NAME: JAMES WASHINGTON

COURSE: DATA STRUCTURES

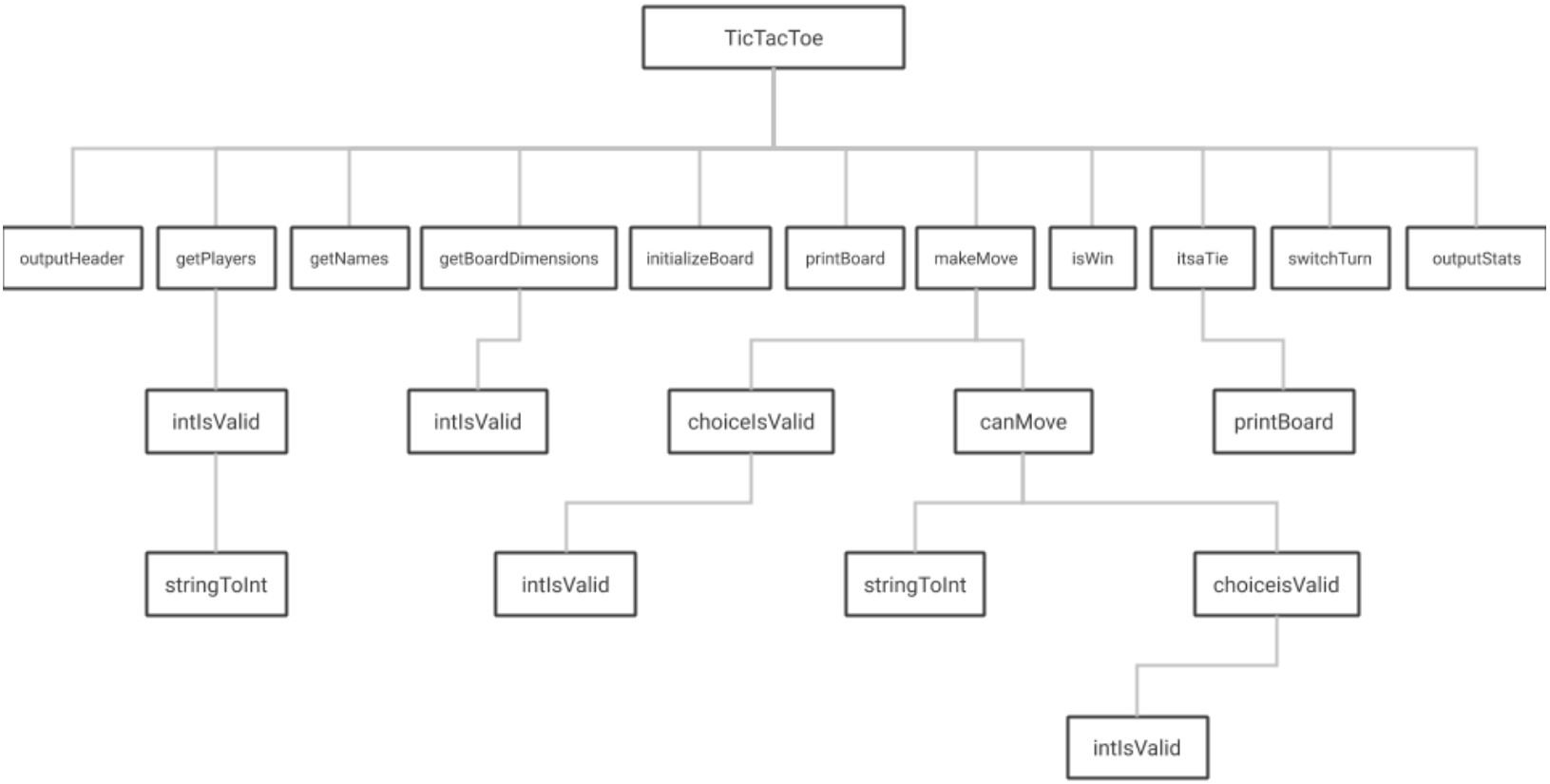
DUE DATE: 2/10/20

PROFESSOR: DR.NG

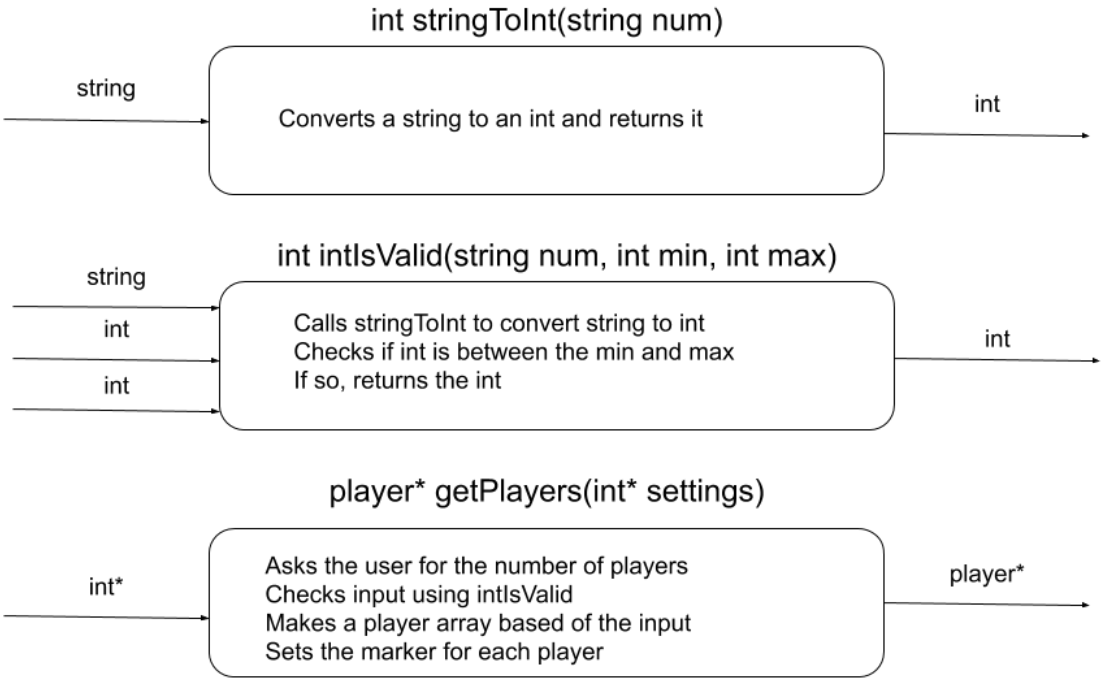
Program Design

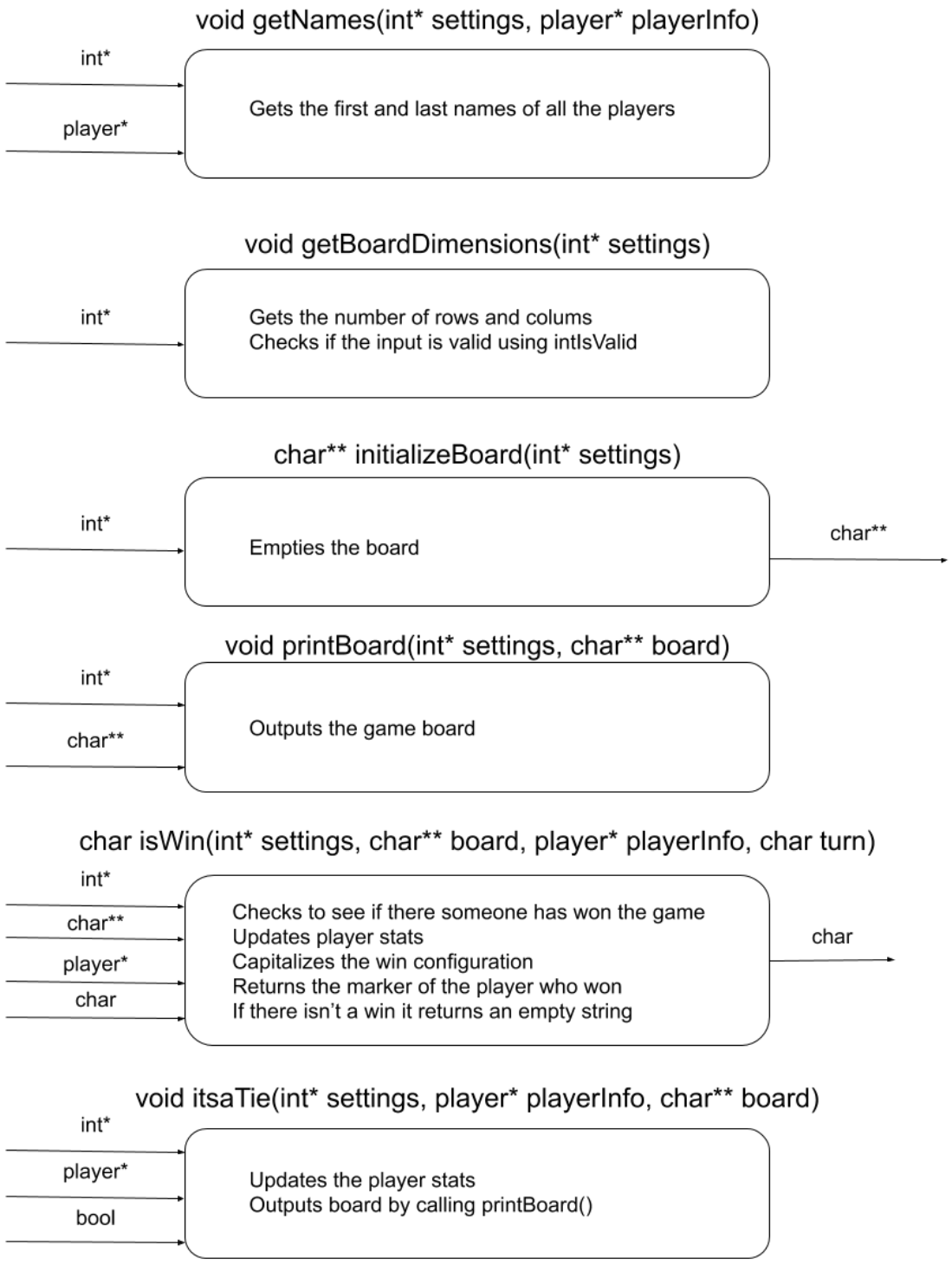
Declares the settings and board array for the game
Initializes a struct player array (playerInfo) by calling the getPlayers function
Initializes playAgain, turn, isTie variable
Declares the whoStarted, winner, and count variable
Calls the getNames function to get the player names
While the user wants to keep playing
    Calls the printBoard function to print the game board
    Calls the makeMove function so the player can make a move
    Calls the isWin function to check for a win
    While the game isn't over
        If there was a winner
            Calls the printBoard function to print the game board
            Sets the winner variable to true
        If there was a tie
            Calls the itsaTie function to do various tie functions
            Sets the isTie function to true
        Calls the switchTurn method to change turns
    If there was a tie
        Initializes the whoStarted variable to the person that didn't start the previous game
        Sets the turn variable to whoStarted
        Sets the isTie variable to false
    Else
        Sets the turn variable to the next person after the winner
    Calls the outputStats function to output the stats for the games
    Asks the user if they want to play again

