

# University of St.Gallen

School of Management, Economics, Law, Social Sciences, International Affairs and Computer Science

---

## Programming a Blackjack game using Python

Group 15

Lorenzo De Martino 21-612-817

Chloé Léa Isoline Paterac 21-620-398

Karan Rastogi 21-612-411

University of St.Gallen

Skills: Programming - Introduction Level

3,793,1.00

Dr. Mario Silic

23 December 2022

### Our idea

During the semester, our team went to the casino to enjoy a Wednesday evening. Whilst there, we discussed about potential project ideas for the Introduction to Programming class. When we saw Lorenzo losing his money on Blackjack, the idea of writing the game in Python came to us.

### Blackjack rule summary

Blackjack is a betting game between a player and a dealer. For our game these are the rules we used.

In the card game of blackjack, the objective is to get as near to 21 as you can without going over (busting). The players are dealt standard cards which have the following value:

Ace: 1 or 11 depending on the player

2 through 10: Face value

King, Queen, and Jack: 10

Each player and the dealer are dealt two cards at the start of the game. One of the dealer's cards is dealt face down, while the player's cards are dealt face up. The player's goal is to out-handicap the dealer by having a hand worth that is higher while staying under 21.

The option to "hit" (draw another card) or "stand" is given to the player (keep their current hand value). If a player hits and their hand totals more than 21, the round is over for them (bust). The dealer's turn starts if the player decides to stand.

If the dealer's hand value is less than 17, they must hit; if it is 17 or more, they must stand. The player wins the round if the dealer busts. The dealer wins the round if she or he stands and has a better hand value than the player. The round ends in a tie if both the player and the dealer have the same hand value.

### **Code description**

This Python code is a blackjack program. The "random" module, which will be used to generate card values at random, is imported at the start of the game. Following that, the code creates a number of variables that will be utilized during the game, including "total" and "totalc," which stand for the sum of the player's cards and the dealer's cards, respectively. The player's current wager is indicated by the "bet" variable, while their overall bankroll is indicated by the "money" variable. The player's desire to carry on playing after the current round concludes will be determined by the value of the "play again" variable, which can be 1 (play again) or 0 (stop playing).

The "value card" function, which accepts a card value as an input and returns the proper numerical value for that card, is then defined in the code. The function will return a value of 10 for "Jack," "Queen," or "King," for instance, if the card is one of them. Depending on the player's selection, the function will return a value of either 1 or 11 if the card is an Ace.

The game starts by welcoming the player and asking him to enter a valid amount of money (between \$100 and \$1000) to play the game, the code then starts a loop that keeps going indefinitely.

When the player runs out of money or decides to stop playing, the main game loop starts. The player is asked to place a bet during this loop, and the code verifies that the amount is appropriate (less than or equal to the player's existing funds and larger than zero).

The player is then dealt two cards at random, and the code outputs their values. The player is given the option of selecting a value of 1 or 11 if either card is an Ace. The worth of the player's cards as a whole is then determined and shown. The dealer is dealt two cards at random by the code, and the first card's value is then shown. The player is not informed of the overall value of the dealer's cards, which is likewise determined.

After that, the player has the choice of "hitting" (drawing another card) or "standing" (end their turn). The code generates another random card for the player and shows its value if they decide to hit. Until the player decides to stand, or their total value exceeds 21, they can keep hitting (busting).

The dealer's turn starts when the player decides to stand. The second dealer card and the overall value of the dealer's cards are revealed by the code. Until their combined value is more than or equal to 17, the dealer will keep hitting. The player wins the round and receives their wager and the dealer's wager if the dealer busts (has a total value greater than 21). The round is won by the dealer and the player loses their bet if the dealer stands and has a higher total value than the player or the player has gone over 21. The round is declared a tie and the player's wager is refunded if both the player and the dealer have the same total value.

The code invites the gamer to continue playing after the round is over. If the player chooses to keep playing, a fresh round is added to the game loop. The game terminates and the player's remaining balance is shown if they decide not to continue.

### **How to run the code – step-by-step**

1. Download the current version of python from the official python website (for this program, we used Python 3.10.9)
2. Save our game to your computer
3. Open Command Prompt
4. Type in: python "*LOCATION OF THE GAME FILE*"  
For example: python "C:\Users\User\Documents\blackjack.py"
5. The game should now be running, follow the instructions you receive and have fun!