Test cases for My Empire Project

Methods are tested using the GameTest class. The sample input column below will have references to a (group of) line(s) of code in the GameTest.java file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | | | | |
| **Method** | **Test Case #** | **Test Description** | **Expected Output**  **or**  **Expected Return Value** | **Expectations**  **met?** |
| Bank | | | | |
| addOrDeductBank  (initial value = 2500) | 1 | Changes bank by a positive whole number | 10000.0 + 100 = 10100 | ✔ |
| 2 | Changes bank by a positive floating point number | 10100 + 2.2 = 10102.2 | ✔ |
| 3 | Changes bank by a negative whole number | 10102.2 - 50 = 10052.2 | ✔ |
| 4 | Changes bank by a negative floating point number | 10052.2 - 5.5 = 10046.7 | ✔ |
| CardGroup1 | | | | |
| applyCardToPlayer | 1 | Currently 0 cards | jailChance cards = 1 | ✔ |
| 2 | Currently 1 card | jailChance cards = 2 | ✔ |
| 3 | Currently 2 cards | jailChance cards = 3 | ✔ |
| CardGroup2 | | | | |
| findRandomProperty | 1 | Random # 1 | Property at index 1 | ✔ |
| 2 | Random # 2 | Property at index 11 | ✔ |
| 3 | Random # 3 | Property at index 15 | ✔ |
| findNearestRail | 1 | Nearest # 1 | Railroad at index 13 | ✔ |
| 2 | Nearest # 2 | Railroad at index 26 | ✔ |
| 3 | Nearest # 3 | Railroad at index 14 | ✔ |
| findNearestUtil | 1 | Nearest # 1 | Utility at index 17 | ✔ |
| 2 | Nearest # 2 | Utility at index 9 | ✔ |
| 3 | Nearest # 3 | Utility at index 9 | ✔ |
| CardGroup5 | | | | |
| isCardApplicable | 1 | # of Properties = 3  SPEC\_TYPE = 2 | true | ✔ |
| 2 | # of Properties = 0  SPEC\_TYPE = 3 | false | ✔ |
| 3 | # of Railroads = 0  SPEC\_TYPE = 4 | false | ✔ |
| 4 | # of Utilities = 1  SPEC\_TYPE = 4 | true | ✔ |
| applyCardToSpace | 1 | SPEC\_TYPE = 1 | rent = rent \* 2  bDoubleRent = true  doubleRentHolder = this card | ✔ |
| 2 | SPEC\_TYPE = 2  Property has hotel | toPay = 50  rent = rent \* 1.5 | ✔ |
| 3 | SPEC\_TYPE = 2  Property has 3 houses | toPay = 25 \* 3 = 75  rent = rent \* 1.5 | ✔ |
| 4 | SPEC\_TYPE = 3  rent = 16  Dilapidated houses | rent = 14.4 | ✔ |
| 5 | SPEC\_TYPE = 4  rent =100  Increase 10% | rent = 110 | ✔ |
| 6 | SPEC\_TYPE = 4  rent = 50  Decrease 10% | rent = 45 | ✔ |
| Deck | | | | |
| drawCard | 1 | Draw #1  activeDeck size = 28 | drawn = remove from activeDeck  activeDeck size = 27 | ✔ |
| 2 | Draw #2  activeDeck = 1 card left  Double rent card and get out of jail card taken | drawn = remove from activeDeck  activeDeck size = 0  Shuffle cards  activeDeck new size = 26 | ✔ |
| 3 | Draw #3  activeDeck size = 26 | drawn = remove from activeDeck  activeDeck size = 25 | ✔ |
| discardCard | 1 | Discard #1 | Discard size = 1 | ✔ |
| 2 | Discard #2 | Discard size = 2 | ✔ |
| 3 | Discard #3 | Discard size = 3 | ✔ |
| replaceDeckAndShuffle | 1 | Shuffle #1 | discardDeck will become activeDeck  discardDeck cleared | ✔ |
| 2 | Shuffle #2  activeDeck is reduced | discardDeck will become activeDeck  discardDeck cleared | ✔ |
| 3 | Shuffle #3 | discardDeck will become activeDeck  discardDeck cleared | ✔ |
| TaxSpace | | | | |
| payTax | 1 | Income Tax  Player’s cash = 1500  Player’s cash \* 0.1 < 200 | dToPay = 200 | ✔ |
| 2 | Income Tax  Player’s cash = 2260  Player’s cash \* 0.1 > 200 | dToPay = 200 | ✔ |
| 3 | Luxury Tax | dToPay = 75 | ✔ |
| 4 | Player does not have enough cash (Luxury) | dToPay = -75 | ✔ |