Hierarchical Diagrams/ Structure Charts

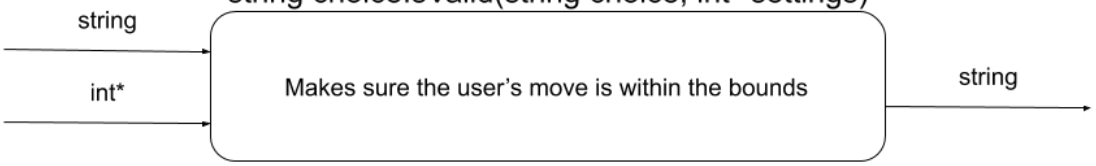


Procedure Specification

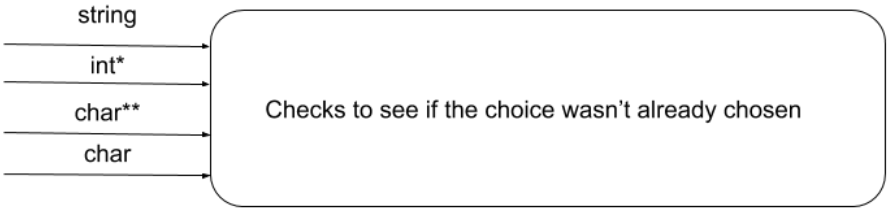



void getNames(int* settings, player* playerInfo)
int*
player*
Gets the first and last names of all the players
void getBoardDimensions(int* settings)
int*
Gets the number of rows and colums
Checks if the input is valid using intIsValid
char** initializeBoard(int* settings)
int*
Empties the board
char**
void printBoard(int* settings, char** board)
int*
char**
Outputs the game board
char isWin(int* settings, char** board, player* playerInfo, char turn)
int*
char**
player*
char
Checks to see if there someone has won the game
Updates player stats
Capitalizes the win configuration
Returns the marker of the player who won
If there isn't a win it returns an empty string
char
void itsaTie(int* settings, player* playerInfo, char** board)
int*
player*
bool
Updates the player stats
Outputs board by calling printBoard()

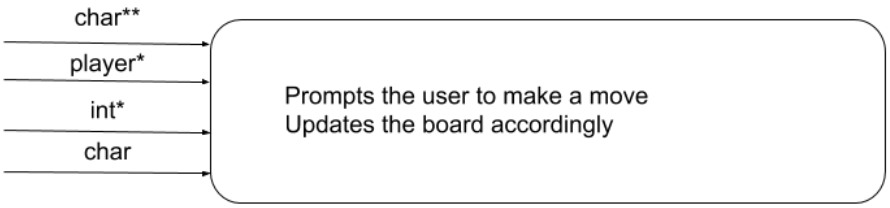
string choiceIsValid(string choice, int* settings)



