Card Game

10/28/2022

90/100 Points

Attempt 1

Review Feedback
10/28/2022

Attempt 1 Score:
90/100

View Feedback

Anonymous Grading: **no**

~~Unlimited Attempts Allowed~~
12/9/2022

Details

Preparation

Learn through videos, lectures, and/or book up to arrays

Objective

The objective of this lab is to create the beginnings for a python-based card game. We DO NOT expect a fully functioning card game. What we do expect is that you create a main function and various functions that will accomplish the following goals:

- Build a single-dimension array to keep track of the location of every card
- DO NOT move cards around... Just use the array to keep track of where each card is
- All card data is really integers - Use other arrays to translate integers to suits, ranks, and player names
- All cards will start in the DECK
- Write a function that translates a card number to a card name. HINT - look at the suitName and rankName arrays
- Write a function to assign a card to a given player
- Dealing a card involves picking a card number and assigning a new location to the corresponding element of cardLoc
- Write a function that displays the location of every card. (Early versions should show numeric values for the card number and location. Later versions can include string values for prettier output.)
- Write a function that prints the name of every card in a given hand

Starter Code

Please begin by copying the following code to your editor. You will not need to change my code at all, (in fact, you **may not** change the main code) but you will need to add several new functions to make it work correctly.

```
""" cardGame.py
    basic card game framework
    keeps track of card locations for as many hands as needed
"""
from random import *

NUMCARDS = 52
DECK = 0
```

```
PLAYER = 1
COMP = 2

cardLoc = [0] * NUMCARDS
suitName = ("hearts", "diamonds", "spades", "clubs")
rankName = ("Ace", "Two", "Three", "Four", "Five", "Six", "Seven",
            "Eight", "Nine", "Ten", "Jack", "Queen", "King")
playerName = ("deck", "player", "computer")

def main():
  clearDeck()

  for i in range(5):
    assignCard(PLAYER)
    assignCard(COMP)

  showDeck()
  showHand(PLAYER)
  showHand(COMP)
```

Sample output

Your program output should look something like this. (It's perfectly fine if things don't line up perfectly. It's the general structure I'm looking for.)

```
Location of all cards
#      card           location
0    Ace of hearts       deck
1    Two of hearts       computer
2    Three of hearts     deck
3    Four of hearts      deck
4    Five of hearts      deck
5    Six of hearts       player
6    Seven of hearts     deck
7    Eight of hearts     player
8    Nine of hearts      computer
9    Ten of hearts       deck
10   Jack of hearts      deck
11   Queen of hearts     deck
12   King of hearts      deck
13   Ace of diamonds     deck
14   Two of diamonds     deck
15   Three of diamonds   deck
16   Four of diamonds    deck
17   Five of diamonds    deck
18   Six of diamonds     deck
19   Seven of diamonds   deck
20   Eight of diamonds   deck
```

```
21  Nine of diamonds    deck
22  Ten of diamonds     computer
23  Jack of diamonds    deck
24  Queen of diamonds   deck
25  King of diamonds    deck
26  Ace of spades       deck
27  Two of spades       deck
28  Three of spades     deck
29  Four of spades      deck
30  Five of spades      deck
31  Six of spades       deck
32  Seven of spades     player
33  Eight of spades     deck
34  Nine of spades      deck
35  Ten of spades       deck
36  Jack of spades      deck
37  Queen of spades     deck
38  King of spades      deck
39  Ace of clubs        deck
40  Two of clubs        deck
41  Three of clubs      deck
42  Four of clubs       deck
43  Five of clubs       deck
44  Six of clubs        deck
45  Seven of clubs      deck
46  Eight of clubs      deck
47  Nine of clubs       computer
48  Ten of clubs        computer
49  Jack of clubs       player
50  Queen of clubs      player
51  King of clubs       deck
```

```
Displaying player hand:
Six of hearts
Eight of hearts
Seven of spades
Jack of clubs
Queen of clubs
```

```
Displaying computer hand:
Two of hearts
Nine of hearts
Ten of diamonds
Nine of clubs
Ten of clubs
```

Hints

This assignment tends to trip people up. It is not difficult to write, but I'm asking you to think in a way that may be completely unfamiliar to you. The key to this program is the way the cards are organized in computer memory, which is NOT the way they work in real life.

If you search for online help on this project, I'm pretty certain you'll get advice that will confuse you more than it helps. It's no shame to be confused. I've seen this assignment on stackoverflow and reddit, and most advice on these sites is going to confuse you more than it helps. The solution I'm looking for here is much cleaner than most of the card implementations you'll find online once you understand it. Use that kind of resource for general information, but never as a tool to answer homework problems. If you have found a solution online, we have probably already seen it too, and we'll know if you turn it in. Just ask for clarification from me or your TA if you're getting lost.. (this would be a bad week to miss recitation...)

- Most people make this way too complicated
- You don't need any arrays I didn't already give you
- Do not use a two-dimensional array (unless you want to for the blackbelt challenge)
- Computer memory does *not* work like actual cards. You are not really moving things around.
- You are storing a DATABASE of card LOCATIONS. This is actually more powerful than storing cards themselves, but it's not the way humans do it.

Card numbers

It may help to think of the card numbers like this:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51
```

Note there are 4 rows and 13 columns. I wonder if there is an easy mathematical way to find the row and column of a number in this matrix? How could that be helpful?

Submission

Please submit the following on Canvas:

- Your .py file (NOT a link to your pythonanywhere page and NOT a word document)
- A .txt file describing your algorithm (congruent with the requirements for algorithm files described in an earlier assignment about algorithm files)
- If you are turning in a blackbelt version, submit your blackbelt as a separate .py file from your basic .py file

Blackbelt

Use what you have created and extend it to actually implement a working card game. War is pretty easy, as is BlackJack. If you're feeling more ambitious, you might try something like "Go Fish." Poker is easy to write, but the scorekeeping can be tricky.

Or keep the program as it is and try another implementation closer to the way humans play cards (To do this most appropriately, you'll need to look up a data structure called a 'stack.')

File Name	Size

File Name	Size	
cardgamecode.txt	3.26 KB	
cardgame.py.txt	1.38 KB	