Matthew Walden B. Lua

Brian Jezreel So

MyEmpire Test Script

**Player class**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| purchase() | Player 1 must purchase Rodeo Drive so player 2 cannot purchase it and must pay rent | Turn 1: Matts  You landed on Rodeo Drive  1) Purchase  2) Ignore  Input 1 | Purchase Successful!  Owner: Matts  Turn 2: Brian  You landed on Rodeo Drive  Rodeo Drive is owned by Matts  You must pay rent | Purchase Successful!  Owner: Matts  Turn 2: Brian  You landed on Rodeo Drive  Rodeo Drive is owned by Matts  You must pay rent | Pass |
| Player 1 must not purchase Rodeo Drive so player 2 can purchase it | Turn 1: Matts  You landed on Rodeo Drive  1) Purchase  2) Ignore  Input 2  Turn 2: Brian  You landed on Rodeo Drive  1) Purchase  2) Ignore  Input 1 | Purchase Successful!  Owner: Brian | Purchase Successful!  Owner: Brian | Pass |
| Player 1 and Player 2 must ignore the property and nothing will happen | Turn 1: Matts  You landed on Rodeo Drive  1) Purchase  2) Ignore  **Input 2**  Turn 2: Brian  You landed on Rodeo Drive  1) Purchase  2) Ignore  **Input 2** | No output | No output | Pass |
| updateCash() | Final Cash must be equal to the sum of initial cash and added cash | Matts' initial cash: 1500.0  Input cash to add: **45** | Final cash: 1545.0 | Final cash: 1545.0 | Pass |
| Final cash must be equal to Initial cash when 0 is added | Matts' initial cash: 1500.0  Input cash to add: **0** | Final cash: 1500.0 | Final cash: 1500.0 | Pass |
| updateCash accepts negative values as well (in cases where the player pays another player or the bank). | Matts' initial cash: 1500.0  Input cash to add: **-10** | Final cash: 1490.0 | Final cash: 1490.0 | Pass |
| isBankrupt() | Should return false if player has cash greater than 0 | Matts's initial cash: 1500.0  Input cash to subtract: **1499** | Final cash: 1.0 | Final cash: 1.0 | Pass |
| Should return true if player has cash equal to 0 | Matts's initial cash: 1500.0  Input cash to subtract: **1500** | Final cash: 0.0  Matts is bankrupt | Final cash: 0.0  Matts is bankrupt | Pass |
| Should return true if player has cash less than 0 | Matts's initial cash: 1500.0  Input cash to subtract: **1510** | Final cash: -10.0  Matts is bankrupt | Final cash: -10.0  Matts is bankrupt | Pass |
| develop() | A house should be built if there are enough resources and property is not fully developed | Turn 1: Matts  You landed on Rodeo Drive, Owner: Matts  **Number of houses: 0**  **Number of hotels: 0** | Property Developed  Rodeo Drive, Owner: Matts  Number of houses: 1  Number of hotels: 0 | Property Developed  Rodeo Drive, Owner: Matts  Number of houses: 1  Number of hotels: 0 | Pass |
| A hotel should be built if the property is fully developed | Turn 1: Matts  You landed on Rodeo Drive, Owner: Matts  **Number of houses: 4**  **Number of hotels: 0** | Property Developed  Rodeo Drive, Owner: Matts  Number of houses: 4  Number of hotels: 1 | Property Developed  Rodeo Drive, Owner: Matts  Number of houses: 4  Number of hotels: 1 | Pass |
| The property cannot be developed anymore if it has a hotel | Turn 1: Matts  You landed on Rodeo Drive, Owner: Matts  **Number of houses: 4**  **Number of hotels: 1** | Property already has hotel  Rodeo Drive, Owner: Matts  Number of houses: 4  Number of hotels: 1 | Property already has hotel  Rodeo Drive, Owner: Matts  Number of houses: 4  Number of hotels: 1 | Pass |
| trade() | Player 1 initiates a trade with player 2. Player 2 accepts. | Player 1’s property list contains **Almond Drive.**  Player 2’s property list contains **Rodeo Drive.** | Player 1’s property list now has **Rodeo Drive.**  and is the new owner.  Player 2’s property list now has **Almond Drive,**  and is the new owner. | Player 1’s property list now has **Rodeo Drive.**  and is the new owner.  Player 2’s property list now has **Almond Drive,**  and is the new owner. | Pass |
|  | Player 1 initiates a trade with player 2. Player 2 declines. | Player 1’s property list contains **Almond Drive.**  Player 2’s property list contains **Rodeo Drive.** | Player 1’s property list still contains **Almond Drive** and is still the owner.  Player 2’s property list still contains **Rodeo Drive** and is still the owner. | Player 1’s property list still contains **Almond Drive** and is still the owner.  Player 2’s property list still contains **Rodeo Drive** and is still the owner. | **Fail** |
| payRent() | The player should pay rent if property is already owned and cash should be updated | Turn 2: Brian  You landed on Rodeo Drive, Owner: Matts  Number of houses: 0  Number of hotels: 0  You must pay rent  Matts Initial Cash: **1500.0**  Brian Initial Cash: **1500.0** | Matts Final Cash: 1506.0  Brian Final Cash: 1494.0 | Matts Final Cash: 1506.0  Brian Final Cash: 1494.0 | Pass |
| If renter cash becomes negative, property owner should still receive full amount | Turn 2: Brian  You landed on Rodeo Drive, Owner: Matts  Number of houses: 0  Number of hotels: 0  You must pay rent  Matts Initial Cash: **1500.0**  Brian Initial Cash: **5.0** | Matts Final Cash: 1506.0  Brian Final Cash: -1.0  Brian is bankrupt | Matts Final Cash: 1506.0  Brian Final Cash: -1.0  Brian is bankrupt | Pass |
| If renter has a double rent chance card, he must pay double the amount and owner should also receive double the amount | Turn 2: Brian  **You have double rent**  You landed on Rodeo Drive, Owner: Matts  Number of houses: 0  Number of hotels: 0  You must pay rent  Matts Initial Cash: **1500.0**  Brian Initial Cash: **1500.0** | Matts Final Cash: 1512.0  Brian Final Cash: 1488.0 | Matts Final Cash: 1512.0  Brian Final Cash: 1488.0 | Pass |
| drawCard() | Player draws top card on Chance deck. This tests only whether the card is added to the player’s list of cards | Top card drawn is a Get Out of Jail card. | The card is added to the player’s chanceList. The chanceList now contains the card. | The card is added to the player’s chanceList. The chanceList now contains the card. | Pass |