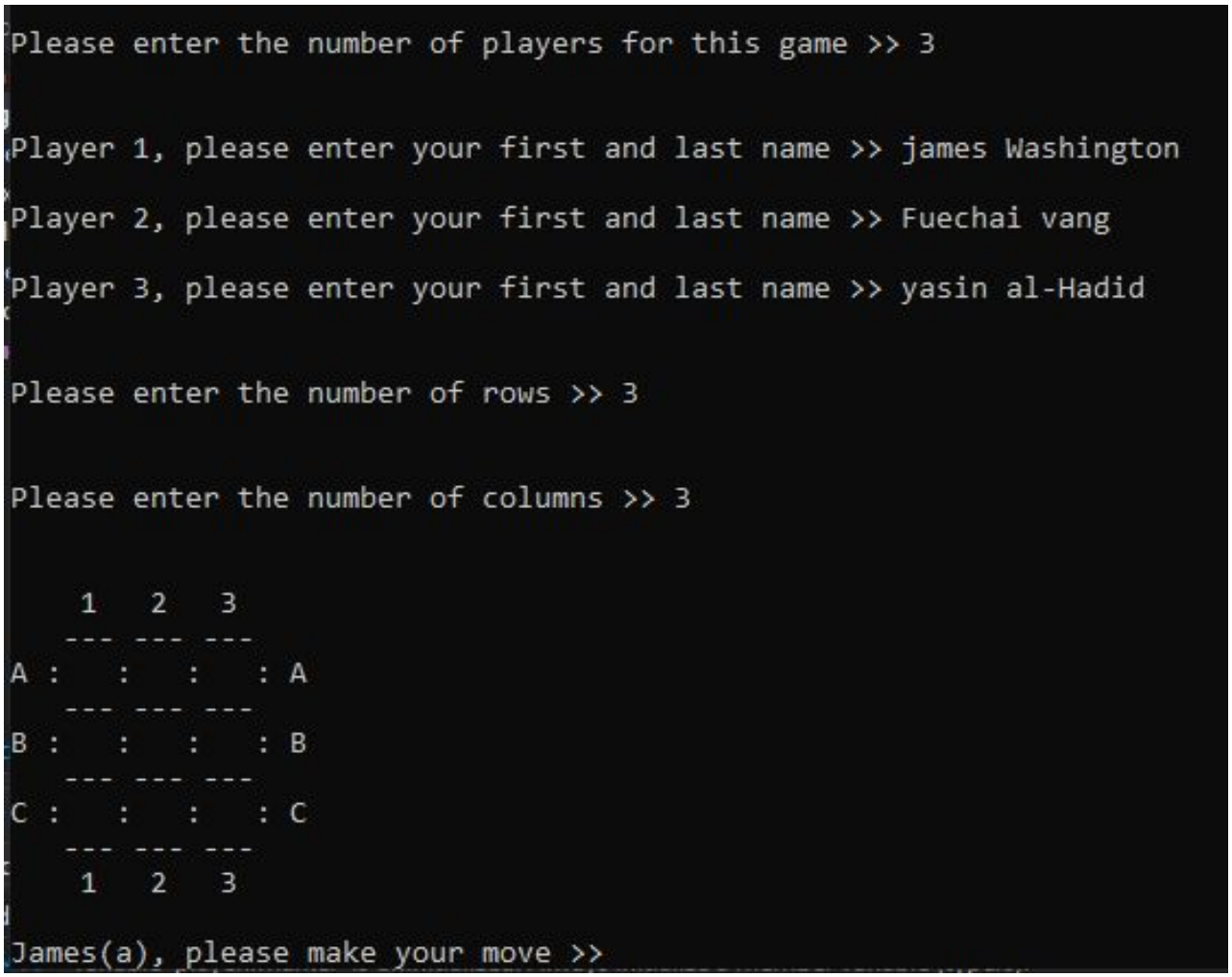
void canMove(string choice, int* settings, char** board, char turn)



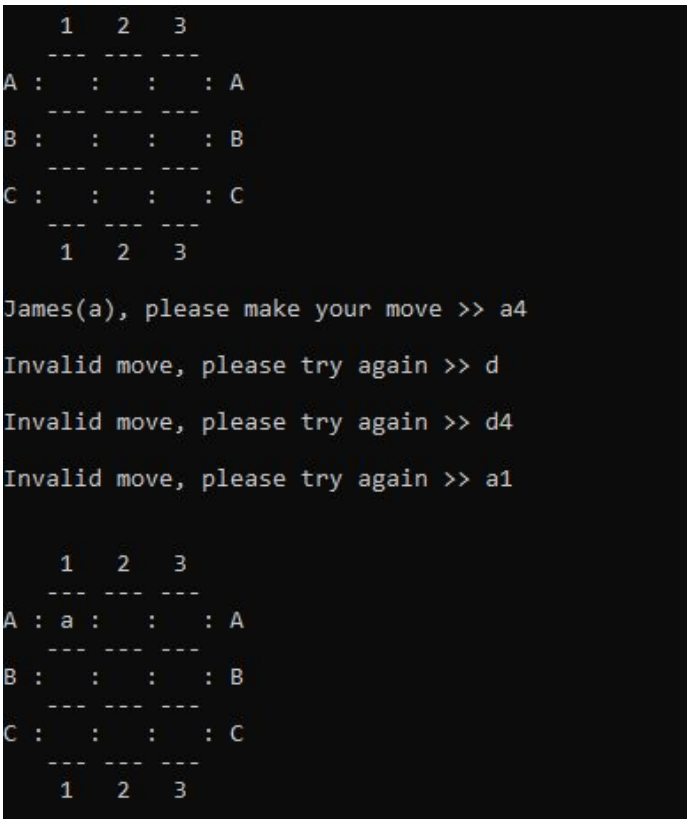
void makeMove(char** board, player* playerInfo, int* settings, char turn)



