



Front End Technologies Week 5 Coding Assignment

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.
 - 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:

```
Assignment 11 > tictactoe.html > tictactoe.js > style.css
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <link rel="stylesheet" href="style.css">
9     <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
10    <script src="tictactoe.js" defer></script>
11    <!-- defer: allows you to run javascript in order and no parsing is blocked -->
12  </head>
13  <body>
14
15  <div class="container py-3">
16    <header>
17      <h1>Tic-Tac-Toe</h1>
18      <h1>player <span class="display-player">rona</span>'s turn.</h1>
19    </header>
20    <div class="button py-3">
21      <button id="restart" class="btn btn-outline-light">take me back to 2019</button>
22    </div>
23
24    <div class="container alert alert-light fade show hidden w-50 p-3 text-center"></div>
25  </div>
26
27  <!-- The data-* attribute gives us the ability to embed custom data attributes on HTML elements
28      | The data attribute is providing each cell with an individual index to reference in JavaScript
29
30  <div class="board" id="board">
31    <div class="cell" data-cell></div>
32    <div class="cell" data-cell></div>
33    <div class="cell" data-cell></div>
34    <div class="cell" data-cell></div>
35    <div class="cell" data-cell></div>
36    <div class="cell" data-cell></div>
37    <div class="cell" data-cell></div>
38    <div class="cell" data-cell></div>
39    <div class="cell" data-cell></div>
40  </div>
41
```



```
Assignment 11 > JS tictactoe.js > ...
1  const winningCombinations = [
2      [0, 1, 2], // there are 9 spots but index starts with 0
3      [3, 4, 5],
4      [6, 7, 8],
5      [0, 3, 6],
6      [1, 4, 7],
7      [2, 5, 8],
8      [0, 4, 8],
9      [2, 4, 6],
10 ]
11
12 const playerX = 'rona';
13 const player0 = 'earth';
14 const restart = document.getElementById('restart');
15 const cells = document.querySelectorAll('[data-cell]'); // selecting all cell elements
16 const winner = document.querySelector('.alert');
17 const announcingWinner = document.querySelector('[data-announcing-winner]');
18 const displayPlayer = document.querySelector('.display-player');
19 let currentPlayer = 'rona';
20 let earthTurn;
21
22 startGame();
23
24 restart.addEventListener('click', startGame);
25
26 function startGame () {
27     earthTurn = false
28     cells.forEach(cell => { // looping through the cells
29         cell.classList.remove(playerX);
30         cell.classList.remove(player0);
31         cell.removeEventListener('click', whenClicked); // syntax: element.addEventListener(event, f
32         cell.addEventListener('click', whenClicked, { once: true }); // every time we click on a c
33     });
34     winner.classList.remove('show');
35 }
36
```



PROMINEO TECH

```
< tictactoe.html   JS tictactoe.js # style.css
Assignment 11 > JS tictactoe.js > ...

37 function whenClicked(event) {
38     const cell = event.target
39     const currentPlayer = earthTurn ? player0 : playerX // current class = if it's circle return cir
40     mark(cell, currentPlayer);
41     if (checkWin(currentPlayer)) {
42         endGame(false);
43     } else if (isDraw()){
44         endGame(true);
45     } else {
46         swapTurns();
47     }
48 }
49
50 function mark(cell, currentPlayer) {
51     cell.classList.add(currentPlayer); // add the the mark according to who's turn it is
52 }
53
54 function checkWin(currentPlayer) {
55     return winningCombinations.some(combination => { // return true if any of the winning combina
56         return combination.every(index => { // checking if all the index in the winning combinat
57             return cells[index].classList.contains(currentPlayer); // if true return the current p
58         })
59     })
60 }
61
62 function endGame(draw) {
63     if (draw) {
64         winner.innerText = 'nobody wins, better luck in 2023!'
65         // if it is a draw make the text announce ...
66     } else {
67         winner.innerText = `${earthTurn ? "you're in the clear earth" : "cough, cough rona"} wins!` 
68     }
69     winner.classList.add('show');
70 }
71 }
```



PROMINEO TECH

```
< tictactoe.html   JS tictactoe.js # style.css
Assignment 11 > JS tictactoe.js > ...
72 function isDraw() {
73     return [...cells].every(cell => {
74         return cell.classList.contains(playerX) || cell.classList.contains(player0);
75     })
76 }
77
78 function swapTurns() {
79     earthTurn = !earthTurn      // swapping between X and O turns
80     displayPlayer.classList.remove(`player${currentPlayer}`); // after they take their turn swaps
81     currentPlayer = currentPlayer === 'rona' ? 'earth' : 'rona'; // to next player
82     displayPlayer.innerText = currentPlayer; // changes the
83     displayPlayer.classList.add(`player${currentPlayer}`);
84 }
85
86
87
88 // console.log(playerX);
89 // console.log(player0);
90 // console.log(restart);
91 // console.log(cells);
92 // console.log(winner.classList);
93 // console.log(winner);
94 // console.log(displayPlayer);
95 // console.log(earthTurn);
96 // console.log(checkWin(player0));
```



