Evan Henderson

CPSC 121; Section 04

December 5, 2018

My contributions to the tic tac toe program were the contents of main(), the displayBoard() function and the replay() function. The displayBoard() function is how the tic tac toe board is displayed to the user, taking in the character array that stores the information for the board, the size of the first parameter of the array, and two strings each being the input for the names of player one and player two. The function then uses cout to display the board, which can accurately be updated for every time the function is called since the data within the array and variables is being called every time the function is called. The replay() function was made to simplify the process re-initializing the tic tac toe array every time that the user wants to play again. This is done with the same nested for loop that is used in main that sets everything in the character array to \_. Then the function calls displayBoard() to show the user the re-initialized board. Both functions are of the void return type. The main() function starts off with the creation of the integer variables: go, loop1, loop2, and ans. With loop1 and loop2 initialized to 0 to prevent the use of garbage variables. Then the creation of a 2-D character array that had 3 rows and 3 columns, this will be what stores the information for the game board. Then two string variables that hold the user inputted player names and last of the arrays is a Boolean cont. Then there is a nested for loop to initialize the board array to \_. Then the program prompts the user to input a name for player 1. Followed by player 2, with the option of having player to be the computer by inputting computer or Computer as the player 2 name. Then comes the meat of the main function, the function calls and the playing of the game, all contained in a do while loop if the Boolean cont equals true. The program first checks if player 2 should be the computer by checking string name2 for Computer or computer in an if statement. If the player 2 is the computer, the program goes into a nested for loop based upon the incrementation of the variables loop1 and loop2. Then the input function for player 1 is called, then displayBoard() function is called and finally the winner() function is called to output into the go variable so that the following if statements can check if the game is over. If the game is indeed over, the if statements will ask the user if they wish to play again; if the answer is no then main returns 0, if the answer is yes then the loop variables are set so that the loops run at least one more time properly and the replay function is called to re-initialize the board. Then, the compInput() function is called, which randomly inputs the move for player 2. Lastly, in the inner loop the board is then displayed to the user after the computer’s move. The winner function then updates the go variable in the outer for loop. Followed by the if statement checking if the game should continue, if so, the loop variables are set to -1 and 2 respectively. Otherwise, there is another prompt of whether to play again, the same as the inner for loop. Next is when the player 2 is not the computer with the only difference being the player 2 input function being called instead of the computer input function.