

```

1 // 2D Retro Game - Box Hunt
2 // Some code sourced from 'Whac-A-Mole' game by Ania Kubow
3
4 // DOM event listener - all HTML elements to be read before loading JS file
5 // makes sure events happen in order, because I
6 // had issues with Jest testing as DOM hadn't loaded
7 document.addEventListener('DOMContentLoaded', () => {
8
9     // game object
10    let game = {
11        score: 0,
12        numberOfTiles: 9,
13        timePeriod: 60,
14        countdownPeriod: 1000,
15        startPeriod: 1000
16    };
17
18    const grid = document.querySelector('.grid'); // get the game elements with their clas
19    const timeLeft = document.querySelector('#time-left');
20    const scoreDisplayed = document.querySelector('#score');
21
22    const start = document.querySelector('#start-game');
23    const reset = document.querySelector('#reset-game');
24    // ensure when 'start' is available, 'reset' is not and vice versa
25    start.disabled = false;
26    reset.disabled = true;
27
28    // game variables
29    let hitPosition;
30    let randomTargetPeriod = game.startPeriod; // on start, move target every 1000ms (1.0s
31    let randomTargetTimer;
32    let countdownTimer;
33
34    function createGrid() {
35        //Build your own CANDY CRUSH using JavaScript, HTML and CSS | Ania Kubow
36        // create the game grid
37        for (let n = 0; n < game.numberOfTiles; n++) {
38            // create the <div> tag for each tile
39            const tile = document.createElement('div');
40            tile.classList.add('tile');
41            tile.setAttribute('id', n); // each tile gets it's 'id'. Also use 'n' for loop
42            grid.appendChild(tile);
43        }
44    }
45    createGrid();
46    const tiles = document.querySelectorAll('.tile'); // select all elements with class .t
47
48    function randomTarget() {
49        // generate random target tile
50        // get each tile (0-8) and remove the class .target
51        tiles.forEach(tile => {
52            tile.classList.remove('target');
53        });
54        let randomNumber = Math.floor(Math.random() * 9); //0-8
55        let randomTile = tiles[randomNumber];
56        randomTile.classList.add('target');
57        hitPosition = randomTile.id;
58    }
59
60    function countdown() {
61        // countdown timer and check if game is over
62        game.timePeriod--; //decrement
63        timeLeft.textContent = game.timePeriod;
64
65        if (game.timePeriod == 0) {
66            clearInterval(countdownTimer);
67            clearInterval(randomTargetTimer);
68            //end game
69            gameOver();
70        }
71    }
72
73    function isTargetHit() {
74        // check if random target has been clicked or touched
75        tiles.forEach(tile => {
76            tile.addEventListener('click', () => {
77                // 'click' for mouse click or touch on touchscreen
78                if (tile.id == hitPosition) {
79                    clearInterval(randomTargetTimer); // clear repeat period
80
81                    game.score++;
82                    scoreDisplayed.textContent = game.score;
83                    hitPosition = null;
84
85                    // speed up the moving target depending on score
86                    randomTargetPeriod = game.startPeriod - (game.score * 50); //set the n
87                    randomTargetTimer = setInterval(randomTarget, randomTargetPeriod); //
88                } // if you hit the coloured box
89            });
90        });

```

CONFIGURE

Metrics

There are 15 functions in this file.

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

Largest function has 24 statements in it, while the median is 6.

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



version 2.13.6

([https://github.com/jshint/j](https://github.com/jshint/jshint/)

[About \(/about\)](#)

[Documentation \(/docs\)](#)

[Install \(/install\)](#)

[Contribute \(/contribute\)](#)

[Blog \(/blog\)](#)

```

91     }
92
93     function removeTarget() {
94         // remove event listener and ability to restart game by random clicking on grid
95         tiles.forEach(tile => {
96             tile.classList.remove('target');
97             hitPosition = null;
98             tile.removeEventListener('click', isTargetHit);
99         });
100     }
101
102     function resetGame() {
103         // reset game when Reset button is clicked/touched
104         reset.onclick = () => {
105             clearInterval(countDownTimer);
106             clearInterval(randomTargetTimer);
107             scoreDisplayed.textContent = 0;
108             timeLeft.textContent = 60;
109
110             removeTarget(); // remove coloured tile and deactivate clicking
111
112             // if the 'Game Over!' is displayed, remove it
113             const element = document.querySelector('#game-over');
114             element.textContent = "";
115
116             start.disabled = false;
117             reset.disabled = true;
118         };
119     }
120     resetGame();
121
122     function startGame() {
123         // start game when Start button is clicked/touched
124         start.onclick = () => {
125             game.score = 0;
126             game.timePeriod = 60;
127             start.disabled = true;
128             reset.disabled = false;
129
130             countDownTimer = setInterval(countDown, game.countDownPeriod); // every 1 sec
131             randomTargetTimer = setInterval(randomTarget, game.startPeriod);
132
133             isTargetHit(); // check if target is hit
134         };
135     }
136     startGame();
137
138     function gameOver() {
139         // create the text 'Game Over'
140         const gameOverText = document.querySelector('#game-over');
141         gameOverText.textContent = "Game Over!";
142
143         removeTarget(); // remove coloured tile and deactivate clicking
144
145         start.disabled = true;
146         reset.disabled = false;
147     }
148
149     // END of file - place module exports here
150     // remember to import these into app.test.js
151     //module.exports = { game, createGrid, startGame }; // Uncomment when running Jest tests
152
153     });
154
155

```



version 2.13.6

Metrics

There are 15 functions in this file.

Function with the largest signature take 1 arguments, while the

median is 0.

Largest function has 24 statements in it, while the median is 6.

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 \(/blog\)](#)

