Final Project: Blackjack

# Game Rules (Modified):

Blackjack, also known as twenty-one, is the most widely played casino banking game in the world. Blackjack is a comparing card game between a player and dealer, meaning players compete against the dealer but not against other players. It is played with one or more decks of 52 cards. The objective of the game is to beat the dealer in one of the following ways:

* Get 21 points on the player’s first two cards (called a “blackjack”), without a dealer blackjack.
* Reach a final score higher than the dealer without exceeding 21.
* Let the dealer draw additional cards until their hand exceeds 21.

**Note:**

* Suite (Spade, Heart, Club, and Diamond) is insignificant in Blackjack.
* All face cards “J”, “Q”, “K” are 10 points. Cards “2” through “10” are of their own face values.
* “A” can be 1 or 11 to make the optimized hand.
* Player vs Dealer 1-on-1
* Dealer hits below 17, and stands on both soft and hard 17s (Soft 17 Ex: [A] [6]. Hard 17 Ex: [10] [7]) or above.
* No bonus payout for blackjack.
* No surrender, double-down, split, or insurance.

# Specifications and Requirements:

Write a script that will allow a user to play the popular casino game Blackjack (aka Twenty-One) with rules slightly modified to reduce some complexities of the full featured casino version of the game.

1. Let the player enter the amount of money to start with. The bank holds both the dealer and the player’s money in equal amount.
2. Let the player choose the number of decks to play with. (The standard is 6 decks in most California casinos)
3. The dealer shuffles the decks, then asks the player to place a bet.
4. The dealer draws cards from the decks to display on screen for the dealer and the player respectively, then checks if anyone has blackjack. If there’s blackjack, make the payout accordingly, then ask the player if continue to play the next round; if not, move on.
5. The dealer gives the player options to hit, stay, or check the amount of money available.
6. The dealer draws more cards or stop according to the player selected action. The round stops if anyone busts (going over 21), and payout is made accordingly, then the player chooses if continue to play.
7. In the case where no one has blackjack or busts, the dealer and the player’s hands are to be compared to decide who the winner is, and the payout is made accordingly. Then the player chooses if continue to play.
8. At any point after each payout, if anyone runs out of money in the bank, the game is over.

# Design/Pseudocode:

Print banner

Prompt the player to enter amount of money to start to initialize bank

Prompt the player to enter the number of decks to play with to initialize decks

Initialize dealer and assign the decks to the dealer

Dealer shuffles (randomize the elements in the list) the decks

While user continues to play

Place the bet

Dealer deals the first 2 cards for both dealer and player

Dealer checks for blackjack

If there’s blackjack

Make payout to winner

If anyone’s broke

Breaks the loop, game over

If player chooses NOT to continue

Breaks the loop, game over

Else go back to the beginning of the loop

Player selects option

While player not burst and option is hit

Player gets 1 card

Dealer calculates points

If player bursts

Make payout to the dealer

Set condition for player not burst to false

Else player continues to select option

If anyone’s broke

Breaks the loop, game over

If player bursts

If player chooses NOT to continue

Breaks the loop, game over

Else go back to the beginning of the loop

Dealer reveals the bottom card

If dealer hand is not under 17

Set condition for under 17 to false

Else set condition for under 17 to true

While dealer hand is under 17 and dealer not burst

Dealer gets 1 card

If dealer hand is not under 17

Set condition for under 17 to false

If dealer bursts

Make payout to the player

Set condition for dealer not bust to false

If anyone’s broke

Breaks the loop, game over

If dealer busts

If player chooses NOT to continue

Breaks the loop, game over

Else go back to the beginning of the loop

Calculate dealer’s points

Calculate player’s points

Decide winner

Make payout to the winner

If anyone’s broke

Breaks the loop, game over

Player chooses if to continue

Game over

# Classes and Functions:

**Class:** Deck

**Member Functions:**

* \_\_init\_\_
* getDeck

**Class:** Bank

**Member Functions:**

* \_\_init\_\_
* getDealerMoney
* getPlayerMoney
* setDealerMoney
* setPlayerMoney
* setBet

**Class:** Dealer

**Member Functions:**

* \_\_init\_\_
* shuffleDeck
* initialDeal
* getDealerHand
* getPlayerHand
* playerHit
* isBlackjack
* flipBottomCard
* dealerHit
* calculatePoints
* bustOrNot
* isUnderSeventeen
* displayCards
* checkForBlackjack
* optimalHand
* payout
* decideWinner

**Public Functions:**

* isInt
* getNumOfDecks
* placeBet
* chipCount
* ifContinue
* playerOptions
* ifBroke
* buildBank
* printBanner