BattleShip

All right we have been learning a lot of things. Now it is time to create something fun. Do you know the game battleship? Yes? well, we are going to create something similar.

Battleship is a war-themed game for two players in which the opponents try to guess the location of their opponent's warships and sink them.

Players take turns calling out row (x) and column (y), those are coordinates (x,y) identify a cell that contains a ship.

As we are creating our own version, all of our ships are located in random cells of the gameBoard (each one in a cell). To make this possible you need to initialized the player's gameBoard (4x4) with zeros and to create the random positions of the warships

4 different coordinates x and y (one random number for 'x' between 0-3, one random number for 'y' between 0-3) need to be generated to indicate that there is going to be a ship and it is representated by a number 1 in the gameboard.

On each turn, the player has to enter two numbers that identify a row (x) and column (Y) of the opponent target grid. if the coordinate "hit" a ship (find a number 1 in the gameBoard) the opponent has one less ship. And an alert should be shown saying "you hit a warship"

the winner is the first player that eliminates all the four ships of the oponent (ships=0)

In order to store data we want you to:

- Create two Players objects (one for each player). The Player object has the following properties:
    1. name (ask to the users) {string}
    2. ships (we are going to play with 4) {number}
    3. gameBoard (initialized with zeros) {number} (matrix 4x4)

so in general the game is a loop that takes a player turn and asks for the cordinates of the oponent's ships. Loop is over when one of the players has 0 ships.

return final value contains a string with the winner name

Example of the random scoreBoard

	0	1	2	3
0	0	0	0	0
1	1	0	1	0
2	0	1	0	0
3	0	0	0	1

Coordinates (x,y) of ships:

- (1,0) ship 1
- (2,1) ship 2
- (1,2) ship3
- (3,3) ship4