



Front End Technologies Week 5 Coding Assignment

Lisa Smith — 2021-10-05-fesd-nashua

Points possible: 70

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



PROMINEO TECH

- 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:

```
JS index.js M  index.html M  # main.css x
FE-Week5-HW > # main.css > ...
1  body {
2      background-color: lightgray;
3      text-align: center;
4      font-family: 'Merriweather', serif;
5      border: 1px solid black;
6      border-radius: 50%;
7      margin: 2em;
8      padding: 2em;
9  }
10
11  div {
12      border: 1px solid black;
13  }
14
15  #tic-tac-toe-grid {
16      background-color: #131700;
17  }
18
19  h1 {
20      color: #131700;
21      background-color: black;
22  }
23
24  h3 {
25      text-decoration-color: pink;
26      color: blueviolet;
27      padding: 0.5em;
28      margin: 0.5em;
29  }
30
31  button {
32      border: 3px solid #000080;
33  }
34
35  .button-1 {
36      border-color: #000080;
37      background-color: lightgray;
38      border-radius: 35%;
39      padding: 2em;
40      margin: 0.5em;
41      color: #800080;
42      font-size: 18px;
43      text-shadow: gray;
44      font-weight: 5;
45  }
46
47  .button-1:hover {
48      background-color: white;
49      color: #800080;
50      box-shadow: 0 12px 16px 0 #000080, 0 17px 50px 0 #000080;
51  }
52
53  .button-2 {
54      border-color: #000080;
55      background-color: #131700;
56      color: #800080;
57      font-size: 16px;
58      text-shadow: gray;
59      margin: 1em;
60      padding: 1em;
61      border-radius: 50%;
62  }
63
64  .button-2:hover {
65      background-color: white;
66      color: #800080;
67      box-shadow: 0 12px 16px 0 #000080, 0 17px 50px 0 #000080;
68  }
69
70
```



PROMINEO TECH

index.html

```
JS index.js M    index.html M x    # main.css
FE-Week5-HW > index.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>FET Week 5 Coding Assignment</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/css2?family=Dancing+Script&display=swap" rel="stylesheet">
10   <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
11   <link rel="stylesheet" href="main.css">
12 </head>
13
14
15 <body class="container-fluid m-5 mx-auto center rounded">
16   <div id="game-page">
17     <!-- The assignment is to create a Tic-Tac-Toe Game
18     1. develop 3 * 3 grid layout
19     2. apply some CSS effects on the same.
20     NOTE: Show text showing whose turn it is...
21     NOTE: Define a button to reset the game. -->
22     <div class="container-fluid center mx-auto rounded">
23       <h1 style="color: rgb(13, 170, 233)">Tic-Tac-Toe Game</h1>
24     </div> <!-- end of Title div-->
25
26     <div class="container-fluid center border-dark rounded">
27       <div class="container-fluid center">
28         <h3 id="turn" style="color: rgb(105, 13, 233)">Game starting: X's turn</h3>
29       </div> <!-- end of X's or O's Turn div-->
30
31       <div class="container-fluid rounded">
32         <div class="row">
33           <div class="col-sm-4 "></div>
34
35           <div class="col-sm-4 ">
36             <table id="tic-tac-toe-grid" class="container-fluid center border border-dark rounded">
37               <tr>
38                 <td>
39                   <button id="btn-p-1" class="button-1">1</button>
40                 </td>
41                 <td>
42                   <button id="btn-p-2" class="button-1">2</button>
43                 </td>
44                 <td>
45                   <button id="btn-p-3" class="button-1">3</button>
46                 </td>
47               </tr>
48               <tr>
49                 <td>
50                   <button id="btn-p-4" class="button-1">4</button>
51                 </td>
52                 <td>
53                   <button id="btn-p-5" class="button-1">5</button>
54                 </td>
55                 <td>
56                   <button id="btn-p-6" class="button-1">6</button>
57                 </td>
58               </tr>
59               <tr>
60                 <td>
61                   <button id="btn-p-7" class="button-1">7</button>
62                 </td>
63                 <td>
64                   <button id="btn-p-8" class="button-1">8</button>
65                 </td>
66                 <td>
67                   <button id="btn-p-9" class="button-1">9</button>
68                 </td>
69               </tr>
70             </table>
71           </div> <!-- end of table div-->
72
73         </div> <!-- end of row div -->
74       </div>
75     </div> <!-- end of Tic-Tac-Toe Grid div-->
76
77     <br><br>
78     <button id="reset-game" type="button" class="button-2 rounded">Reset Game</button>
79   </div> <!-- end of Game Page -->
80
81   <!-- Need to include jquery.js first, and then my .js file afterwards-->
82   <script src="node_modules/jquery/dist/jquery.js"></script>
83   <script src="index.js"></script>
84   <script src="node_modules/bootstrap/dist/js/bootstrap.bundle.js"></script>
85 </body>
86 </html>
```