**Ownables**

**Property class**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| hasHotel() | The property is undeveloped. | Number of houses = 0  **Number of hotels = 0** | false | false | Pass |
| The property is partially developed. | Number of houses = 3  **Number of hotels = 0** | false | false | Pass |
| The property has four houses and one hotel | Number of houses = 4  **Number of hotels = 1** | true | true | Pass |
| setRent() | The player has one property of the same color | Number of homogenous properties = 1  **Rent price = $160** | New rent price =  $160 + $0 = $160 | New rent price =  $160 + $0 = $160 | Pass |
| The player has two properties of the same color | Number of homogenous properties = 2  **Rent price = $160** | New rent price =  $160 + $10 = $170 | New rent price =  $160 + $10 = $170 | Pass |
| The player has three properties of the same color | Number of homogenous properties = 3  **Rent price = $160** | New rent price =  $160 + $20 = $180 | New rent price =  $160 + $20 = $180 | Pass |
| canBuildHotel() | The player has one property of a color group. The property is fully developed.  Build a hotel on the first property. | Number of homogenous properties = 1  Number of houses = 4 | canBuildHotel() returns true | canBuildHotel() returns true | Pass |
| The player has two properties of a color group. Both properties are fully developed.  Build a hotel on the first property. | Number of homogenous properties = 2  1st property houses = 4  2nd property houses = 3 | canBuildHotel() returns false | canBuildHotel() returns false | Pass |
| The player has two properties of a color group. Both properties are fully developed.  Build a hotel on the first property. | Number of homogenous properties = 2  1st property houses = 4  2nd property houses = 4 | canBuildHotel() returns true | canBuildHotel() returns true | Pass |
| build() | The player has two properties of a color group. One property is fully developed, but the other is not. Build a hotel on the first property. | Number of homogenous properties = 1  Number of houses = 4 | Number of buildings =  4 + 1 = 5.  Number of houses = 4  Number of hotels = 1 | Number of buildings = 5.  Number of houses = 4  Number of hotels = 1 | Pass |
| The player has two properties of a color group. Both properties are fully developed.  Build a hotel on the first property. | Number of homogenous properties = 2  1st property houses = 4  2nd property houses = 3 | For the first property:  Number of buildings = 4  Number of houses = 4  Number of hotels = 0 | Number of buildings = 4  Number of houses = 4  Number of hotels = 0 | Pass |
| The player has two properties of a color group. Both properties are fully developed.  Build a hotel on the first property. | Number of homogenous properties = 2  1st property houses = 4  2nd property houses = 4 | For the first property:  Number of buildings = 4  Number of houses = 4  Number of hotels = 1 | For the first property:  Number of buildings = 4  Number of houses = 4  Number of hotels = 1 | Pass |

