

CheckWin.cpp

```

1  /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1B
5  * SECTION    : MW 7:30pm
6  * Assign #2  : tic-tac-toe game (multi-dimensional arrays)
7  * DUE DATE   : 19 September 2019
8  *****/
9
10 #include "MyHeader.h"
11
12 /*****
13 * CheckWin
14 * This function checks to see if either player has won. Once it is
15 * possible for a win condition to exist, this should run after each a
16 * player makes a play.
17 *
18 * RETURNsthe character value of the player that won or a value that
19 * indicates a tie.
20 *****/
21
22 char CheckWin(const char boardAr[][3]) //PROCESS - the function checks boardAr
23 {
24
25     /*****
26     * VARIABLES *
27     *****/
28
29     bool    xWinWays; //PROCESS - How token X is considered won
30     bool    oWinWays; //PROCESS - How token 0 is considered won
31
32     int     i; // PROCESS - used in loop
33     int     j; // PROCESS - used in loop
34
35     char wonPlayer; // PROCESS - returns the won token
36
37     /*****
38     * PROCESS - There are 9 ways that the token X can win the game
39     *****/
40     xWinWays = (((boardAr[0][0] == 'X') &&
41                  (boardAr[1][0] == 'X') &&
42                  (boardAr[2][0] == 'X')) ||
43
44                ((boardAr[0][1] == 'X') &&
45                 (boardAr[1][1] == 'X') &&
46                 (boardAr[2][1] == 'X')) ||
47
48                ((boardAr[0][2] == 'X') &&
49                 (boardAr[1][2] == 'X') &&
50                 (boardAr[2][2] == 'X')) ||
51
52                ((boardAr[0][0] == 'X') &&
53                 (boardAr[0][1] == 'X') &&
54                 (boardAr[0][2] == 'X')) ||
55

```

CheckWin.cpp

```

56         ((boardAr[1][0] == 'X') &&
57         (boardAr[1][1] == 'X') &&
58         (boardAr[1][2] == 'X')) ||
59
60         ((boardAr[2][0] == 'X') &&
61         (boardAr[2][1] == 'X') &&
62         (boardAr[2][2] == 'X')) ||
63
64         ((boardAr[0][0] == 'X') &&
65         (boardAr[1][1] == 'X') &&
66         (boardAr[2][2] == 'X')) ||
67
68         ((boardAr[0][2] == 'X') &&
69         (boardAr[1][1] == 'X') &&
70         (boardAr[2][0] == 'X')));
71
72
73     /*****
74     * PROCESS - There are 9 ways that the token 0 can win the game
75     *****/
76     oWinWays = (((boardAr[0][0] == '0') &&
77         (boardAr[1][0] == '0') &&
78         (boardAr[2][0] == '0')) ||
79
80         ((boardAr[0][1] == '0') &&
81         (boardAr[1][1] == '0') &&
82         (boardAr[2][1] == '0')) ||
83
84         ((boardAr[0][2] == '0') &&
85         (boardAr[1][2] == '0') &&
86         (boardAr[2][2] == '0')) ||
87
88         ((boardAr[0][0] == '0') &&
89         (boardAr[0][1] == '0') &&
90         (boardAr[0][2] == '0')) ||
91
92         ((boardAr[1][0] == '0') &&
93         (boardAr[1][1] == '0') &&
94         (boardAr[1][2] == '0')) ||
95
96         ((boardAr[2][0] == '0') &&
97         (boardAr[2][1] == '0') &&
98         (boardAr[2][2] == '0')) ||
99
100        ((boardAr[0][0] == '0') &&
101        (boardAr[1][1] == '0') &&
102        (boardAr[2][2] == '0')) ||
103
104        ((boardAr[0][2] == '0') &&
105        (boardAr[1][1] == '0') &&
106        (boardAr[2][0] == '0')));
107
108
109
110

```

CheckWin.cpp

```
111  /*****
112  * PROCESS - Initializes the wonPlayer based on which token has won the game
113  *****/
114  if(xWinWays)
115  {
116      wonPlayer = 'X';
117  }
118
119  else if(oWinWays)
120  {
121      wonPlayer = 'O';
122  }
123  else
124  {
125      for(i = 0; i < ROW_SIZE; i++)
126      {
127          for(j = 0; j < COL_SIZE; j++)
128          {
129              if((boardAr[i][j] == ' ') && (!xWinWays && !oWinWays))
130              {
131                  wonPlayer = 'K'; // For KEEP PLAY
132              }
133          }
134      }
135  }
136
137
138
139  return wonPlayer;
140
141 }
142
143
144
145
```