import random

from take\_card import player\_takes\_card

from take\_card import choosing\_for\_pc

from take\_card import choosing\_for\_player

from take\_card import greater\_than\_21

from take\_card import say\_hello

import os

other\_game = True

while other\_game == True:

    # warm welcome

    say\_hello()

    # list with cards

    cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

    # variables to use

    computer = []

    computer\_total = 0

    player = []

    player\_total = 0

    # Choosing cards fot computer

    computer, computer\_total = choosing\_for\_pc()

    # Choosing cards for player

    player, player\_total = choosing\_for\_player()

    # player taking more cards

    player, player\_total = player\_takes\_card(player, player\_total)

    # the player's total is greater than 21

    if player\_total >= 22:

        while 11 in player:

            player, player\_total = greater\_than\_21(player, player\_total)

        # player taking more cards

    # if computer less than

    while not computer\_total > 13 and player\_total < 21:

        if computer\_total < 13:

            print ("\nComputer total was less than 13, computer takes other card")

            computer += [random.choice(cards)]

            computer\_total = 0

            for item in computer:

                computer\_total += item

            print (f"Computer: {computer} = {computer\_total}")

    # showing the results of the game

    print ("\nResults:")

    print (f"Computer: {computer} = {computer\_total}")

    print (f"Player: {player} = {player\_total}")

    # deciding who is the winner

    if player\_total == computer\_total:

        print ("Game is a draw!")

    elif player\_total > computer\_total and not player\_total > 21 or computer\_total > 21:

        print ("Player wins!")

    else:

        print ("Computer wins!")

    if input ("Do you want to play again? y or n: ") == "n":

        other\_game = False

    os.system('cls')

input ("\nThanks for playing, press the enter key to exit.")

**Functions**

import random

cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

# Choosing cards fot computer

def choosing\_for\_pc():

    computer = []

    computer\_total = 0

    computer += [random.choice(cards)]

    print (f"\nComputer = {computer} + ???")

    computer += [random.choice(cards)]

    for item in computer:

        computer\_total += item

    return computer, computer\_total

# Choosing cards for player

def choosing\_for\_player():

    player = []

    player\_total = 0

    player += [random.choice(cards)]

    player += [random.choice(cards)]

    for item in player:

        player\_total += item

    print (f"Player: {player} = {player\_total}")

    return player, player\_total

# player taking more cards

def player\_takes\_card(player, player\_total):

    more = True

    other\_card = ""

    while more == True and not player\_total > 21:

        other\_card = input("Do you want to take other card? y or n? ")

        if other\_card == "n":

            more = False

        elif other\_card == "y":

            player += [random.choice(cards)]

            player\_total = 0

            for item in player:

                player\_total += item

            print (f"Player: {player} = {player\_total}")

    return player, player\_total

# player is greater than 21

def greater\_than\_21(player, player\_total):

    if player\_total > 21:

        for position in range(len(player)):

            if player[position] == 11:

                player[position] = 1

            player\_total = 0

            for item in player:

                player\_total += item

        print (f"Your As becomes 1, now you have {player} = {player\_total}")

    player, player\_total = player\_takes\_card(player, player\_total)

    return player, player\_total

# computer is less than 17

def say\_hello():

    print (''' \_     \_            \_    \_            \_

| |   | |          | |  (\_)          | |

| |\_\_ | | \_\_ \_  \_\_\_| | \_\_\_  \_\_ \_  \_\_\_| | \_\_

| '\_ \| |/ \_` |/ \_\_| |/ / |/ \_` |/ \_\_| |/ /

| |\_) | | (\_| | (\_\_|   <| | (\_| | (\_\_|   <

|\_.\_\_/|\_|\\_\_,\_|\\_\_\_|\_|\\_\ |\\_\_,\_|\\_\_\_|\_|\\_\

                       \_/ |

                      |\_\_/  '''

     )

    input ("\nWelcome to the BlackJack game. Press the enter key to start.")

############### Blackjack Project #####################

#Difficulty Normal 😎: Use all Hints below to complete the project.

#Difficulty Hard 🤔: Use only Hints 1, 2, 3 to complete the project.

#Difficulty Extra Hard 😭: Only use Hints 1 & 2 to complete the project.

#Difficulty Expert 🤯: Only use Hint 1 to complete the project.