PROMINEO TECH

index.js:

```
JS index.js M X  index.html M  # main.css
FE-Week5-HW > JS index.js > ...
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
17 // DONE: 1. H1 Title for the Game
18 // DONE: 2. Develop 3 * 3 grid layout --- table with 9 buttons
19 // DONE: 3. Apply some CSS effects on the same.
20 // HB --- Show text showing whose turn it is.
21 // DONE:
22 // 4. Define a button to reset the game.
23 // * redraw the table
24 // * alert box which appears and goes away
25 // * reset winning strategy
26 // game_status['NO WINNER',' ',' ',' ',' ',' ',' ',' ',' ']
27 // NOTE: index will be the position of the button
28 // Winning combinations:
29 // Horizontal:
30 // [1,2,3]
31 // [4,5,6]
32 // [7,8,9]
33 // Vertical:
34 // [1,4,7]
35 // [2,5,8]
36 // [3,6,9]
37 // Diagonal:
38 // [1,5,9]
39 // [3,5,7]
40
41
42
43
44 let gameId = 0; // always increments, never resets
45 let gameOver = false; // boolean, used to end the game
46
47 // increments during a game, reset when game is reset
48 let numOfTurn = 1; //turn counter
49
50 // play position array
51 let gameStatus = ['NO WINNER',' ',' ',' ',' ',' ',' ',' ',' '];
52
53 onClick('reset-game', () => {
54   let headerText = document.getElementById('turn');
55   // let winAlert = document.getElementById('game-over');
56   // winAlert.remove();
57   // clear the winning strategy
58   gameStatus = ['NO WINNER',' ',' ',' ',' ',' ',' ',' ',' '];
59   // reset turn counter 'numOfTurn' to 1
60   numOfTurn = 1;
61   gameOver = false;
62   // clear the game and reset
63   drawDOM();
64   console.log('Tic-Tac-Toe Game Reset!');
65   headerText.innerHTML = "Game restarted! X's turn";
66
67 });
68
69 function onClick(id, action) {
70   let element = document.getElementById(id);
71   element.addEventListener('click', action);
72   return element;
73 }
74
75 function drawDOM() {
76   // clear the view, and re"draw" the table
77   let tictactoeDiv = document.getElementById('game-page');
78   for (let index=1; index<=9; index++) {
79     //reset each button to original value
80     document.getElementById('btn-p-${index}').innerHTML=index;
81   }
82   if (gameId > 0) {
83     $('#restart-game').show();
84   } else {
85     let lineBreak = document.createElement('br');
86     tictactoeDiv.appendChild(lineBreak);
87     lineBreak = document.createElement('br');
88     tictactoeDiv.appendChild(lineBreak);
89     let alert = document.createElement('alert');
90     alert.setAttribute('class', 'alert alert-success');
91     alert.setAttribute('role', 'alert');
92     alert.setAttribute('id', 'restart-game');
93     alert.innerHTML = 'Tic-Tac-Toe Game has been restarted!';
94     tictactoeDiv.appendChild(alert);
95   }
96   gameId++;
97   // Use setTimeout() to hide it again after 5 seconds
98   setTimeout(() => $('#restart-game').hide(), 2000);
99 }
100
101
```



