```
1
      /* jshint esversion: 6 */
2
      // global constants
3
 4
      // modals
 5
      const RULES_MODAL = document.getElementById("modal-rules");
 6
      const RULES MODAL BTN = document.getElementById("btn-rules");
      const RULES_MODAL_CLOSE = document.getElementById("close-rules");
7
8
      const START GAME MODAL = document.getElementById("modal-start-game");
9
      const START GAME MODAL BTN = document.getElementById("btn-start-game");
10
      const START_GAME_MODAL_CLOSE = document.getElementById("close-start-game");
11
      const CONGRATULATIONS_MODAL = document.getElementById("modal-congratulations");
      const CONGRATULATIONS_TEXT = document.getElementById("congratulations-text");
12
      const CONGRATULATIONS_DOG = document.getElementById("congratulations-dog");
13
      const CONGRATULATIONS_BOWL = document.getElementById("congratulations-bowl");
14
      const START_GAME_FORM = document.getElementById("start-game-form");
15
16
      // dom elements
      const MAIN = document.getElementsByTagName("main")[0];
17
      const BODY = document.getElementsByTagName("body")[0];
18
19
      const MENU_CONTAINER = document.getElementById("menu-container");
20
      const MENU_NAV = document.getElementById("game-menu");
21
      const MENU NAV BTN = document.getElementById("btn-menu");
      const BACK_NAV_BTN = document.getElementById("btn-back");
22
23
      const MAZE_CONTAINER = document.getElementById("outer-maze-container");
24
      const TREATS = document.getElementById("treats");
      const TIME = document.getElementById("time");
25
      const HIDDEN_TREATS = document.getElementsByClassName("treat");
26
27
      const SOUND_BTN = document.getElementById('sound');
28
      // control arrows
29
      const CONTROL_UP = document.getElementById("up");
30
      const CONTROL_LEFT = document.getElementById("left");
      const CONTROL DOWN = document.getElementById("down");
31
      const CONTROL_RIGHT = document.getElementById("right");
32
33
      // highscore table
      const TABLE = document.getElementById("highscore");
34
35
      const TABLE BODY = TABLE.getElementsByTagName("tbody")[0];
      const RESET_HIGHSCORE = document.getElementById("reset-highscore");
36
      const RESTART = document.getElementById("restart");
37
38
39
      // global variables needed during the whole game
40
      let isPlaying = false;
41
      let treatsCollected = 0;
42
      let treatsAvailable;
43
      let timePassed;
44
      let seconds;
45
      let timer = false;
      let timerInterval;
46
47
      // sounds
48
      let sound = false;
49
      let soundTreat = new Audio("assets/sounds/treat.mp3");
      let soundEating = new Audio("assets/sounds/eating.mp3");
50
      let soundPanting = new Audio("assets/sounds/panting.mp3");
51
52
      soundEating.loop = true;
53
      soundPanting.loop = true;
54
55
      // necessary to move the dog character
56
      // adapted from https://www.the-art-of-web.com/mazing.js
      let Position = function (x, y) {
57
58
          this.x = x;
           this.y = y;
59
60
      Position.prototype.toString = function () {
61
62
          return this.x + ":" + this.y;
63
64
      document.addEventListener("DOMContentLoaded", function () {
65
66
           \ ^{*} Open the modal according to the given event.
67
68
             @param {event} e
69
          function openModal(e) {
70
```



version 2,13.6 There are 24 function (https://github.com/jshint/j: Function with the larg

About (madian is 0.5.

Documentation (/docs)

Install (/install)
of 15 while the media
Contribute (/contribute)

Blog (/blog)

