
README

Black Jack

Allan Tsay, Kristine Sun

3038959471, 3038046663

Project TA: Harry Qian

Intro (optional)

Traditional rules of the card game Black Jack.

Project Approval

Project was approved by Harry Qian. No changes to the original proposal were necessary.

Instructions

Users should first run the file. Next, a prompt should appear showing the user's hand and the first card of the dealer's hand. It then will ask whether or not the user wants to hit or stay. The prompt will keep appearing until the user chooses to stay. After choosing to stay, the program will reveal the dealer's hand and tell the user if they won, lost, or tied. Lastly, the program will ask if the user wants to play again. The game will run again if yes.

Features

- *A list of the **2 - 3 features** that you listed in your project proposal. If this has changed, only discuss the most up-to-date features*
 1. The program has a working deck of cards.
 2. There is a win condition of being as close to 21 as possible without going over the number as well as forcing the dealer to hit if it is a soft 16 (if the two cards they have add up to 16 or less).
 3. The user is able to restart the game if they want.

Justification for Complexity

- *A brief justification for why the project should receive full credit for complexity:*
 - *How did your features contribute to complexity?*
 - *The program uses F strings to show the current hands of each player.*
 - *Did you use trickier concepts that we taught in class and what were they? How did you implement them?*
 - *It uses object oriented programming in order to reset the game when you want to replay it.*
 - *Other points that show your project has complexity*
 - *When calculating the total of a hand, the program considers whether an ace card should be worth 1 or 11.*
 - *When drawing cards from the deck, it removes the card from the original deck list and adds it into your hand list.*

Lists & Script Variables

- *Description of the lists and script variables in your project, and how you utilized them (what makes the lists non-trivial, etc.).*
 - *The deck is used to store all the cards*
 - *The player hand stores the user's hand*
 - *The deal hand stores the hand of the dealer (computer)*
 - *Playerin which is set to true and becomes false when the player chooses to stay and end your turn.*
 - *Player total which is the total of the user's hand which is used to compare to the dealer's total to see who wins.*

Function Table

Each row will describe the functionality of one custom block/function, with the relevant information placed in the relevant columns (as defined above). You must include a separate row for EVERY custom block/function you create

Block / Function Name	Domain (inputs)	Range (outputs)	Behavior (role in the context of the project)
deal	lists	lists	Deals cards into player's and dealer's hand
total	list	integer	Adds up the value of the player's and dealer's hand
print_hands	nothing	string	Displays one card of the dealer's hand and player's hand and total
play_game	nothing	string	This function contains most of the main game play allowing the user to hit or stand and decides winner
main	nothing	nothing	Resets the game state and allows user to stop the game

Video

<https://www.youtube.com/watch?v=qR6659SSkyU>