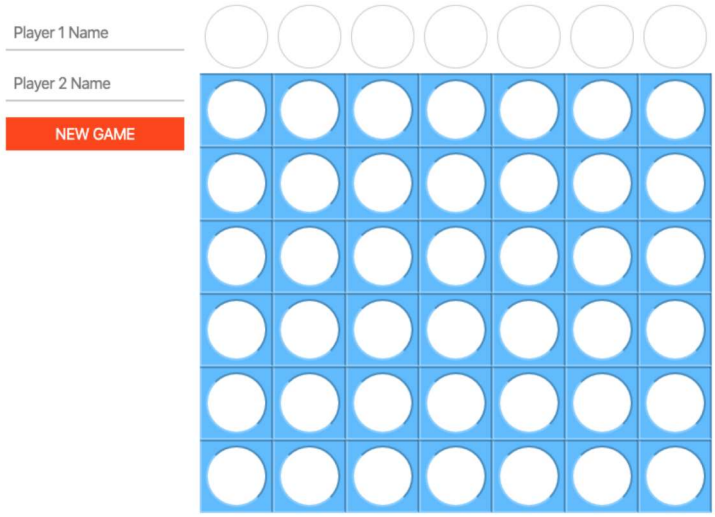
UI

The UI for the game is already created for you in the game. Review it so that you become familiar with how the HTML and CSS work together to make the playable board.

Reviewing is good, but typing is better. Although it may seem strange, you are encouraged to rename **index.html**and **site.html**to **index-complete.html**and **site-complete.html**, respectively. Then, create a new file named**index.html**and reproduce the **index-complete.html**in it to see what it looks like unstyled. Then, create a new file named **site.html**and add the CSS rules one at a time to see how they affect the layout. Play around with them to see how that affects the visualization.

If you're pair programming, try to talk through the structure of the HTML before adding the CSS. Review `display: grid`together. Learn about `transition`. Figure out together how that blue board is situated.



Points of interest

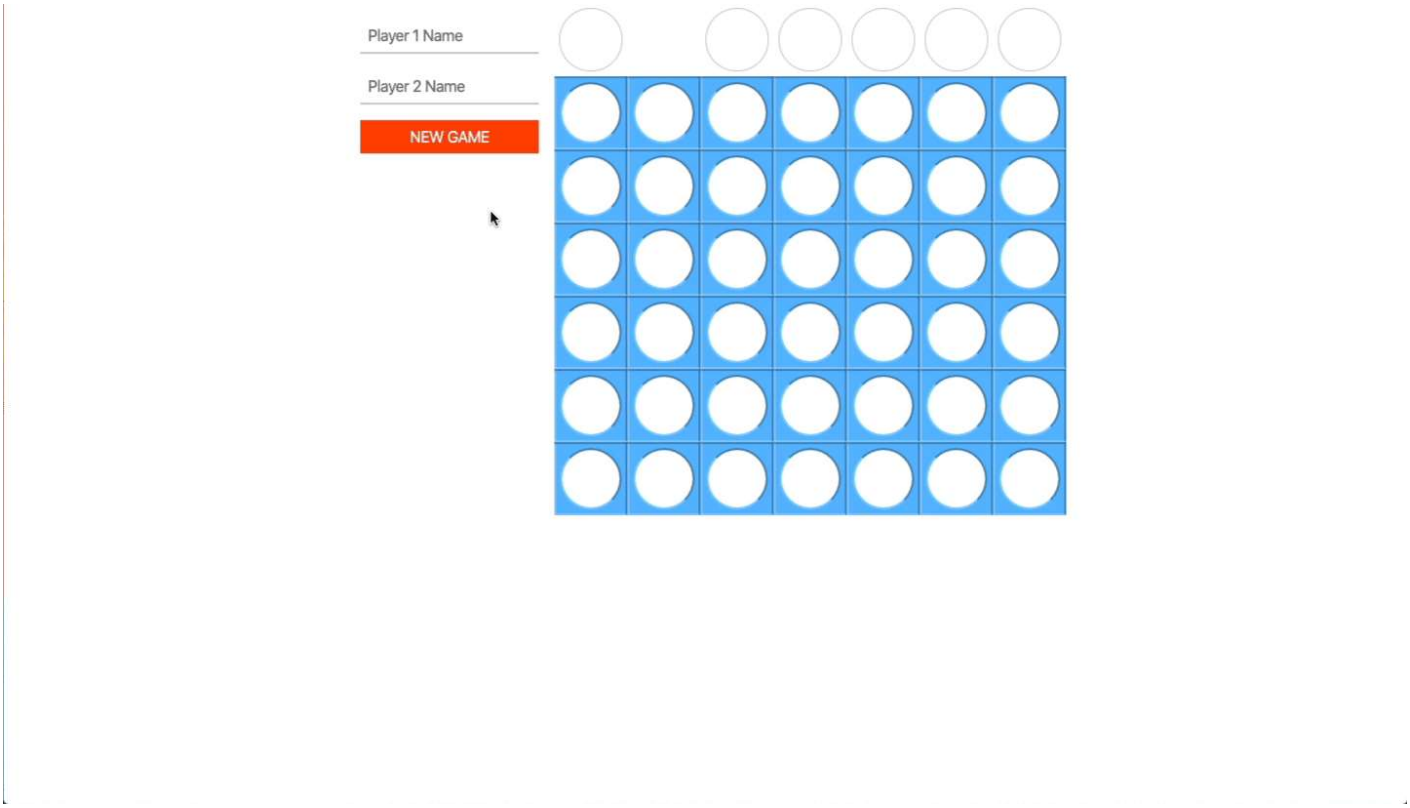
You want the UI to respond to the state of the game in such a way that the players feel as if they're actually playing Connect Four, but with the added features of what a digital human-computer interface can provide. Here are the ways that you can modify the UI to respond to changes in game state.

