MARQUEZ, GABRIEL C.
ESCOBAR, MARK ADRIAN H.



**CLASS:** MainGame

| METHOD | # | DESCRIPTION | SAMPLE INPUT / STATUS | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| gameIntro() | 1 | 2 Player selected | On GUI: Player 2 is selected | numberOfPlayersInGame = 2<br>2 Player objects are instantiated within Player[] array. | numberOfPlayersInGame = 2<br>2 Player objects are instantiated within Player[] array. | P |
| | | 3 Players selected | On GUI: Player 2 is selected | numberOfPlayersInGame = 3<br>3 Player objects are instantiated within Player[] array. | numberOfPlayersInGame = 3<br>3 Player objects are instantiated within Player[] array. | P |
| | | Start game exited | On GUI: Click on X button | Function not called, program terminates | Function not called, program terminates | P |
| throwDice() | 2 | Number generated is not 0 (always generate between 1 - 10) | -mt terminal argument NOT included | Generated number is between 1 - 10, assigned to current player | Generated number is between 1 - 10, assigned to current player | P |
| | | Number generated is 0 | -mt terminal argument NOT included | Number is generated again until non-zero result, then is assigned to current player | Number is generated again until non-zero result, then is assigned to current player | P |
| | | Number generated is user specified | -mt terminal argument included;<br><br>4 | Assign 4 to current player, move 4 times. | Assign 4 to current player, move 4 times. | P |
| movePlayer() | 3 | Number of moves is 5, current location is [0,0] | Career life path | Player moves to [5,0] and lands at orange space | Player moves to [5,0] and lands at orange space | P |
| | | Number of moves is 8, player is currently at junction space [5,7] | Family life path | Player moves to [7,13[ and lands at blank space | Player moves to [7,13[ and lands at blank space | P |
| | | Number of moves causes the player to travel through a magenta space<br><br>Current location: [7,13] | Family life path | Player stops at [9,12] and forfeits remaining moves. Get Married event is initiated. Game allows current player to generate number and move again after this. | Player stops at [9,12] and forfeits remaining moves. Get Married event is initiated. Game allows current player to generate number and move again after this. | P |
| | | Number of moves causes the player to move past the end space at [15,19]<br><br>Current location: [12,19] | Career life path | Player stops at the end space [15,19], forfeits remaining moves. Player retires. | Player stops at the end space [15,19], forfeits remaining moves. Player retires. | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| movePastMagentaSpace() | 4 | Player stops at Graduate magenta space | College life path | Shift player coordinate upward, decrement number of moves assigned to player | Shift player coordinate upward, decrement number of moves assigned to player | P |
| | | Player stops at Get Married magenta space | Career life path | Shift player coordinate leftwards, decrement number of moves assigned to player | Shift player coordinate leftwards, decrement number of moves assigned to player | P |
| | | Player stops at Baby magenta space | Family life path | Shift player coordinate downwards, decrement number of moves assigned to player | Shift player coordinate downwards, decrement number of moves assigned to player | P |
| executeStartLifePathCareer() | 5 | Drawn career card requires a degree | Career life path | Ignore card, draw next career card | Ignore card, draw next career card | P |
| | | Drawn career card does not require a degree | Career life path | Accept card | Accept card | P |
| | | Player 2 chooses career starting path | Career life path | Career and salary card assigned to player 2 | Career and salary card assigned to player 2 | P |
| executeStartLifePathCollegeCareer() | 6 | Player 1 chooses college starting path | College life path | Player 1 has starting loan of $40000 and interest $5000 * 2 | Player 1 has starting loan of $40000 and interest $5000 * 2 | P |
| | | Player 2 chooses college starting path | College life path | Player 2 has starting loan of $40000 and interest $5000 * 2 | Player 2 has starting loan of $40000 and interest $5000 * 2 | P |
