
texasholdem

mark truett

Sep 08, 2021

CONTENTS:

1	Operations	1
1.1	Purpose:	1
1.2	challenges and successes:	1
1.3	Link	1
1.4	functions used in operations	1
2	PlayHoldEm	3
2.1	PlayHoldEm module	3
2.2	RankTheHand module	4
2.3	validator module	4
3	Indices and tables	5
	Python Module Index	7
	Index	9

OPERATIONS

1.1 Purpose:

Texas holdem is a 1 vs 1-8 computer player game. the game randomly generates your cards. you're able to bet \$1 - \$1000. at the end of the game all cards are shown and a winner is returned

1.2 challenges and successes:

I was successful with creating my deck and creating my players in a dictionary I was able to accomplish that fairly quickly and felt confident in using functions to pass the data throughout my program.

it was very challenging to use the lambdas and combination to check the different combinations against the functions in RankTheHand.py.

Unfortunately, I could not figure out how to get my program to pick a winner.

1.3 Link

[This is a link to gitlab](https://git.cybbh.space/170D/wobc/student-folders/21_002/truett/texasholdem/-/blob/master/)

1.4 functions used in operations

functions used in texasholdem

gen_player_hand ()

generate_the_flop()

generate_a_card()

generate_deck()

create_players()

handle_a_bet()

the_hand_is_a()

return_all_players__best_hand()

main()

PLAYHOLDEM

2.1 PlayHoldEm module

PlayHoldEm Game.

Texas holdem is a 1 vs 1-8 computer player game. the game randomly generates your cards. you're able to bet 1 - 1000. at the end of the game all cards are shown and a winner is returned

PlayHoldEm.**create_players**(*the_deck, num_players*)
create_players.

a dictionary of players are created and given 2 cards, \$1000. the amount of players is input by the user.

PlayHoldEm.**gen_player_hand**(*the_deck, number_of_cards*)
Gen_player_hand.

Generates Players Hand the number of cards are how many cards are to be returned those cards are removed to only be used once.

PlayHoldEm.**generate_a_card**(*the_deck*)
generate_a_card.

Generates 1 card. the 2nd and third round.

PlayHoldEm.**generate_deck**()
generate_deck.

This function creates the deck

PlayHoldEm.**generate_the_flop**(*the_deck*)
generate_the_flop.

Generates 3 cards

These are the first three community cards

PlayHoldEm.**handle_a_bet**(*a_player, amount_to_bet, the_pot*)
handle_a_bet.

Player 1s bet is handled here and counted in the pot

PlayHoldEm.**handle_other_bets**(*a_player, amount_bet, the_pot*)
handle_other_bets.

This function brings amount_bet from player 1 and applied that to the rest of the players

PlayHoldEm.**main**()

`PlayHoldEm.return_all_players_best_hand(player_list)`
`return_all_players_best_hand.`

we iterate through each players best hand. each players hand is ran through the players best hand and the highest value is returned to pick_a_winner

`PlayHoldEm.the_hand_is_a(player_hand)`
`the_hand_is_a.`

the hand is uses the rankthehand.py file to rank each hand

`PlayHoldEm.the_players_best_hand(player_hand)`
`the_players_best_hand.`

We take all possible combinations of the players hand and run it against rankthehand.py. we then sort the tuple

2.2 RankTheHand module

`RankTheHand.cards_all_same_suit(a_hand)`

`RankTheHand.is_flush(a_hand)`

`RankTheHand.is_four_kind(a_hand)`

`RankTheHand.is_full_house(a_hand)`

`RankTheHand.is_pair(a_hand)`

`RankTheHand.is_straight(a_hand)`

`RankTheHand.is_straight_flush(a_hand)`

`RankTheHand.is_three_kind(a_hand)`

`RankTheHand.is_two_pairs(a_hand)`

`RankTheHand.return_cards_rank_as_straight(a_hand)`

`RankTheHand.return_most_frequent_and_num_counter_elements(a_hand)`

2.3 validator module

Module that validates user input Has various functions that check integer, float, is number in range, is entry contained in

Functions continue to prompt until input is valid

`validator.enter_a_float(prompt)`

`validator.enter_an_integer(prompt)`

`validator.enter_correct_type(prompt, type_function)`

`validator.enter_float_in_range(prompt, low, high)`

`validator.enter_integer_in_range(prompt, low, high)`

`validator.enter_number_in_range(prompt, low, high, entry_function)`

`validator.enter_valid_character(prompt, set_of_valid_chars: str, ignore_case=True)`

`validator.main()`

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

p

PlayHoldEm, 3

r

RankTheHand, 4

v

validator, 4

INDEX

C

`cards_all_same_suit()` (in module *RankTheHand*), 4
`create_players()` (in module *PlayHoldEm*), 3

E

`enter_a_float()` (in module *validator*), 4
`enter_an_integer()` (in module *validator*), 4
`enter_correct_type()` (in module *validator*), 4
`enter_float_in_range()` (in module *validator*), 4
`enter_integer_in_range()` (in module *validator*), 4
`enter_number_in_range()` (in module *validator*), 4
`enter_valid_character()` (in module *validator*), 4

G

`gen_player_hand()` (in module *PlayHoldEm*), 3
`generate_a_card()` (in module *PlayHoldEm*), 3
`generate_deck()` (in module *PlayHoldEm*), 3
`generate_the_flop()` (in module *PlayHoldEm*), 3

H

`handle_a_bet()` (in module *PlayHoldEm*), 3
`handle_other_bets()` (in module *PlayHoldEm*), 3

I

`is_flush()` (in module *RankTheHand*), 4
`is_four_kind()` (in module *RankTheHand*), 4
`is_full_house()` (in module *RankTheHand*), 4
`is_pair()` (in module *RankTheHand*), 4
`is_straight()` (in module *RankTheHand*), 4
`is_straight_flush()` (in module *RankTheHand*), 4
`is_three_kind()` (in module *RankTheHand*), 4
`is_two_pairs()` (in module *RankTheHand*), 4

M

`main()` (in module *PlayHoldEm*), 3
`main()` (in module *validator*), 4
module
 PlayHoldEm, 3
 RankTheHand, 4
 validator, 4

P

PlayHoldEm
 module, 3

R

RankTheHand
 module, 4
`return_all_players_best_hand()` (in module *PlayHoldEm*), 3
`return_cards_rank_as_straight()` (in module *RankTheHand*), 4
`return_most_frequent_and_num_counter_elements()` (in module *RankTheHand*), 4

T

`the_hand_is_a()` (in module *PlayHoldEm*), 4
`the_players_best_hand()` (in module *PlayHoldEm*), 4

V

validator
 module, 4