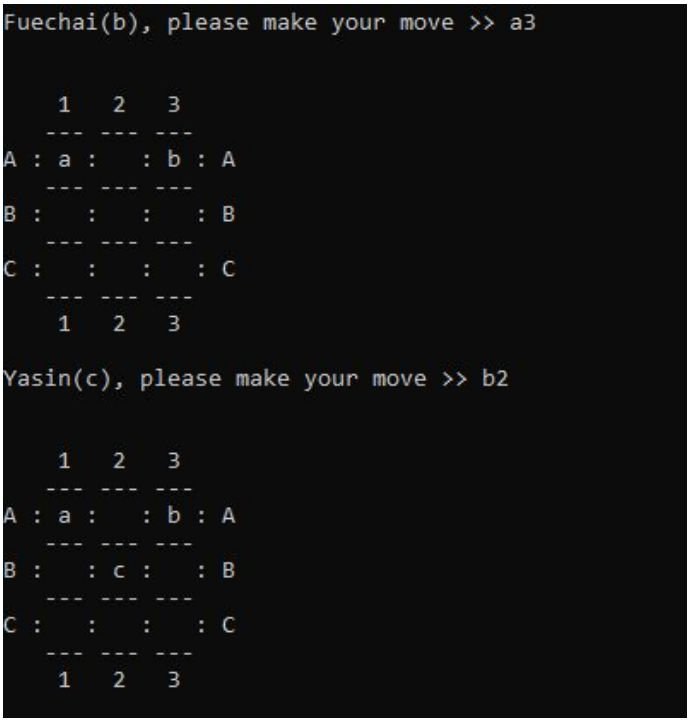
Program Testing



Gets all the players' names and reformats them. The first player starts the first game

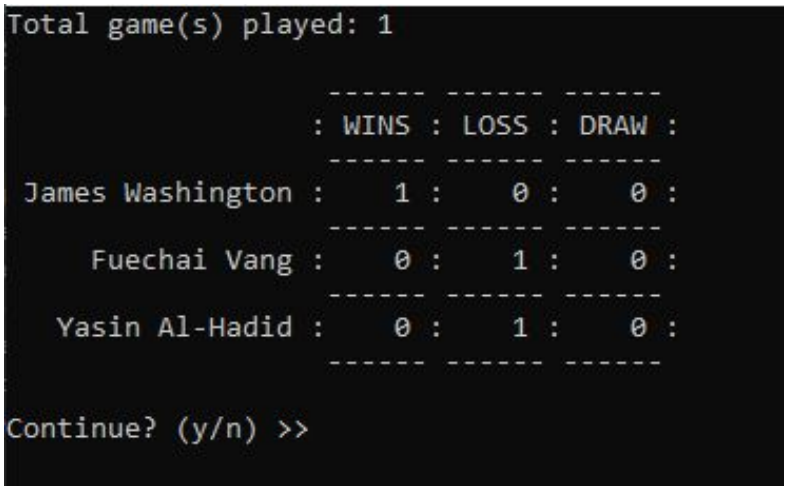


Shows input validation works correctly

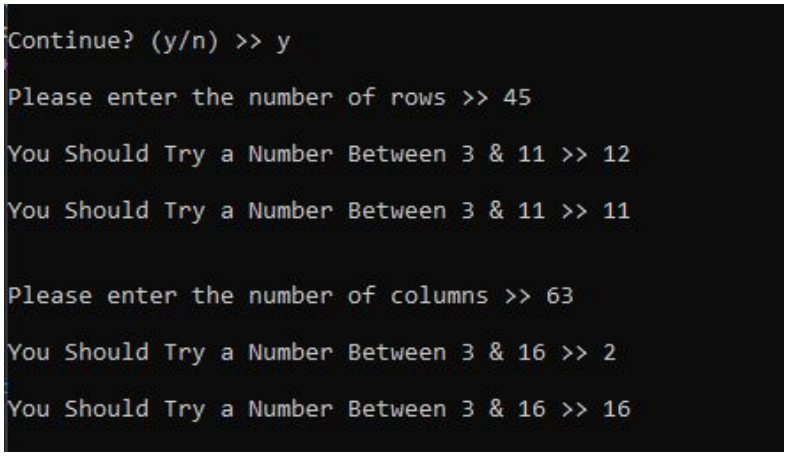


```
James(a), please make your move >> b1


    1   2   3
   --- --- ---
A : a :   : b : A
   --- --- ---
B : a : c :   : B
   --- --- ---
C :   :   :   : C
   --- --- ---
    1   2   3

Fuechai(b), please make your move >> b3


    1   2   3
   --- --- ---
A : a :   : b : A
   --- --- ---
B : a : c : b : B
   --- --- ---
C :   :   :   : C
   --- --- ---
    1   2   3
```

