Homework 7 (due Thursday Mar. 23th, 11:00 AM)

Please include the resultSave.mat generated after running the autograder in your zip file. In this assignment, you will be filling in the chess game code provided to use events; you will not use code from your previous assignments. It is very similar but some changes have been made to the code you are being given; take some time to review it and make sure you understand the changes. Here is a list of some of the changes:

- Pieces should now be added through **`ChessGame`**'s **`addPiece`** method, instead of directly to the **`ChessBoard`**.
- **`ChessGame`** now has a property for how many teams are involved in the game.
- **`ChessBoard`** now has a property for the size of the board; the piece classes use this to compute their moves.
- **`ChessPiece`** now has a **`Game`** property instead of **`Board`**. This is set by **`ChessGame`** and can only be set once.
- The **`checkmate`** method in **`ChessGame`** and **`removePiece`** in **`ChessBoard`** are now set up to be callbacks.
- Various access settings have been modified.

What needs to be done for this assignment:

- **`ChessPiece`** must have an event called **`Death`** that gets triggered when the **`die`** method is called.
    - In **`ChessBoard`**, the **`addPiece`** method must register a listener for the piece's **`Death`** event, with the **`removePiece`** method as the callback.
    - In **`ChessGame`**, the **`addKing`** method must add to **`KingList`** and register a listener for the king's **`Death`** event (note that a king will have *two* listeners for its **`Death`** event), with the **`checkmake`** method as the callback. The **`addKing`** method must also check to see if the team already has a king and give an error if it does.
- You will implement *queening* (see http://en.wikipedia.org/wiki/Promotion_(chess)). In **`ChessBoard`**, you must add a method called **`pawnToQueen`** that can be a callback function. The object passed to the callback will be a **`Pawn`** object. The **`pawnToQueen`** method must kill the pawn and replace it with a new **`Queen`** object in the same position.
    - **`Pawn`** must have an event called **`ReachedBoardEdge`** that gets trigged when the pawn reaches the edge of the board (the notification must be made in the **`move`** method).
    - In **`ChessGame`**, you must modify the **`addPiece`** method so that if the piece is a pawn, it registers a listener for the pawn's **`ReachedBoardEdge`** event, with the **`pawnToQueen`** method of the game's **`ChessBoard`** as the callback.

Note: You can assume that the user inputs positions like “[1,2]” correctly.

Submit the homework on bCourses. You should create a folder named **`lastname_firstname_hw7`**. Place all of your m-files in this folder and zip it. Please upload this single zip file.