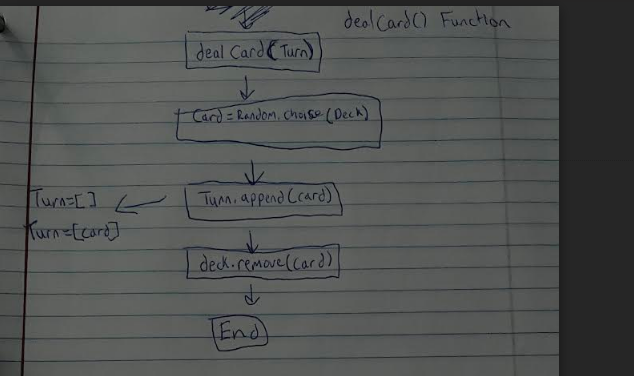
Programmers Journal - Mid-Unit Assignment

BlackJack!

September 28, 2023

[3:06] To start off my program, I am going to create a simple function that deals out a card and adds it to the list of who asked for it and then it will remove that card from the deck so there aren’t any duplicates. After the card was added to the list it added the whole code so that it didn't look right when it was printed so I had to make it only append the card value and card suit to the list, and now that printed out nicely. Next, I will create another function to calculate the total for both the player hand and dealer hand while keeping in mind the face cards all equal 10, and for the ace, I’ll make it so that it assumes it’s worth 11 but if the total is greater than 11 and contains an “A”, it will change to a 1. I need to make one more function for the dealer that reveals one card and keeps the other one hidden and if it has more than 2 cards it reveals both cards.



def dealCard(turn): #picks a random card from deck

card = random.choice(deck)

turn.append(card)

deck.remove(card)

def total(turn): #calculate the total of the hands

total = 0

num\_aces = 0 # Count of Aces in the hand

for card in turn:

card\_value = card[1] # Extract the face value of the card[my case: 1]

if card\_value.isdigit():

total += int(card\_value)

elif card\_value in ['J', 'K', 'Q']:

total += 10

elif card\_value == 'A':

num\_aces += 1

total += 11 # Assume Ace is worth 11 initially

def revealHand():

if len(dealerCards) == 2:

return dealerCards[0]

elif len(dealerCards) > 2:

return dealerCards[0], dealerCards[1]

[4:05] After creating these functions, I realized I forgot to add ‘T’ to the list of face cards, so it wasn’t adding to the total. Now It was time to create some variables to help the game flow, and also, to create the deck to show the welcome screen and show the instructions of how the game will be played. I'm also going to ask the user if they would like to play before the main game loop. I made a rule where the dealer must stand on 17 and cannot hit above 17.

playerCards = [] #users cards score if above 21: lose

dealerCards = [] #computers cards score if above 21: win

play = True #whether or not to start the game loop - Assume True

playerPlay = True #player not stand or bust

dealerPlay = True #dealer not stand or bust

for face in faces:

for suit in suits:

deck.append("["+face+suit+"]") #Creates all the values for each suit in the deck

print(Fore.GREEN)

#Welcome screen and Intructions for player

print('''

█▀▀▄ █░░ █▀▀█ █▀▀ █░█ ░░▀ █▀▀█ █▀▀ █░█

█▀▀▄ █░░ █▄▄█ █░░ █▀▄ ░░█ █▄▄█ █░░ █▀▄

▀▀▀░ ▀▀▀ ▀░░▀ ▀▀▀ ▀░▀ █▄█ ▀░░▀ ▀▀▀ ▀░▀

-Get as close to 21 as you can to win

-Dealer must stand on 17 or up

-Player can choose to hit or stand

-Aces can be worth 1 point or 11 points

-Face cards are all worth 10 points

''')

print(Fore.RESET)

while True:

start\_game = input("Would you like to start a new game? [Y/y] or [N/n]: ")

if start\_game.lower() == 'y':

break # Exit the loop and start the game

elif start\_game.lower() == 'n':

exit() # Exit the program if the user doesn't want to play

else:

print("Error: Invalid response. Please enter 'Y' to play or 'N' to quit.")

