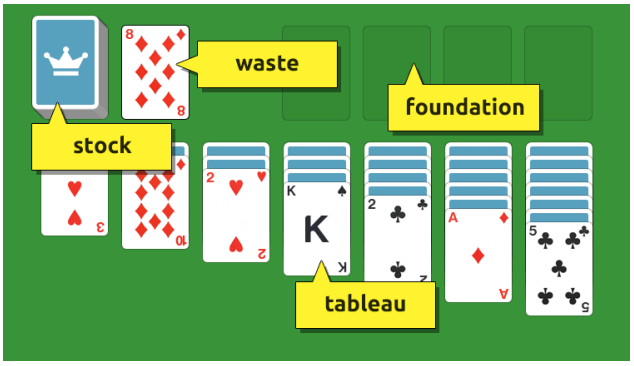
Design document

Overview

I have always enjoyed playing card games. However, my go-to game would be the popular card game, Solitaire whenever I feel bored or I'm just looking to kill some time when I'm on my own.

Classes



1) Playing Card
    a. Purpose: Initialize the deck of 52 cards and assign values for each rank.
    b. Methods/Functions:
        i. Is opposite suit color
            - **Input:** The four suits (club, diamond, spade and heart)
            - **Output:** Club and diamond cards are red whilst spade and heart cards are black
        ii. Is faced up card
            - **Input:** Card
            - **Output**: Card facing up has to be opposite color and one rank below the previous faced up card
        iii. Is transferable to Tableau pile
            - **Input:** Is opposite suit color and is faced up card functions
            - **Output**: Returns true if both inputs are satisfied
2) Deck
    a. Purpose: Distribute cards between the four different spaces: Stock, Waste, Foundation and Tableau.
    b. Methods/Functions:
        i. Deals cards
            - **Input:** Card count
            - **Output**: Dealt cards to either Tableau or Stock or Waste
        ii. Removes flipped card from deck
            - **Input:** Card
            - **Output**: Removed flipped card from Stock or Waste pile.
3) Tableau
    a. Purpose: Manages the seven piles of cards and interactions with other spaces.
    b. Methods/Functions:
        i. Flips card under each of the seven columns
            - **Input:** Card
            - **Output**: Flipped card for each column

ii. Adds cards
    - **Input:** Permitted card and column specified by player
    - **Output**: Added permitted card under specified column
iii. Transfers card from column to column
    - **Input:** Permitted card and column specified by player
    - **Output**: Successful transfer of card specified by player
iv. Transfers card to Foundation
    - **Input:** Permitted card and column specified by player
    - **Output**: Successful transfer of card specified by player
v. Transfers card from Waste Pile
    - **Input:** Permitted card and column specified by player
    - **Output**: Successful transfer of card specified by player

4) Stock and Waste
    a. Purpose: Manages the cards in the Stock and Waste piles.
    b. Methods/Functions:
        i. Transfers card from Stock to Waste pile
            - **Input:** Permitted card and column specified by player
            - **Output**: Successful transfer of card specified by player
        ii. Retrieves top card from Waste
            - **Input:** Top card from Waste
            - **Output**: Removed card from Waste deck
5) Foundation
    a. Purpose: Manages the placement of cards in the Foundation deck.
    b. Methods/Functions:
        i. Retrieves card from Tableau or Waste pile
            - **Input:** Ace card in the faced up position in Tableau or top card in Waste pile
            - **Output**: Added card in Foundation stack
        ii. Notifies player if game won
            - **Input:** Empty Tableau stack
            - **Output**: Printed display of congratulatory message

How the user will interact with the program/objects
There will be a help menu printed to illustrate the command key functions and input syntax. The user will have control on executing the commands if the action is permitted as per the official rules of Solitaire. A message for winning will be displayed if user wins the game. If not, the user will be able to quit the game or keep on playing.

Features
The project will have at least five classes as stated above. I am planning on using getter/setter properties in the stock and waste class. There will be complex flow control between each object as to follow the official Solitaire rules. Modularization of the project will be implemented to provide ease of use and sustainable flow control. It will be an interactive game in which the user gets to input commands as well as quit if necessary. The data structure of this project is mostly implemented using lists and a dictionary to organize, handle placement and retain the order or position of the card elements. I believe these features included will be of sufficient complexity to meet the project goals.