Report Summary

* ***Author:*** Raymond Okyere-Forson (Rofy)
* ***Motivation***: I expanded the code from the *Card Game War* Assignment. The reason I chose this project is that I loved the notion behind the game when I first encountered the assignment. Also, the experience of writing code for that assignment to make it playable created the feeling of almost building a game from scratch. Therefore, I decided to take this project to the next level by creating a visualization interface.
* ***Purpose***: The game conducts two players: the user and the computer. Between the two players, each one picks up a card from their given playing pile and then place them in the center of the game board, whoever has the bigger card number wins. The core purpose is to display the cards with their numbers and the movement on the screen instead of playing through the console and having to type to play basically. So, the control of the entire game is done on the screen by using turtle objects and easygui for GUI interactions.
* ***Audience***: The intended audience for this game would be anyone who loves playing card games but just want to do so virtually. This program would allow that. A core benefit is that you can always play solo.
* ***Instructions:*** Please refer to the README file [here](https://github.com/Rofy-Ray/stack-queue-war-game/blob/master/README.md).
* ***Design***: Using CRC cards and pseudocode. Then, created a design plan which meets the computational requirements.
* ***Enhancement***: The game was displayed on the screen, by using turtle libraries and the easygui GUI interface. I basically animated the entire game where the users can interact with the game on the screen instead of running it in the IDE and playing through the console. Likewise, users can visualize everything happening in the game while playing it. I also created buttons in a popup window for the users to click whether or not they want to continue to play the game or quit the game.
* ***Functionality***: The primary functionality of the game will be two players (typically the user and the computer) playing in a card game where each one of them has a pile of cards. In order to play the game, each one of them picks a card from their pile and then puts on the board game. Whoever has the bigger number wins the game at that round and then they go the next round of the game until they all run out of cards. When they run of cards, each one picks a card from their storage pile and put them in their playing pile again in order to continue to play the game.
* ***Files***: The files I am submitting with this project are as follows:
  + cards.py
  + easygui.py
  + myQueue.py
  + myStack.py
  + war.py
  + exhibit.py
  + ply\_easygui.py [***easygui* might have to be downloaded to run program]**
  + head-driver.py
  + okyereforsonr\_AEC.docx
  + README.md
* ***Utilized Data Structures:*** I am using two data structures: Stacks and Queues. I used stacks for both playing piles because you have to pick the card from the top of the pile and we know items are taken from the top of stacks*.* On the other hand, I used queues for storages piles because when we store the cards into the storage piles we need to have same order as they popped from the other piles. And I want to have pile such that the first card can be popped first and the idea of using queues is first in first out (LIFO) structure it possesses. Therefore, I decided to use these two data structures in order to play our game properly.
* ***Big O Analysis:*** Since we used the stacks and queues to insert and remove data from one pile to another, the big O of this data structures will be constant because we know that big O of inserting and remove item both stacks and queue is constant O(1)**.**Reason being that you are not doing much and just removing something from the top of the stack.
* ***Resources***: I used several software sources for my project in order to conduct a successful project. I used the Python language for this project and run the code in the PyCharm IDE. I also used several python libraries including easygui, turtle and random. The combination these libraries assisted in running a smooth program. I used the random library to shuffle the cards during the game. I also used turtle library to help me draw the visualization of the cards on the game board and used easygui for easy interaction on the screen instead of playing in the IDE console. I also used some base code from the original class assignment, as well as some code that the T.A and some other colleagues helped provide.
* ***Challenges***: The main challenge was trying to figure out how to animate the game easily and create an interface for it. It was quite difficult to make game an interacting game where the user can play the game on the screen using the easygui library in the beginning. That took a while to figure out but things came together finally. Animating the movement of the cards and color-coding the cards of the user and those of the computer to assist easy distinction took some time to figure out.
* ***Testing***: For testing, I changed the size of the deck for each player and made sure choosing to continue to play or quit playing worked properly and continued to run or stopped the program accordingly.
* ***Errors***: There is no known errors as at now since the animation and visualization interface both run as expected.
* ***Measures and Assessment***: The main idea of the project is to display the Card War Game on the screen and to visualize everything that is happening in the game. I think the program does that satisfyingly.
* ***Summary***: Overall, I spent between 5-7 hours working on this project as an individual to get these results aside all the help and guidance I received.
* ***Comments***: I used easygui for our pop up box, so you might have to download it to run it when you do. This was generally fun to do!