



Week 11: Coding Assignment

URL to GitHub Repository:

<https://github.com/Rcruzn33/tic-tac-toe-game.git>

URL to Your Coding Assignment Video:

<https://youtu.be/bcSoAReptno>

Instructions:

- In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignments and push this document, with your project code, to the repository.
- Include the URLs for this week's repository and video where instructed.
- Submit this document as a .PDF file in the LMS.

Coding Steps:

- Using any of the tools you've worked with so far, create a game of Tic-Tac-Toe.
 - Create a Tic-Tac-Toe game 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.
 - A heading should say whether it is X's or O's turn and change with each move made.
 - A button should be available to clear the grid and restart the game.
 - 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.



Week 11: Coding Assignment

Video Steps:

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
- This video should be done using screen share and voice over.
- This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
 - You can create a new meeting, start screen sharing, and start recording.
 - This will create a video recording on your computer.
- This should then be uploaded to a publicly accessible site, such as YouTube.
 - Ensure the link you share is **PUBLIC** or **UNLISTED**!
 - If it is not accessible by your grader, your project will be graded based on what they can access.



PROMINEO TECH

Week 11: Coding Assignment

```
1 // Store the current player's turn
2 let currentPlayer = "X";
3
4 // Store the number of moves played so far
5 let moveCount = 0;
6
7 // Add an event listener to each button on the tic-tac-toe board
8 $(".btn-sq-responsive").click(function() {
9   // If the box has not already been played
10  if (!$(this).text()) {
11    // Add the current player's symbol to the square
12    $(this).text(currentPlayer);
13
14    // Update the player's turn
15    if (currentPlayer === 'X'){
16      currentPlayer = 'O';
17    }
18    else {
19      currentPlayer = 'X';
20    }
21
22    // Increase the move count
23    moveCount++;
24
25    // Check for a win or draw
26    checkGameStatus();
27  }
28 });
29
30 // Reset the game
```



Week 11: Coding Assignment

```
JS tictactoe2.js > checkGameStatus > hasWinningCombo2
29
30 // Reset the game
31 function reset() {
32   // Clear the board
33   $(".btn-sq-responsive").text("");
34   $(".btn-sq-responsive").prop("disabled", false);
35
36   // Reset the player's turn and move count
37   currentPlayer = "X";
38   moveCount = 0;
39
40   // Update the screen to show player 1's turn
41   $("#screen").text("PLAYER 1 TURN");
42 }
43
44 // Check if the game has ended
45 function checkGameStatus() {
46   // Possible winning combinations
47   let boxes = $(".btn-sq-responsive").toArray();
48   //let hasWinningCombo = false;
49
50   let hasWinningCombo1 = false;
51   let hasWinningCombo2 = false;
52   const winningCombos = [
53     [0, 1, 2], [3, 4, 5], [6, 7, 8], // horizontal
54     [0, 3, 6], [1, 4, 7], [2, 5, 8], // vertical
55     [0, 4, 8], [2, 4, 6] // diagonal
56   ];
57   for (let i = 0; i < winningCombos.length; i++) {
58     const [a, b, c] = winningCombos[i];
59     const Box1 = $(boxes[a]);
60     const Box2 = $(boxes[b]);
61     const Box3 = $(boxes[c]);
62
63     /* if (Box1.text() === currentPlayer && Box2.text() === currentPlayer && Box3.text() === currentPlayer) {
64       hasWinningCombo = true;
65     }
66     */
67
68     if (Box1.text() === "X" && Box2.text() === "X" && Box3.text() === "X") {
69       hasWinningCombo1 = true;
70       console.log("1" + hasWinningCombo1);
71     }
72     else if (Box1.text() === "O" && Box2.text() === "O" && Box3.text() === "O") {
73       hasWinningCombo2 = true;
74       console.log("2" + hasWinningCombo2);
75     }
76   }
77
78   if (hasWinningCombo1) {
79     // Display the winner
80     $("#screen").text("Player 1 WINS!");
81     // Disable all remaining squares
82     $(".btn-sq-responsive").filter(function() {
83       return !$(this).text();
84     }).prop("disabled", true);
85   }
86   return;
```



PROMINEO TECH

Week 11: Coding Assignment

```
JS tictactoe2.js > checkGameStatus > hasWinningCombo2
90 // Disable all remaining squares
91 $(".btn-sq-responsive").filter(function() {
92   return !$(this).text();
93 }).prop("disabled", true);
94
95 return;
96
97
98
99
100 /*if (hasWinningCombo) {
101   // Display the winner
102   $("#screen").text(`${currentPlayer} WINS!`);
103   // Disable all remaining squares
104   $(".btn-sq-responsive").filter(function() {
105     return !$(this).text();
106   }).prop("disabled", true);
107
108   return;
109 }*/
110
111
112
113 // If no winner yet and all squares have been played, declare a draw
114 if (moveCount === 9) {
115   $("#screen").text("DRAW");
116 } else {
117   // Update the screen to show the current player's turn
118   $("#screen").text(`PLAYER ${currentPlayer === "X" ? "1" : "2"} TURN`);
119 }
```