| | | Player 3 chooses college starting path | College life path | Player 3 has starting loan of $40000 and interest $5000 * 2 | Player 3 has starting loan of $40000 and interest $5000 * 2 | P |
| executeOrangeSpace() | 7 | Action Card drawn is Tax Refund | Current player: 1 | Player 1 money balance +30000.00 | Player 1 money balance +30000.00 | P |
| | | Action card drawn is Lawsuit | Current player:1<br>Chosen player: 3 | Player 1 money balance -30000.00<br><br>Player 3 money balance +30000.00 | Player 1 money balance -30000.00<br><br>Player 3 money balance +30000.00 | P |
| | | Action card drawn is the last in the deck | N/A | Deck of action card is shuffled, reset head value to point to the first card | Deck of action card is shuffled, reset head value to point to the first card | P |
| executeGetMarried() | 8 | Player 1 is not married | N/A | Get Married magenta space event is called, return number generated | Get Married magenta space event is called, return number generated | P |
| | | Player 3 is not married | N/A | Get Married magenta space event is called, return number generated | Get Married magenta space event is called, return number generated | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Player 3 is already married | N/A | Return -1 | Return -1 | P |
| executeGraduate() | 9 | Player 1 stops on Graduate magenta space | N/A | Set Player 1 to be graduated | Set Player 1 to be graduated | P |
| | | Player 2 stops on Graduate magenta space | N/A | Set Player 2 to be graduated | Set Player 2 to be graduated | P |
| | | Player 3 stops on Graduate magenta space | N/A | Set Player 3 to be graduated | Set Player 3 to be graduated | P |
| executeCollegeCareerChoice _DrawCareer() | 10 | Drawn career card is currently being used by another player | N/A | Ignore card, draw again | Ignore card, draw again | P |
| | | Dawn career card is currently being used by the current player | N/A | Ignore card, drawn again | Ignore card, drawn again | P |
| | | Dawn career card is not currently being used by anyone | N/A | Accept career card and allow it to show on GUI for the player to choose | Accept career card and allow it to show on GUI for the player to choose | P |
| executeCollegeCareerChoice _DrawSalary() | 11 | Drawn salary card has salary 10000.00 and tax due 1000.00 | N/A | Accept salary card and allow it to show on GUI for the player to choose | Accept salary card and allow it to show on GUI for the player to choose | P |
| | | Drawn salary card has salary 20000.00 and tax due 2000.00 | N/A | Accept salary card and allow it to show on GUI for the player to choose | Accept salary card and allow it to show on GUI for the player to choose | P |
| | | 2 salary cards are already drawn | | Shuffle deck | Shuffle deck | P |
| executeCollegeCareerChoice _CareerChosen(String typeOfCardChosen) | 12 | Accepted career card is LAWYER | LAWYER | Set LAWYER career card ownership status to true, set LAWYER as current player's career | Set LAWYER career card ownership status to true, set LAWYER as current player's career | P |
| | | Accepted career card is ACCOUNTANT | ACCOUNTANT | Set ACCOUNTANT career card ownership status to true, set ACCOUNTANT as current player's career | Set ACCOUNTANT career card ownership status to true, set ACCOUNTANT as current player's career | P |
| | | Accepted career card is SERVER | SERVER | Set SERVER career card ownership status to true, set SERVER as current player's career | Set SERVER career card ownership status to true, set SERVER as current player's career | P |
| executeCollegeCareerChoice _SalaryChosen(String salary) | 13 | Accepted Salary is $10000.00 | 10000.00 | Set current player salary to 10000.00 and tax due to 1000.00 | Set current player salary to 10000.00 and tax due to 1000.00 | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Accepted Salary is $20000.00 | 20000.00 | Set current player salary to 20000.00 and tax due to 2000.00 | Set current player salary to 20000.00 and tax due to 2000.00 | P |
| | | Accepted Salary is $30000.00 | 30000.00 | Set current player salary to 30000.00 and tax due to 3000.00 | Set current player salary to 30000.00 and tax due to 3000.00 | P |
| executeJobSearch() | 14 | Drawn career card is currently being used by another player | N/A | Ignore card, draw again | Ignore card, draw again | P |