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder

https://jshint.com 1/9

```
71
               isPlaying = false;
72
               // set aria-hidden for the main content hidden behind the modal for accessibility
               MAIN.setAttribute('aria-hidden', 'true');
73
               // set overflow: hidden for the body to prevent scrolling the hidden content
74
               BODY.style.overflow = "hidden";
75
76
               // open the model according to the clicked button
               switch (e.target) {
77
                   // open rules modal
78
                   case RULES_MODAL_BTN:
79
                       RULES MODAL.style.display = "block";
80
                       RULES_MODAL.setAttribute('aria-hidden', 'false');
81
82
                       break:
83
                   // open start game modal
                   case START_GAME_MODAL_BTN:
84
85
                       START_GAME_MODAL.style.display = "block";
86
                       START_GAME_MODAL.setAttribute('aria-hidden', 'false');
87
                       handleRadioButtons();
88
                       break;
89
                   default:
90
                       break;
91
               }
92
           }
93
94
95
            * Close the modal according to the given event.
            * @param {event} e
96
97
98
           function closeModal(e) {
99
               MAIN.setAttribute('aria-hidden', 'false');
100
               BODY.style.overflow = "unset";
               \ensuremath{//} close the model according to the clicked button
101
102
               switch (e.target) {
103
                   // close rules modal
104
                   case RULES MODAL CLOSE:
105
                       RULES_MODAL.style.display = "none";
                       RULES MODAL.setAttribute('aria-hidden', 'true');
106
107
                       break;
108
                   // close start game modal
109
                   case START GAME MODAL CLOSE:
110
                       START_GAME_MODAL.style.display = "none";
111
                       START_GAME_MODAL.setAttribute('aria-hidden', 'true');
112
                       break;
                   // game started - close modal
113
114
                   case START_GAME_FORM:
                       START_GAME_MODAL.style.display = "none";
115
116
                       START_GAME_MODAL.setAttribute('aria-hidden', 'true');
117
                       break;
118
                   // game restarted - close congratulations modal
119
                   case RESTART:
                       CONGRATULATIONS_MODAL.style.display = "none";
120
121
                       CONGRATULATIONS_MODAL.setAttribute('aria-hidden', 'true');
122
                       break;
123
                   default:
124
                       break;
125
               }
126
           }
127
128
            ^{st} Start the game with the selected options from the form.
129
130
              @param {event} e
131
132
          function startGame(e) {
133
               if (timer) {
134
                   clearInterval(timerInterval);
135
136
               isPlaying = true;
               if (sound) {
137
                   soundPanting.play();
138
139
140
141
               e.preventDefault();
142
               closeModal(e);
```

JSL int

version 2,13.6 There are 24 functior (https://github.com/jshint/j: Function with the larg

About (median is 0.5.

Documentation (/docs)

Install (/install)
of 15 while the media
Contribute (/contribute)

Blog (/blog)

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder

https://jshint.com 2/9

```
143
               MENU_NAV.style.display = "flex";
144
               MENU_NAV.setAttribute('aria-hidden', 'false');
145
               closeMenu();
               // initialize the maze based on the selected difficulty
146
147
               let Maze;
148
               switch (START_GAME_FORM.difficulty.value) {
                   case "easy":
149
150
                       Maze = new MazeBuilder(8, 4);
151
                       break:
152
                   case "medium":
                       Maze = new MazeBuilder(10, 5);
153
154
                       break:
                   case "hard":
155
156
                       Maze = new MazeBuilder(14, 8);
157
                       break;
158
                   case "mobile":
159
                       Maze = new MazeBuilder(5, 6);
160
                       break;
161
                   default:
                       break;
162
163
               Maze.display("maze-container");
164
165
               // place treats inside the maze
               checkTreats();
166
167
               treatsAvailable = HIDDEN TREATS.length;
               // place the selected dog character
168
               placeDog(START_GAME_FORM.dog.value);
169
170
               startTimer();
171
          }
172
173
            * Start the timer and update the value every second.
174
175
176
           function startTimer() {
177
               timer = true;
               let start = Date.now();
178
179
               timerInterval = setInterval(function () {
180
                   let time = Date.now() - start;
181
                   let timeFormatted = new Date(time).toISOString().substring(14, 19);
182
                   TIME.innerText = timeFormatted;
183
                   timePassed = timeFormatted;
184
                   seconds = Math.round(time / 1000);
               }, 1000);
185
186
          }
187
188
            ^{st} Check which key was pressed on the keyboard.
189
            * Return without action if the value of isPlaying is false.
190
191
              @param {event} e
192
193
           function checkKey(e) {
194
               if (!isPlaying) {
195
                   return:
196
               } else {
197
                   e = e || window.event;
198
                   switch (e.keyCode) {
199
                       // up arrow
200
                       case 38:
                            e.preventDefault();
201
202
                            moveDog("up");
203
                           break;
                       // down arrow
204
205
                       case 40:
206
                            e.preventDefault();
207
                           moveDog("down");
208
                            break;
209
                       // left arrow
210
                       case 37:
211
                            e.preventDefault();
212
                           moveDog("left");
                            break;
213
214
                       // right arrow
```



version 2,13.6 There are 24 function (https://github.com/jshint/j: Function with the larg

About (madian is 0.5.

