1. Big Data 1 - Exercise 2

Create a card based game using Python.
See detailed problem statement at the end of document.

1.1 Approach for the problem:

I misunderstood the problem statement regarding "AFTER starting player has seen his card and selected the characteristic for the round, THEN he chooses whether to invoke GodSpell". I have implemented the game to be as "BEFORE seeing the card, the starting player is given the choices to play as GodSpell/ Resurrect/ Regular mode".

I decided to get some experience with file handling and therefore the trump cards data is picked up from a cards data txt file. Being new to OOP way, the game was first designed on paper and I tried think of various entities and the actions they would take. I identified three classes: Card, Deck and Player. Then identified methods to be built to emulate actions like roll of dice, throwing a card, picking up a random card and adding to outdated deck were added to the classes.

The trump cards are based on Movies.

The winner when comparing characteristics is only based on higher value wins for all characteristics.

Two global variables are used: one for the #characteristics (set to 6) and the input data file location.

The input file is a comma separated .txt file with 1 header row and 10 data rows and is also part of the Github repo.
A short description follows showing the first 3 lines of the file as an example:

1) RatingSc,ActingSc,StorySc,OriginalitySc,PositiveReviews,Budget,Movie,CardNo
   8.2,7.8,8.94,8.8,45,1.55,Primer,1
   5.5,8.25,8.4,7.6,68,2.56,TheMist,2
2) It has 6 characteristics that are in the starting positions.
3) The second last value is the Movie name for the card.
4) The last value (CardNo) is only as a serial number and will not be loaded as part of the program logic.

The objects are:

1) Player:
   a. Player 1 and Player 2.
   b. Winning and Losing player objects that will point to either Player 1 or 2 for each round, based on the winner of previous round or the roll of dice for first round.
2) Deck:
   a. Data read from the input text file and stored in to AllCards Deck. This will be shuffled and used to distribute to the two players.
   b. One deck each for the two players

    c. One Outdated Deck.
3) Cards:
    a. one card for each row of input data file. Together these are in the AllCards Deck.
    b. One more copy of all the cards that are actually moved around during gameplay between the Player1Deck, Player2Deck and the Outdated Deck.

1.2 Algorithm:

1. Complete game setup process:
    a. Load the card details from the input file. Exiting program if the file is not found or if the number of data rows is not even (as we will not be able to equally distribute the cards).
    b. Set the points tally for both players to zero (player1Points = player2Points = 0).
    c. Roll a dice to decide who starts the round as the WinningPlayer. We are referring to the deck of cards of the winner as the winPlayDeck and the other players deck as the losePlayDeck.
2. Complete the setup process to start a new round:
    a. Collect all the cards from the individual players and outdated deck. Let us call these p1Deck, p2Deck, outDeck.
    b. Distribute the cards randomly and equally to the two players. So number of cards in p1Deck = p2Deck, and outDeck=0.
    c. If this is not the very first round where dice throw based assignment is required, assign the winPlayDeck and losePlayDeck to p1Deck and p2Deck as per whichever player is the Winner from previous round.
3. Winning player decides to play regularMode or godSpell or resurrectSpell with following conditions enforced as pre-checks:
    a. regularMode: No checks required
    b. godSpell: Winning player should not have used it so far
    c. resurrectSpell: Winning player should not have used it so far; and there must be at least 1 card in the outDeck
4. If resurrectMode has been chosen by Winning player:
    a. Generate a random card selection from outDeck and remove this card from outDeck
    b. Assign this card to the top of the winPlayDeck top card
    c. Go to step 7
5. If godSpellMode is chosen by Winning player:
    a. Get Winning player to choose which card position of losePlayDeck will be thrown by Losing player
6. Set the Winning player thrown card as the one at top of Winning players deck. So winPlayThrownCard is = the top card of winPlayDeck.
7. Display the card that will be thrown by Winning player. Winning player chooses the characteristic for comparison for the round.
8. Default the Losing player thrown card as the card at the top of Losing players deck. So losePlayThrownCard is = the top card of losePlayDeck.

