

```
1  var grid = [  
2    // each item is a row  
3    ["none", "none", "none"],  
4    ["none", "none", "none"],  
5    ["none", "none", "none"]];  
6  var currentPlayer, gameType, allowed, winLocation;  
7  /* gameType values:  
8     0: 2-Player  
9     1: 1P Easy  
10    2: 1P Hard */  
11  var xWins = 0;  
12  var oWins = 0;  
13  
14  function resetGame() {  
15    grid = [  
16      ["none", "none", "none"],  
17      ["none", "none", "none"],  
18      ["none", "none", "none"]];  
19    for (a in grid) { //reset table on the page  
20      for (b in grid[a]) {  
21        document.getElementById(b.toString() + a.toString()).src = "none.png";  
22      }  
23    }  
24    document.getElementById('message').innerHTML = " ";  
25  
26  }  
27  function newGame() {  
28    resetGame()  
29    allowed = true;  
30    if (gameType == 0) {  
31      document.getElementById('message').innerHTML = currentPlayer + "'s turn...";  
32    } else if (currentPlayer == 'x') {  
33      document.getElementById('message').innerHTML = "Your turn.";  
34    } else {  
35      easyGame()  
36    }  
37  }  
38  function winner(player) { //function checks if the given player is the winner  
39    for (i in grid) {  
40      //checks rows  
41      if (grid[i][0] == player && grid[i][1] == player && grid[i][2] == player) {  
42        return true;  
43      }  
44      //checks columns  
45      if (grid[0][i] == player && grid[1][i] == player && grid[2][i] == player) {  
46        return true;  
47      }  
48    }  
49    //checks diagonals  
50    if (grid[0][0] == player && grid[1][1] == player && grid[2][2] == player) {  
51      return true;  
52    } else if (grid[0][2] == player && grid[1][1] == player && grid[2][0] == player) {  
53      return true;  
54    }  
55    // If player hasn't won..  
56    for (a in grid) { //if board isn't full, return false (game isn't over)  
57      if (grid[a].includes('none')) {  
58        return false;  
59      }  
60    }  
61    //undefined means tie  
62    return undefined;  
63  }  
64  function updateScoreboard() {  
65    document.getElementById('xWins').innerHTML = xWins;  
66    document.getElementById('oWins').innerHTML = oWins;
```

```

67 }
68
69 function startGame(type) {
70     resetGame()
71     xWins = 0;
72     oWins = 0;
73     updateScoreboard()
74     //choose starting player
75     if (Math.random() >= .5) {
76         currentPlayer = "o";
77     } else {
78         currentPlayer = "x";
79     }
80     gameType = type;
81     allowed = true;
82     if (type == 0) {
83         document.getElementById('message').innerHTML = currentPlayer.toUpperCase() +
            "'s turn.";
84     } else if (currentPlayer == 'x') {
85         document.getElementById('message').innerHTML = "Your turn.";
86     } else {
87         easyGame()
88     }
89 }
90 }
91
92 /* HUMAN'S TURN, Main Game */
93
94 function playTurn(trow, tcol, player) {
95     //only lets player take turn if it's their turn
96     if ((currentPlayer == 'x' || gameType == 0) && grid[trow][tcol] == "none" &&
        allowed) {
97         grid[trow][tcol] = player;
98         document.getElementById(trow.toString() + tcol.toString()).src = player + ".png";
99         //switch players
100        if (player == 'x') {
101            currentPlayer = 'o';
102        } else {
103            currentPlayer = 'x';
104        }
105        //end game if player wins
106        if (winner(player)) {
107            if (gameType == 0) {
108                document.getElementById('message').innerHTML = player + " wins!<br
                    /><button onclick='newGame()'>New Game</button>";
109            } else {
110                document.getElementById('message').innerHTML = "You win!<br /><button
                    onclick='newGame()'>New Game</button>";
111            }
112            allowed = false;
113            //increase score of winner
114            if (player == 'o') {
115                oWins++;
116            } else {
117                xWins++;
118            }
119            updateScoreboard()
120        } else if (winner(player) == undefined) {
121            //tie
122            document.getElementById('message').innerHTML = "Tie!<br /><button
                    onclick='newGame()'>New Game</button>";
123            allowed = false;
124        } else if (gameType == 0) {
125            document.getElementById('message').innerHTML = currentPlayer.toUpperCase()
                + "'s turn.";
126        }

```

```
127         //both only run if in the correct game type
128         easyGame()
129     }
130 }
131
132 /* "AI"!