A column is full

When a column is full, then you will want to indicated to the players that the option to drop a token in that column is no longer valid. You can do this by setting the class "full" on the appropriate column. For example, if you open the **index.html**page and add the class "full" to the `div`with the id of "column-1", it will make the click target disappear and change the cursor to the "Uh uh! You can't do that here!" icon.

The following is a hard-coded example. You will *not*want to do this in your code other than to try it out. You will want to set the "full" class on the column click target programmatically from your JavaScript. Don't forget to undo this change before continuing!

```
<div id="click-targets">
  <!-- Add the "full" class to any of these click targets -->
  <div id="column-0" class="click-target"></div>
  <div id="column-1" class="click-target full"></div>
  <div id="column-2" class="click-target"></div>
  <div id="column-3" class="click-target"></div>
  <div id="column-4" class="click-target"></div>
  <div id="column-5" class="click-target"></div>
  <div id="column-6" class="click-target"></div>
</div>
```



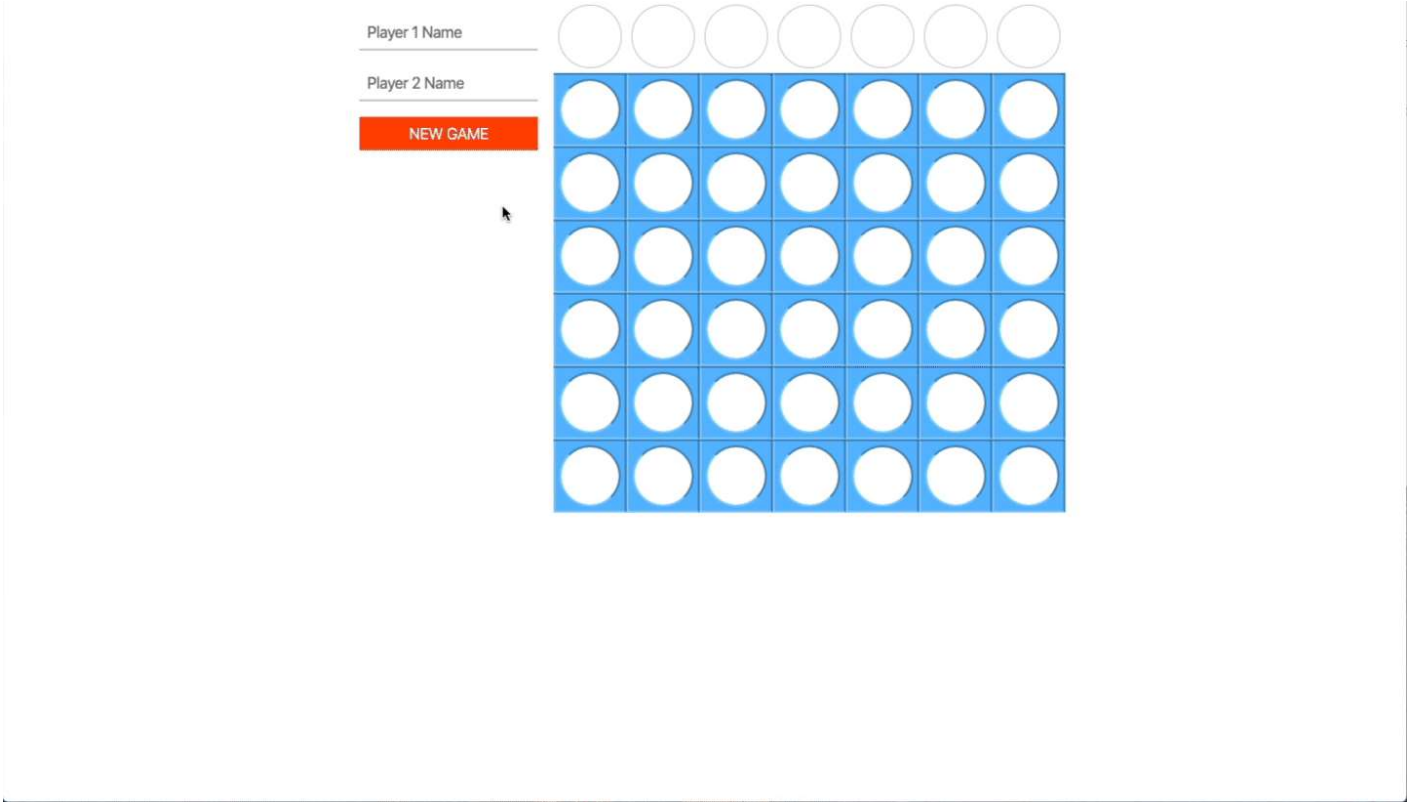
Indicating whose turn it is

If it's player one's turn, then you will want to give an indication of that by having the click targets change color to black when you hover over them. If it's player two's turn, you'll want them to turn red. You can specify this by adding the "black" or "red" class to the div with the "click-targets" id.

The following is a hard-coded example. You will *not*want to do this in your code other than to try it out. You will want to set the "full" class on the column click target programmatically from your JavaScript. Don't forget to undo this change before continuing!

```
<!-- Add the "black" or "red" class to the "click-targets" div -->
<div id="click-targets" class="black">
  <div id="column-0" class="click-target"></div>
  <div id="column-1" class="click-target"></div>
  <div id="column-2" class="click-target"></div>
  <div id="column-3" class="click-target"></div>
```

```
    <div id="column-4" class="click-target"></div>
    <div id="column-5" class="click-target"></div>
    <div id="column-6" class="click-target"></div>
</div>
```



Putting a token in a square

When you want to place a token in a square, select the square that you want to place it in, create a `div`, set the class to have both "token" and the color that you want the token to be ("black" or "red"), and then make the `div`the child element of that square.

The squares all have id values of the format `square-«row»-«column»`, so it should be ok for you to find the correct square. Note that the values for *row*and *column*are zero-based just like a JavaScript array.

To get a sense of what it looks like, modify the last two squares in the HTML file to show a black token and a red token. The following is a hard-coded

example. You will *not*want to do this in your code other than to try it out. You will want to set the "full" class on the column click target programmatically from your JavaScript. Don't forget to undo this change before continuing!

```
<div id="square-5-5" class="token-square">
  <div class="token black"></div>
</div>
<div id="square-5-6" class="token-square">
  <div class="token red"></div>
</div>
```