[4:41] Next, it was time to move on to the main game loop. To start off the loop I needed to deal 2 cards to the playercards and the dealercards and print off one of the dealer cards and show the player cards and their total.

while play:

for i in range(2):

dealCard(playerCards)

dealCard(dealerCards)

while playerPlay or dealerPlay:

print(f"Dealer has {revealHand()} and x")

print(f"You have {playerCards} for {total(playerCards)}")

[5:03] It must ask the player if they want to hit or stand. After writing the code, I tried to test the game and I kept getting an index error when standing and the hitting was all off too. I realized its because it kept looping and shooting out cards whenever and the deck kept running out so to fix this I got rid of the while play loop and set all the indentation and it seems to work fine now.

#main game loop

while play:

while playerPlay or dealerPlay:

print(f"Dealer has {revealHand()} and ?")

print(f"You have {playerCards} for {total(playerCards)}")

if playerPlay:

data = input("(s)tand\n(h)it\n:")

if total(dealerCards) >= 17:

dealerPlay = False

else:

dealCard(dealerCards)

if data.lower() == 's':

playerPlay = False

elif data.lower() == 'h':

dealCard(playerCards)

else:

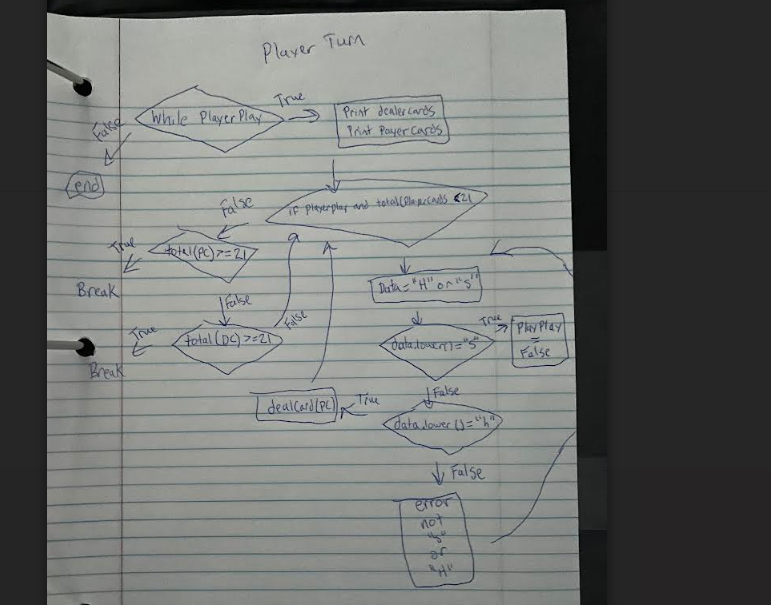
print("Error: Invalid input.")

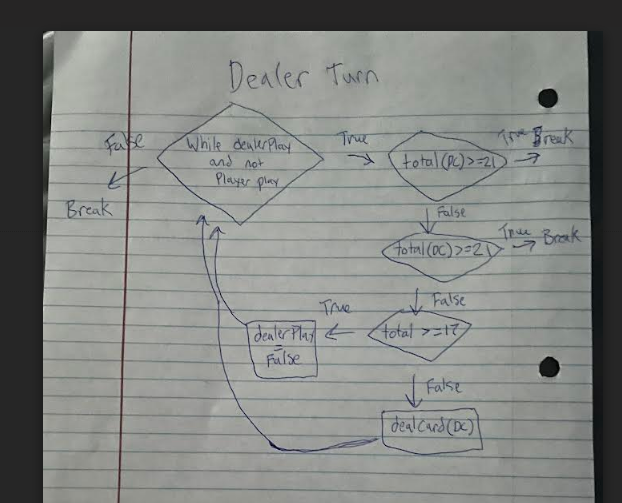
if total(playerCards) >= 21:

break

elif total(dealerCards) >= 21:

