FIT9133

Introduction to Python

Assignment 2

1) **Card Class**:

Card class has two attributes face and suit of a card. Face determines the rank of the card for e.g. A, 2, 3…etc. While suit determines the type of card e.g. Heart, Spade, Diamond and Club.

Card class also has a function def __str__(self) which displays the card on console in string format.

Card class has also got a mutators and accessors function that returns the suit and face of card.

2) **Deck Class**:

Initially the card class is imported in Deck class to generate a Deck of cards. A Deck class has three attributes start value, end value and number of suits. Basically, Deck class can generate 'N' number of cards and 'N' number of suits in a Deck. However, I have used two loops to generate a Deck of card. First loop generates a Deck of 52 cards with Heart, Spade, Diamond and Club suit. While another loop generates a deck of 'N' number of cards and suits. I have taken range of suits in numerical format. For e.g. suits for 'N' number of cards will start from 0,1,2,3…etc.
For e.g.
d = Deck(1,14, 1)
for above values Deck will generate values like :
['01', '02', '03', '04', '05', '06', '07', '08', '09', '010', '011', '012', '013']
Here '0' represents suit and 1,2,3 etc. represents face value of a card.

For d = Deck(1, 13, 4)
The deck will generate 52 cards with 4 suits.
['C7', 'D1', 'C10', 'D4', 'D12', 'C1', 'H3', 'H11', 'S12', 'C5', 'S5', 'C9', 'D7', 'C2', 'H6', 'C11', 'S1', 'D10', 'H10', 'S6', 'H4', 'S13', 'D6', 'S4', 'C4', 'D13', 'H8', 'C3', 'D11', 'S10', 'H5', 'C13', 'H9', 'H13', 'D9', 'D2', 'D3', 'D5', 'D8', 'C8', 'S3', 'H2', 'C12', 'S9', 'S2', 'S11', 'S7', 'S8', 'H12', 'H7', 'H1', 'C6']

The Deck class contains a shuffled card function that uses random library function to shuffle the cards in deck list.

The Deck class also contains drawCard function that returns the shuffled deck of cards list.
Note: I have used drawcard function in Freecell class to fill each of 8 cascade with cards from deck list returned from drawcard

function. Also I have used 11, 12 and 13 for J, Q and K for comparing their face values. However, Later the cards are displayed in J, Q and K format only.

3) FreeCell Class:

FreeCell is the main class of FreeCell game. It has also set of rules and conditions applied on Deck of cards to play FreeCell game. Deck class is imported in freecell class to use shuffled cards from deck and fill in the cascade.

__init__(self) function is used to initialize variables that are used in functions in freecell class.

I have used 8 for loops to fill 4 cascades each with 6 cards and 4 cascades each with 7 cards from the shuffled list.

Cascade list 1 with 6 cards.

```
for i in range(6):
    y = x.pop(-1)
    self.cascade1.append(y)
```

Cascade list 2 with 7 cards.

```
for i in range(7):
    y = x.pop(-1)
    self.cascade5.append(y)
```

I have used list of list of list for representing cascades. List of list is easy to interpret and call while running any functions.

```
self.stackOfStack = [[]] * 8
```

Below are the functions used for FreeCell Game:

a. Move to Cascade function:

This function moves a card from one cascade list to another cascade list. It checks if the card to be moved is smaller than the target location card. Also, the function allows only opposite suits to append. For e.g. Heart and Diamond cards can be placed only on Spade and Club suit card. This function basically, checks the face and suit while moving a card. If the face value is smaller and suit is opposite, it allows to move a card.

This function also checks if cascade is empty it allows to append card of any suit and any colour.
Heart card can be placed on spade and club suit cards.
Club card can be placed on Heart and Diamond suit cards.
Spade card can be placed on Heart and Diamond suit cards
Diamond card can be placed on Spade and Club suit cards.
The code prompts the player to enter the list number from where he wants to move the card. There are total 8 cascade list which starts from cascade 0, cascade 1, cascade 2 and so on. The list is displayed in horizontal way. Hence, any operation can be performed only on last cards of each list.