133 (2 player mode needs no function since playTurn only runs for X or 2 player)
134 uses algorithm, not real AI
135 The name easyGame() is a remnant from the original plan to make separate easy and
136 hard AI modes
137 however, the easy AI was TOO easy, so now it's just 1, medium difficulty AI. */
138 function nextWin(player) { //makes it possible to block or win when opportunity exists
139     for (i in grid) {
140         //top or left is blank
141         if (grid[i][0] == "none" && grid[i][1] == player && grid[i][2] == player) {
142             winLocation = [i, 0]
143             return true;
144         }
145         if (grid[0][i] == "none" && grid[1][i] == player && grid[2][i] == player) {
146             winLocation = [0, i]
147             return true;
148         }
149         //middle blank
150         if (grid[i][0] == player && grid[i][1] == "none" && grid[i][2] == player) {
151             winLocation = [i, 1]
152             return true;
153         }
154         if (grid[0][i] == player && grid[1][i] == "none" && grid[2][i] == player) {
155             winLocation = [1, i]
156             return true;
157         }
158         //bottom or right blank
159         if (grid[i][0] == player && grid[i][1] == player && grid[i][2] == "none") {
160             winLocation = [i, 2]
161             return true;
162         }
163         if (grid[0][i] == player && grid[1][i] == player && grid[2][i] == "none") {
164             winLocation = [2, i]
165             return true;
166         }
167     }
168     //DIAGONALS
169     //top is blank
170     if (grid[0][0] == "none" && grid[1][1] == player && grid[2][2] == player) {
171         winLocation = [0, 0]
172         return true;
173     } else if (grid[0][2] == "none" && grid[1][1] == player && grid[2][0] == player) {
174         winLocation = [0, 2]
175         return true;
176     }
177     // center is blank
178     if (grid[0][0] == player && grid[1][1] == "none" && grid[2][2] == player) {
179         winLocation = [1, 1]
180         return true;
181     } else if (grid[0][2] == player && grid[1][1] == "none" && grid[2][0] == player) {
182         winLocation = [1, 1]
183         return true;
184     }
185     //bottom is blank
186     if (grid[0][0] == player && grid[1][1] == player && grid[2][2] == "none") {
187         winLocation = [2, 2]
188         return true;
189     } else if (grid[0][2] == player && grid[1][1] == player && grid[2][0] == "none") {
190         winLocation = [2, 0]
191         return true;
```

```
192     }
193     return false;
194 }
195 function easyGame() {
196     if (gameType == 1 && currentPlayer == 'o' && allowed) {
197         document.getElementById('message').innerHTML = "Thinking...";
198         setTimeout( //timeout to create illusion of thinking
199             function(){
200                 if (nextWin('o')) { //if AI will win
201                     grid[winLocation[0]][winLocation[1]] = 'o';
202                     document.getElementById(winLocation[0].toString() +
203                         winLocation[1].toString()).src = "o.png";
204                 } else if (nextWin('x')) { //block player from winning
205                     grid[winLocation[0]][winLocation[1]] = 'o';
206                     document.getElementById(winLocation[0].toString() +
207                         winLocation[1].toString()).src = "o.png";
208                 } else {
209                     //if blocking or winning unnecessary, pick a random spot
210                     var x, y;
211                     var worked = false;
212                     while (!worked) { //pick random spots until one works
213                         x = Math.floor(Math.random() * 3);
214                         y = Math.floor(Math.random() * 3);
215                         if (grid[x][y] == "none") {
216                             grid[x][y] = 'o';
217                             document.getElementById(x.toString() + y.toString()).src =
218                                 "o.png";
219                             worked = true;
220                         } else {
221                             worked = false;
222                         }
223                     }
224                 }
225                 //switching turn
226                 currentPlayer = 'x';
227                 document.getElementById('message').innerHTML = "Your turn.";
228                 //check if AI won
229                 if(winner('o')) {
230                     document.getElementById('message').innerHTML = "I win!<br /><button
231                         onclick='newGame()'>New Game</button>";
232                     oWins++;
233                     updateScoreboard()
234                     allowed = false;
235                 } else if (winner('o') == undefined) {
236                     document.getElementById('message').innerHTML = "Tie!<br /><button
237                         onclick='newGame()'>New Game</button>";
238                     allowed = false;
239                 }
240             }, 700
241         )
242     }
243 }
```

```
1 <html>
2 <head>
3   <title>Tic-Tac-Toe</title>
4   <script src="game.js"></script>
5   <style>
6     h2 {font-family: calibri; font-size: 36; margin-bottom: 0px;}
7     h3 {font-family: calibri; font-size: 26; margin-bottom: 0px;}
8     p {font-family: calibri; font-size: 16;}
9
10    /* button CSS from
11    https://www.w3schools.com/css/tryit.asp?filename=trycss_buttons_group_vertical */
12    .btn-group .button {
13      background-color: #4CAF50; /* Green */
14      border: 1px solid green;
15      color: white;
16      padding: 15px 32px;
17      text-align: center;
18      text-decoration: none;
19      display: inline-block;
20      font-size: 16px;
21      cursor: pointer;
22      width: 150px;
23      display: block;
24    }
25
26    .btn-group .button:not(:last-child) {
27      border-bottom: none; /* Prevent double borders */
28    }
29
30    .btn-group .button:hover {
31      background-color: #3e8e41;
32    }
33  </style>
34  <link rel="icon" href="icon.png" type="image/x-icon" />
35 </head>
36 <body onload="resetGame()">
37   <table width=100% height=100% rules=cols>
38     <tr>
39       <td width=25% valign=center rowspan=2>
40         <table align=center>
41           <tr>
42             <td colspan=2><h2><u>Wins</u></h2></td>
43           </tr>
44           <tr align=center>
45             <td><h3>X</h3></td>
46             <td><h3>O</h3></td>
47           </tr>
48           <tr align=center>
49             <td><p id="xWins">0</p></td>
50             <td><p id="oWins">0</p></td>
51           </tr>
52         </table>
53
54       <td align=center>Start Tic-Tac-Toe</td>
55     </tr>
56     <tr align=center>
57       <td align=center>
58         <div class="btn-group" align=center>
59           <button onclick="startGame(1)" class=button>1 Player</button>
60           <button onclick="startGame(0)" class=button>2 Players</button>
61         </div>
62       </td>
63     </tr>
64   </table>
```

```
64         
66     </td>
67     <td>
68         
70     </td>
71     <td>
72         
74     </td>
75 </tr>
76 <tr>
77     <td>
78         
80     </td>
81     <td>
82         
84     </td>
85     <td>
86         
88     </td>
89 </tr>
90 <tr>
91     <td>
92         
94     </td>
95     <td>
96         
98     </td>
99     <td>
100         
102     </td>
103 </tr>
</table>
</td>
</tr>
<tr height=20%>
    <td align=center><h2 id="message"> </h2></td>
</tr>
</table>
</body>
</html>
```