

CINS 465: Assignment 2

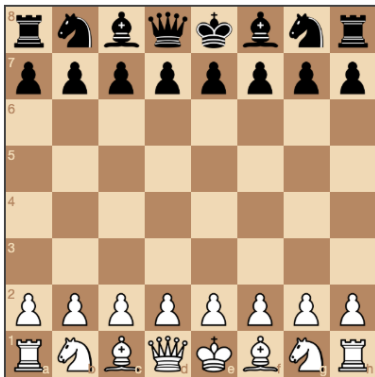
This is an individual assignment. **Do your own work. Do not copy / paste anything from any outside source (other than the bootstrap import tags within your <head> section).** You may brainstorm with others if needed. You may come to office hours to get help if needed.

Objectives

- Practice using basic HTML, CSS, Javascript & bootstrap to create an **interactive** web page containing a chess board with **movable** pieces.

Instructions

- Starting with your solution for Assignment 1 (static web page containing a chessboard):
 - If you defined your CSS in an external file, move it in to the <style></style> section of your HTML page, within the <head></head> section (this makes grading the assignment easier, thank you).
 - Add row and column labels to the chessboard. Use numbers for the row labels, and lowercase letters for the column labels. Render the row numbers in the upper-left corner of each cell in the first column. Render the column numbers in the lower-right corner of each cell in the last row. Here is an example board showing the correct labelling:



Notes:

- You do not need to match these exact colors, continue with whatever you picked for assignment.
 - You do not need to match the exact layout for the row and column labels shown above. Any scheme you like is fine, as long as the rows are clearly labeled with 1-8 and the columns are clearly labelled with a-h.
- Using bootstrap (which is built on top of the JQuery javascript libraries) & custom javascript within the <script></script> section of your HTML page, add a set of interactive controls just below the chessboard that allow any piece to be moved from any source square to any destination square. Use the row and column labels to identify squares. You do not need to prevent illegal moves. If the destination square has a piece on it already, replace that piece with the piece being moved. Clear the piece from the source square. There are many possible ways you might implement this UI. You could have separate text input controls for the source and destination squares. You could use dropdown controls to allow the selection of the source and destination row & column. You could use a single text input control that takes both the source and destination separated by white-space, e.g. "1b 3c" would mean "move the piece at row 1, column b to row 3 column c". There are other possibilities too. Get creative and create a simple interface that you like. You might think of doing a nice drag/drop interface for moving the pieces, which is also possible, but for this assignment we want to get experience using the bootstrap form controls so you must implement the movement via form controls. If you decide to also implement drag/drop moves that is fine too, but not necessary.
 - Add a button below the chess board that resets the pieces to the starting position as shown above.

Hints

- In order to move the pieces around you will need to write javascript code connected to your form controls, to manipulate your chessboard HTML content dynamically, using the DOM (Document Object Model). See https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction for details and examples on the DOM. <continued on next page...>

Requirements

- Do not include any external CSS other than the required bootstrap CSS. Any custom CSS styles you define must be defined within the head section of your single HTML page.
- Do not include any external Javascript files other than the ones required by bootstrap / JQuery. All of your Javascript and CSS must be embedded within the head section of your single HTML page.