Cascade:

```
['HQ', 'C6', 'C5', 'D9', 'C7', 'CJ']
['H2', 'C9', 'H4', 'C3', 'SJ', 'D6']
['S10', 'S6', 'D5', 'CA', 'SA', 'HJ']
['D3', 'DJ', 'S3', 'CQ', 'D8', 'D10']
['DK', 'H3', 'S9', 'H5', 'S2', 'DQ', 'CK']
['D2', 'D4', 'HK', 'H8', 'S7', 'DA', 'H9']
['SK', 'S4', 'C2', 'SQ', 'H10', 'S8', 'C8']
['C4', 'C10', 'D7', 'H7', 'H6', 'S5', 'HA']
```

Note: The list is displayed horizontally and only last items or cards are accessible and can be moved. List starts from 0 and ends at 7

Player will be asked for input every time until the Game is over.

```
1 : Move from one Cascade to another Cascade
2 : Move from Cascade to OpenCell
3 : Move from Cascade to Foundation
4 : Move from Opencell to Foundation 4
5 : Move from OpenCell to Cascade
6 : Move from foundation to Cascade
7 : Exit
Enter your choice
```

If the user wants to move a card from one cascade to another cascade, then the inputs will be as follows:

```
Enter your choice1
Enter the source cascade number from where you want to move the card : 6
Enter the target location where you want to move the card : 5
```

Enter your Choice: 1
Enter the source cascade number from where you want to move the card: **6 (List 6)**
Enter the target location where you want to move the card: **5 (List 5)**

The output should be as given below:

```
['D3', 'DJ', 'S3', 'CQ', 'D8', 'D10']
['DK', 'H3', 'S9', 'H5', 'S2', 'DQ', 'CK']
['D2', 'D4', 'HK', 'H8', 'S7', 'DA', 'H9', 'C8']
['SK', 'S4', 'C2', 'SQ', 'H10', 'S8']
['C4', 'C10', 'D7', 'H7', 'H6', 'S5', 'HA']
```

Note: List starts from 0 to 7.

Similarly, inputs can be provided for every choice the players makes to win the game.

b**) Move Card to Open Cell from Cascade:**

This function prompts the user to enter the card to be moved to open cell. If opencell is empty it will allow user to append the card into Open list.

Consider below example:

```
Enter your choice2
Enter the source cascade number from where you want to move the card : 0
Enter the Opencell location where you want to move the card : 1
```

The output should be like below:

```
Open Cell :
[]
['CJ']
[]
[]
```

c) Move Card From Open Cell :

This code in this function allows player to move a card from Open Cell to Foundation and Cascade. This function calls foundation function to move a card from Open cell

to foundation and calls move to stack function to move a card from Open cell to cascade.
To push the card in Open cell the input of Open cell list starts from 0.
For e.g.

```
1 : Move from one Cascade to another Cascade
2 : Move from Cascade to OpenCell
3 : Move from Cascade to Foundation
4 : Move from Opencell to Foundation
5 : Move from OpenCell to Cascade
6 : Move from foundation to Cascade
7 : Exit
Enter your choice5
Enter the OpenCell location from where you want to move the card : 1
Enter the Cascade location where you want to move the card : 4
```

The output should be like as given below:

```
['H2', 'C9', 'H4', 'C3', 'SJ', 'D6']
['S10', 'S6', 'D5', 'CA', 'SA', 'HJ']
['D3', 'DJ', 'S3', 'CQ', 'D8', 'D10']
['DK', 'H3', 'S9', 'H5', 'S2', 'DQ', 'CJ']
['D2', 'D4', 'HK', 'H8', 'S7', 'DA', 'H9', 'C8']
['SK', 'S4', 'C2', 'SQ', 'H10', 'S8']
['C4', 'C10', 'D7', 'H7', 'H6', 'S5', 'HA']
```

d) **Move‘to Foundation Function:**

This function takes source and target inputs from the player. This code takes input from player for source and target and then validates each cards with respect to its face value. The function looks for a Ace card for each suit in 8 list and then appends the card to foundation list. Here, Foundation list starts from 0 to 3. I have taken list of list for moving cards from or to Foundation cell.

