**Project-03: Blackjack (21) Game with Deck of Cards**

**Objective:**

Create a simplified version of the card game Blackjack where the player competes against the computer (dealer) to get as close to 21 as possible without going over. The game should use a deck of cards represented by numerical values, and both the player and dealer must follow standard Blackjack rules.

**Additional Information:**

**Game Rules**

1. The goal is to have a hand value as close to 21 as possible without exceeding it.
2. Each card has a point value:
   * Number cards (2–10) are worth their face value.
   * Face cards (Jack, Queen, King) are worth 10 points.
   * Aces can be worth either 1 or 11 points, depending on which benefits the hand most.
3. The player competes against the dealer (computer), and the dealer follows a basic set of rules:
   * The dealer must hit (draw more cards) until their hand totals at least 17 points.

**Part 1: Deck of Cards Representation**

Use an array (or list) to represent a full deck of 52 cards, with each card being repre sented by its numerical value. You don’t need to represent suits, as this game focuses on the card values for calculating scores.

Here is an example of how to represent the deck:

python

Copy code

# Array for a deck of cards with numerical values

deck\_of\_cards = [

# Hearts

2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11, # Jack, Queen, King = 10, Ace = 11

# Diamonds

2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11,

# Clubs

2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11,

# Spades

2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11

]

# Shuffle the deck at the start of each game:

import random

random.shuffle(deck\_of\_cards)

In this deck:

* **2–10** are represented by their number value.
* **Face cards** (Jack, Queen, King) are represented by 10.
* **Ace** is represented by 11, but in gameplay, you will handle it as either 1 or 11 based on the player's total.

**Part 2: Game Mechanics**

**1. Player's Turn:**

* The player starts by being dealt two cards from the shuffled deck.
* The player can choose to:
  + **Hit** (draw another card).
  + **Stand** (keep their current total).
* If the player's total exceeds 21, they "bust" and lose immediately.
* The player can continue to hit until they are satisfied with their total or until they bust.

**2. Dealer's Turn:**

* The dealer (computer) is also dealt two cards at the start.
* The dealer will continue drawing cards (hit) until their total is at least 17. Once they reach 17 or more, they must stand.
* If the dealer's total exceeds 21, they bust, and the player wins.

**3. Determine the Winner:**

* Once both the player and dealer have finished their turns, their totals are compared:
  + If the player’s total is closer to 21 without exceeding it, the player wins.
  + If the dealer’s total is closer to 21, the dealer wins.
  + If both totals are the same, the game ends in a tie (push).

**4. Ace Handling:**

* Aces can be worth either 1 or 11. If counting an Ace as 11 would make the hand’s total exceed 21, count it as 1 instead.

**Recommended Programming Language:**

* Python

**Recommended by:**

* [SaladStik](https://github.com/SaladStik)

**Requirements:**

1. **Deck of Cards:**
   * Implement an array representing a full deck of cards using the provided numerical values (2–10, face cards as 10, and Aces as 11).
   * Shuffle the deck at the start of each game.
2. **Player and Dealer Logic:**
   * Implement the player’s ability to hit or stand.
   * Implement the dealer’s automatic logic to hit until their total is at least 17.
   * Ensure that the game handles busts (when a total exceeds 21) and Ace logic (Aces can be 1 or 11).
3. **Game Flow:**
   * The game should be interactive, allowing the player to input their choices.
   * After each round, display the winner and ask if the player wants to play again.
   * Optionally, display a running total of wins/losses if multiple rounds are played.

**Program/Menu Example:**

* **Input:** Prompts the user for the numbers and the operator needed for the calculation.
* **Output:** Show the appropriate result.
* **Menu UI:**

Welcome to Blackjack!

The goal is to get as close to 21 as possible without going over.

Your hand: [8, 5] (Total: 13)

Do you want to hit or stand? hit

You drew a 7.

Your hand: [8, 5, 7] (Total: 20)

Do you want to hit or stand? stand

Dealer's hand: [10, 6] (Total: 16)

Dealer hits and draws a 5.

Dealer's hand: [10, 6, 5] (Total: 21)

Dealer wins!

Do you want to play again? (yes/no): yes

**Concepts Covered:**

**Basic Game Logic** – Implementing a turn-based game with a clear win/loss condition.

**Data Structures** – Using lists to represent a deck of cards.

**Randomization** – Shuffling the deck using random.shuffle().

**Conditional Statements** – Handling game decisions like hitting, standing, and checking for busts.

**Looping** – Using loops to manage turns for the player and dealer.

**User Input Handling** – Allowing the player to interact by choosing to hit or stand.

**Functions** – (If implemented) Organizing code into functions for reusability.

**Handling Edge Cases** – Managing Ace values dynamically as 1 or 11 based on the hand's total.

**Hints to Get Started:**

* Use the random.shuffle() function to shuffle the deck at the start of each game.
* Use the pop() method to draw cards from the deck.
* Remember to handle Aces carefully: they should be counted as 11 unless that would make the hand exceed 21, in which case they should be counted as 1

**Bonus Challenges (Optional):**

* **Tracking Wins/Losses:**
  + Keep track of how many rounds the player has won, lost, or tied. Display these statistics at the end of each game or when the player decides to quit.
* **Betting System:**
  + Add a simple betting system where the player can place a bet at the beginning of each round. If they win, they double their bet. If they lose, they lose the bet.
  + Track the player’s total balance over multiple rounds.

**Expected Outcome:**

By the end of this assignment, participants should be able to:

* Form Interactive Gameplay
* Follow Correct Game Flow
* Proper Handling of Game Rules
* Randomized Deck & Fair Play
* Game Continuation