false.
starBonus(): This function is the payout of when a scene closes. It sorts the StarringRoles ArrayList and then casts it to a Queue because we want the functionality of a Queue when the diceValues is greater than StarringRoles.
resetRoles(): This function takes in the current room when all the shotCounters are gone and re-instantiates a new SceneCard for the room. It will reset the ExtraRoles to be unoccupied by all the players.
validBonus(): This function checks if the given SceneCard is occupied by a player, because if it is, they will get the correct payout.
getCurrRoom(): The function will go through the BoardSections and the Rooms to get the title of the current board position of the player.
getAdjRooms(): This function goes through the BoardSections and the Rooms and checks to see which rooms are next to each other. This is from a text file from an already set up Board.
endPlayerTurn(): This function ends the current players turn, and adds the current player back to the Queue.

setDays(): This function sets the days based off of how many players are inputted by the user.
askRole(): Returns the current player's position and if they are in a role or not.
parseSceneCards(): All of our SceneCards are formatted in a text file and it grabs the budget of the scene, the roles and their ranks, and parses the catch-phrase associated with them.
parseRooms(): The function parses our Rooms.txt file and gets the ExtraRoles, gets the adjacentRooms.
refreshBoard(): Will go through each BoardSection and room and reset the original shotCounters (from the very beginning of the game), puts a new SceneCard in Room, and sets the amount of movies left to play. Then it resets the position of each player to go back to the Trailers and resets their attributes.
endGame(): Once all the days are over, this function calculates who has the most money, and whoever does, is the winner of the game.
pullCard(): This function randomly pulls any of the 40 SceneCards and decrements the size.

BoardSection.java: Creates the BoardSection class. There are 4 BoardSections in total with all the appropriate Rooms to each BoardSection.

CastingOffice.java: Extends the Room class. It has a HashMap - the key is the desired Rank that someone can upgrade to, and an Integer Array of the values of the dollars or credits the player needs to pay to upgrade.

Dice.java: Every time we instantiate a new Dice object, it will give a random value between (0-6) + 1, so it doesn't get that 0 value. And we call this for payout of a movie or when a player wants to act.

ExtraRole.java: Extends Roles. Has it's own implementation of payout and it's own implementation of when a player receives a bonus.

Role.java: Stores information on if a role is occupied by the current player, the role's rank, and can reset the booleans of the occupant and the role itself.

Room.java: Places all the adjacent rooms accordingly, and has the setters and getters of the current room.

SceneCard.java: All of the SceneCards have the StarringRoles of the game so there are getters and setters for all of those roles and also the budget of their movie, too.

SceneRoom.java: Extends Room class. A SceneRoom is not considered Trailers or the CastingOffice. It holds all the shotCounters for each room, the SceneCard with all the StarringRoles, the ExtraRoles from each room, and the overall movieStatus.
SctarringRole.java: Extends Role. Has it's own implementation of payout and it's own implementation of when a player receives a bonus.

Trailer.java: Checks what is adjacent to the Trailers room.

UI.java: Talks to the user and grabs input for when the game starts, what actions that actors want to do, or if they want to check the status of themselves.
moveCheck(): If you are in Trailers (start of the game) or if you are still eligible to move.
roleCheck(): Prompt if the player wants to take a role or not after moving. A player can take a role after moving but then their turn ends.
roleChoose(): The UI will only tell the user what roles are available based off of their rank.
moveTo(): The UI shows the adjacent rooms that a player can move to based off of their current position.
workCheck(): If a player is in a role, the UI will ask what they want to do on their turn. If the user keeps checking their statuses, their turn won't end unless they act or rehearse. The user should be able to check their statuses however many times they want.
finalCheck(): If the user chooses to act or rehearse, a menu will show up to see if they want to see their statuses and it will keep prompting this menu unless they choose to end their turn.
upgradeCheck(): When the current player is in the CastingOffice they can move or upgrade or check their status.
upgradeWithDollarsOrCredits(): Follow up with upgradeCheck(), if the user chooses to upgrade their rank, it will show the ranks they can upgrade to and the dollar/credit amount to pay. The UI returns an entire line of input to the BoardController.
printPlayerStatus(): This is an option to see all the current player's attributes.
printMovieStatus(): This is an option to see the status of the current room.
printDiceRoll(): This shows the dice value when you act.

Wallet.java: Stores all of the actor's earnings with getters and setters. Also has decrement methods for when an actor wants to upgrade.