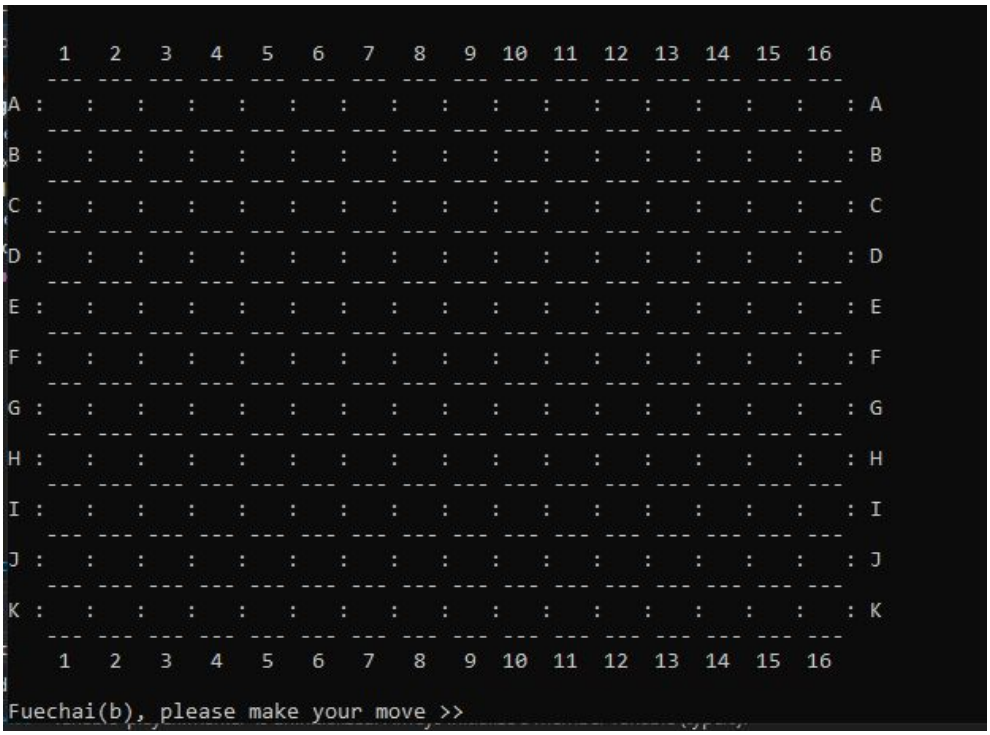
```
Yasin(c), please make your move >> c2


    1   2   3
   --- --- ---
A : a :   : b : A
   --- --- ---
B : a : c : b : B
   --- --- ---
C :   : c :   : C
   --- --- ---
    1   2   3

James(a), please make your move >> c1


James has won the game!!!

    1   2   3
   --- --- ---
A : A :   : b : A
   --- --- ---
B : A : c : b : B
   --- --- ---
C : A : c :   : C
   --- --- ---
    1   2   3
```



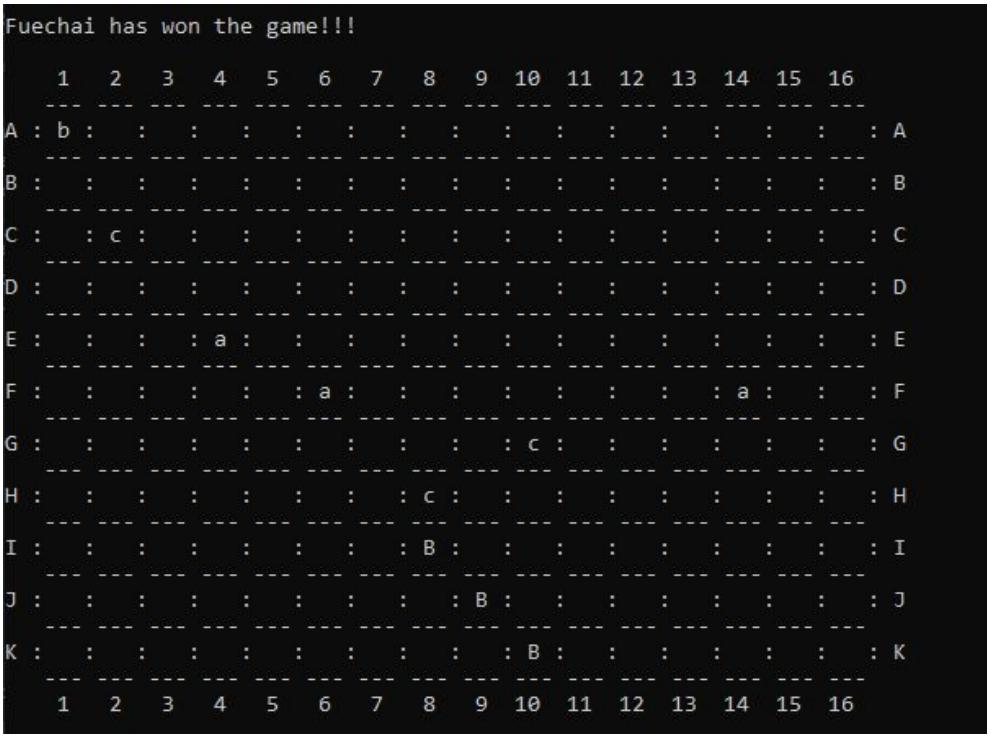
Won the game, shows the winning configuration. Stats are outputted correctly. First game shows the game works properly

