Link to Repository:
https://github.com/arungupta21/BD_Programming_Assignment_2_CardGame.git

Diary of the approach followed during the implementation

There were a few links that I visited before starting on implementing the game. They gave some insight towards how the games are implemented. However the advice on the following discussion link suggested that the typical mistake that most of the programmers do, while implementing a game, is that they try to implement everything at once.

https://softwareengineering.stackexchange.com/questions/206609/how-do-i-design-a-card-game

Therefore, I followed the bottom-up approach i.e. build a basic *working* system. Once that is implemented, build on top of it following one step forward at a time.

Therefore, following this advice, I built the game with the basic rounds, where one player draws a card and chooses a characteristic that is deemed to be the best one, and then the scores are given to the winner of each round.

Once that was done, I planned to implement the God Spell and Resurrect spells on top of it. However, as I went ahead with the implementation of the spells, there were more bugs being introduced. And eventually it was not looking like a good solution. This was probably the bad design and planning. Therefore, I just kept it to the basic rounds of game – without spells.

Following is the description of implementation and the pseudo-code:

Classes used

- **Card:** It represents an individual card having various characteristics and their values.
- **Deck:** Contains a list of objects of type Card.
- **Player:** Class representing the Player. It contains the attributes such as the Name of the player and score.

Methods

- **AddCard:** Returns a card object with the attributes' values passed to it.

Pseudocode of the implementation

1. Define and Initialize Players, **P1** & **P2**. Initialise the following attributes:
    a. Name
    b. Deck
    c. IsGodSpellAvailable
    d. IsResurrectSpellAvaiable
    e. Score

2. Create 20 cards using AddCard(), and distribute 10 cards to each player.

3. Roll the dice, to determine which player gets the chance to start.

    a. Generate one random number for each player.
    b. Take the "Modulo 6" of the random number.
    c. Whichever player gets the higher number, gets the chance to start

4. While both the players have cards in hand
    a. Print the scorecard for both the players
    b. *PlayerWithTurn* = the player that has the turn
    c. *SecondaryPlayer* = The other player
    d. Pick the top card of *PlayerWithTurn*. Call is *DrawnCardOfPlayerWithTurn*
    e. Show the details of *DrawnCardOfPlayerWithTurn*.
    f. Let the *PlayerWithTurn* choose a characteristic.
    g. Pick the top card of *SecondaryPlayer*. Call it *DrawnCardOfSecondaryPlayer*
    h. *Charateristic1* = Value of chosen characteristic of Player 1
    i. *Charateristic2* = Value of chosen characteristic of Player 2
    j. Match the values of *Charateristic1* & *Charateristic2*.
    k. Determine the winner based on the values of *Charateristic1* & *Charateristic2*.
    l. Winner get +1 in his/her score.
    m. Remove the used card from each player's deck, and add them to outdated deck.
    n. Swap the turn – if it was player 1's turn in this round, shift it to Player 2; and vice versa.

5. Visit the final score, and print the winner of the game

Arun Gupta

Matriculation #: 11012482