PROMINEO TECH

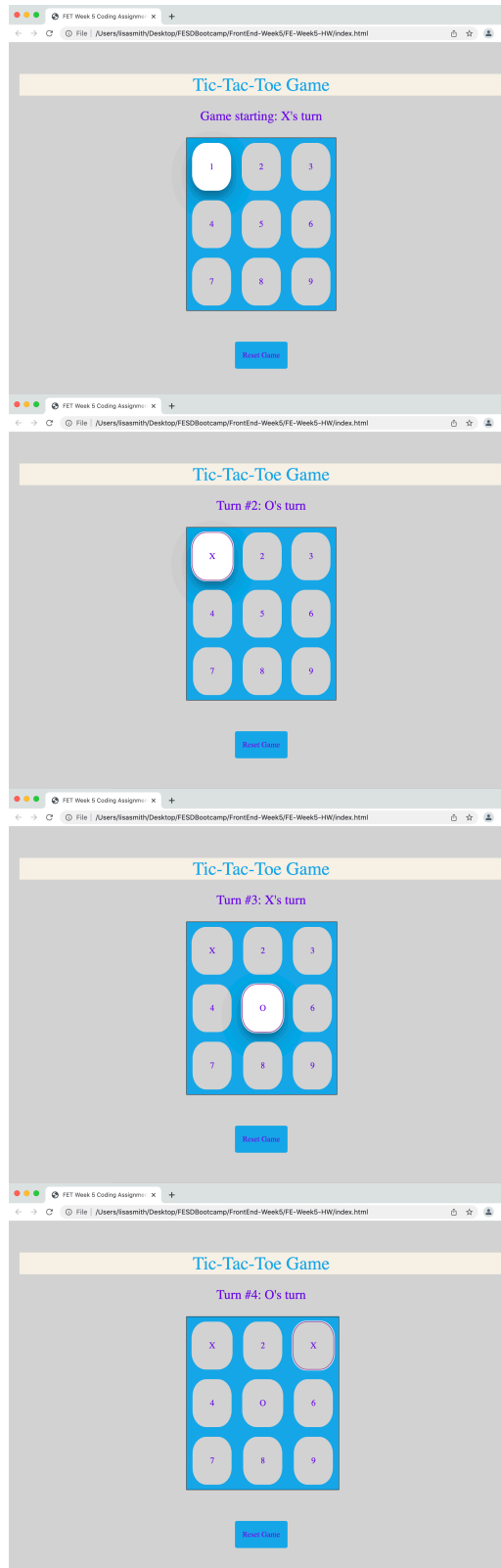
index.js cont'd:

```
100 // Set-up an EventListener for each btn in the game.
101
102 // Set-up an EventListener for each btn in the game.
103
104 for (let btnIndex=1; btnIndex <=9; btnIndex++) {
105   onClick('btn-p-' + btnIndex), () => {
106     // Change Button to display players symbol
107     if (!gameOver) {
108       let headerText = document.getElementById('turn');
109       console.log(gameStatus);
110       if (gameStatus[btnIndex] === '') {
111         console.log('Turn #' + numOTurn);
112         if (numOTurn % 2) {
113           // X's turn
114           document.getElementById('btn-p-' + btnIndex).innerHTML = 'X'
115           //activeButton1.innerHTML = 'X';
116           gameStatus[btnIndex] = 'X';
117           console.log('X's turn!');
118           headerText.innerHTML = 'Turn #' + numOTurn + 1; O's turn';
119         } else {
120           // O's turn
121           document.getElementById('btn-p-' + btnIndex).innerHTML = 'O'
122           //activeButton1.innerHTML = 'O';
123           gameStatus[btnIndex] = 'O';
124           console.log('O's turn!');
125           headerText.innerHTML = 'Turn #' + numOTurn + 1; X's turn';
126         }
127       }
128       numOTurn++;
129       // =====
130       // NOTE: After each turn, check for a winner!
131       // Check the winner
132       // When the game is a draw, or there is a winner --
133       // Bootstrap alert or message across the screen
134       // to announce the winner!
135       // =====
136       let winner = checkForWinner();
137       console.log(winner);
138       if (winner !== '') {
139         let headerText = document.getElementById('turn');
140         headerText.innerHTML = 'Game over! ' + winner;
141         console.log('Game over! ' + winner);
142         // prompt/alert
143         if (gameId > 0) {
144           $('#game-over').show();
145         } else {
146           let tictactoeDiv = document.getElementById('game-page');
147           let lineBreak = document.createElement('br');
148           tictactoeDiv.appendChild(lineBreak);
149           let gameOverAlert = document.createElement('alert');
150           gameOverAlert.setAttribute('class', 'alert alert-success');
151           gameOverAlert.setAttribute('role', 'alert');
152           gameOverAlert.setAttribute('id', 'game-over');
153           gameOverAlert.innerHTML = 'Tic-Tac-Toe Game has ended!';
154           tictactoeDiv.appendChild(gameOverAlert);
155           // Use setTimeout() to hide it again after 5 seconds
156           setTimeout(() => $('#game-over').hide(), 2000);
157         }
158       } else {
159         console.log('INVALID CHOICE');
160       }
161     } else {
162       numOTurn++;
163       console.log('Game Over');
164     }
165   }
166 }
167
168 // Winning combinations:
169 // Horizontal:
170 // [1,2,3] OR [4,5,6] OR [7,8,9]
171 // Vertical:
172 // [1,4,7] OR [2,5,8] OR [3,6,9]
173 // Diagonal:
174 // [1,5,9] OR [3,5,7]
175
176 // =====
177 // =====
178
179 function checkForWinner() {
180   let winner = '';
181   console.log(gameStatus);
182   if ((gameStatus[1] === gameStatus[2]) && (gameStatus[1] === gameStatus[3])
183     && (gameStatus[1] !== '')) {
184     winner = 'Winner is ' + gameStatus[1];
185     gameOver = true;
186   } else if ((gameStatus[4] === gameStatus[5]) && (gameStatus[4] === gameStatus[6])
187     && (gameStatus[4] !== '')) {
188     winner = 'Winner is ' + gameStatus[4];
189     gameOver = true;
190   } else if ((gameStatus[7] === gameStatus[8]) && (gameStatus[7] === gameStatus[9])
191     && (gameStatus[7] !== '')) {
192     winner = 'Winner is ' + gameStatus[7];
193     gameOver = true;
194   } else if ((gameStatus[1] === gameStatus[4]) && (gameStatus[1] === gameStatus[7])
195     && (gameStatus[1] !== '')) {
196     winner = 'Winner is ' + gameStatus[1];
197     gameOver = true;
198   } else if ((gameStatus[2] === gameStatus[5]) && (gameStatus[2] === gameStatus[8])
199     && (gameStatus[2] !== '')) {
200     winner = 'Winner is ' + gameStatus[2];
201     gameOver = true;
202   } else if ((gameStatus[3] === gameStatus[6]) && (gameStatus[3] === gameStatus[9])
203     && (gameStatus[3] !== '')) {
204     winner = 'Winner is ' + gameStatus[3];
205     gameOver = true;
206   } else if ((gameStatus[1] === gameStatus[5]) && (gameStatus[1] === gameStatus[9])
207     && (gameStatus[1] !== '')) {
208     winner = 'Winner is ' + gameStatus[1];
209     gameOver = true;
210   } else if ((gameStatus[3] === gameStatus[5]) && (gameStatus[3] === gameStatus[7])
211     && (gameStatus[3] !== '')) {
212     winner = 'Winner is ' + gameStatus[3];
213     gameOver = true;
214   } else if (numOTurn > 9) {
215     winner = 'The game is a draw!';
216     gameOver = true;
217   }
218   return winner;
219 }
220 }
```



