CSC17A Project: **War** - Based on the Popular Card Game

The program uses a dynamically allocated 2D array in pointer notation in order to keep track of which cards are in each player’s hand. The dynamic 2D array is used inside of a structure called Hands inside the Hands.h header file, and it included in the main.cpp source code file. The basis of the game is to end up with all the cards by drawing higher cards than the other player, which is all caused by chance of course. You can speed through the entire program’s run sequence by pressing and holding down the SPACE bar on the keyboard.

There are several important functions my program utilizes, the first being the \*fillAry function. This function is used to initialize the main pointer variable that keeps track of which cards each player is holding. The cards are originally stored inside the dynamic 2D array as numbers 1 through 52, but they are then translated into their actual values using the % operator. This allows the program to query which card the players have rather than having to assign them in some other disorderly fashion. In other words, each number 1 through 52 represents some card in the deck, whether its a Two, Three, Seven, Ten, Queen, King, or Ace. The next important function is the used function. This function checks to make sure no duplicate cards are made while assigning random numbers from 1 to 52. The toString function translates the card numbers into their card names. The other function such as getTop, setTop, and toBot all focus on moving the cards around, such as moving cards to the bottom of player’s hands, resetting their top cards, and getting the values of their top cards. The destroy function deallocates the memory that the program allocates at the very beginning so no memory leaks are created.

The program writes the winner’s name to a binary file once the game has ended to keep track of winners. The game also provides a basic menu, 0 is to exit the program, 1 is to play, and 2 is to view the rules. Other than that, the commends explain pretty much everything else on a more detailed level within the source code itself.