|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **#** | **Test Description** | **Sample test case** | **Expected result** | **Actual Result** | **P/F** |
| Action | 1 | Card is type ColB | Balance = 100000  Amount = 20000 | Balance = 120000 | Balance = 120000 | P |
|  | 2 | Card is type PayB | Balance = 100000  Amount = 20000 | Balance = 80000 | Balance = 80000 | P |
|  | 3 | Card is type ColP, P1 collects 20000 from P2 | P1.Balance = 100000  P2.Balance = 100000  Amount = 20000 | P1.Balance = 120000  P2.Balance = 80000 | P1.Balance = 120000  P2.Balance = 80000 | P |
|  | 4 | Card is type PayP, P1 pays 20000 to P2 | P1.Balance = 100000  P2.Balance = 100000  Amount = 20000 | P1.Balance = 80000  P2.Balance = 120000 | P1.Balance = 80000  P2.Balance = 120000 | P |
|  | 5 | Card is type ColP, the game have 3 players, the card allows the player to collect from all | P1.Balance = 100000  P2.Balance = 100000  P3.Balance = 100000  Amount = 5000 | P1.Balance = 110000  P2.Balance = 95000  P3.Balance = 95000 | P1.Balance = 110000  P2.Balance = 95000  P3.Balance = 95000 | P |
|  | 6 | Card is type PayP, the game have 3 players, the card makes the player to pay from all | P1.Balance = 100000  P2.Balance = 100000  P3.Balance = 100000  Amount = 5000 | P1.Balance = 90000  P2.Balance = 105000  P3.Balance = 105000 | P1.Balance = 90000  P2.Balance = 105000  P3.Balance = 105000 | P |
| ActionCard | 1 | Generating the Deck |  | Actions.get(0) != null | Actions.get(0) != null | P |
|  | 2 | Get top card and execute action | Actions.get(0) = Buy an Item  Actions.get(1) = Hiking  P1.Balance = 100000  Amount = 20000 | P1.Balance = 80000  Actions.get(0) = Hiking  Actions.get([lastIndex]) = Buy an Item | P1.Balance = 80000  Actions.get(0) = Hiking  Actions.get([lastIndex]) = Buy an Item | P |
| BlueCard | 1 | P1 gets a blueCard that matches his own career | P1.Balance = 100000 | P1.Balance = 115000 | P1.Balance = 115000 | P |
|  | 2 | P1 gets a blueCard that does not match the careers of any Player | P1.Balance = 100000  Amount = 10000 | P1.Balance = 90000 | P1.Balance = 90000 | P |
|  | 3 | P1 gets a blueCard that matches the career of P2 | P1.Balance = 100000  P2.Balance = 100000  Amount = 25000 | P1.Balance = 75000  P2.Balance = 125000 | P1.Balance = 75000  P2.Balance = 125000 | P |
| BlueDeck | 1 | Generating the Deck |  | Blues.get(0) != null | Blues.get(0) != null | P |
|  | 2 | Get top card and execute action | blues.get(0) = Buy an Item  blues.get(1) = Hiking  P1.Balance = 100000  Amount = 20000 | P1.Balance = 80000  blues.get(0) = Hiking  blues.get([lastIndex]) = Buy an Item | P1.Balance = 80000  blues.get(0) = Hiking  blues.get([lastIndex]) = Buy an Item | P |
| CareerDeck | 1 | Generating the Deck |  | Careers.get(0) != null | Careers.get(0) != null | P |
|  | 2 | Player without degree gets top card | Careers.get(0) = Lawyer  Career.get(1) = Doctor  Career.get(2) = Server  Career.get(3) = Athlete | Careers.get(0) = Lawyer  Career.get(1) = Doctor  Career.get(2) = Athlete  P1.Career = Server | Careers.get(0) = Lawyer  Career.get(1) = Doctor  Career.get(2) = Athlete  P1.Career = Server | P |
|  | 3 | Player with degree gets top card | Careers.get(0) = Lawyer  Career.get(1) = Doctor  Career.get(2) = Server  Career.get(3) = Athlete | Careers.get(0) = Doctor  Career.get(1) = Server  Career.get(2) = Athlete  P1.Career = Lawyer | Careers.get(0) = Doctor  Career.get(1) = Server  Career.get(2) = Athlete  P1.Career = Lawyer | P |