Documentation (/docs)

The most complex fu
Install (/install)
of 15 while the media
Contribute (/contribute)

Blog (/blog)

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder

https://jshint.com 3/9

```
215
                       case 39:
216
                            e.preventDefault();
217
                            moveDog("right");
218
                            break;
219
                       default:
220
                            break;
221
                   }
222
               }
           }
223
224
225
            * Check which arrow button was clicked.
226
            \ensuremath{^{*}} Return without action if the value of isPlaying is false.
227
            * @param {event} e
228
229
230
           function checkArrow(e) {
231
               if (!isPlaying) {
232
                   return;
               } else {
233
234
                   switch (e.target.id) {
235
                       // up arrow
                       case "up":
236
237
                            moveDog("up");
238
                            break;
239
                       // down arrow
                       case "down":
240
241
                            moveDog("down");
242
                            break;
243
                       // left arrow
244
                       case "left":
                            moveDog("left");
245
246
                            break;
247
                        // right arrow
248
                       case "right":
249
                            moveDog("right");
250
                            break;
251
                       default:
252
                            break;
253
                   }
254
               }
255
           }
256
257
258
            * Move the dog in the direction given as a parameter.
            * Return without action if the next position is a wall.
259
            * (Function adapted from https://www.the-art-of-web.com/mazing.js)
260
            * @param {string} direction
261
262
           function moveDog(direction) {
263
264
               let dogPos = findDog();
265
               let nextPos = new Position(dogPos.x, dogPos.y);
266
               let dir;
267
               let nextClass;
268
269
               if (dogPos) {
270
                   // set the value of dir to the current value, only change it if the dog is mov
271
                   if (this.maze[dogPos].classList.contains("right")) {
                       dir = "right";
272
                   } else if (this.maze[dogPos].classList.contains("left")) {
273
                       dir = "left";
274
275
                   switch (direction) {
276
                       case "up":
277
278
                            nextPos.x--;
279
                            break:
280
                       case "down":
281
                            nextPos.x++;
282
                            break;
                       case "left":
283
284
                            nextPos.y--;
285
                            dir = "left";
286
                            break;
```



version 2.13.6 There are 24 function (https://github.com/jshint/j: Function with the larg

About (Median is 0.5.

Documentation (/docs) has
The most complex fu
Install (/install)
of 15 while the media
Contribute (/contribute)

Blog (/blog)

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder

https://jshint.com 4/9

```
case "right":
287
288
                            nextPos.y++;
                            dir = "right";
289
290
                            break;
291
                        default:
292
                            break;
                   }
293
294
295
               if (this.maze[nextPos]) {
                   nextClass = this.maze[nextPos].className;
296
                   if (nextClass == "wall") {
297
298
                        return:
299
                   }
300
301
                   if (dir === "right") {
302
                        this.maze[nextPos].classList.remove("left");
303
                        this.maze[nextPos].classList.add("right");
                   } else if (dir === "left") {
304
305
                        this.maze[nextPos].classList.remove("right");
306
                        this.maze[nextPos].classList.add("left");
307
308
309
                    // remove treat from maze if it was collected
                   if (nextClass.includes("treat")) {
310
311
                        if (sound) {
                            soundTreat.play();
312
313
314
                        this.maze[nextPos].classList.remove("treat");
315
                        treatsCollected++;
316
                        TREATS.innerText = treatsCollected;
                   }
317
318
                   this.maze[dogPos].classList.remove("dog");
319
                   this.maze[nextPos].classList.add("dog", START_GAME_FORM.dog.value);
320
321
322
                   // end the game
                   if (nextClass.includes("exit")) {
323
324
                        endGame();
325
                   }
326
               }
327
           }
328
329
330
            * The user has reached the exit.
            * Stop the timer, open the congratulations modal.
331
332
333
           function endGame() {
               isPlaying = false;
334
335
               if (sound) {
                   soundPanting.pause();
336
337
                   soundEating.play();
               }
338
339
               clearInterval(timerInterval);
340
               timer = false;
               // set aria-hidden for the main content hidden behind the modal for accessibility
341
               MAIN.setAttribute('aria-hidden', 'true');
342
343
               // set overflow: hidden for the body to prevent scrolling the hidden content
               BODY.style.overflow = "hidden";
344
345
               // calculate and display the score
346
               let score = calculateScore();
347
               updateHighscore(score);
348
               // open congratulations modal
349
               CONGRATULATIONS MODAL.style.display = "block";
               CONGRATULATIONS_MODAL.setAttribute('aria-hidden', 'false');
CONGRATULATIONS_TEXT.innerText = "You collected " + treatsCollected + " out of " +
350
351
               CONGRATULATIONS_DOG.innerHTML = '<img src="assets/images/character_' + START_GAME_
352
353
               // calculate treat amount
354
               let bowl;
355
               let points = treatsAvailable / treatsCollected;
               if (points === 1) {
356
                   bowl = "extra_full";
357
               } else if (points == "Infinity") {
358
```