| ***OwnableSpace*** | | | | |
| canBeBought | 1 | price > player’s cash | false | ✔ |
| 2 | price < player’s cash | true | ✔ |
| 3 | price = player’s cash | false | ✔ |
| Property | | | | |
| canBeDeveloped | 1 | Developing fully developed property | false | ✔ |
| 2 | Developing a hotel to a property with foot traffic greater than the required number | true | ✔ |
| 3 | Developing a house to a property with total collected equal to the price of house development | true | ✔ |
| 4 | Developing a house to a property without having sufficient funding | true  user will be forced to end the game | ✔ |
| getPropertyRent | 1 | Property is only property owned by the player  Base rent = $10 | Returns 10 | ✔ |
| 2 | Player owns the property plus one other property of the same color  Base rent = $20 | Returns 30 | ✔ |
| 3 | Player owns the property plus one other property of the same color, current property has double rent status  Base rent = $20 | Returns 50  Double rent status will now be false | ✔ |
| 4 | Player owns the property plus two other properties of the same color, current property has double rent status  Base rent = $100 | Return 220  Double rent status will now be false | ✔ |
| Player | | | | |
| movePlayer | 1 | Index = 31 (before START)  Move 6 steps (Income Tax) | New index = 5  Player’s cash + 200  Bank’s cash - 200 | ✔ |
| 2 | Index = 4  Move 4 steps (Free Parking) | New index = 8 | ✔ |
| 3 | Index = 0 (START)  Move 7 steps (Property Space) | New index = 7  Property foot traffic + 1 | ✔ |
| payPlayer | 1 | Player 1 pays Player 2 $26  Player 1 cash = 1200  Player 2 cash = 1390 | Player 1 cash = 1174  Player 2 cash = 1416 | ✔ |
| 2 | Player 4 pays Player 1 $100  Player 4 cash = 1250  Player 1 cash = 1174 | Player 4 cash = 1150  Player 1 cash = 1274 | ✔ |
| 3 | Player 3 pays Player 2 $49.5  Player 3 cash = 1460  Player 2 cash = 1416 | Player 3 cash = 1410.5  Player 2 cash = 1465.5 | ✔ |
| payBank | 1 | Player 1 pays Bank $75  Player 1 cash = 1500  Bank = 10000 | Player 1 cash = 1425  Bank = 10075 | ✔ |
| 2 | Player 2 pays Bank = $300  Player 2 cash = 1500  Bank = 10075 | Player 1 cash = 1200  Bank = 10375 | ✔ |
| 3 | Player 1 pays Bank $202.7  Player 1 cash = 1500  Bank = 9980.8 | Player 1 cash = 1297.3  Bank = 10183.5 | ✔ |
| adjustPropertyRents | 1 | nType = 5 | Nothing happens | ✔ |
| 2 | nType = 7 (Railroad)  Railroads owned = 1 | Rent = 25 | ✔ |
| 3 | nType = 7 (Railroad)  Railroads owned = 3 | Rent = 150 | ✔ |
| 4 | nType = 8 (Utility)  Utilities owned = 1 | Rent = 4 | ✔ |
| 5 | nType = 8 (Utility)  Utilities owned = 2 | Rent = 10 | ✔ |
| buyProperty | 1 | South Line (Railroad) bought by Player 2  South Line price = 200  Player 1 cash = 1500  Bank = 7500 | Property owner = Player 2  Player 1 cash = 1300  Bank = 7700 | ✔ |
| 2 | Almond Drive bought by Player 1  Almond Drive price = 60  Player 2 cash = 1500  Bank = 7700 | Property owner = Player 1  Player 2 cash = 1440  Bank = 7760 | ✔ |
| 3 | Electric (Utility) bought by Player 3  Electric price = 150  Player 3 cash = 1500  Bank = 7760 | Property owner = Player 3  Player 3 cash = 1350  Bank = 7910 | ✔ |
| developProperty | 1 | Almond Drive Development = 2 | Almond Drive rent = 90  Almond Drive Development = 3 | ✔ |
| 2 | Almond Drive Development = 4  Hotel price = 240 | Almond Drive rent = 250  Almond Drive Development = 5  Property worth + 240 | ✔ |
