



## Front End Technologies Week 5 Coding Assignment

Lisa Smith — 2021-10-05-fesd-nashua

**Points possible:** 70

Category	Criteria	% of Grade
<b>Functionality</b>	Does the code work?	25
<b>Organization</b>	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
<b>Creativity</b>	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
<b>Completeness</b>	All requirements of the assignment are complete.	25

**Instructions:** In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

### Coding Steps:

1. Using any of the tools you've worked with so far, create a game of tic-tac-toe.
  - a. A heading should say whether it is X's or O's turn and change with each move made.
  - b. Create a tic-tac-toe grid using your HTML element of choice. When a cell in the grid is clicked, an X or O should appear in that spot depending on whose turn it is.



- c. A button should be available to clear the grid and restart the game.
- d. When a player has won, or the board is full and the game results in a draw, a Bootstrap alert or similar Bootstrap component should appear across the screen announcing the winner.

## Screenshots of Code:

### main.css:

```
FE-Week5-HW > # main.css > body
1  body {
2    background-color: #lightgray;
3    border: 2px solid blueviolet;
4    text-align: center;
5    font-family: 'Merriweather', serif;
6  }
7
8  alert {
9    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
10 }
11
12 table {
13   border-color: blueviolet;
14   border-style: inset;
15   border-width: 1px;
16 }
17
18 col-sm-6 {
19   margin: 2px;
20   padding: 2px;
21 }
22
23
24 #tic-tac-toe-grid {
25   background-color: #rgb(13, 170, 233);
26   border-color: blueviolet;
27   border-style: inset;
28   border: 2px solid #105, 13, 233;
29   box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
30 }
31
32 h1 {
33   color: #rgb(13, 170, 233);
34   text-decoration-color: blueviolet;
35   text-shadow: 2px 2px 5px blueviolet;
36   border-style: inset;
37   border-width: 1px;
38   border-color: blueviolet;
39   background-color: #linen;
40   box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
41 }
42
43 h5,h6 {
44   text-decoration-color: pink;
45   color: blueviolet;
46   background-color: #linen;
47   border-color: blueviolet;
48   border-style: inset;
49   border-width: 1px;
50   box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
51 }
52 button {
53   border: 2px solid #41, 41, 56;
54   border-color: #105, 13, 233;
55   border-style: inset;
56   border-width: 1px;
57   box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
58 }
59
60
61 .button-1 {
62   background-color: #lightgray;
63   border-radius: 35px;
64   padding: 2em;
65   margin: 0.5em;
66   color: #rgb(105, 13, 233);
67   font-size: 16px;
68   text-shadow: 2px 2px 5px gray;
69   font-weight: 5;
70 }
71
72 .button-1:hover {
73   background-color: #white;
74   color: #rgb(105, 13, 233);
75   border-color: #rgb(105, 13, 233);
76   box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
77 }
78
79 .button-2 {
80   background-color: #rgb(13, 170, 233);
81   color: #rgb(105, 13, 233);
82   font-size: 24px;
83   text-shadow: 2px 2px 5px gray;
84   margin: 1em;
85   padding: 1em;
86   border-radius: 50px;
87 }
88
89 .button-2:hover {
90   background-color: #white;
91   color: #rgb(105, 13, 233);
92   border-color: #rgb(105, 13, 233);
93   box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
94 }
95
96
```



