

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <title>Tic Tac Toe Game</title>
5  <meta charset="utf-8">
6  <meta name="viewport" content="width=device-width">
7  <!--
8      Author: Professor
9      Date created: Summer 2016
10     Date updated: May 2019
11     Version: 19.519.9
12         FIRST VERSION: procedural unobtrusive Javascript, using console output
13         SECOND VERSION: procedural unobtrusive Javascript, using DOM methods and
14         unmodified HTML
15         THIRD VERSION: functional unobtrusive Javascript, using DOM methods to create
16         HTML
17         FORTH VERSION: object-based unobtrusive Javascript, using DOM methods to create
18         HTML
19     Purpose:
20         Demonstrate the approach and rigor required in designing and implementing a web
21         application. Start with documentation to provide the framework for coding the
22         solution.
23     NOTE! THIS IS NOT A FINAL VERSION!
24         You need to continue to improve it as you get closer to the complete
25         version. Continue to refactor the code. Use TODO: DONE: TOFIX: FIXED: until
26         the application is complete, tested, and ready for production.
27     REMEMBER: there are many solutions and multiple ways of doing any single
28         task. Write down all your observations and alternatives (which you can
29         expand on and rearrange throughout the process) so that you can validate and
30         verify you chose the appropriate one at some point.
31     Copyright:
32         This work is the intellectual property of Sheridan College. Any further copying
33         and distribution outside of class must be within the copyright law. Posting to
34         commercial sites for profit is prohibited.
35     Objective:
36         design a generic tic tac toe game (technology-agnostic) and then implement with
37         HTML5, CSS3, and Javascript (client-side).
38     Citations:
39         [1] https://en.wikipedia.org/wiki/Tic-tac-toe
40         [2] https://www.quora.com/Why-is-a-draw-game-in-tic-tac-toe-called-a-cats-game
41         [3] Course Material, Section 7846x, SYST10199, Summer 2019, Sheridan College
42     Description:
43         A two-player game based on [1].
44         The players are presented with a gameboard of three-by-three grid and a button
45         to start a new game.
46         Once the game starts, players take turn selecting one of the nine
47         open squares (available or empty cells) until one player wins or the game is "a
48         scratch" [2]. A player wins when they complete a row, a column, or a diagonal
49         first. When a winning state occurs, a message is presented that the game is
50         over and who the winner is.
51
52         * A message is displayed, specifying which player's turn is presented during
53         the game.
54         * A gameboard representation used
55
56             0 | 1 | 2
57             ---+---+---
58             3 | 4 | 5
59             ---+---+---
60             6 | 7 | 8
61
62     Algorithm: see Readme.pdf file for earlier versions.
```

```

46   Algorithm (pass N-1):
47       (event-driven)
48       1. Start game
49       2. Select cell
50       3. Check if winner
51       3a.   if winner, End game
52       3b.   if no winner and cells available, switch player
53       4. Repeat 2-3 until no cell empty
54       5. End game
55
56   Algorithm (pass N):
57       (event-driven)
58       1. Start game
59       2. Repeat until no cell empty
60       3.   Select cell
61       4.   Check if winner
62       4a.   if winner, End game
63       4b.   if no cells available, End game
64       4c.   if no winner and cells available, switch player
65       5. End game
66
67       CAUTION:  x can complete a win combo using the last available cell
68
69   Testing:
70       create test cases...
71
72       "When considering only the state of the board, and after taking into account
73       board symmetries (i.e. rotations and reflections), there are only 138 terminal
74       board positions. A combinatorics study of the game shows that when "X" makes
75       the first move every time, the game is won as follows:
76       91 distinct positions are won by (X)
77       44 distinct positions are won by (O)
78       3 distinct positions are drawn (often called a "cat's game")" [1]
79
80       

|                                                                    |                                                                  |                                                                  |
|--------------------------------------------------------------------|------------------------------------------------------------------|------------------------------------------------------------------|
| <pre> o       o ---+---+---          o ---+---+---      o   </pre> | <pre>      o   ---+---+--- o   o   ---+---+---          o </pre> | <pre>      o   ---+---+---      o   ---+---+--- o       o </pre> |
|--------------------------------------------------------------------|------------------------------------------------------------------|------------------------------------------------------------------|


82   -->
83   <style>
84   body {
85       width: 680px;
86       margin: 0 auto;
87       text-align: center;
88   }
89   table {
90       margin: auto;
91   }
92   td {
93       border: 1px solid blue;
94       height: 50px;
95       width: 50px;
96       font-size: 1.4em;
97   }
98   tr:first-child td {
99       border-top: none;
100  }
101  tr:nth-child(3) td {
102      border-bottom: none;
103  }
104  td:first-child {
105      border-left: none;

```

