Ruby Han
W200 Fall 2020
Project 1: Solitaire

INTRODUCTION:

Solitaire is a popular card game that anyone can play on their own to kill time. The solitaire game program I created follows the official rules of Solitaire.

INSTRUCTIONS ON HOW TO USE:

1. Please enter “python solitaire.py” in the command line.
2. The user interface including the help menu for input command keys will be displayed.

ALGORITHM:

There are eight classes in total (PlayingCard, Deck, Tableau, StockAndWaste, Foundation, Help, Solitaire, Engine). Both the PlayingCard and Deck classes are for initializing valid deck of 52 cards. The main classes Tableau, StockAndWaste and Foundation manage the interaction and different spaces based on the official rules of Solitaire. The Help class displays the help menu to illustrate the command key functions and input syntax. The Solitaire class sets up the solitaire board environment and the Engine class loops gameplay and user/player interactions. Two library modules (Random and Colorama) are imported in order to enable random shuffling of the deck and colorization of the suits and user interface.

CHALLENGES AND OVERCOMING THEM:

The flow control aspect in terms of how the card objects interact with each other in different spaces is incredibly complex. The most challenging part is thinking up the logic and conditional statements for transferring valid cards within the columns in the Tableau space as well as adding valid cards from the Waste to the Tableau space. I performed tons of research online, did plenty of trial and errors when debugging the code and reversed the steps taken in order to come up with the conditional statements. I was also having a tough time keeping track of the cards and their positions. In order to overcome the challenge, I largely used lists and dictionaries to organize, handle placement and retain order or card element positions. All in all, I did have a blast doing this assignment. I learnt a lot about OOP and if provided more time, I would like to be able to implement the following features to enhance gameplay for my program.

FUTURE IMPROVEMENT OPPORTUNITIES:

- Consider using PyGame to enhance UX/UI (fancy user interface graphics, congratulatory animations etc.)
- Implement scoring/timer/undo features
- Consider displaying three cards simultaneously in the Waste pile
- Improve user experience by implementing drag-and-drop feature so that there is no need to type in command keys to activate game moves.