

```

1  /**
2   * 2D Retro Game - Box Hunt
3   * Some code sourced from games by Ania Kubow
4   */
5
6  // game object
7  const GAME_CONFIG = {
8    numberOfTiles: 9,
9    timePeriod: 60,
10   countdownPeriod: 1000,
11   startPeriod: 1000
12 };
13
14 let score = 0;
15 let period = GAME_CONFIG.timePeriod; // length of the game
16
17 let grid = document.querySelector('.grid'); // get the game elements with their classes and
18 let timeLeft = document.querySelector('#time-left');
19 let scoreDisplayed = document.querySelector('#score');
20 let tiles = null;
21
22 let start = document.querySelector('#start-game'); // Play button
23 let reset = document.querySelector('#reset-game'); // Replay button
24 // ensure when 'Start' is available, 'Replay' is not and vice versa
25
26 // game variables
27 let hitPosition = null;
28 let randomTargetPeriod = GAME_CONFIG.startPeriod; // on start, move target every 1000ms (1
29 let randomTargetTimer = GAME_CONFIG.startPeriod;
30 let countdownTimer = GAME_CONFIG.countdownPeriod;
31
32 function initialiseGame() {
33   // 'Start'/'Replay' buttons
34   start.disabled = false; // 'Start' is enabled
35   reset.disabled = true; // 'Replay' is disabled
36
37   createGrid();
38   tiles = document.querySelectorAll('.tile'); // select all elements with class .tile
39
40   reset.addEventListener('click', resetGame);
41   start.addEventListener('click', startGame);
42 }
43
44 // DOM event listener - all HTML elements to be read before loading JS file, as it makes s
45 document.addEventListener('DOMContentLoaded', initialiseGame);
46
47 function createGrid() {
48   // code from 'Build your own CANDY CRUSH using JavaScript, HTML and CSS' by Ania Kubow
49   // create the game grid of 'n' number of tiles
50   for (let n = 0; n < GAME_CONFIG.numberOfTiles; n++) {
51     // create the <div> tag for each tile
52     const tile = document.createElement('div');
53     tile.classList.add('tile');
54     tile.setAttribute('id', n); // each tile gets it's 'id'. Also use 'n' for looping
55     grid.appendChild(tile); // append the tile to the grid <div>
56   }
57 }
58
59 function randomTarget() {
60   // generate random target tile
61   // get each tile (0-8) and remove the class .target
62   // code from 'Learn JavaScript by building 7 games: Whac-A-Mole' by Ania Kubow
63
64   tiles.forEach(tile => {
65     // removes target before producing new one, otherwise will have several targets at
66     tile.classList.remove('target');
67   });
68   let randomNumber = Math.floor(Math.random() * 9); //generate random number 0-8
69   let randomTile = tiles[randomNumber]; //generate random target tile position
70   randomTile.classList.add('target'); // assign target class to position
71   hitPosition = randomTile.id; // assign potential hit position with tile and it's id
72 }
73
74 function clearTheIntervals() {
75   // clear when game has timed out for game over or when 'Replay' button is hit
76   clearInterval(countdownTimer);
77   clearInterval(randomTargetTimer);
78 }
79
80 function countdown() {
81   // countdown timer and check if game is over
82   // some code from 'Learn JavaScript by building 7 games: Whac-A-Mole' by Ania Kubow
83   period--; //decrement
84   timeLeft.textContent = period;
85   if (period == 0) {
86     clearTheIntervals();
87     gameOver(); //end the game
88   }
89 }
90

```

CONFIGURE

Metrics

There are 16 functions in this file.

Function with the largest signature take 1 arguments, while the median is 0.

Largest function has 7 statements in it, while the median is 3.5.

The most complex function has a cyclomatic complexity value of 2 while the median is 1.



version 2.13.6

(<https://github.com/jshint/jshint>)

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

```

91 function onTileClicked() {
92     // if target is hit, increase the score and speed of the game
93     clearInterval(randomTargetTimer); // clear current repeat period
94
95     score++;
96     scoreDisplayed.textContent = score;
97     hitPosition = null; // clear the hit position - no multiple scoring
98
99     // speed up the moving target depending on score
100     randomTargetPeriod = GAME_CONFIG.startPeriod - (score * 50); //set the new repeat period
101     randomTargetTimer = setInterval(randomTarget, randomTargetPeriod); // run faster
102 }
103
104 function isTargetHit() {
105     // check if random target has been clicked or touched
106     // some code from 'Learn JavaScript by building 7 games: Whac-A-Mole' by Ania Kubow
107     tiles.forEach(tile => {
108         tile.addEventListener('click', () => {
109             // 'click' is used for mouse click or touch on touchscreen
110             if (tile.id == hitPosition) {
111                 onTileClicked(); // if target is hit
112             }
113         });
114     });
115 }
116
117 function removeTarget() {
118     // remove event listener and ability to restart game by random clicking on grid
119     tiles.forEach(tile => {
120         tile.classList.remove('target');
121         hitPosition = null;
122         tile.removeEventListener('click', isTargetHit);
123     });
124 }
125
126 function resetGame() {
127     // reset game when 'Replay' button is clicked/touched
128     clearTheIntervals();
129     scoreDisplayed.textContent = 0;
130     timeLeft.textContent = GAME_CONFIG.timePeriod;
131
132     removeTarget(); // remove coloured tile and deactivate clicking
133
134     // if the 'Game Over!' is displayed, remove it
135     const element = document.querySelector('#game-over');
136     element.textContent = "";
137
138     startGame();
139 }
140
141 function enableReplay() {
142     // 'Replay' button is active, whilst 'Start' is inactive
143     start.disabled = true;
144     reset.disabled = false;
145 }
146
147 function startGame() {
148     // this is the first play of the game
149     score = 0;
150     period = GAME_CONFIG.timePeriod;
151
152     enableReplay();
153
154     countdownTimer = setInterval(countDown, GAME_CONFIG.countDownPeriod); // countdown even
155     randomTargetTimer = setInterval(randomTarget, GAME_CONFIG.startPeriod); // create target
156
157     isTargetHit(); // check if target is hit
158 }
159
160 function gameOver() {
161     // create the text 'Game Over'
162     const gameOverText = document.querySelector('#game-over');
163     gameOverText.textContent = "Game Over!";
164
165     removeTarget(); // remove coloured tile and deactivate clicking
166     enableReplay();
167 }
168

```



version 2.13.6

Metrics

There are 16 functions in this file.

Function with the largest signature take 1 arguments, while the

median is 0.

Largest function has 7 statements in it, while the median is

3.5.

The most complex function has a cyclomatic complexity value of 2 while the median is 1.

(<https://github.com/jshint/jshint>);

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