I have allocated specific location for each suit in foundation cell. For e.g. Ace of Spade will be moved to zero location in foundation cell. Followed by Heart suit in 1st location, then club suit in 2nd location and Diamonds at 3rd location.
Foundation Cell : [‘SA’, ‘HA, ‘CA’, ‘DA’]

Cards cannot be moved to different location. If the player tries to move a card in other location than the above specified, then the function will throws an error saying that *“Spade card can move to 0th cell, Heart card in 1st cell, Club in 2nd cell and Diamond card in 3rd foundation cell”.*

```
['HQ', 'C6', 'C5', 'D9', 'C7']
['H2', 'C9', 'H4', 'C3', 'SJ', 'D6']
['S10', 'S6', 'D5', 'CA', 'SA', 'HJ']
['D3', 'DJ', 'S3', 'CQ', 'D8', 'D10']
['DK', 'H3', 'S9', 'H5', 'S2', 'DQ', 'CJ']
['D2', 'D4', 'HK', 'H8', 'S7', 'DA', 'H9', 'C8']
['SK', 'S4', 'C2', 'SQ', 'H10', 'S8']
['C4', 'C10', 'D7', 'H7', 'H6', 'S5', 'HA']

Foundation cell :
[]
[]
[]
[]
```

```
 1 : Move from one Cascade to another Cascade
 2 : Move from Cascade to OpenCell
 3 : Move from Cascade to Foundation
 4 : Move from Opencell to Foundation
 5 : Move from OpenCell to Cascade
 6 : Move from foundation to Cascade
 7 : Exit
 Enter your choice3
 Enter the source cascade number from where you want to move the card : 7
 Enter the Foundation location where you want to move the card : 1
```

The output for above inputs is:

```
['HQ', 'C6', 'C5', 'D9', 'C7']
['H2', 'C9', 'H4', 'C3', 'SJ', 'D6']
['S10', 'S6', 'D5', 'CA', 'SA', 'HJ']
['D3', 'DJ', 'S3', 'CQ', 'D8', 'D10']
['DK', 'H3', 'S9', 'H5', 'S2', 'DQ', 'CJ']
['D2', 'D4', 'HK', 'H8', 'S7', 'DA', 'H9', 'C8']
['SK', 'S4', 'C2', 'SQ', 'H10', 'S8']
['C4', 'C10', 'D7', 'H7', 'H6', 'S5']

Foundation cell :
[]
['HA']
[]
[]
```

e) gameFreecell function.

This function is generally used for display in the FreeCell game. While loop is used to prompt the user for next operation to be performed, until the player wins the Game. This function also has code to display foundation list and Open Cell list every time to the Player to keep the track of the cards to be moved. For every input particular

function is called. For e.g. to Move a card from one cascade to another cascade, move to stack function Is called.

```
if self.num == 1:
    source = int(input("Enter the source cascade number from where you want to
move the card : "))
    target = int(input("Enter the target location where you want to move the
card : "))
    self.moveTostack(source, target)
```

The function also keeps a track of number of moves a player uses to win this game. I have used a counter to keep a track on number of moves. Also, I have inserted a quit or Exit option in case the player wants to quit the game. Player needs to enter a values between 1 to 7 for each options.

f) FaceCheck function:

This function is used to display face values of S11, S12, S13 as SJ, SQ and SK.
For every card it checks its face value. If face value is 11 then the function converts 1 into A, 11 into ‘J’, 12 into ‘Q’ and 13 into ‘K’.

```
['DQ', 'S10', 'S8', 'DA', 'SQ', 'H3']
['HA', 'SJ', 'CQ', 'H10', 'H6', 'D7']
['S6', 'C2', 'S7', 'H4', 'DJ', 'S2']
['D5', 'SK', 'D10', 'H9', 'D9', 'S9']
['HJ', 'HQ', 'D3', 'S5', 'C7', 'C10', 'D8']
['D6', 'D2', 'CJ', 'DK', 'CA', 'HK', 'D4']
['C9', 'C6', 'H7', 'S3', 'CK', 'C4', 'H5']
['C8', 'H2', 'SA', 'C3', 'C5', 'H8', 'S4']
```