

Project #2: Tic-Tac-Toe

Prof. John P. Baugh – CIS 435 – Fall 2018

Points: _____ / 100

Due: **November 6, 2018 at 11:59 p.m.**

Instructions

Write a web page that enables the user to play the game of tic-tac-toe. This involves a 3 x 3 grid (which you can create using an HTML table), on which users will click the cells, then displaying a symbol representing that player's move. The symbol should be either an X or an O. The objective is to get 3 in a row of your symbol.

Requirements

Program Requirements

1. Your program should create a board with all the cells of the table having empty game pieces. The user's goal is to get three in a row. The program will alternate between X and O, starting with either (chosen at random) and displaying, under the table, "It is X's turn" or "It is O's turn."
2. Using the DOM and the `click` event, write a script that allows the user to place their game piece into the cell they click. [*Hint: The `onclick` event should be specified for each table cell.*]
3. Your program should determine when the game is over, and then prompt the user to determine whether to play again (using buttons.) The buttons "Start/Restart" and "Play Again?" should be available, with the "Play Again?" button only being available when a game is over (due to a win/loss or "the cat's game" (i.e., when nobody wins but the board is full).
4. Use the `Math.random` method to determine if X or O plays first at the beginning of a game.
5. Your program should allow a user to quit the current game and start a new game ("Start/Restart")
6. Your program should allow the user to select one of two radio buttons:
 - a. Player v Player
 - b. Player v Computer
7. If the game is a "Player v Computer", the Player always gets X and the Computer always gets the O symbol, but who moves first depends on the randomization just as it would if it was Player v Player
8. The AI isn't really all that intelligent. The Computer simply looks for an open spot on the board and randomly selects one (again, using `Math.random`.) Again, the **spot must be open – unoccupied – you must NOT write over a symbol if one already exists in a particular spot.**
9. Your program should record and display the amount of time the game has been played for.
10. Your program must work in Microsoft Edge, Mozilla Firefox, and Google Chrome.
11. Your program must be written in JavaScript. You are not allowed to use any JavaScript libraries such as jQuery.
12. Your program should indicate a winning or tie condition when such a situation occurs.

Testing / Analysis / Design Requirements

1. You must include a test plan with **at least 3 days worth** of testing recorded
2. You must include an **appropriate UML diagram**, such as an **Activity** diagram, or **Use-Case** diagram, in a separate .docx or .pdf document
3. **In the same document as the UML diagram is in, described above**, you should include a short (one or two paragraphs – no more than four short paragraphs) written description/analysis of the *problem* and then some information on how you chose to *design* it.
 - a. For example, one paragraph could be a brief explanation of the *problem* and how you are breaking it up / thinking about it (analysis)
 - b. The second paragraph could be about how you developed the *solution* (design)

Sample GUI Prototype

Two moves each, now it's X's turn.

X	O	O
X		

X wins the game

X	O	O
X		
X		

Another example, in which O wins the game

X	O	O
X	X	O
		O

An example of the “Cat’s game” (i.e., nobody wins, and the board is full.)

X	O	O
X	X	O
O	O	X

Deliverables

Please zip up the entire folder containing the appropriate HTML, CSS, etc., also containing subfolders for things like scripts, styles, and images. Include your test plan and analysis document as well. Upload the zip file containing all these items to the appropriate project directory on Canvas.