

Poker Simulator Demonstration Notebook

By: Shrey Ghai



Introduction

In this demo notebook, I will demonstrate the different aspects of my poker simulator. You will find the key functionalities of the tool I built below. You will see how my code has replicated a game of poker in user friendly way. I built this so I could teach someone what is happening at each step of the game. This allows a user to do scenario analysis and calculate probabilities to better learn the game.

In case you are unfamiliar with Texas hold 'em poker:

Texas hold 'em poker is a variance of the card game poker. Each player is dealt two cards face down and then five community cards are dealt face up. The community cards come out in stages in a series of three cards known as the flop, an additional single card known as the turn, and the final card known as the river. The goal of a Texas hold'em game is to use both, one or none of your hole cards in combination with the community cards to make the best possible five-card poker hand.

If you want to learn more, click here: <https://www.cardplayer.com/rules-of-poker/how-to-play-poker/games/texas-holdem> (<https://www.cardplayer.com/rules-of-poker/how-to-play-poker/games/texas-holdem>)

The Simulator

In the following code execution, you can see that I am simulating a game of Texas Hold'em Poker with myself and three of my friends from Scranton. `test_game` is the name of the first game I will create. The simulator is capable of running as many players as the user chooses.

```
In [60]: players = ['Shrey', 'Jim', 'Pam', 'Dwight',]  
test_game = Game(players)
```

In the following block we will simulate dealing a hand of poker. The method `deal()` deals each player two cards from a digital deck of shuffled cards.

```
In [61]: test_game.deal()  
  
Shrey:[3♣, K♣]  
Jim:[4♣, A♦]  
Pam:[8♠, 2♥]  
Dwight:[5♥, 7♥]
```

The method `current_winner()` was implemented in the program to determine who is the current winner based on the cards dealt and what poker hand ranking each player has. This method can be called at any time so users can learn and see how the winner changes throughout the course of the game.

If you want to learn more about the rankings, click here: <https://www.cardplayer.com/rules-of-poker/hand-rankings> (<https://www.cardplayer.com/rules-of-poker/hand-rankings>)

```
In [62]: current_winner(test_game)  
  
Jim is the current winner  
Jim has Ace high
```

In addition to `current_winner()`, users can utilize `determine_hand()` which informs the user what hand each player has. We will call this method to determine Pam's current hand.

```
In [63]: determine_hand('Pam', test_game)  
  
Pam has Eight high
```

The flop will deal three community cards on the table

```
In [64]: test_game.flop()  
  
[9♦, K♦, 4♥]
```

Notice how the current winner changes after the flop has been dealt

```
In [65]: current_winner(test_game)  
  
Shrey is the current winner  
Shrey has a Pair of Kings with Nine high
```

The method `determine_probability()` returns the percentage of a players odds of winning the specific hand. Similar to the other methods, `determine_probability()` can be called during any part of the game so users can see how their odds increase or decrease as the game progresses. The computer will run the exact number of simulations the user chooses to input into the method.

In the cell below, I will run 1,000 simulations of the current game. The output will tell you the probability of each person winning.

```
In [66]: determine_probability(test_game, number_simulations = 1000, print_sim = False)
Out[66]: {'Shrey': '88%', 'Jim': '12%', 'Pam': '0%', 'Dwight': '0%', 'split': '0%'}
```

This method has the functionality to print scenarios of the simulator as well.

In the cell below, I will run 10 simulations of the current game. The output will tell you the probability of each person winning along with text including the winner of each simulation and what poker hand they won with.

```
In [67]: determine_probability(test_game, number_simulations = 10, print_sim = True)
```

Simulation 1

[9♦, K♦, 4♥, 10♥, Q♥]

Dwight is the current winner

Dwight has a Queen high Flush

Simulation 2

[9♦, K♦, 4♥, 4♠, 3♥]

Jim is the current winner

Jim has a Three of a Kind of Fours with Ace high

Simulation 3

[9♦, K♦, 4♥, 10♠, 2♠]

Shrey is the current winner

Shrey has a Pair of Kings with Ten high

Simulation 4

[9♦, K♦, 4♥, 10♥, 10♠]

Shrey is the current winner

Shrey has a Two Pair with a pair of Kings and pair of Tens with Nine high

Simulation 5

[9♦, K♦, 4♥, K♠, J♦]

Shrey is the current winner

Shrey has a Three of a Kind of Kings with Jack high

Simulation 6

[9♦, K♦, 4♥, 5♦, 2♦]

Jim is the current winner

Jim has a Ace high Flush

Simulation 7

[9♦, K♦, 4♥, K♥, Q♦]

Shrey is the current winner

Shrey has a Three of a Kind of Kings with Queen high

Simulation 8

[9♦, K♦, 4♥, 4♦, A♠]

Jim is the current winner

Jim has a Full House with Fours full of Aces

Simulation 9

[9♦, K♦, 4♥, Q♠, J♥]

Shrey is the current winner

Shrey has a Pair of Kings with Queen high

Simulation 10

[9♦, K♦, 4♥, 10♠, A♥]

Jim is the current winner

Jim has a Two Pair with a pair of Aces and pair of Fours with King high

```
Out[67]: {'Shrey': '50%', 'Jim': '40%', 'Pam': '0%', 'Dwight': '10%', 'split': '0%'}
```

```
In [68]: test_game.turn()
```

```
[9♦, K♦, 4♥, 10♥]
```

```
In [69]: test_game.river()
```

```
[9♦, K♦, 4♥, 10♥, Q♥]
```

```
In [70]: current_winner(test_game)
```

```
Dwight is the current winner  
Dwight has a Queen high Flush
```

If you are interested in learning more:

Check out a video where I provide a voiceover of the simulator: <https://youtu.be/O71kB6guwNk>
(<https://youtu.be/O71kB6guwNk>)

Check out the code and contribute on Github:

https://github.com/shreyghai/poker_simulator/blob/master/poker_simulator.py
(https://github.com/shreyghai/poker_simulator/blob/master/poker_simulator.py)