**Railroad**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **getRent()** | Player has 1 railroad.  Other player has $2000 in cash. | Number of owned railroads = 1  Other player’s cash =  $2000 | getRent() returns $25.  Other player’s cash = $1975 | getRent() returns $25.  Other player’s cash = $1975 | Pass |
| Player has 2 railroads  Other player has $2000 in cash. | Number of owned railroads = 2  Other player’s cash =  $2000 | getRent() returns $50  Other player’s cash =  $1950 | getRent() returns $50  Other player’s cash =  $1950 | Pass |
| Player has 3 railroads  Other player has $2000 in cash. | Number of owned railroads = 3  Other player’s cash =  $2000 | getRent() returns $150  Other player’s cash =  $1850 | getRent() returns $150  Other player’s cash =  $1850 | Pass |

**Utility**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **getRent()** | Player has 1 utility.  Other player has $2000 in cash.  Dice roll shows a 7. | Number of owned utilities = 1  Other player’s cash =  $2000  rollDice() = 7 | getRent() returns  4 \* 7 = **$28.**  Other player’s cash = $2000 - $28 = $1972 | getRent() returns  4 \* 7 = **$28.**  Other player’s cash = $2000 - $28 = $1972 | Pass |
| Player has 2 utilities  Other player has $2000 in cash.  Dice roll shows a 7. | Number of owned  utilities = 2  Other player’s cash =  $2000  rollDice() = 3 | getRent() returns  10 \* 7 = **$70**  Other player’s cash =  $2000 - $28 = **$1972** | getRent() returns  10 \* 7 = **$70**  Other player’s cash =  $2000 - $28 = **$1972** | Pass |

**Bank**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **updateBank()** | Player 1 pays bank $12. | **Bank initial cash = $2500**  **Player initial cash = $1500** | **Bank final cash = $2512**  **Player final cash = $1488** | **Bank final cash = $2512**  **Player final cash = $1488** | **Pass** |
| Player 1 receives $200 from bank | **Bank initial cash = $2500**  **Player initial cash = $1500** | **Bank final cash = $2300**  **Player final cash = $1700** | **Bank final cash = $2300**  **Player final cash = $1700** | **Pass** |

**Corners**

**Community Service**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **actOnPlayer()** | Player lands on Community Service space with a current cash value of $1500. | Current cash = $1500  Land on Community Service space.  Bank’s cash = $5000 | New cash = $1450  Bank’s cash = $5050 | New cash = $1450  Bank’s cash = $5050 | Pass |

**Free Parking**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **actOnPlayer()** | Player 1 lands on Free Parking | Player 1 lands on Free Parking | **Player 1 end turn**  **Player 2 start turn** | **Player 1 end turn**  **Player 2 start turn** | **Pass** |

**Jail**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **actOnPlayer()** | Player lands on JAIL and ends turn with a $50 debt. | Player position = JAIL  Player cash = $1500 | Player’s new cash = $1450 on next turn | Player’s new cash = $1450 on next turn | **Pass** |

**Start**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **actOnPlayer()** | Player lands on START and collects $200. | Player cash = $1500  Bank cash = $5000  Player lands on START. | Player’s cash = **$1700.**  Bank’s new cash = **$4800** | Player’s cash = **$1700.**  Bank’s new cash = **$4800** | Pass |

