Homework 6 (due Thursday Mar. 16th, 11:00 AM)

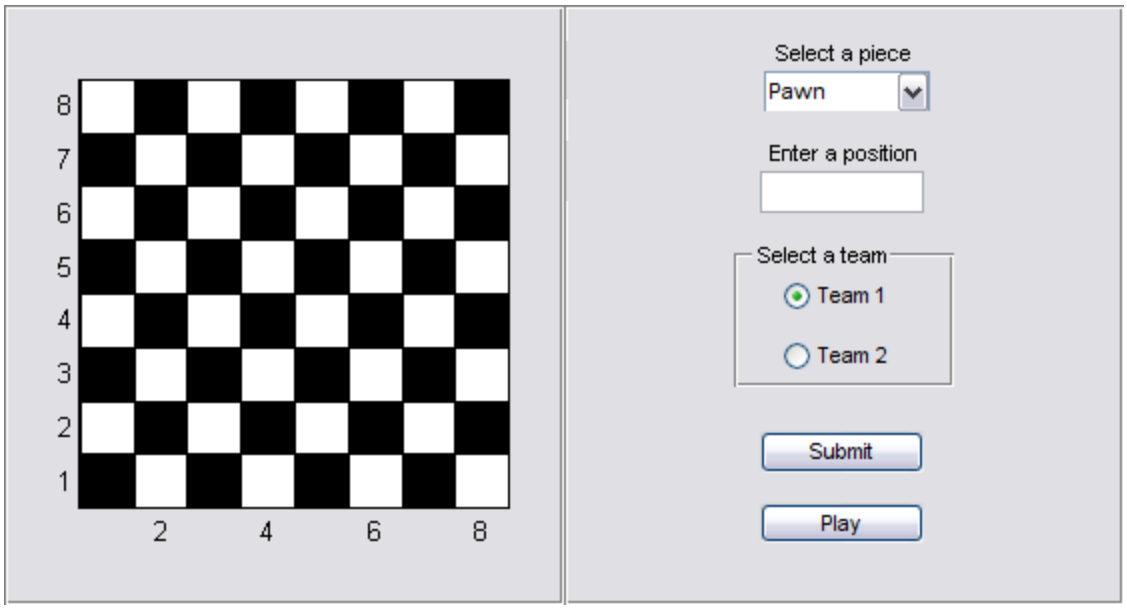
In this homework, you will create a programmatic GUI (don't use GUIDE) that allows you to add chess pieces to the board before you start the game (instead of using the normal chess setup).

For this assignment, you will need to use the code attached.

The display method in **`ChessBoard`** has been changed to take a new input argument: a handle to an axes object (for displaying the board).

A new method has been added to **`ChessGame`**: **`playui`**. This is the method to which you will add code to make your GUI.

Your GUI should look like this:

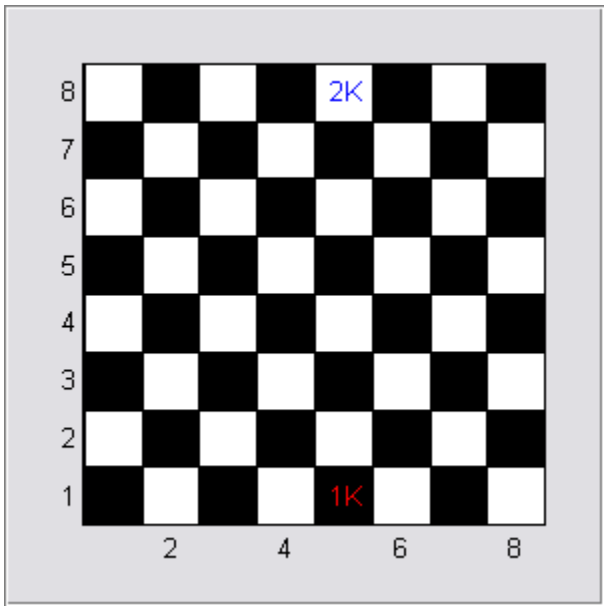


The figure needs to be big enough to contain the UI components shown above. The figure contains two panels, each of 50% width of the figure. The left panel contains an axes object for the board to display on. The right panel contains nine elements. There are two static text labels, one pop-up menu, one edit text box, a radio button group with two radio buttons, and two push buttons. The pop-up menu for the piece should contain the full names of the pieces. The position text box takes a string array with the format "[X, Y]".