| | | Dawn career card is not currently being used by anyone | N/A | Accept career card and allow it to show on GUI for the player to choose | Accept career card and allow it to show on GUI for the player to choose | P |
| | | Drawn salary card has salary 10000.00 and tax due 1000.00 | N/A | Accept salary card and allow it to show on GUI for the player to choose | Accept salary card and allow it to show on GUI for the player to choose | P |
| executeBuyHouse(String chosenTypeOfHouse) | 15 | Chosen House card is Split-Level | Split-Level | Reduce current player money by -40000.00, assign Split-Level to current player | Reduce current player money by -40000.00, assign Split-Level to current player | P |
| | | Chosen House card is Mobile Home | Mobile Home | Reduce current player money by -60000.00, assign Mobile Home to current player | Reduce current player money by -60000.00, assign Mobile Home to current player | P |
| | | Chosen House card is Log Cabin | Log Cabin | Reduce current player money by -80000.00, assign Log Cabin to current player | Reduce current player money by -80000.00, assign Log Cabin to current player | P |
| executeBlueCard() | 16 | Drawn blue card is LAWYER, current player career is LAWYER | N/A | return “Same career! Paid yourself.” | return “Same career! Paid yourself.” | P |
| | | Drawn blue card is SALARY TAX DUE, other player career is ACCOUNTANT | N/A | Return “Paid to Player *<number>*” | Return “Paid to Player *<number>*” | P |
| | | Drawn blue card is TIP THE SERVER, no other player has SERVER as career | N/A | Return “Paid to Bank” | Return “Paid to Bank” | P |
| executeGreenSpace() | 17 | Player lands on Pay Day | Change Career Path | Returned space ID = 0 | Returned space ID = 0 | P |
| | | Player lands on Pay Raise | Change Career Path | Returned space ID = 1 | Returned space ID = 1 | P |
| | | Player lands on a non-green space | N/A | Method not called | Method not called | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| executeBaby() | 18 | Generated number is even | 6 | Set current player to have twins | Set current player to have twins | P |
| | | Generated number is odd | 3 | Set current player to have one baby | Set current player to have one baby | P |
| | | Player has twins, current number of players is 3 | N/A | Current player collects 10000.00 * 3<br>Other players money balance reduce by 10000.00 | Current player collects 10000.00 * 3<br>Other players money balance reduce by 10000.00 | P |
| retire() | 19 | Player 1 first reaches the end space, has a baby, and has a house | N/A | 100000.00 added to balance<br>$10000 collected from bank<br>House sold and amount is added to balance<br>All loans are paid | 100000.00 added to balance<br>$10000 collected from bank<br>House sold and amount is added to balance<br>All loans are paid | P |
| | | Player 2 first reaches the end space, has twins, and has a house | N/A | 50000.00 added to balance<br>$20000 collected from bank<br>House sold and amount is added to balance<br>All loans are paid | 50000.00 added to balance<br>$20000 collected from bank<br>House sold and amount is added to balance<br>All loans are paid | P |
| | | Player 2 first reaches the end space, has no baby, and has a house | N/A | 20000.00 added to balance<br>All loans are paid | 20000.00 added to balance<br>All loans are paid | P |
| determineWinner() | 20 | Player 1 has the most money balance | Player 1: $675000.00<br>Player 2: $490000.00 | Player 1 declared winner | Player 1 declared winner | P |
| | | Player 3 has the most money balance | Player 1: $675000.00<br>Player 2: $490000.00<br>Player 3: $712000.00 | Player 3 declared winner | Player 3 declared winner | P |
| | | Player 1 and Player 2 equal money balance | Player 1: $490000.00<br>Player 2: $490000.00<br>Player 3: $712000.00 | Declare tie | Declare tie | P |
| nextTurn() | 21 | Current player is 1<br>Number of players 2 | N/A | Hand turn to player 2 | Hand turn to player 2 | P |
| | | Current player is 3<br>Number of players 3 | N/A | Hand turn to player 1 | Hand turn to player 1 | P |
| | | Current player is 2<br>Number of players 3<br>Player 3 has reached end space | N/A | Hand turn to player 1 | Hand turn to player 1 | P |

