Black Jack Game using Pycharm

Rajesh

September 2019

1 Objective

The main objective of this report is to code a game of black jack for any number of players

1.1 Functions for each decision

1.1.1 Hit

To append an extra card to the corresponding hand from dictionary named deck

1.1.2 BlackJack

To check the condition for black jack

1.1.3 printResult

To print the cards in and score of hand

1.1.4 start

To start game with asking number of players and name and best on each hand

1.2 Simplified version without bet

The next sub objective was to code a simple blackjack for multiplayer without exceptions of split and bet and double bet.

1.3 Actual Black Jack game with all conditions

Adding features of Bet on hand , push ups, double bet , split etc

2 Methodology

Firstly took number of players from user and saving it in a list of names,and appending each hand for a player in another list. Then called the function to initialize the game with each hand of player with two cards including dealer.

Then start calls function game which iterates over each hand of each player. Then it first checks if there is black jack if not then the player is given option for double bet or hit or stand, if player selects double the bet on hand is doubled and all playerMoney list is modified as well as the bet and a additional card is added and no further option is available, if it selects hit and additional card is added and again three options are given, if it selects stand it comes out of loop immediately.

If any where in loop total score is greater then 21 the loop breaks and bet on hand is lost ,that is,it is deducted from corresponding lists. if the hand is valid after en of loop then if total of dealer and and hand is compared and the bet is added or deducted accordingly.

If anywhere in loop hand has identical card then user is asked for split and if it's an ace then flag is put on and only one additional card is added and then hands are taken care individually.

3 Results

Successfully achieved all exceptions and the card is selected at random.

4 Discussion

The code efficiency may be optimized.There may be functions for split,double bet, checking decision can be made more clear.Using OOP can make things simpler and understandable.Adding graphics would make it more interesting and user friendly.

5 Conclusion

The code runs quite satisfactorily.Implemented many decision handling dependent on each other.

References

[1]https://gist.github.com/mjhea0/5680216
[2]https://towardsdatascience.com/python-blackjack-simulator-61b591ffb971