|  | 4 | Player with degree choose second career when he is in College Career Choice | Careers.get(0) = Lawyer  Career.get(1) = Doctor  Career.get(2) = Server  Career.get(3) = Athlete | Careers.get(0) = Server  Career.get(1) = Athlete  Career.get([lastIndex]) = Lawyer  P1.Career = Doctor | Careers.get(0) = Server  Career.get(1) = Athlete  Career.get([lastIndex]) = Lawyer  P1.Career = Doctor | P |
|  | 5 | Player steps on job Change and choose his current career | Career.get(0) = Server  Career.get(1) = Doctor  P1.Career = Athlete | Career.get(0) = Doctor  Career.get([lastIndex]) = Server  P1.Career = Athlete | Career.get(0) = Doctor  Career.get([lastIndex]) = Server  P1.Career = Athlete | P |
|  | 6 | Player steps on job Change and choose the new career | Career.get(0) = Server  Career.get(1) = Doctor  P1.Career = Athlete | Career.get(0) = Doctor  Career.get([lastIndex]) = Athlete  P1.Career = Server | Career.get(0) = Doctor  Career.get([lastIndex]) = Athlete  P1.Career = Server | P |
| GameResource | 1 | Generate Players |  | Player.get(0) != null | Player.get(0) != null | P |
|  | 2 | Generate ActionCards |  | Actions.get(0) != null | Actions.get(0) != null | P |
|  | 3 | Generate BlueDeck |  | Blues.get(0) != null | Blues.get(0) != null | P |
|  | 4 | Generate CareerDeck |  | Careers.get(0) != null | Careers.get(0) != null | P |
|  | 5 | Generate SalaryDeck |  | Salaries.get(0) != null | Salaries.get(0) != null | P |
|  | 6 | Generate HouseDeck |  | Houses.get(0) != null | Houses.get(0) != null | P |
|  | 7 | Generate board |  | startingCollegePath != null  startingCareerPath != null | startingCollegePath != null  startingCareerPath != null | P |
|  | 8 | Get current player on a 2 player game with 1 of them already retired | playerIndex = 0  players.get(0) = P1  players.get(1) = null  retired.get(0) = P2 | currPlayer = P1 | currPlayer = P1 | P |
|  | 9 | Get other players | Players.get(0) = P1  Players.get(1) = P2  Players.get(2) = P3  currPlayer = P1 | otherPlayers.get(0) = P2  otherPlayers.get(1) = P3  currPlayer = P1 | otherPlayers.get(0) = P2  otherPlayers.get(1) = P3  currPlayer = P1 | P |
|  | 10 | Retire the first player on a 3 player game | Players.get(0) = P1  Players.get(1) = P2  Players.get(2) = P3  currPlayer = P1 | Retired.get(0) = P1  Players.get(0) = P2  Players.get(1) = P3 | Retired.get(0) = P1  Players.get(0) = P2  Players.get(1) = P3 | P |
|  | 11 | Retire the last player on a 3 player game | Retired.get(0) = P1  Retired.get(1) = P2  Players.get(0) = P3  CurrPlayer = P3 | Retired.get(0) = P1  Retired.get(1) = P2  retired.get(2) = P3 | Retired.get(0) = P1  Retired.get(1) = P2  Players.get(2) = P3 | P |
| GreenSpace | 1 | P1 step on a Pay Day Space | P1.Salary = 20000  P1.Balance = 100000 | P1.Balance = 120000 | P1.Balance = 120000 | P |
|  | 2 | P1 step on a Pay Raise Space | P1.Salary = 20000  P1.Balance = 100000 | P1.Salary = 30000  P1.Balance = 100000 | P1.Salary = 30000  P1.Balance = 100000 | P |
|  | 3 | P1 step on a Pat Raise Space but he already have the maximum amount of pay raises | P1.Salary = 20000  P1.Balance = 100000  P1.raiseCount = 1  P1.Career.raisese = 1 | P1.Salary = 20000  P1.Balance = 100000 | P1.Salary = 20000  P1.Balance = 100000 | P |