PROMINEO TECH

```
Assignment 11 > # style.css > ...
1  body {
2      text-align: center;
3      background-image: url("images/nightsky.jpg");
4      background-repeat: no-repeat;
5  }
6
7  h1 {
8      color: ■white;
9      font-size: 65px;
10     text-align: center;
11 }
12
13 .display-player {
14     color: ■yellow;
15 }
16
17 .button {
18     text-align: center;
19 }
20
21 .board {
22     display: grid;
23     grid-template-columns: auto auto auto;
24     padding: 10px;
25     justify-content: center;
26 }
27
28 .cell {
29     width: 135px;
30     height: 135px;
31     border: 3px solid ■white;
32     display: flex;
33     justify-content: center;
34     align-items: center;
35     position: relative;
36     cursor: pointer;
37 }
```



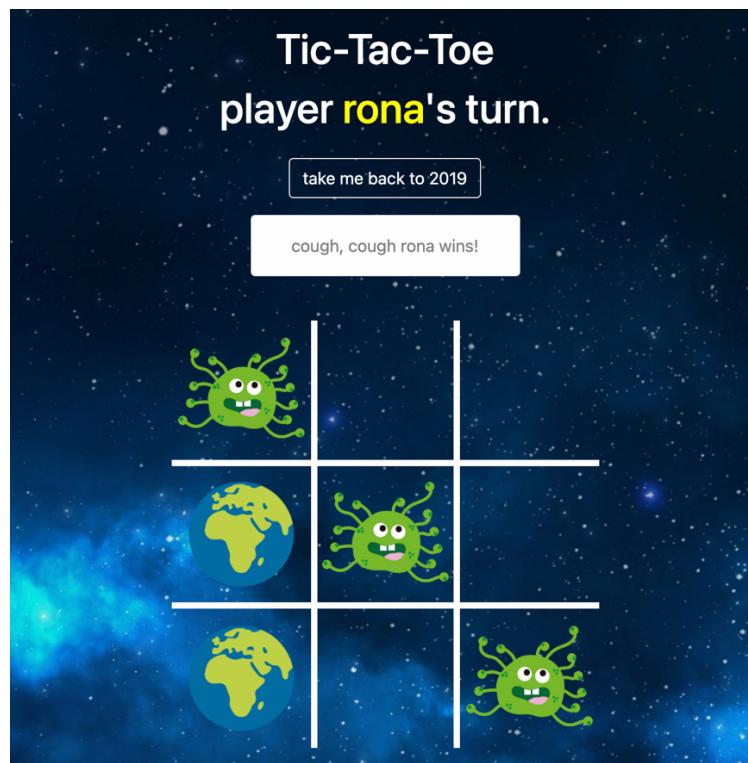
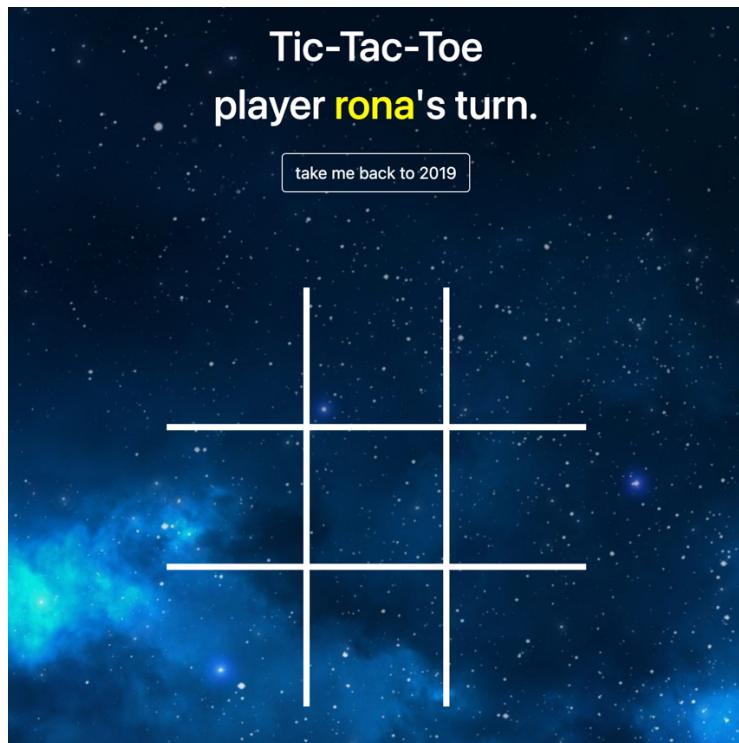
PROMINEO TECH

```
↳ tictactoe.html  ↳ tictactoe.js  # style.css ×  
Assignment 11 > # style.css > ...  
39 .cell.earth {  
40   background-image: url("images/earth1.png");  
41   background-size: 120px 120px;  
42   align-items: center;  
43   justify-content: center;  
44   background-position: center;  
45 }  
46  
47 .cell.rona {  
48   background-image: url("images/rona2.png");  
49   background-size: 130px 130px;  
50   align-items: center;  
51   justify-content: center;  
52   background-position: center;  
53 }  
54  
55 /* REMOVING BORDER */  
56 .cell:nth-child(1),  
57 .cell:nth-child(2),  
58 .cell:nth-child(3) {  
59   border-top: none;  
60 }  
61  
62 .cell:nth-child(1),  
63 .cell:nth-child(4),  
64 .cell:nth-child(7) {  
65   border-left: none;  
66 }  
67  
68 .cell:nth-child(7),  
69 .cell:nth-child(8),  
70 .cell:nth-child(9) {  
71   border-bottom: none;  
72 }
```