# PROMINEO TECH

## newindex.html

```
main.css M  newIndex.html u  JS newIndex.js u
-Week5-HW > newindex.html > ...
1  |<!DOCTYPE html>
2  |<html lang="en">
3  |<head>
4  |    <meta charset="UTF-8">
5  |    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |    <title>Tic-Tac-Toe Week5 HW</title>
7  |    <link rel="preconnect" href="https://fonts.googleapis.com">
8  |    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
9  |    <link href="https://fonts.googleapis.com/css?family=Dancing+Script&display=swap" rel="stylesheet">
0  |    <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
1  |    <link rel="stylesheet" href="main.css">
2  |</head>
3  |
4  |
5  |<body class="container-fluid m-5 mx-auto center rounded">
6  |    <div id="game-page">
7  |        <!-- The assignment is to create a Tic-Tac-Toe Game
8  |            1. develop 3 * 3 grid layout
9  |            2. apply some CSS effects on the same.
0  |                NOTE: Show text showing whose turn it is...
1  |                NOTE: Define a button to reset the game. -->
2  |        <div class="col-sm-4"></div>
3  |
4  |        <div class="col-sm-4 container-fluid center mx-auto rounded">
5  |            <h1 style="color: #rgb(8, 160, 220)">Tic-Tac-Toe Game</h1>
6  |        </div> <!-- end of Title div-->
7  |
8  |        <div class="container-fluid center border-dark rounded">
9  |            <div class="col-sm-4"></div>
0  |            <div class="col-sm-4 container-fluid center">
1  |                <h5 id="turn-header" style="color: #rgb(105, 13, 233)">Game starting -- Turn #1: X's turn</h5>
2  |                <h6 id="tally" style="color: #rgb(8, 160, 220)">SCORE: Team X: 0 vs. Team O: 0</h6>
3  |                <br>
4  |            </div> <!-- end of X's or O's Turn div-->
5  |            <div class="col-sm-4"></div>
6  |
7  |            <div class="container-fluid rounded">
8  |                <div class="row">
9  |                    <div class="col-sm-4">
0  |
1  |                        <div class="col-sm-4">
2  |                            <table id="tic-tac-toe-grid" class="container-fluid center border rounded">
3  |                                <tr>
4  |                                    <td>
5  |                                        <button id="btn-p-1" class="button-1">1</button>
6  |                                    </td>
7  |                                    <td>
8  |                                        <button id="btn-p-2" class="button-1">2</button>
9  |                                    </td>
0  |                                    <td>
1  |                                        <button id="btn-p-3" class="button-1">3</button>
2  |                                    </td>
3  |                                </tr>
4  |                                <tr>
5  |                                    <td>
6  |                                        <button id="btn-p-4" class="button-1">4</button>
7  |                                    </td>
8  |                                    <td>
9  |                                        <button id="btn-p-5" class="button-1">5</button>
0  |                                    </td>
1  |                                    <td>
2  |                                        <button id="btn-p-6" class="button-1">6</button>
3  |                                    </td>
4  |                                </tr>
5  |                                <tr>
6  |                                    <td>
7  |                                        <button id="btn-p-7" class="button-1">7</button>
8  |                                    </td>
9  |                                    <td>
0  |                                        <button id="btn-p-8" class="button-1">8</button>
1  |                                    </td>
2  |                                    <td>
3  |                                        <button id="btn-p-9" class="button-1">9</button>
4  |                                    </td>
5  |                                </tr>
6  |                            </table>
7  |                        </div> <!-- end of table div-->
8  |
9  |                    <div class="col-sm-4"></div>
0  |                </div> <!-- end of row div -->
1  |
2  |            <br>
3  |            <button id="reset-game" type="button" class="button-2 rounded">Reset Game</button>
4  |        </div> <!-- end of Game Page -->
5  |
6  |        <!-- Need to include jquery.js first, and then my .js file afterwards-->
7  |        <script src="node_modules/jquery/dist/jquery.js"></script>
8  |        <script src="newIndex.js"></script>
9  |        <script src="node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
0  |
1  |
2  |</body>
3  |</html>
```