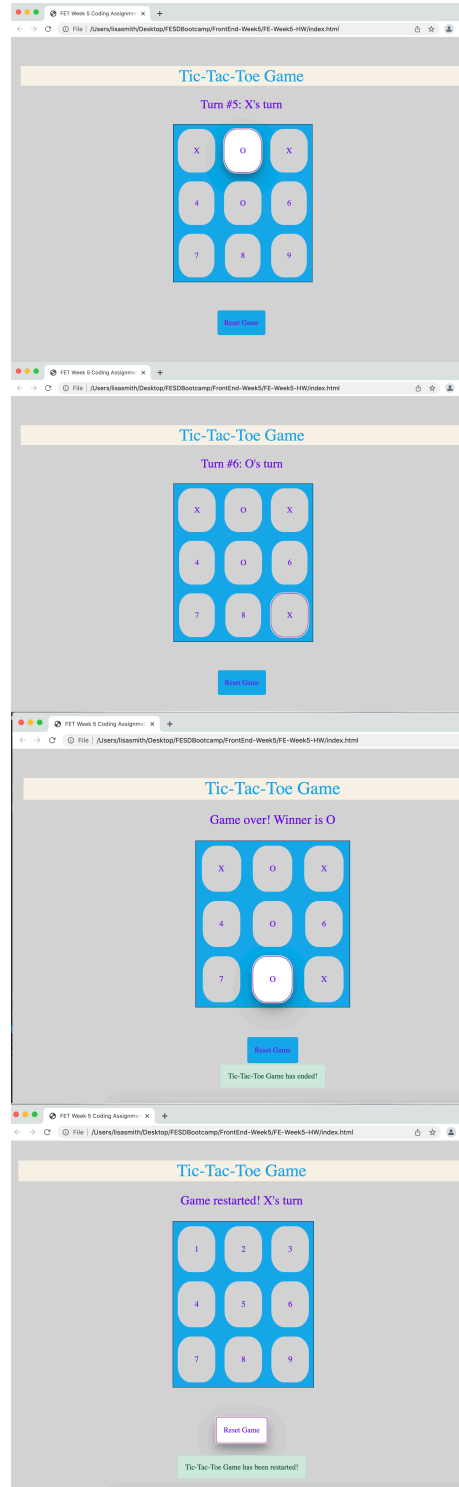
PROMINEO TECH

Screenshots of Running Application:





PROMINEO TECH



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