

Homework #2: Due September 21**Overview:**

What is this assignment about?

You are going to use arrays, structures, functions, and loops to implement a simple board game called GoHome!. The game is described on the course website. Each student will construct their own board (you may draw it by hand or on the computer). Your board must have at least 20 spaces and at least 6 obstacles. Once you have your board, you will write a C++ program to play the game. We will start with the simplest version, and as the semester progresses, we will add more features to this program.

User interface**Who plays the game?**

The "players" in this game are all simulated as instances of a structure in your program. The actual people playing do not do anything to interact with this computer game, other than entering their names at the beginning.

All output will display on the screen. Each time a "player," picks a card, the value on the card is displayed. Then, the location on the board, including the value of the obstacle if there is one, is also displayed to the user. When a player reaches the end, the winner is displayed on the screen as well.

Data Structures

The board should be implemented as a 1D array. Each location of the board array can store its obstacle.

The players should be implemented as a structure. The structure's members should be the player's first name, last name, and the space on the board which their token is currently placed on.

However, in the first version of the game, you may implement only TWO players for which you do not need an array.

Card Pile: The card pile should also be implemented as an array. The cards should be shuffled at the start of the game, and then each time the pile gets used up, it should be shuffled again.

Necessary Functions

Here is a list of functions that must be written: (Note, this is not a complete list -- you will need to write other functions).

- `displayRules()`: a function called at the beginning of the program to state the basic rules of the game.
- `play()`: the function that actually plays the game.
- `taketurn()`: the function that does a single turn for a player.
- `showState()`: print to the screen the locations of all players.
- `switchPlaces()`: swaps the values of the two given players.

Limitations and Advice:

You may limit your game to play with exactly two players every time. You may also allow two players to land on the same space at the same time. Additionally, a player can win as long as it passes "home," i.e. it does not have to exactly land on the home space.

Extra Credit:

You will receive extra credit if your program provides a graphical view of the board (even if the view is generated completely by ascii text)