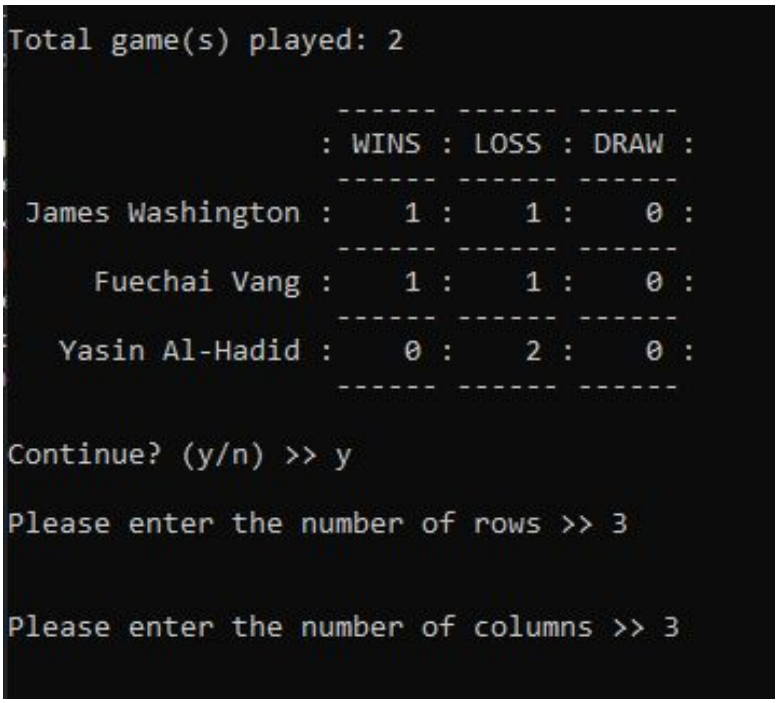


Input validation working properly, also shows the game continues as long as the player requests