break





[6:00] I had an issue where the dealer would still pull a card after the player had hit, and the dealer shouldn’t pull a card until the player has decided to stand. So to fix this I separated the game loop and created a while loop of while dealer in and player not in, then and only then will the dealer pull a card:

while dealerPlay and not playerPlay:

if total(dealerCards) >= 17: # Dealer stands on 17 or up

dealerPlay = False

else:

dealCard(dealerCards)

if total(playerCards) >= 21: # Player bust or blackjack

break

elif total(dealerCards) >= 21: # Dealer bust or blackjack

break

[7:16] It needs logic to determine if the player has won, lost, or tied, or if someone has got 21. I had an issue where the dealer was sometimes standing on 16 and that's not supposed to happen so I changed the while player “and” dealer play loop to a “or” loop. Also, when the user inputs anything other than ‘s’ or ‘h’ it would hit regardless so I made an else error to go along with that(in blue). If the player hit up to 21, it would automatically end the game and the player would instantly win but in normal blackjack you would still have to choose to stand and the dealer would have a chance to push you. So to fix this I added an ‘and’ len(card) == 2 as well for blackjack, and that seems to have done the trick.

#Check for winner!

if total(dealerCards) == total(playerCards):

print(f"Push! You had {playerCards} for {total(playerCards)} and the dealer had {dealerCards} for {total(dealerCards)}")

print("Tied.")

elif total(dealerCards) == 21:

print(f"BlackJack! Dealer got 21. {dealerCards}")

print("You lose.")

elif total(playerCards) == 21:

print(f"BlackJack! You got 21. {playerCards}")

print("You win!")

elif total(playerCards) > 21:

print(f"Bust! You had {playerCards} for a total of {total(playerCards)} and the dealer had {dealerCards} for a total of {total(dealerCards)}")

print("You lose.")

elif total(dealerCards) > 21:

print(f"Bust! You had {playerCards} for a total of {total(playerCards)} and the dealer had {dealerCards} for a total of {total(dealerCards)}")

print("You win!")

elif total(playerCards) > total(dealerCards):

print(f"You had {playerCards} for a total of {total(playerCards)} and the dealer had {dealerCards} for a total of {total(dealerCards)}")

print("You win!")

elif total(playerCards) < total(dealerCards):

print(f"You had {playerCards} for a total of {total(playerCards)} and the dealer had {dealerCards} for a total of {total(dealerCards)}")

print("You lose.")

[8:53] With the code all working as intended so far, the last thing I am going to do is add a play again function if the player would like to start a new game. I will do this with a simple return true or false code and if true, I will reset all the variables and make sure to reset the deck. I had to create an outer loop for the game for multiple rounds to be able to be played so it could go back to it and if the function returned true it would reset all the variables for a fresh new game. Try again sounded a bit off so I changed it to “new game?”.

def tryAgain():

while True:

play\_again = input("Try again? [Y/y] or [N/n]: ")

if play\_again.lower() == 'y':

return True # Player wants to play again

elif play\_again.lower() == 'n':

return False # Player wants to quit

else:

print("Error: Invalid response. Please enter 'Y' to play again or 'N' to quit.")

if not tryAgain():

print("Thanks for playing! Goodbye.")

exit()

else:

playerCards = []

dealerCards = []

playerPlay = True

dealerPlay = True

# Reset deck

deck = []

for face in faces:

for suit in suits:

deck.append("["+face+suit+"]")

September 29, 2023

[12:00] When it was displaying the hands it was displaying weirdly, to my understanding it's showing the whole list and actually not printing the cards, so to fix this I’m going to create a function to display all the items in the list with the end= “ “. I will also be changing everything to make it look a lot neater with the \n command and messing around with the color settings in colorama to make it look cool to me.



[3:00] Conclusion: After making a bunch of changes to the way everything looks, I’m very happy with how it turned out and using the color for looks. I had a lot of fun working on this assignment because I always found game 21 so entertaining and it was a great way to practice the new functions I have learned. I feel like this assignment further helped my understanding and improved my skills as a coder overall.