**CLASS:** Player

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| addMoneyBalance(double amount) | 1 | Adds amount to the current money balance of the player | 30000.00 | <30000.00 is added to the current balance of the player> | <30000.00 is added to the current balance of the player> | P |
| | | Adds amount to the current money balance of the player | 50000.00 | <50000.00 is added to the current balance of the player> | <50000.00 is added to the current balance of the player> | P |
| | | Adds amount to the current money balance of the player | 2000.00 | <2000.00 is added to the current balance of the player> | <2000.00 is added to the current balance of the player> | P |
| addLoan() | 2 | Player 1 currently has $20000 balance, draws VISIT A PLACE (deduct $30000) | N/A | Warning! A loan added to a player. Make sure to pay it back.<br>Updated loan and interest for a player total is now 25000.0<br>$30000 deducted from Player 1<br>Updated current MONEY: 10000.0 | Warning! A loan added to a player. Make sure to pay it back.<br>Updated loan and interest for a player total is now 25000.0<br>$30000 deducted from Player 1<br>Updated current MONEY: 10000.0 | P |
| | | Player 1 needs to make more than one loan to pay the other players. Drawn card is CHRISTMAS BONUS (Pay all players $30000) | N/A | Warning! A loan added to a player. Make sure to pay it back.<br>Updated loan and interest for a player total is now 25000.0<br>PAID Player 2 $30000.00<br>Warning! A loan added to a player. Make sure to pay it back.<br>Updated loan and interest for a player total is now 50000.0<br>PAID Player 3 $30000.00<br>Updated current MONEY: $0.0 | Warning! A loan added to a player. Make sure to pay it back.<br>Updated loan and interest for a player total is now 25000.0<br>PAID Player 2 $30000.00<br>Warning! A loan added to a player. Make sure to pay it back.<br>Updated loan and interest for a player total is now 50000.0<br>PAID Player 3 $30000.00<br>Updated current MONEY: $0.0 | P |
| reduceMoneyBalance(double amount) | 3 | Subtracts amount to the current money balance of the player. If the amount goes lower than 0, addLoan() will be called. Balance is currently at $200000.00 | 40000.00 | <40000.00 is subtracted from the balance of the player> | <40000.00 is subtracted from the balance of the player> | P |
| | | Subtracts amount to the current money balance of the player. If the amount goes lower than 0, | 200000.00 | $0.00 balance left | $0.00 balance left | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | addLoan() will be called. Balance is currently at $200000.00 | | | | |
| | | Subtracts amount to the current money balance of the player. If the amount goes lower than 0, addLoan() will be called. Balance is currently at $200000.00 | 220000.00 | <Method calls payLoan(), adds $20000 loan and $5000 interest to player, update current balance $0.0> | <Method calls payLoan(), adds $20000 loan and $5000 interest to player, update current balance $0.0> | P |
| setLoan() | 4 | Player chooses College Life Path as life path | College Life Path | moneyLoan += 40000.00 moneyLoanInterest += 10000.00 | moneyLoan += 40000.00 moneyLoanInterest += 10000.00 | P |
| | | Player chooses Career Life Path as life path | Career Life Path | Method not called | Method not called | P |

