Name: *John Hellrung*

This reflection is to be completed individually (or with your pair-partner), though consultations with TAs and classmates are encouraged as long as they are appropriately acknowledged.

This assignment was intended for you to work with stack and queues in order to play the game of "War". Although there are several data structures that CAN be used to implement the game, there are certain ones that are the most appropriate.

This reflection document is intended to help you think about how to decide when a stack, queue or other data structure is best for a specific need in the program. For each pile listed below, which data structure did you decided to use and what was the reason for the choice? Do not use the hints in the assignment webpage as an explanation, but instead focus on the operations that you needed and how the data structure supports them.

|  |
| --- |
| 1. Dealing pile was a stack because I felt that a deal all way shuffles and pulls the last card. 2. a player's playing pile was a stack because I have always played war with a stack of cards 3. a player's storage pile was a stack because I have always played war with a stack of cards 4. the opponent's playing pile was a stack because I have always played war with a stack of cards 5. the opponent's storage pile was a stack because I have always played war with a stack of cards 6. loot pile was a queue because the loot is collected with the first card first out when I always played the game when I was younger. |

Precisely describe what the data structure should be for the initial pile that needs to be shuffled, and why it may or may not be different from the data structure for the dealing pile.

|  |
| --- |
| I believe the initial pile should be the same data structure. For example if the initial pile would be a stack I think the dealing pile should also be a stack because the data structures would be stick to the idea of recreating the card game of war. |

Most design choices have advantages as well as disadvantages. Describe the primary advantages of using a stack or a queue data structure, which is admittedly restricted in how you would use it, versus a Python list for the last three piles in the list above.

|  |
| --- |
| Most stacks and queue have the advantage are having better management of memory and speed when executing. Now, a python allows me to edit and drop any part of itself but this means that python lists are easier not better for the user. |

Describe at least one disadvantage of using a stack or a queue data structure, versus a Python list for the last three piles in the list above.

|  |  |
| --- | --- |
| If I used most stacks in my code and that disadvantages is that I was limited to just using the lasted object in the stack. For my loot method I used a queue which gave me the first one which would restricted to the use of editing only the first item in the list. Now, a python allows me to edit and drop any part of itself which is its key advantage. |  |

Imagine you are in a hackathon where you are restricted to using a single data structure for all of the piles and you have to choose between using all stacks and using all queue. Explain whether you would choose all stacks or all queues if you want to have the smallest impact on how the game functioned? Explain what changes and why you made the choice you did.

|  |
| --- |
| I believe I would choice depending on my assignment but I choice based on my liking I would choose stacking. I know how to use stacks the best and I feel the best about stacks rather than queue. Note that I know how to use queues but I like stakes better. |