version 2,13.6 There are 24 function (https://github.com/jshint/j: Function with the larg

About (median is 0.5.

Documentation (/docs)

Install (/install)

of 15 while the media

Contribute (/contribute)

Blog (/blog)

One undefined va

- 150 MazeBuilder
- 153 MazeBuilder
- 156 MazeBuilder
- 159 MazeBuilder

https://jshint.com 5/9

```
bowl = "empty";
359
360
               } else if (points < 2) {</pre>
                    bowl = "full";
361
362
               } else {
363
                    bowl = "half full";
364
                                                                                                    g" a version 2,13.6
There are 24 function
(https://github.com/jshint/j:
Function with the larg
               CONGRATULATIONS_BOWL.innerHTML = '<img src="assets/images/bowl_' + bowl + '.svg" a
365
366
           }
367
368
                                                                                                     About (median is 0.5.
            st Calculate the score of the current attempt.
369
370
            * @returns {number} score
                                                                                                     Documentation (/docs) has
371
                                                                                                           The most complex fu
           function calculateScore() {
372
                                                                                                            of 15 while the media
373
               let treatsWeight = 0.6;
                                                                                                     Contribute (/contribute)
374
               let timeWeight = 0.4;
375
               let difficultyWeight = {
                                                                                                     Blog (/blog)
376
                    easy: 0,
                    medium: 0.1,
377
                                                                                                            One undefined va
378
                    hard: 0.2,
                    // same weight as hard for mobile maze as navigating with the buttons takes mo
379
                                                                                                            150 MazeBuilder
380
                    mobile: 0.2
381
               };
                                                                                                            153
                                                                                                                 MazeBuilder
382
                                                                                                                MazeBuilder
383
               let score = ((treatsCollected / treatsAvailable) * treatsWeight) + (1 / seconds) *
                                                                                                            156
384
               return Math.round(score * 100);
                                                                                                                 MazeBuilder
385
           }
386
387
            * Save the current highscore values to localStorage.
388
            * Display the highscore in the modal.
389
            * @param {number} score
390
            */
391
392
           function updateHighscore(score) {
393
               TABLE BODY.innerHTML = "";
               let highscoreData = [];
394
395
               let timestamp = new Date().getTime();
396
               let currentAttempt;
397
398
               let storedHighscoreData = localStorage.getItem("highscoreData");
399
400
               if (storedHighscoreData) {
401
                    highscoreData = JSON.parse(storedHighscoreData);
402
               }
403
404
               highscoreData.push({
                    dog: START_GAME_FORM.dog.value,
405
406
                    difficulty: START_GAME_FORM.difficulty.value,
407
                    score: score,
408
                    timestamp: timestamp
409
               });
410
411
               highscoreData.sort(function (a, b) {
412
                    return b.score - a.score;
413
               });
414
415
               if (highscoreData.length > 10) {
                    highscoreData.pop();
416
417
418
               for (let i = 0; i < highscoreData.length; i++) {</pre>
419
                    let newRow = TABLE_BODY.insertRow(TABLE_BODY.rows.length);
420
421
                    let posCell = newRow.insertCell(0);
422
                    let dogCell = newRow.insertCell(1);
423
                    let difficultyCell = newRow.insertCell(2);
424
                    let scoreCell = newRow.insertCell(3);
425
                    if (highscoreData[i].dog === "retriever") {
426
                        highscoreData[i].dog = "golden Retriever";
427
428
429
                    posCell.innerText = i + 1 + ".";
430
```