**Taxable**

**Income Tax**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **chargeTax()** | Player lands on Income Tax space with a current cash value of $1500.  10% of cash is $150, less than $200.  The bank has a value of $5000 | Current cash = $1500  Land on Income Tax space.  Bank’s cash = $5000 | Player new cash =  $1500 – $200 = **$1300**  Bank’s cash =  $5000 + $200 = **$5200** | Player new cash =  $1500 – $200 = **$1300**  Bank’s cash =  $5000 + $200 = **$5200** | Pass |
| Player lands on Income Tax space with a current cash value of $2500.  10% of cash is $250, more than $200.  The bank has a value of $5000 | Current cash = $2500  Land on Income Tax space.  Bank’s cash = $5000 | Player new cash =  $1500 – $250 = **$1250**  Bank’s cash =  $5000 + $250 = **$5250** | Player new cash =  $1500 – $250 = **$1250**  Bank’s cash =  $5000 + $250 = **$5250** | Pass |

**Luxury Tax**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **chargeTax()** | Player lands on Luxury Tax space with a current cash value of $1500.  The bank has a value of $5000 | Current cash = $1500  Land on Luxury Tax space.  Bank’s cash = $5000 | Player new cash =  $1500 – $75 = **$1425**  Bank’s cash =  $5000 + $75 = **$5075** | Player new cash =  $1500 – $200 = **$1425**  Bank’s cash =  $5000 + $75 = **$5075** | Pass |

**Chance**

**ChanceCard**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **discard()** | Player discards a RentChance card.  ChanceDeck’s discard pile size is 0. | RentChance rc = new RentChance();  rc.discard(); | ChanceDeck’s discard pile size = 1 | ChanceDeck’s discard pile size = 1 | Pass |

**DonateChance and PayTaxChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **giveMoney()** | Player draws a DonateChance card. Player’s cash is $1500.  Bank’s cash is $5000.  The random value to deduct is $100. | Current cash = $1500  Draw a DonateChance card  Bank’s cash = $5000 | New cash = $1400  Draw a DonateChance card  Bank’s cash = $5100 | New cash = $1400  Draw a DonateChance card  Bank’s cash = $5100 | Pass |

**ProceedToOwnable**

**ProceedToProperty**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **proceedTo**  **Ownable()** | Player 1 draws a ProceedToProperty Card. Player proceeds to unowned property | Player 1 goes to Almond Drive from ChanceCard random choice. Almond Drive is unowned | Player 1 can purchase Almond Drive and purchases it. | Player 1 can purchase Almond Drive and purchases it. | **Pass** |
| Player 2 draws a ProceedToProperty Card. Player 2 proceeds to owned property | Player 2 goes to Almond Drive from ChanceCard random choice. Almond Drive is owned by Player 1. | Player 2 has to either trade, or pay rent | Player 2 has to either trade, or pay rent | **Pass** |

**ProceedToUtility**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **proceedTo**  **Ownable()** | Player 1 draws a ProceedToUtility  Card. Player proceeds to nearest utility | Player 1 draws a ProceedToUtility  Card.  Nearest utility is Electric | Player 1 lands on Electric | Player 1 lands on Electric | **Pass** |

**ProceedToRailroad**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **proceedTo**  **Ownable()** | Player 1 draws a ProceedToUtility  Card. Player proceeds to nearest utility | Player 1 draws a ProceedToUtility  Card.  Nearest utility is Electric | Player 1 lands on Electric | Player 1 lands on Electric | **Pass** |

**ReceiveMoneyChance**

**AdvanceToStartChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **collectMoney()** | Player advances to START | Player advances to START.  Player cash = $1500 | Player new cash = **$1700** | Player new cash = **$1700** | **Pass** |

**BankPaysPlayerChance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **collectMoney()** | Player draws BankPaysPlayerChance. | Player draws BankPaysPlayer  Chance.  Bank’s cash = $5000  Player cash = $1500 | Bank new cash = $4950  Player new cash = $1550 | | Bank new cash = $4950  Player new cash = $1550 | **Pass** |

**BirthdayChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **collectMoney()** | Player draws BirthdayChance | Player draws BirthdayChance  Bank’s cash = $5000  Player cash = $1500 | Bank new cash = $4700  Player new cash = $1800 | Bank new cash = $4700  Player new cash = $1800 | **Pass** |

**TaxRefundChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **collectMoney()** | Player draws TaxRefundChance | Player draws TaxRefundChance  Bank’s cash = $5000  Player cash = $1500 | Bank new cash = $4900  Player new cash = $1600 | Bank new cash = $4900  Player new cash = $1600 | **Pass** |

**WonCompetitionChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **collectMoney()** | Player draws WonCompetitionChance | Player draws WonCompetitionChance  Bank’s cash = $5000  Player cash = $1500 | Bank new cash = $4850  Player new cash = $1750 | Bank new cash = $4850  Player new cash = $1750 | **Pass** |

**Rent**

**DilapidatedPropertyChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **applyRentTo**  **Ownable()** | Brian draws this card.  The card is applied to his Almond Drive | Brian draws card  Almond Drive  initial rent = $2 | Almond Drive  New rent = $1.8 | Almond Drive  New rent = $1.8 | **Pass** |
| Brian draws this card.  Brian has no properties | Brian draws card | Card is added to discard pile | Card is added to discard pile | **Pass** |

**DevelopedPropertyChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **applyRentTo**  **Ownable()** | Brian draws this card.  The card is applied to his J.Abad Santos | Brian draws card  Almond Drive  init house cost = $50  init hotel cost = $250  init rent price = $12 | Almond Drive  new house cost = $25  new hotel cost = $50  new rent price = $18 | Almond Drive  new house cost = $25  new hotel cost = $50  new rent price = $18 | **Pass** |
| Brian draws this card.  Brian has no properties | Brian draws card | Card is added to discard pile | Card is added to discard pile | **Pass** |

**DilapidatedUtilRailChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **applyRentTo**  **Ownable()** | Brian draws this card.  The card is applied to his Metro Railroad | Brian draws card  Metro  init rent price = $2 | Brian’s Metro  new rent price = $1.8 | Brian’s Metro  new rent price = $1.8 | **Pass** |
| Brian draws this card.  Brian has no properties | Brian draws card | Card is added to discard pile | Card is added to discard pile | **Pass** |

**DevelopedUtilRailChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **applyRentTo**  **Ownable()** | Matts draws this card.  The card is applied to his Water Utility. Matts only has one | Matts draws card  Matts’ Water  init rent price = $2 | Matts Water  new rent price = $1.8 | Matts’ Water  new rent price = $1.8 | **Pass** |

**Double Rent – unit and integration test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **applyRentTo**  **Ownable()** | Matts draws this card.  Matts chooses his Almond Drive to apply rent to. | Matts draws this card.  Almond Drive rent = $2 | Almond Drive rent = $4 | Almond Drive rent = $4 |  |
| Brian lands on Matts’ Almond Drive | Almond Drive current rent = $4 | Brian pays Matts $4.  Almond Drive rent is reverted to $2 | Brian pays Matts $4.  Almond Drive rent is reverted to $2 |  |
| Matts draws this card and applies to his Almond Drive.  Game continues.  Matts draws the card again and applies again to Almond Drive. | Almond Drive rent = $2  Matts draws this card and applies to Almond Drive.  Almond Drive new rent = $4  Matts draws this card again and applies to Almond Drive. | Almond Drive  new rent = $8 | Almond Drive  new rent = $8 |  |
| Brian lands on Matts’ Almond Drive | Almond Drive current rent = $8 | Brian pays Matts $8  Rent is reverted to $4 | Almond Drive new rent = $4 |  |

**TakeATripChance**

**TripToPropertyChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **takeATrip()** | Matts draws this card.  Target property is Kasoy Street, unowned.  Player collects money when passing START | Matts draws this card  Matts’ initial cash = $1750  Matts passes thru START | Matts’ new cash = $1950  Matts can purchase or ignore Kasoy St. | Matts’ new cash = $1950  Matts can purchase or ignore Kasoy St. | **Pass** |

**GoToJailChance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **takeATrip()** | Brian draws this card.  Brian passes START while going to Jail | Brian draws this card  Brian cash = $1500  Brian goes to Jail | Brian ends turn  Brian’s cash = $1500  Brian has a debt of $50 | Brian ends turn  Brian’s cash = $1500  Brian has a debt of $50 | **Pass** |

**GetOutOfJail**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| **getOutOfJail()** | Brian is in jail.  Brian has this card with him and uses it. | Brian’s cash = $1500  Brian uses GetOutOfJail | Brian’s new cash = $1500 | Brian’s new cash = $1500 | **Pass** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function name** | **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass**  **Or**  **Fail** |
| applyRentToPlayer() | The target player lands on the property where the card is applied. | Property rent value = $10  Double chance card is applied | New rent = $20 | New rent = $20 | Pass |