# PROMINEO TECH

## newindex.js:

```
main.css M  newindex.html u  newindex.js u
FE-Week5-HW > JS newindex.js > ⓘ callback > checkForWinner
1  // Coding Steps:
2  //
3  // Using any of the tools you've worked with so far,
4  // Create a game of tic-tac-toe.
5  // A heading should say whether it is X's or O's turn and change with
6  // each turn.
7  // Create a tic-tac-toe grid using your HTML element of choice. When a
8  // cell in the grid is clicked, an X or O should appear in
9  // that spot depending on whose turn it is.
10 // A button should be available to clear the grid and restart the game.
11 // When a player has won, or the board is full and the game results in
12 // a draw, a Bootstrap alert or similar Bootstrap
13 // component should appear across the screen announcing the
14 // winner.
15 //
16 // DONE: 1. HD Title for the Game
17 // DONE: 2. Develop 3x3 grid layout -- table with 9 buttons
18 // DONE: 3. Apply some CSS effects on the same
19 // HD -- Show text showing whose turn it is.
20 // DONE: 4. Define a button to reset the game,
21 // * redraw the table
22 // * alert box which disappears and goes away
23 // * reset winning strategy
24 //   -- possible values: "NO WINNER", "X WINNER", "O WINNER"
25 //   NOTE: index will be the position of the button
26 //   Winning combinations:
27 //   Horizontal:
28 //     [1,2,3] OR [4,5,6] OR [7,8,9]
29 //   Vertical:
30 //     [1,4,7] OR [2,5,8] OR [3,6,9]
31 //   Diagonal:
32 //     [1,5,9] OR [3,5,7]
33 //
34
35 $function() {
36   let gameId = 0; // always increments, never resets
37   let gameOver = false; // boolean, used to end the game
38   let xWin = 0;
39   let oWin = 0;
40
41   // increments during a game, reset when game is reset
42   let numOfTurn = 1; // turn counter
43
44   // play position array
45   let gameStatus = ['NO WINNER', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '];
46
47   onClick('reset-game', () => {
48     // clear the winning strategy
49     gameStatus = ['NO WINNER', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '];
50     // reset turn counter 'numOfTurn' to 1
51     numOfTurn = 1;
52     gameover = false;
53     // clear the game and reset
54     drawOM();
55     console.log('Tic-Tac-Toe Game Reset!');
56     $( "#turn-header").text("Game restarted — Turn #1: X's turn");
57     $( "#tally").text('SCORE: Team X: ' + xWin + ' vs. Team O: ' + oWin);
58   });
59
60   function onClickId(action) {
61     return $( "#$id").on('click',action);
62   }
63
64   function drawOM() {
65     // clear the view, and re"draw" the table
66     let tictactoeDiv = document.getElementById('game-page');
67     //let tictactoeDiv = $('#game-page');
68     for (let index=1; index<9; index++) {
69       //reset each button to original number value
70       $( '##0'+index).text( $( '##0'+index).attr('id') );
71     }
72     if (gameId > 0) {
73       $( "#restart-game").show();
74     } else {
75       let linebreak = document.createElement('br');
76       tictactoeDiv.appendChild(linebreak);
77       linebreak = document.createElement('br');
78       tictactoeDiv.appendChild(linebreak);
79       let alert = document.createElement('alert');
80       alert.setAttribute('class', 'alert alert-info');
81       alert.setAttribute('id', 'alert');
82       alert.setAttribute('id', 'restart-game');
83       alert.innerHTML = 'Tic-Tac-Toe Game has been restarted!';
84       tictactoeDiv.appendChild(alert);
85     }
86     gameId++;
87     // Use setTimeout() to hide it again after 5 seconds
88     setTimeout(() => $( "#restart-game").hide(), 2000);
89   } // end of function drawOM()
90
91 // Set up an EventListener for each bit in the game.
92 for (let btnIndex=0; btnIndex<9; btnIndex) {
93   onClick('#btn-p'+btnIndex), (evt) => {
94     // Change button to display players symbol
95     if (!gameover) {
96       if (gameStatus[btnIndex] == '') {
97         console.log('Turn #' + numOfTurn);
98         if (numOfTurn % 2) {
99           evt.target.innerText = 'X';
100           gameStatus[btnIndex] = 'X';
101           console.log('Turn X's turn');
102           $( "#turn-header").text("Turn #"+numOfTurn+1): O's turn");
103         } else {
104           // O's turn
105           evt.target.innerText = 'O';
106           gameStatus[btnIndex] = 'O';
107           console.log('Turn O's turn');
108           $( "#turn-header").text("Turn #"+numOfTurn+1): X's turn");
109         }
110       }
111     }
112     // *****
113     // NOTE: Starting with turn #5, after each turn,
114     // Check for a WINNER!
115     //
116     // When the game is a draw, OR there is a winner --
117     // Bootstrap alert or message across the screen
118     // to announce the winner!
119     // *****
120     if (numOfTurn > 5) {
121       let winner = checkForWinner();
122       console.log(winner);
123       if (winner != '') {
124         $( "#turn-header").text('Game over! ' + winner);
125         console.log('Game over! ' + winner);
126         // prompt/alert
127         if (gameId == 0) {
128           $( "#game-over").show();
129         }
130         let tictactoeDiv = document.getElementById('game-page');
131         $( "#game-over").append(tictactoeDiv);
132         tictactoeDiv.appendChild(linebreak);
133         let gameOverAlert = document.createElement('alert');
134         gameOverAlert.setAttribute('class', 'alert alert-info');
135         gameOverAlert.setAttribute('id', 'alert');
136         gameOverAlert.setAttribute('id', 'game-over');
137         gameOverAlert.innerHTML = 'Tic-Tac-Toe Game has ended!';
138         tictactoeDiv.appendChild(gameOverAlert);
139       }
140       // Use setTimeout() to hide it again after 3 seconds
141       setTimeout(() => $( "#game-over").hide(), 3000);
142     }
143   }
144 }
145 } else {
146   console.log(INVALID CHOICE);
147 }
148 else {
149   numOfTurn++;
150   console.log('Game Over');
151 }
152 });
153
154 // Winning combinations for Tic-Tac-Toe:
155 //   Horizontal:
156 //     [1,2,3] OR [4,5,6] OR [7,8,9]
157 //   Vertical:
158 //     [1,4,7] OR [2,5,8] OR [3,6,9]
159 //   Diagonal:
160 //     [1,5,9] OR [3,5,7]
```