https://jshint.com 6/9

version 2,13.6 There are 24 function

The most complex fu

of 15 while the media

One undefined va

MazeBuilder

150 MazeBuilder

156 MazeBuilder

159 MazeBuilder

About (median is 0.5.

Contribute (/contribute)

Blog (/blog)

153

Documentation (/docs) has

```
431
                   dogCell.innerText = highscoreData[i].dog.charAt(0).toUpperCase() + highscoreDa
432
                   difficultyCell.innerText = highscoreData[i].difficulty.charAt(0).toUpperCase()
                   scoreCell.innerText = highscoreData[i].score;
433
434
435
                   if (highscoreData[i].timestamp === timestamp) {
436
                        currentAttempt = newRow;
437
               }
438
                                                                                                   (https://github.com/jshint/j:
Function with the larg
439
440
               if (currentAttempt) {
                   currentAttempt.classList.add("current-attempt");
441
442
443
               localStorage.setItem("highscoreData", JSON.stringify(highscoreData));
444
445
           }
446
447
            \ensuremath{^*} Delete the highscore information in localStorage and empty the table.
448
            */
449
450
           function resetHighscore() {
451
               localStorage.clear();
               TABLE_BODY.innerHTML = "";
452
453
           }
454
455
           /**
            st Reset game values and open menu.
456
            * @param {event} e
457
458
459
           function restartGame(e) {
460
               closeModal(e);
461
               soundEating.pause();
462
               treatsCollected = 0;
463
               TREATS.innerText = 0;
464
               TIME.innerText = "00:00";
               openMenu();
465
466
               MENU NAV.style.display = "none";
               MENU NAV.setAttribute('aria-hidden', 'true');
467
468
           }
469
470
            ^{st} Loop through the maze and find the current position of the dog character.
471
            * (Function adapted from https://www.the-art-of-web.com/mazing.js)
472
473
              @returns {object} dog position
474
475
           function findDog() {
476
               this.mazeContainer = document.getElementById("maze");
477
478
               for (let i = 0; i < this.mazeContainer.children.length; i++) {</pre>
479
                   for (let j = 0; j < this.mazeContainer.children[i].children.length; j++) {</pre>
480
                       let el = this.mazeContainer.children[i].children[j];
481
                        this.maze[new Position(i, j)] = el;
                       if (el.classList.contains("dog")) {
482
483
                            /* get position of the dog */
484
                            this.dogPos = new Position(i, j);
485
                       }
486
                   }
487
               }
488
489
               return this.dogPos;
490
           }
491
492
            * Add class names to the treats to use different treat images for each.
493
494
495
           function checkTreats() {
               let treatClasses = ["bone", "can", "cookies", "food", "water"];
496
               let counter = 0;
497
               for (let treat of HIDDEN_TREATS) {
498
499
                   treat.classList.add(treatClasses[counter]);
500
                   counter++:
                   if (counter > treatClasses.length - 1) {
501
502
                       counter = 0:
```