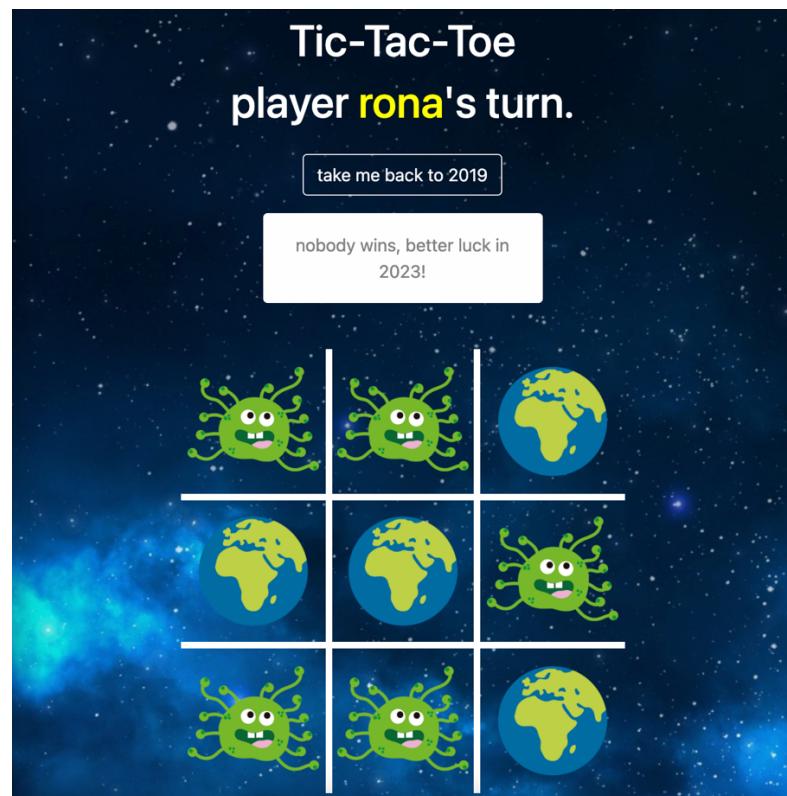
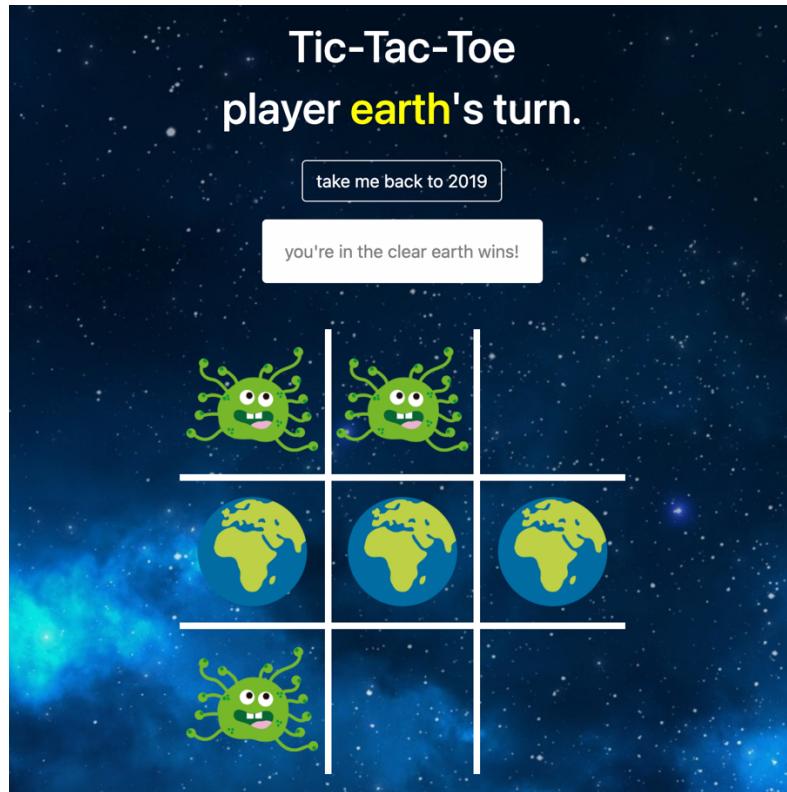
PROMINEO TECH

Screenshots of Running Application:





PROMINEO TECH



URL to GitHub Repository: <https://github.com/zenidev/Coding-Assignment-Week-11.git>