**CLASS:** ActionCard

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| assignDescriptions() | 1 | MainID : 0<br>subID : 0<br>(Collect From Bank card) | N/A | TypeOfCard = TAX REFUND<br>Description: Get your money back from the taxpayers!<br>toDoAction: Collect $30000 From The Bank | TypeOfCard = TAX REFUND<br>Description: Get your money back from the taxpayers!<br>toDoAction: Collect $30000 From The Bank | P |
| | | MainID : 1<br>subID : 1<br>(Pay The Bank card) | N/A | TypeOfCard: VISIT A PLACE<br>Description: Go spend some good time in a good place!<br>toDoAction: Pay $30000 To The Bank | TypeOfCard: VISIT A PLACE<br>Description: Go spend some good time in a good place!<br>toDoAction: Pay $30000 To The Bank | P |
| | | MainID : 2<br>subID : 0<br>(Pay The Player card) | N/A | TypeOfCard: LAWSUIT<br>Description: Settle a case<br>toDoAction: Choose A Player, then Pay $30000 | TypeOfCard: LAWSUIT<br>Description: Settle a case<br>toDoAction: Choose A Player, then Pay $30000 | P |
| doAction(Player[] players, int turn, int numberOfPlayersInGame) | 2 | Drawn card is a Collect From Player card | N/A | PLAYER 1's turn! \| MONEY: 230000.0 \| LOAN: 0.0<br>Action card drawn: FILE A LAWSUIT!<br>Bring them to court!<br>Updated current | PLAYER 1's turn! \| MONEY: 230000.0 \| LOAN: 0.0<br>Action card drawn: FILE A LAWSUIT!<br>Bring them to court!<br>Updated current MONEY: | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | MONEY: $260000.0 | $260000.0 | |
| | | Drawn card is a Collect From Bank card | N/A | PLAYER 1's turn! \| MONEY: 200000.0 \| LOAN: 0.0\| INTEREST: 0.0<br>Action card drawn: TAX REFUND<br>Get your money back from the taxpayers!<br><br>$30000 added to Player 1<br>Updated MONEY: 230000.0 | PLAYER 1's turn! \| MONEY: 200000.0 \| LOAN: 0.0\| INTEREST: 0.0<br>Action card drawn: TAX REFUND<br>Get your money back from the taxpayers!<br><br>$30000 added to Player 1<br>Updated MONEY: 230000.0 | P |
| | | Drawn card is a Pay The Bank card | N/A | PLAYER 1's turn! \| MONEY: 200000.0 \| LOAN: 0.0\| INTEREST: 0.0<br>Action card drawn: TRAFFIC VIOLATION<br>Oopsies! No escaping the law!<br><br>$30000 deducted from Player 1<br>Updated current MONEY: 170000.0 | PLAYER 1's turn! \| MONEY: 200000.0 \| LOAN: 0.0\| INTEREST: 0.0<br>Action card drawn: TRAFFIC VIOLATION<br>Oopsies! No escaping the law!<br><br>$30000 deducted from Player 1<br>Updated current MONEY: 170000.0 | P |

**CLASS:** BlueCard

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| assignDescription() | 1 | mainID = 0 (LAWSUIT) | N/A | typeOfCard = "LAWSUIT"<br><br>toDoAction = "Pay the amount indicated." | typeOfCard = "LAWSUIT"<br><br>toDoAction = "Pay the amount indicated." | P |
| | | mainID = 1 (SALARY TAX DUE) | N/A | typeOfCard = "SALARY TAX DUE"<br><br>toDoAction = "Pay the tax due for current salary." | typeOfCard = "SALARY TAX DUE"<br><br>toDoAction = "Pay the tax due for current salary." | P |
| | | mainID = 2 (TIP THE SERVER) | N/A | typeOfCard = “TIP THE SERVER”<br><br>toDoAction = "Press for random number, then pay $1000 * generated number." | typeOfCard = “TIP THE SERVER”<br><br>toDoAction = "Press for random number, then pay $1000 * generated number." | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| doAction (Player[] players, int turn, int nRand) | 2 | Drawn card is LAWSUIT, current player career is LAWYER | N/A | 15000.00 added to current player | 15000.00 added to current player | P |
| | | Drawn career card is LAWSUIT, other player career is LAWYER | N/A | Current player pays to the player with LAWYER career between $50000 to $150000 | Current player pays to the player with LAWYER career between $50000 to $150000 | P |
| | | Drawn career card is LAWSUIT, no other player has LAWYER career | N/A | Pay to bank between $50000 to $150000 | Pay to bank between $50000 to $150000 | P |

**CLASS:** DeckOfBlueCards, DeckOfActionCards, DeckOfCareerCards, DeckOfSalaryCards

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| pop() | 1 | Head points at the topmost card | N/A | Return the topmost card | Return the topmost card | P |
| | | Head points somewhere in the middle of the deck | N/A | Return the card pointed by the head | Return the card pointed by the head | P |
| | | Head points at the last card | N/A | Return the card, shuffle the deck, reset head value | Return the card, shuffle the deck, reset head value | P |