# PROMINEO TECH

index.js cont'd:

```
163
164     function checkForWinner() {
165         let winner = '';
166         gameOver = false;
167         teamWin = '';
168         console.log("Checking gameStatus: ");
169         console.log(gameStatus);
170         if ((gameStatus[1] === gameStatus[2]) && (gameStatus[1] === gameStatus[3])) {
171             // If (gameStatus[1] == gameStatus[2]) && (gameStatus[1] == gameStatus[3])
172             // Then, winner is ${gameStatus[1]}';
173             winner = ${gameStatus[1]}';
174             teamWin = gameStatus[1];
175         } else if ((gameStatus[4] === gameStatus[5]) && (gameStatus[4] === gameStatus[6])) {
176             // If (gameStatus[4] == gameStatus[5]) && (gameStatus[4] == gameStatus[6])
177             // Then, winner is ${gameStatus[4]}';
178             winner = ${gameStatus[4]}';
179             teamWin = gameStatus[4];
180         } else if ((gameStatus[7] === gameStatus[8]) && (gameStatus[7] === gameStatus[9])) {
181             // If (gameStatus[7] == gameStatus[8]) && (gameStatus[7] == gameStatus[9])
182             // Then, winner is ${gameStatus[7]}';
183             winner = ${gameStatus[7]}';
184             teamWin = gameStatus[7];
185         } else if ((gameStatus[1] === gameStatus[4]) && (gameStatus[1] === gameStatus[7])) {
186             // If (gameStatus[1] == gameStatus[4]) && (gameStatus[1] == gameStatus[7])
187             // Then, winner is ${gameStatus[1]}';
188             winner = ${gameStatus[1]}';
189             teamWin = gameStatus[1];
190         } else if ((gameStatus[3] === gameStatus[6]) && (gameStatus[3] === gameStatus[9])) {
191             // If (gameStatus[3] == gameStatus[6]) && (gameStatus[3] == gameStatus[9])
192             // Then, winner is ${gameStatus[3]}';
193             winner = ${gameStatus[3]}';
194             teamWin = gameStatus[3];
195         } else if ((gameStatus[1] === gameStatus[5]) && (gameStatus[1] === gameStatus[9])) {
196             // If (gameStatus[1] == gameStatus[5]) && (gameStatus[1] == gameStatus[9])
197             // Then, winner is ${gameStatus[1]}';
198             winner = ${gameStatus[1]}';
199             teamWin = gameStatus[1];
200         } else if ((gameStatus[2] === gameStatus[6]) && (gameStatus[2] === gameStatus[8])) {
201             // If (gameStatus[2] == gameStatus[6]) && (gameStatus[2] == gameStatus[8])
202             // Then, winner is ${gameStatus[2]}';
203             winner = ${gameStatus[2]}';
204             teamWin = gameStatus[2];
205         }
206         if (teamWin === 'X') oWins++;
207         if (teamWin === 'O') xWins++;
208         $(`#italy`).text(`SCORE: Team X: ${xWins} vs. Team O: ${oWins}`);
209     } // end of function checkForWinner()
210
211 });
212 );
```