| HouseDeck | 1 | Generate the House Deck |  | Houses.get(0) != null | Houses.get(0) != null | P |
| MagentaSpace | 1 | Marry a Player | P1.status = Single | P1.status = Married | P1.status = Married | P |
| Path | 1 | Generate a Path without next path |  | Path.getPath1 == null | Path.getPath1 == null | P |
|  | 2 | Generate a Path with 1 next path |  | Path.getPath1 != null  Path.getPath2 == null | Path.getPath1 != null  Path.getPath2 == null | P |
|  | 3 | Generate a Path with 2 next paths |  | Path.getPath1 != null  Path.getPath2 != null | Path.getPath1 != null  Path.getPath2 != null | P |
| Player | 1 | P1 pays 20000 | P1.Balance = 100000  Amount = 20000 | P1.Balance = 80000 | P1.Balance = 80000 | P |
|  | 2 | P1 receives 20000 | P1.Balance = 100000  Amount = 20000 | P1.Balance = 120000 | P1.Balance = 120000 | P |
|  | 3 | P1 pays P2 20000 | P1.Balance = 100000  P2.Balance = 100000  Amount = 20000 | P1.Balance = 80000  P2.Balance = 120000 | P1.Balance = 80000  P2.Balance = 120000 | P |
|  | 4 | P1 buys a house | P1.Balance = 500000  House.price = 300000  House.Name = Apartment | P1.Balance = 200000  P1.house = Apartment | P1.Balance = 200000  P1.house = Apartment | P |
|  | 5 | P1 sells a house | P1.Balance = 200000  P1.House.worth = 500000  P1.House = Apartment | P1.Balance = 700000  P1.House != null | P1.Balance = 700000  P1.House != null | P |
|  | 6 | P1 gets married | P1.status = Single | P1.status = married | P1.status = married | P |
|  | 7 | P1 pays all his loans | P1.Balance = 100000  P1.Loans = 1 | P1.Balance = 75000  P1.Loans = 0 | P1.Balance = 75000  P1.Lonas = 0 | P |
|  | 8 | P1 changes careers and therefore resets his raises | P1.raiseCount = 2 | P1.raiseCount = 0 | P1.raiseCount = 0 | P |
|  | 9 | P1 graduates | P1.hasDegree = false | P1.hasDegree = true | P1.hasDegree = true | P |
|  | 10 | P1 gets a loan | P1.Balance = 10000  Amount = 15000  P1.Loans = 0 | P1.Balance = 15000  P1.Loans = 1 | P1.Balance = 15000  P1.Loans = 1 | P |
|  | 11 | P1 gets a child on a 2 player game | P1.Balance = 100000  P1.child = 0  P2.Balance = 100000 | P1.Balance = 105000  P1.child = 1  P2.Balance = 95000 | P1.Balance = 105000  P1.child = 1  P2.Balance = 95000 | P |
|  | 12 | P1 sells his child on retirement | P1.Balance = 105000  P1.child = 1 | P1.Balance = 115000 | P1.Balance = 115000 | P |
|  | 13 | Player moves | P1.getSpace = 1  P1.moves = 2 | P1.getSpace = 3 | P1.getSpace = 3 | P |
|  | 14 | P1 retires first and P2 second and P3 last | P1.Balance = 100000  P2.Balance = 100000  P3.Balance = 100000 | P1.Balance = 200000  P2.Balance = 150000  P3.Balance = 120000 | P1.Balance = 200000  P2.Balance = 150000  P3.Balance = 120000 | P |
| Salary | 1 | Increase salary | Salary.amount = 20000  Salary.tax = 2000 | Salary.amount = 30000  Salary.tax = 3000 | Salary.amount = 30000  Salary.tax = 3000 | P |
| SalaryDeck | 1 | Generate Salary Deck |  | Salaries.get(0) != null | Salaries.get(0) != null | P |
|  | 2 | Get top card | Salary.get(0).amount = 20000  Salary.get(1).amount = 30000 | Salary.get(0).amount = 30000 | Salary.get(0).amount = 30000 | P |

Instructions to run:

1. Javac contents of model, javac contents of controller, and GUI
2. Java GUI