Both the Submit and Play buttons require callbacks.

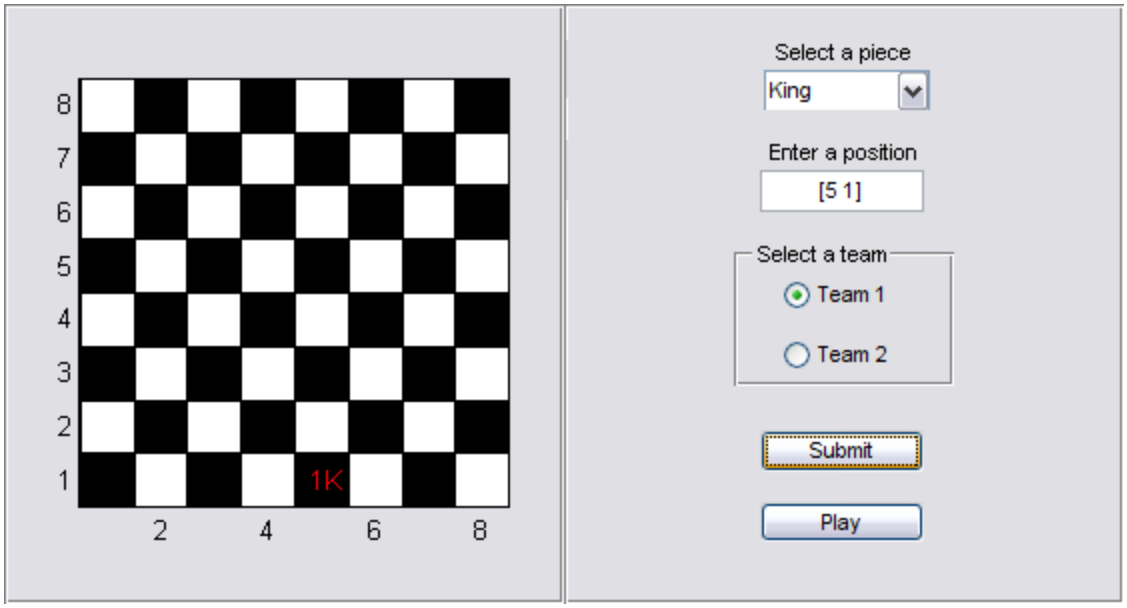
> The Submit button has a callback function named **`submit_callback`**. This is a nested function in **`playui`**. It has already been partially written; you must write the code to create the appropriate piece and get the image of the board on the figure to update.
>
> For the Play button, you must write a callback function named **`play_callback`** that accomplishes two things:

1) The callback should visualize and center the figure's left panel. This will look like the following:



2) The play callback must call the play method.

After hitting the Submit button, your figure should look like this:



As you continue to add pieces, the board will fill up. When you click Play, the board will take up the entire figure and the normal game play will commence. The game should give you an error if the GUI play button is pressed and each team does not have a king. Special Case: If no pieces have been added a Standard Game should be initialized and play begun.

Your GUI does not need to match these pictures exactly. Your UI elements should be evenly spaced and nicely aligned, and the margin around your board should be fairly even, but beyond that they don't need to be placed precisely in the same positions as those shown in the diagrams.

You will need to comment out the initializeBoard call in the ChessGame constructor and it is OK to add properties to ChessGame.

Submit the homework on bCourses. You should create a folder named **`lastname_firstname_hw6`**. Place all of your m-files in this folder and zip it. Please upload this single zip file.

DO NOT USE "eval" ANYWHERE IN YOUR CODE! FOR THIS ASSIGNMENT AND PRETTY MUCH ANYWHERE....EVER

1. Put every type of Piece for each Team on the Board
    1.1. Throw Error if piece already in position
    1.2. Error when pressing Play Button if each team does not have a King
2. Put two kings of opposing team on the Board
    2.1. Play with button
    2.2. Kill one to end game
3. Without clearing or closing anything:
    3.1. Run playui and recover original gui
    3.2. Add two kings of opposing team
    3.3. Play using gui button
    3.4. Kill one to end the game
4. Without clearing or closing anything:
    4.1. Run playui
    4.2. Play using gui button
    4.3. Game should initialize to Standard Game
5. Fun Bonus (no extra points or deductions): Implement submit_callback with exactly one additional "if" statement and no other logic.