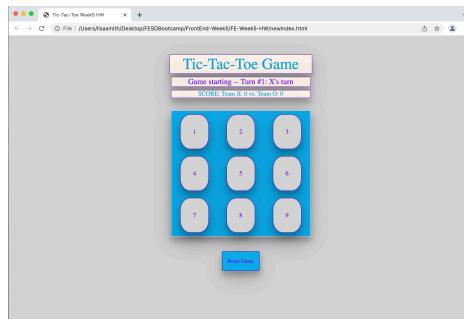


# PROMINEO TECH

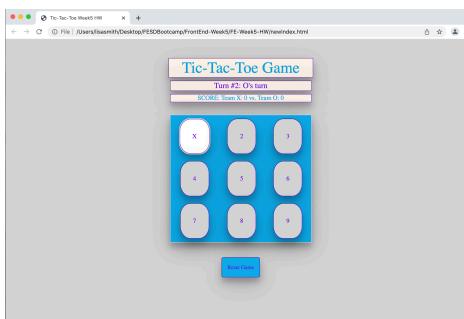
## Screenshots of Running Application:

### Game #2

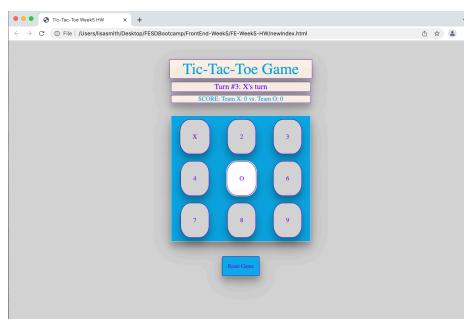
**Start of Game =>**



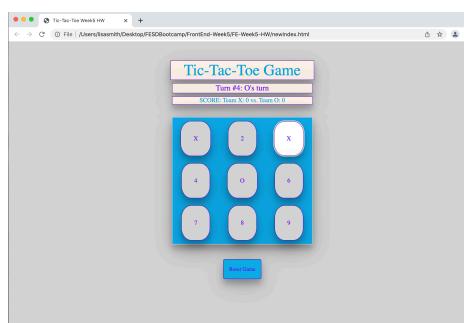
**Turn 1 has happened =>**  
**(Notice X in location 1)**



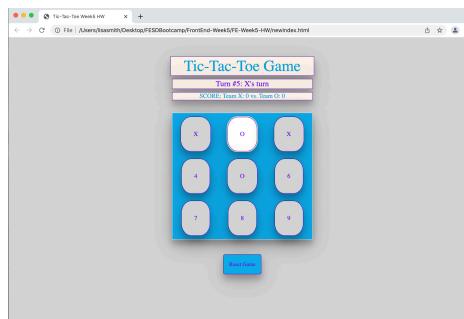
**Turn 2 has happened =>**  
**(Notice O in location 5)**