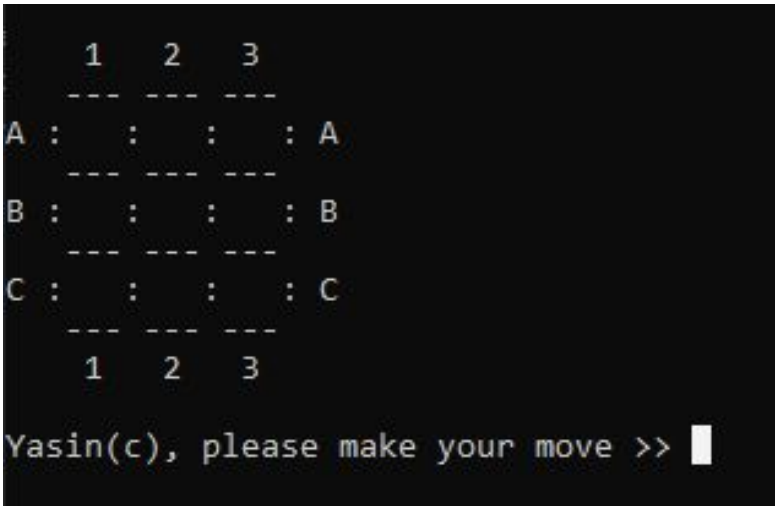


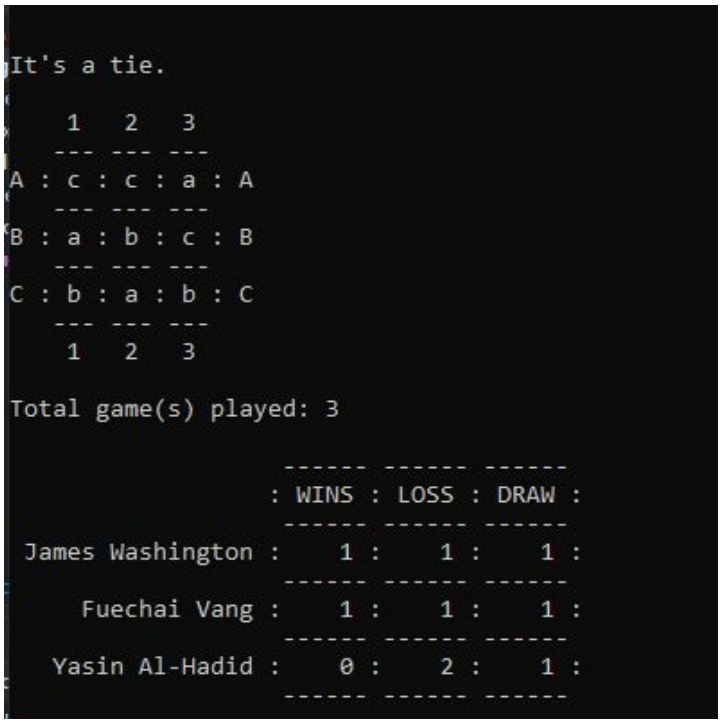
Shows that player after the winner starts and that the board can change every game



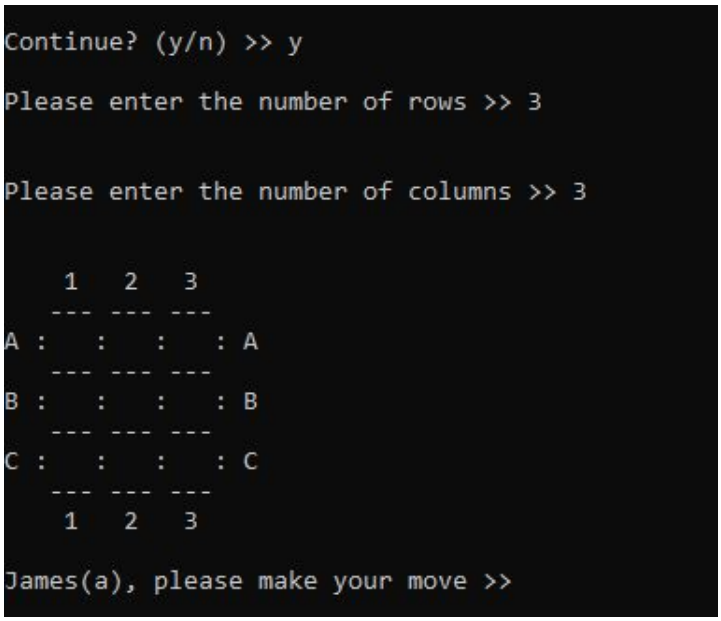


Stats are updated correctly, Player two one so player 3 should start the next game

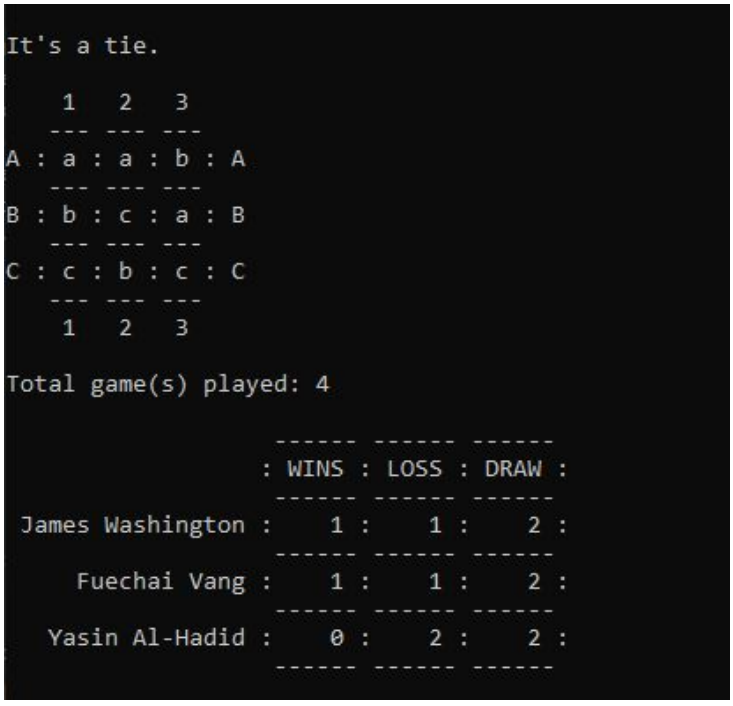




Got a tie to show not only the stats updated correctly but that the person who starts the next game is correct



Player 3 started the previous game so Player 1 starts the next



Played another tie game to show stats are right, and that Player 2 should start the next game since Player 1 started the previous tie game