COSI 10A Autumn 2020

Video Draw Poker

This programming assignment combines advanced list operations, complex logic, and creation and use of dictionaries. The goal is to build a video poker game as one might find at a casino or app store.

Starting with a balance of $100, in a while loop while the balance > 0, you ask the user for a bet. The bet has to be <= the current balance, but a bet of 0 ends the game.

Next, you create a new deck of 52 cards in random order, and deal a hand of 5 cards, drawing cards in order from the deck using list.pop(0). `Newdeck` and `deal` should be separate functions.

I supply `Printcard(card)` which uses Unicode characters to print the suits. You write `Printhand(hand)` which goes thru the 5 cards, printing cards, or 2 spaces if the card is negative.

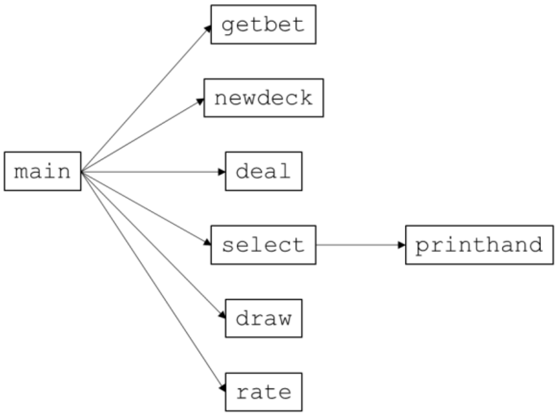
Next engage in `select(hand)` which is a discard loop where the user decides to keep or throw his/her cards (using keys 1-5) then enters 0 to draw new replacement cards from the deck. When the user picks 3, for example, set the third card to – the third card, which will make it "disappear" when `Printhand` is executed again, unless the user changes her mind and picks 3 again.

(Note that in casino videopoker, the user selects the cards to hold while for ease of implementation we are selecting the cards to throw away.)

After the user is done discarding, she enters 0, and you replace the negative cards with new cards drawn from the deck. This is called drawing.

Finally, using the included function rate `RATE` and 2 dictionaries, tell the user what hand they got and their new balance, which is balance + bet * payoff,. then loop back to get the next bet.

Each phase should be written as a separate function to help manage complexity of the main function. This figure shows a likely arrangement of subfunctions:



`NEWDECK` can use a nested loop or a list comprehension to generate 52 numbers 1.1, 1.2, 1.3, 1.4, 2.1…13.4. (each card is face value 1-13 plus suit/10.) Use `random.shuffle` to mix the cards.

`Getbet(stake, prevbet)` should allow user to bet the previous amount by inputting an empty string. Betting 0 should end the game, so you need to be crafty about testing the new bet and breaking out of the main while loop. Betting less than 0 or more than their balance should generate an out of range error and ask again. Betting their full balance should print out "Risky!"

I include model code for some of the more esoteric parts of this programming assignment. Constant `SUITS` sets up Unicode characters for the 4 suit symbols. `PRINTCARD` prints out one card using string indexing, and `FRAC` gets the fractional part of a card as an integer.

`RATE` is so complex that I wrote it myself. It turns a hand into a number, increasing with the rarity of a hand. As in most video poker games, only jacks or better counts for a pair, and a royal flush is considered higher than straight flush. Here's what the numbers returned by `rate` actually mean:

1	Busted
2	Jacks or Better pair
4	Two Pair
8	Three of a kind
9	Straight
10	Flush
16	Full House
64	Four of a kind
90	Straight Flush
110	Royal Flush

You will turn this into a dictionary and look up the result of `RATE` to get the name of the hand.

You will build a second dictionary just of numbers, which are the payouts for when the player gets a particular hand. The only constraint is that Busted should lose their bet and when they win it adds some multiple of bet to their balance. Here are the numbers I used:
.

1	-1
2	1
4	2
8	4
9	6
10	8
16	15
64	25
90	40
110	50

Here is a trace of my game play showing a few different bets

```
You have 100. whats your bet (0 to quit)?[10]25
3♦ 5♦ 9♣ 9♠ 3♠ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!2
3♦    9♣ 9♠ 3♠ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!0
3♦ 4♥ 9♣ 9♠ 3♠ Your hand is  Two Pair
You win 50
You have 150. whats your bet (0 to quit)?[25]
K♦ K♠ K♥ 8♠ 3♦ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!4
K♦ K♠ K♥    3♦ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!5
K♦ K♠ K♥       Enter 1 thru 5 to discard (or recover) a card, 0 to draw!0
K♦ K♠ K♥ 6♥ 9♠ Your hand is  Three of a Kind
You win 100
You have 250. whats your bet (0 to quit)?[25]500
You don't have that much!
You have 250. whats your bet (0 to quit)?[25]50
2♠ A♣ 3♦ 4♣ 4♦ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!5
2♠ A♣ 3♦ 4♣    Enter 1 thru 5 to discard (or recover) a card, 0 to draw!0
2♠ A♣ 3♦ 4♣ T♣ Your hand is  Busted
You win -50
You have 200. whats your bet (0 to quit)?[50]
J♠ A♣ 8♥ 3♠ Q♠ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!3
J♠ A♣    3♠ Q♠ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!4
J♠ A♣       Q♠ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!5
J♠ A♣          Enter 1 thru 5 to discard (or recover) a card, 0 to draw!0
J♠ A♣ 9♦ 7♠ T♠ Your hand is  Busted
You win -50
You have 150. whats your bet (0 to quit)?[50]150
Risky!
9♦ 8♠ 3♣ 5♠ 8♣ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!1
   8♠ 3♣ 5♠ 8♣ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!3
   8♠    5♠ 8♣ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!4
   8♠       8♣ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!0
2♠ 8♠ 2♣ 3♠ 8♣ Your hand is  Two Pair
You win 300
You have 450. whats your bet (0 to quit)?[150]450
Risky!
Q♠ A♠ 9♥ 7♦ T♦ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!3
Q♠ A♠    7♦ T♦ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!4
Q♠ A♠       T♦ Enter 1 thru 5 to discard (or recover) a card, 0 to draw!5
Q♠ A♠          Enter 1 thru 5 to discard (or recover) a card, 0 to draw!0
Q♠ A♠ 4♦ 2♦ 6♥ Your hand is  Busted
You win -450
Thank you for playing. You are exiting with $ 0
```

Here is the given code:

```
SUITS=u'\u2660'+u'\u2665'+u'\u2666'+u'\u2663'

def printcard(card):
    print('A23456789TJQK'[int(card)-1],SUITS[frac(card)-1],sep='',end=' ')

def frac(card):
    return round(card %1 * 10)

def rate(hand):  #sort and rate a poker hand
    shand=sorted(hand)
    i,j,prod = 0,1,1
    while j < 5:
        if int(shand[i]) == int(shand[j]):
            prod=prod*2**(j-i)
            c=int(shand[j])  #what card made the 2,3,4 of a kind
        else:
            i=j
        j+=1
    if prod == 1: #check for strait and flush and high strait
        d,f,e = 9,10,11
        for i in range(1,5):
            d = 1 if int(shand[i]) != int(shand[i-1])+1 else d
            e = 1 if int(shand[i]) != 8+i+int(shand[0]) else e
            f = 1 if frac(shand[0]) != frac(shand[i]) else f
        prod=d*e*f
    prod = 9 if prod == 11 else prod # there aint no high strait
    prod = 1 if prod == 2 and 1 < c < 11 else prod #jacks or better
    return prod
```