**CLASS:** CareerCard

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| assignDescriptions() | 1 | mainID = 0 (LAWYER) | N/A | typeOfCard = "LAWYER"<br>DegreeRequired = true<br>payRaise = [5, 8] | typeOfCard = "LAWYER"<br>DegreeRequired = true<br>payRaise = [5, 8] | P |
| | | mainID = 1 (ACCOUNTANT) | N/A | typeOfCard = "ACCOUNTANT"<br>DegreeRequired = true<br>payRaise = [4, 7] | typeOfCard = "ACCOUNTANT"<br>DegreeRequired = true<br>payRaise = [4, 7] | P |
| | | mainID = 2 (COMPUTER CONSULTANT) | N/A | typeOfCard = "COMPUTER CONSULTANT" | typeOfCard = "COMPUTER CONSULTANT" | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | DegreeRequired = true<br><br>payRaise = [3, 7] | DegreeRequired = true<br><br>payRaise = [3, 7] | |

**CLASS:** SalaryCard

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| toDoAction(Player player) | 1 | Assigns the card's value to the player.<br><br>Current values :<br>salary = 20000.00<br>taxDue = 2000.00 | <Player 1> | <20000.00 salary and 2000.00 taxDue is assigned to Player 1> | <20000.00 salary and 2000.00 taxDue is assigned to Player 1> | P |
| | | Assigns the card's value to the player.<br><br>Current values :<br>salary = 40000.00<br>taxDue = 4000.00 | <Player 2> | <40000.00 salary and 4000.00 taxDue is assigned to Player 2> | <40000.00 salary and 4000.00 taxDue is assigned to Player 2> | P |

**CLASS:** HouseCard

| METHOD | # | DESCRIPTION | SAMPLE INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| assignDescriptions() | | mainID = 0<br>(Split-Level) | N/A | houseType =<br>"Split-Level"<br><br>price = 40000.00 | houseType = "Split-Level"<br><br>price = 40000.00 | P |
| | | mainID = 4 (Dutch Colonial) | N/A | houseType = "Dutch Colonial"<br><br>price = 120000.00 | houseType = "Dutch Colonial"<br><br>price = 120000.00 | P |
| | | mainID = 6<br>(Farmhouse) | N/A | houseType =<br>"Farmhouse"<br><br>price = 160000.00 | houseType = "Farmhouse"<br><br>price = 160000.00 | P |

**CLASS:** GreenSpaces

| METHOD | # | DESCRIPTION | SAMPLE INPUT / STATUS | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| doAction(Player player, int spaceID) | 1 | Player lands on a Pay Day space | Current player salary is 10000.00 | Current player money balance raise +10000.00 | Current player money balance raise +10000.00 | P |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Player lands on a Pay Raise space | Current player salary is 10000.00 | Raise salary by +20000.00, current player money raise +30000.00 | Raise salary by +20000.00, current player money raise +30000.00 | P |
| | | Player lands on a pay raise space, current number pay raises is equal to maximum number of pay raises | Pay Raises: 5/5 | Ignore event. | Ignore event. | P |

**CLASS:** MagentaSpaces

| METHOD | # | DESCRIPTION | SAMPLE INPUT / STATUS | EXPECTED OUTPUT | ACTUAL OUTPUT | P/F |
|---|---|---|---|---|---|---|
| getMarried(Player[] players, int turn) | 1 | Current player is not married, number generated is even, number of players in game is 3 | N/A | Current player is set married status to true, collect 10000.00 from the two other players, return the randomly generated number | Current player is set married status to true, collect 10000.00 from the two other players, return the randomly generated number | P |
| | | Current player is not married, number generated is odd, number of players in game is 2 | N/A | Current player is set married status to true, collect 5000.00 from one other player, return the randomly generated number | Current player is set married status to true, collect 5000.00 from one other player, return the randomly generated number | P |
| | | Current player is already married | N/A | Ignore event | Ignore event | P |
| haveBaby(Player[] players, int turn) | **2** | Number generated is odd | N/A | Assign one baby to current player | Assign one baby to current player | P |
| | | Number generated is even | N/A | Assign twin to current player | Assign twin to current player | P |
| | | Current player gets twin, number of players is 3 | N/A | Current player collects 5000.00 * 2 from the two other players | Current player collects 5000.00 * 2 from the two other players | P |
| | | Current player is unmarried | N/A | Ignore event | Ignore event | P |