```

106 }
107 td:last-child {
108     border-right: none;
109 }
110 </style>
111 </head>
112 <body>
113     <header><h1>Tic Tac Toe</h1></header>
114     <table>
115         <tr>
116             <td>X</td>
117             <td>O</td>
118             <td>O</td>
119         </tr>
120         <tr>
121             <td>X</td>
122             <td>X</td>
123             <td>O</td>
124         </tr>
125         <tr>
126             <td>O</td>
127             <td>O</td>
128             <td>X</td>
129         </tr>
130     </table>
131     <p>Player <span id="player">X</span> go!
132     <button id="reset">Start A New Game</button>
133 </p>
134 <div id="message"></div>
135 <footer>Web Programming &copy; Sheridan College</footer>
136 <script>
137 /* ***
138    Tic Tac Toe Game functionality
139    FIRST VERSION: procedural unobtrusive Javascript, using console output
140    SECOND VERSION: procedural unobtrusive Javascript, using DOM methods and unmodified
141    HTML
142    THIRD VERSION: functional unobtrusive Javascript, using DOM methods to create HTML
143    FORTH VERSION: object-based unobtrusive Javascript, using DOM methods to create HTML
144
145    set up all variable and data structures
146    - current player: X or O
147    - array (collection) of 9 objects
148    - all winning combinations, 3 rows, 3 columns, 2 diagonals
149    - number of available (empty) cells
150    - game not in session (false if in process)
151    - handle to <span id="player">
152    - no handle to <button id="reset"> USED only once
153    - handle to <div id="message">
154
155    playTicTacToe
156    - call gameReset() to start a game
157    - when cell is clicked,
158        call function cellWasClicked(whichCell){}
159    - function cellWasClicked(whichCell){}
160        calls function checkIfCurrentPlayerIsWinner(lastCellPlayed)
161    - function checkIfCurrentPlayerIsWinner(lastCellPlayed)
162        calls displayWhoWon() when there is a winner or a scratch (ends game)
163
164
165
166 /* ***
167    function displayWhoWon() is called when the game is over and

```

```
168     the results are displayed: "Game Over! " + player + " wins."
169 */
170 function displayWhoWon() {
171 }
172
173
174 /* ***
175     function checkIfCurrentPlayerIsWinner() is called to check all winning combinations
176     calls displayWhoWon() if one of them is found true.
177 */
178 function checkIfCurrentPlayerIsWinner(lastCellPlayed) {
179     // loop through all combos and check if one has the same values
180     //     but it is not empty
181     // display the end of the game message - game over
182     // highlight the winning combo on screen
183     // if no empty cells, it is a scratch; display game over
184     // check if there are any remaining empty cells
185     // display the end of the game message - game over
186     // start with checking every time; then filter to lastCellPlayed
187 }
188
189
190 /* ***
191     Function cellWasClicked(whichCell) is called
192         when the event listeners for the "td" cells fire which occurs
193         when the user clicks on one of the nine cells of the board
194     1. sets the content of the clicked cell to the current player's mark
195     2. checks whether or not there is a winner
196     3. flips (changes) the current player
197     4. updates the message to the current player
198     TODO: 1-4 should occur only when the selected cell is empty !!
199 */
200 function cellWasClicked(whichCell) {
201     // conditional on game not being over and cells available
202     // place the user character
203     // one less cells is available
204     // check if there is a winning combination
205     // update player turn and display
206 }
207
208
209 /* ***
210     function gameReset() is called when user clicks on the "Start A New Game" button
211     1. sets content of all 9 cells to nothing
212     2. sets the starting player (this version, X always starts the game)
213     3. updates the message to the current player
214     4. resets the number of empty cells to 9
215     5. sets the game over flag to false to indicate that the game is in progress
216     6. reset font color
217 */
218 function gameReset() {
219 }
220
221
222 /* ***
223     Set up event listeners
224     1. when user clicks on the reset button (id="reset")
225     2. when user clicks on one of the 9 cells on the board
226 */
227
228
229 /* ***
230     Further enhancements
```

```
231      - TODO: change the background of the last cell played to indicate what was the last
232      move
233      - TODO: display and style overlays with messages
234      - TODO: create the board (table) with Javascript
235      - TODO: function playTicTacToe() to load and initialize entire game on
236      "DOMContentLoaded" event
237      - TODO: (optional) make the starting player random
238      - TODO: (optional) keep track of statistics (how many times X wins, etc.)
239      - TODO: constrain checking for winner after the forth turn (start on turn 5 when x
240      places third mark)
241      - TODO: constrain checking only the row, column, and diagonal(s) containing the
242      last selected cell
243      - TODO: create and destroy the board with Javascript instead of hard-coding it with
244      HTML
245      - TODO: convert to object-oriented version
246      */
247      </script>
248      </body>
249      </html>
250
251
252
253
```