############### Our Blackjack House Rules #####################

## The deck is unlimited in size.

## There are no jokers.

## The Jack/Queen/King all count as 10.

## The the Ace can count as 11 or 1.

## Use the following list as the deck of cards:

## cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

## The cards in the list have equal probability of being drawn.

## Cards are not removed from the deck as they are drawn.

##################### Hints #####################

#Hint 1: Go to this website and try out the Blackjack game:

#   https://games.washingtonpost.com/games/blackjack/

#Then try out the completed Blackjack project here:

#   http://blackjack-final.appbrewery.repl.run

#Hint 2: Read this breakdown of program requirements:

#   http://listmoz.com/view/6h34DJpvJBFVRlZfJvxF

#Then try to create your own flowchart for the program.

#Hint 3: Download and read this flow chart I've created:

#   https://drive.google.com/uc?export=download&id=1rDkiHCrhaf9eX7u7yjM1qwSuyEk-rPnt

#Hint 4: Create a deal\_card() function that uses the List below to \*return\* a random card.

#11 is the Ace.

import random

from replit import clear

from art import logo

def deal\_card():

  """Returns a random card from the deck."""

  cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

  card = random.choice(cards)

  return card

#Hint 6: Create a function called calculate\_score() that takes a List of cards as input

#and returns the score.

#Look up the sum() function to help you do this.

def calculate\_score(cards):

  """Take a list of cards and return the score calculated from the cards"""

  #Hint 7: Inside calculate\_score() check for a blackjack (a hand with only 2 cards: ace + 10) and return 0 instead of the actual score. 0 will represent a blackjack in our game.

  if sum(cards) == 21 and len(cards) == 2:

    return 0

  #Hint 8: Inside calculate\_score() check for an 11 (ace). If the score is already over 21, remove the 11 and replace it with a 1. You might need to look up append() and remove().

  if 11 in cards and sum(cards) > 21:

    cards.remove(11)

    cards.append(1)

  return sum(cards)

#Hint 13: Create a function called compare() and pass in the user\_score and computer\_score. If the computer and user both have the same score, then it's a draw. If the computer has a blackjack (0), then the user loses. If the user has a blackjack (0), then the user wins. If the user\_score is over 21, then the user loses. If the computer\_score is over 21, then the computer loses. If none of the above, then the player with the highest score wins.

def compare(user\_score, computer\_score):

  #Bug fix. If you and the computer are both over, you lose.

  if user\_score > 21 and computer\_score > 21:

    return "You went over. You lose 😤"

  if user\_score == computer\_score:

    return "Draw 🙃"

  elif computer\_score == 0:

    return "Lose, opponent has Blackjack 😱"

  elif user\_score == 0:

    return "Win with a Blackjack 😎"

  elif user\_score > 21:

    return "You went over. You lose 😭"

  elif computer\_score > 21:

    return "Opponent went over. You win 😁"

  elif user\_score > computer\_score:

    return "You win 😃"

  else:

    return "You lose 😤"

def play\_game():

  print(logo)

  #Hint 5: Deal the user and computer 2 cards each using deal\_card()

  user\_cards = []

  computer\_cards = []

  is\_game\_over = False

  for \_ in range(2):

    user\_cards.append(deal\_card())

    computer\_cards.append(deal\_card())

  #Hint 11: The score will need to be rechecked with every new card drawn and the checks in Hint 9 need to be repeated until the game ends.

  while not is\_game\_over:

    #Hint 9: Call calculate\_score(). If the computer or the user has a blackjack (0) or if the user's score is over 21, then the game ends.

    user\_score = calculate\_score(user\_cards)

    computer\_score = calculate\_score(computer\_cards)

    print(f"   Your cards: {user\_cards}, current score: {user\_score}")

    print(f"   Computer's first card: {computer\_cards[0]}")

    if user\_score == 0 or computer\_score == 0 or user\_score > 21:

      is\_game\_over = True

    else:

      #Hint 10: If the game has not ended, ask the user if they want to draw another card. If yes, then use the deal\_card() function to add another card to the user\_cards List. If no, then the game has ended.

      user\_should\_deal = input("Type 'y' to get another card, type 'n' to pass: ")

      if user\_should\_deal == "y":

        user\_cards.append(deal\_card())

      else:

        is\_game\_over = True

  #Hint 12: Once the user is done, it's time to let the computer play. The computer should keep drawing cards as long as it has a score less than 17.

  while computer\_score != 0 and computer\_score < 17:

    computer\_cards.append(deal\_card())

    computer\_score = calculate\_score(computer\_cards)

  print(f"   Your final hand: {user\_cards}, final score: {user\_score}")

  print(f"   Computer's final hand: {computer\_cards}, final score: {computer\_score}")

  print(compare(user\_score, computer\_score))

#Hint 14: Ask the user if they want to restart the game. If they answer yes, clear the console and start a new game of blackjack and show the logo from art.py.

while input("Do you want to play a game of Blackjack? Type 'y' or 'n': ") == "y":

  clear()

  play\_game()