**Turn 3 has happened =>**  
**(Notice X in location 3)**



**Turn 4 has happened =>**  
**(Notice O in location 2)**

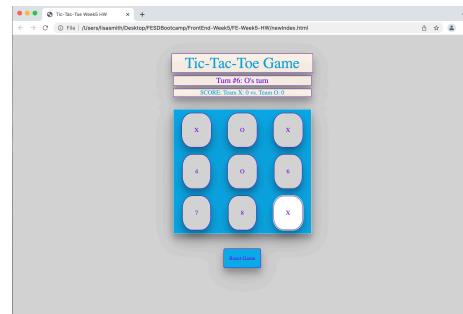




# PROMINEO TECH

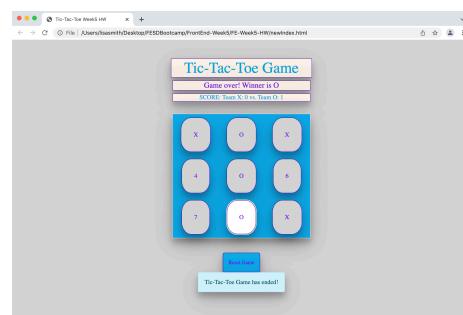
**Turn 5 has happened =>**

**(Notice X in location 9)**



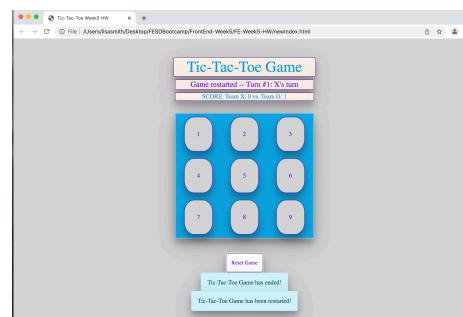
**Turn 6 wins the game for O =>**

**(Notice alert, and change of headers)**

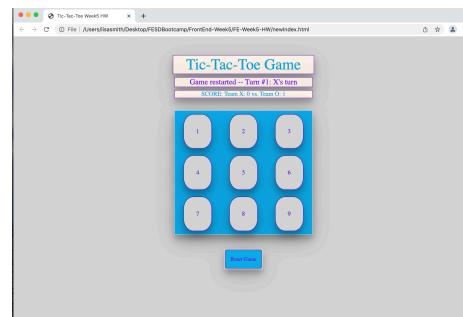


**Reset Game pushed =>**

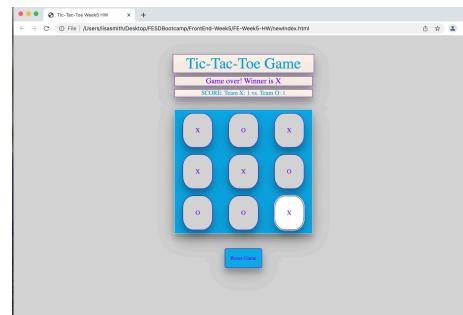
**(Notice second alert)**



**Alerts disappear after some time =>**



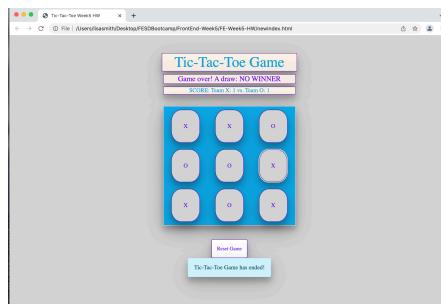
**Game #2: X wins this time! =>**





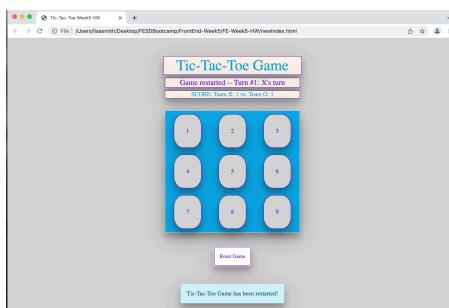
# PROMINEO TECH

**Game #3: It's a draw! =>**

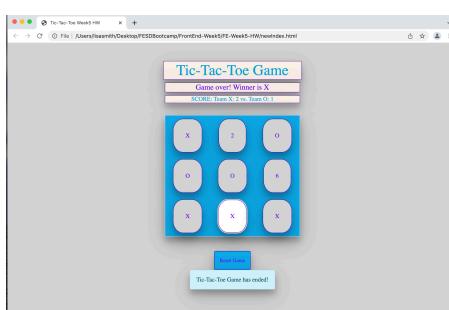


**Waited to Reset Game =>**

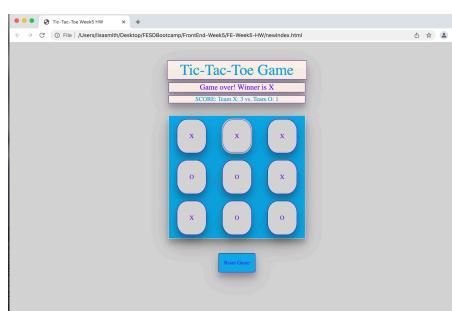
**(Notice no change in score!)**



**Game #4: X wins! =>**



**Game #5: X wins again! =>**



**URL to GitHub Repository:** <https://github.com/sw-dev-lisa-s-nh/FrontEnd-Week5>