

# Black Jack

There are three versions of the black jack card game included in this project.

1) Single Player Text Based:

Can be run using the command: **python3 single\_player\_textbased.py**

2) Multiplayer Text Based:

Can be run using the command: **python3 multiplayer.py**

3) Single Player GUI:

Can be run using the command: **python3 single\_player\_gui.py**

## Design and Algorithms

The primary data structure I used is the Python list. I used a list to keep track of the cards in the deck, as well as the cards in the hand of all players and the dealer. On each round when a player (or dealer) pulls a card out of the deck - a random card is pulled from the deck list. The deck is randomized at the start of the game.

1. The game works by initializing a full deck of 52 cards which are randomized.
2. On each round, the player is able to either 'hit' for a card or 'stay' their turn.
3. If a 'hit' is chosen the first card from the deck (a list) is pulled and the score is added to the player's score. Player's scores are checked to ensure they have not busted - with either the game continuing or ending.
4. Once a card is drawn from the deck it is deleted from the list.
5. The game continues until either dealer or player(s) reach 23 points.

## GUI

I chose to use the Python Tkinter library which enabled me to create a graphical user interface to create an interactive game environment for the user.

Tkinter enabled me to create a master window and add table and button frames to display their respective information.

Tkinter uses a grid based format for creating a layout - and I was able to use this to create the table for cards, as well as buttons for user interaction.

This was my first experience creating a GUI using Tkinter and I enjoyed being able to create a satisfactory visual representation for my game of black jack.

Created by:

Omair Nasri

[omairnasri@gmail.com](mailto:omairnasri@gmail.com) / [omairnasri@berkeley.edu](mailto:omairnasri@berkeley.edu)

(510) 876 - 1645