| 3 | Almond Drive Development = 0 | Almond Drive rent = 10  Almond Drive Development = 1 | ✔ |
| tradeProperty | 1 | Almond Drive owner = Player 1  North Line owner = Player 2 | Almond Drive owner = Player 2  Add Almond Drive to spaces of Player 2  Remove North Line to spaces of Player 2  North Line owner = Player 1  Add North Line to spaces of Player 1  Remove Almond Drive to spaces of Player 1 | ✔ |
| 2 | Electric owner = Player 3  Connecticut owner = Player 1 | Electric owner = Player 1  Add Electric to spaces of Player 1  Remove Connecticut to spaces of Player 1  Connecticut owner = Player 3  Add Connecticut to spaces of Player 3  Remove Electric to spaces of Player 3 | ✔ |
| 3 | Kasoy Street owner = Player 2  Madison owner = Player 4 | Kasoy Street owner = Player 4  Add Kasoy Street to spaces of Player 4  Remove Madison to spaces of Player 4  Madison owner = Player 2  Add Madison to spaces of Player 2  Remove Kasoy Street to spaces of Player 2 | ✔ |
| Game | | | | |
| orderPlayers | 1 | Initial playerList = <A, B, C, D>  diceResults = <5, 7, 3, 10> | Final playerList = <D, B, A, C> | ✔ |
| 2 | Initial playerList = <A, B>  diceResults = <12, 5> | Final playerList = <A, B> | ✔ |
|  | 3 | Initial playerList = <C, A, B>  diceResults = <2, 12, 5> | Final playerList = <A, B, C> | ✔ |
| swapPlayers | 1 | Initial playerList = <A, B, C, D>  pos1 = 3, pos2 = 0 | Final playerList = <D, B, C, A> | ✔ |
| 2 | Initial playerList = <A, B, C>  pos1 = 1, pos2 = 2 | Final playerList = <A, C, B> | ✔ |
|  | 3 | Initial playerList = <A, B, C>  pos1 = 1, pos2 = 1 | Final playerList = <A, B, C> | ✔ |
| endTurn | 1 | activePlayer is at index 2 of playerList  NUM\_PLAYERS = 4 | activePlayer is now playerList.get(3) | ✔ |
| 2 | activePlayer is at index 2 of playerList  NUM\_PLAYERS = 3 | activePlayer is now playerList.get(0) | ✔ |
| 3 | activePlayer is at index 0 of playerList  NUM\_PLAYERS = 2 | activePlayer is now playerList.get(1) | ✔ |
| turn | 1 | activePlayer has nLocationIndex = 29  diceHolder = 3  gameBank.dValue = 4000 | Returns LAND\_ON\_START | ✔ |
| 2 | activePlayer has nLocationIndex = 29  diceHolder = 3  gameBank.dValue = 200 | Returns GAME\_IS\_END | ✔ |
| 3 | activePlayer has nLocationIndex = 0  diceHolder = 12  Space at index 12 is incom tax | Returns LAND\_ON\_TAX | ✔ |
| doesGameResume | 1 | All players own 1 property each | Returns true | ✔ |
| 2 | Player 2 owns 3 properties of type Blue and 2 properties of type Purple | Returns true | ✔ |
| 3 | Player 2 owns 3 properties of type Blue and 3 properties of type Pink | Returns false | ✔ |
| 4 | gameBank.dValue = 200  Player lands on chance and collects 300 from bank. | Returns false | ✔ |
| rankPlayers | 1 | playerList = <D, C, B, A>  Game ends due to player D not being able to pay jail fines.  A has dWorth = 10000  B has dWorth = 5070  C has dWorth = 2500  D has dWorth = 3000 (properties only) | ranked = <A, B, C, D> | ✔ |
| 2 | playerList = <A, B, C, D>  Game ends due to player A completing the gray and purple property sets.  A has dWorth = 2000 (440 are from properties)  B has dWorth = 2500  C has dWorth = 10000  D has dWorth = 1250 | ranked = <C, B, A, D> | ✔ |
| 3 | playerList = <A, B, C, D>  Game ends due to bank getting bankrupted.  A has dWorth = 10000  B has dWorth = 7500  C has dWorth = 9850  D has dWorth = 5370 | ranked = <A, C, B, D> | ✔ |