9. If Winning Player has declared godSpell then update the losePlayThrownCard to the card specified by the Winning player in step 5a.
10. Losing player decides to play regularMode or resurrectSpell with following conditions enforced as pre-checks:
    a. regularMode: No checks required
    b. godSpell: Not allowed for Losing Player
    c. resurrectSpell: Losing player should not have used it so far; and there must be at least 1 card in the outDeck
11. If resurrectMode has been chosen by Losing player:
    a. Generate a random card selection from outDeck and remove this card from outDeck.
    b. Assign this removed card to the top of the losePlayDeck
    c. If Winning player has declared godSpell in current round, then allow Winning player to update choice of card to be thrown by the Losing player. Option is the resurrected card now at the top of losePlayDeck or stick to the original choice from step 5a. If required, update the losePlayThrownCard to the top of the losePlayDeck.
12. Throw the losePlayThrownCard.
13. Compare the characteristic values on the winPlayThrownCard v/s the losePlayThrownCard and decide who wins this round. Update the Winning player to reflect who won this round.
14. Increment the points tally for the Winner of this round (player1Points or player2Points).
15. If winPlayDeck AND losePlayDeck are NOT empty then more rounds are possible, so go to Step 2. Otherwise proceed to next step.
16. Compare the total points earned and declare whichever player has higher points as the Winner of the game.

1.3 Github location of the code:

https://github.com/rbewoor/BigDataAssignment2Submission.git

1.4 Bibliography and References

1) How To Loop Through Pandas Rows? or How To Iterate Over Pandas Rows?
https://cmdlinetips.com/2018/12/how-to-loop-through-pandas-rows-or-how-to-iterate-over-pandas-rows/

2) Extracting key/value pair from dictionary
https://stackoverflow.com/questions/17075848/extracting-key-value-pair-from-dictionary

3) Random Numbers in Python
https://www.geeksforgeeks.org/random-numbers-in-python/

4) Accessing the index in 'for' loops?
https://stackoverflow.com/questions/522563/accessing-the-index-in-for-loops

5) sprintf like functionality in Python
https://stackoverflow.com/questions/5309978/sprintf-like-functionality-in-python
Documentation > Python Standard Library > 5.6.2. String Formatting Operations
https://docs.python.org/2/library/stdtypes.html#string-formatting-operations

1.5 Problem Statement

Exercise 1: Card Game in Python(20 Points)

Game Mechanics: The game is a card game between two entities. The game can be played between two humans or a human and a machine. Choose a set of characters(fictional or real) and their characteristics. Characteristics are represented by a specific numeric value indicating the strength of the characteristic (strenghts may be indicated by negative or positive numeric value). Each card contains only one character and its characteristics.

Characteristics must be same across all characters but can vary in strength. No characteristic across characters can share the same value.

The two entities will be called Player 1 and Player 2 hence.

Distribute the cards equally among player 1 and player 2 face down such that the players cannot see the characters they have been given. Simulate a dice throw by both player 1 and player 2 where the highest number starts first

Constraints : Each player gets two special spells (God and Resurrect) . These spells are not associated to the characters but is assoicated to the player itself

Each round is played as below:

Player 1 : chooses the first card from the deck and can : play a charactertistic he wishes to challenge player 2 with . Player 2 chooses the first card in his deck and compares the charactertistic. The round is won by the Player whose characteristic weighs more and gets 1 point. The 2 cards are then kept faced down on a different deck called 'outdated'(the cards go in the outdated deck in a random order). In round 2 , The player who won the previous round begins by choosing the next card in his deck and the round proceeds as above.

Spells:

In each round a player can decide to play the God spell or the Resurrect spell along with challenging with a characteristic. God Spell: After choosing the next card and characteristic on his deck, player 1 can play the God spell allowing the player to force a card number to be played by the Player 2. Player 1 can choose any card number he wants and player 2 should then compare the charactestic on the choosen card.

Resurrect Spell: Before choosing the next card and characteristic, player 1 can choose Resurrect spell. Choosing Resurrect will create a random number and the card at the that random number position from the 'outdated' deck will be choosen. The card present at the random number at the 'outdated' deck will be added back to the top of the player deck. The player must then play this card.

When Player 1 plays the God spell , Player 2 cannot play the God spell as he is forced to choose the card , however player 2 can play the Resurrect spell and add a new card to the top of his deck. Player 1 can then force the resurrected card to be choosen or can go with his earlier choice.

God and Resurrect spell can only be played once by each Player.

The Game ends when all cards of one / both players have been played.

The game is won by the player with maximum points.