**Instructions**

Run the ers.py file by typing ‘python ers.py’ into the terminal.

**Rules of the Game**

Reference: <https://www.bicyclecards.com/how-to-play/egyptian-rat-screw/>

Welcome to Egyptian Rat Screw (ERS)!

* Objective: clear your deck before your opponent.
* Set Up
  + Deal the 52-card deck face down into two decks, one for the Computer and one for the Player.
* Gameplay
  + Player (the human) starts.
  + Each player pulls the top card off of their own deck and puts it on the middle or common pile face up.
    - If the card is a number card, then the next player also puts down a card.
    - This gameplay continues until either player puts down a face card or an ace (J, Q, K, or A)
      * When a face card is played by one player, then the other player has a certain number of chances to play another face card, or else the player who played the original face card wins the middle pile and begins the next round of play.
        + If an ace is played: the other player gets 4 chances
        + If a king is played: the other player gets 3 chances
        + If a queen is played: the other player gets 2 chances
        + If a jack is played: the other player gets 1 chance
      * If another face card or ace is played within the number of chances, then the gameplay continues and it is the other players turn to take chances at playing another face card.
    - The only rule that overrides the face card or ace rule is the slap rule. If the middle pile is correctly slapped according to these rules, then the slapper gets the pile. If the middle pile is incorrectly slapped, then the slapper must burn a card by pulling the top card of their deck and placing it on the bottom of the middle deck.
      * Slap rules:
        + Double: when two cards of equivalent value are laid down consecutively
        + Sandwich: when two cards of equivalent value are separated by at most one card
        + Marriage: when a queen and king are laid down consecutively in either order
  + Continue playing until one player runs out of cards. The first player to do so wins!

**Design**

* I chose to implement Deck and Card classes to organize the methods initializing the games and of retrieving cards. The reason I used these classes is because the game requires many decks (one for each player and one for the middle) that all use the same properties.
* Deck resembles a list data structure because each Deck has an order that needs to be maintained.
* I chose to code in Python for ease of use and clarity
  + By importing library functions such as shuffle from random, sys and select, I was able to easily wait for user input for the slapping mechanism and perform operations on decks of cards.