7/9 https://jshint.com

```
503
                   }
504
               }
505
           }
506
507
            \ensuremath{^{*}} Add the according dog as a class to the entrance div.
508
            * \mbox{\it @param {string}}\ \mbox{\it dog The selected value from the start game form.}
509
510
511
           function placeDog(dog) {
               let entrance = document.getElementsByClassName("entrance")[0];
512
               entrance.classList.add(dog, "dog", "right");
513
514
           }
515
516
            \ensuremath{^{*}} Open main menu, show back to game button and hide menu button.
517
518
519
           function openMenu() {
520
               isPlaying = false;
521
               MENU_CONTAINER.style.display = "block";
               MENU_CONTAINER.setAttribute('aria-hidden', 'false');
522
               MENU_NAV_BTN.style.display = "none";
523
               MENU NAV BTN.setAttribute('aria-hidden', 'true');
524
525
               BACK_NAV_BTN.style.display = "flex";
               BACK_NAV_BTN.setAttribute('aria-hidden', 'false');
526
527
               MAZE CONTAINER.style.display = "none";
               MAZE_CONTAINER.setAttribute('aria-hidden', 'true');
528
529
           }
530
531
532
            * Close main menu, hide back to game button and show menu button.
            */
533
534
           function closeMenu() {
535
               isPlaying = true;
536
               MENU CONTAINER.style.display = "none";
537
               MENU_CONTAINER.setAttribute('aria-hidden', 'true');
               BACK_NAV_BTN.style.display = "none";
538
               BACK NAV BTN.setAttribute('aria-hidden', 'true');
539
               MENU_NAV_BTN.style.display = "flex";
540
               MENU_NAV_BTN.setAttribute('aria-hidden', 'false');
541
542
               MAZE_CONTAINER.style.display = "block";
543
               MAZE_CONTAINER.setAttribute('aria-hidden', 'false');
544
           }
545
546
            ^{st} Only show the mobile option in the start game form if the screen width is too small
547
548
549
           function handleRadioButtons() {
550
               let screenWidth = window.innerWidth;
551
552
               if (screenWidth < 600) {
553
                   document.getElementById('mobile').checked = true;
               } else {
554
555
                   document.getElementById('medium').checked = true;
556
           }
557
558
559
            * Turn the sound of the game on and off.
560
561
              @param {event} e
562
           function toggleSound(e) {
563
564
               let toggle = e.target;
565
               let state = toggle.getAttribute('aria-checked');
566
               if (state === 'true') {
567
568
                   toggle.setAttribute('aria-checked', false);
                   sound = false;
569
570
                   soundPanting.pause();
571
               } else {
                   toggle.setAttribute('aria-checked', true);
572
                   sound = true;
573
574
                   soundPanting.play();
```



version 2.13.6 There are 24 function (https://github.com/jshint/j: Function with the larg

About (median is 0.5.

Documentation (/docs)

Install (/install)

of 15 while the media

Contribute (/contribute)

Blog (/blog)

One undefined va

- 150 MazeBuilder
- 153 MazeBuilder
- 156 MazeBuilder
- 159 MazeBuilder

https://jshint.com

```
575
                 }
576
577
578
            // event listeners
            // modal listeners
579
            RULES_MODAL_BTN.addEventListener("click", openModal);
580
                                                                                                                version 2,13.6
There are 24 function
(https://github.com/jshint/j:
Function with the larc
            RULES_MODAL_CLOSE.addEventListener("click", closeModal);
581
582
            START_GAME_MODAL_BTN.addEventListener("click", openModal);
583
            START_GAME_MODAL_CLOSE.addEventListener("click", closeModal);
584
            // game and navigation listeners
                                                                                                                 About (madian is 0.5.
            START_GAME_FORM.addEventListener("submit", startGame);
585
            MENU_NAV_BTN.addEventListener("click", openMenu);
586
                                                                                                                 Documentation (/docs) has
            BACK_NAV_BTN.addEventListener("click", closeMenú);
587
                                                                                                                        The most complex fu (/install) of 15 while the media
588
            SOUND_BTN.addEventListener("click", toggleSound);
589
            // key press listener
                                                                                                                 Contribute (/contribute)
            document.onkeydown = checkKey;
590
            // control arrows listeners
591
                                                                                                                 Blog (/blog)
            CONTROL_UP.addEventListener("click", checkArrow);
592
            CONTROL_LEFT.addEventListener("click", checkArrow);
CONTROL_DOWN.addEventListener("click", checkArrow);
CONTROL_RIGHT.addEventListener("click", checkArrow);
593
                                                                                                                        One undefined va
594
595
                                                                                                                         150 MazeBuilder
596
            // reset and restart
            RESET_HIGHSCORE.addEventListener("click", resetHighscore);
597
                                                                                                                         153 MazeBuilder
598
            RESTART.addEventListener("click", restartGame);
                                                                                                                         156 MazeBuilder
599
       });
       -∢-
```

159 MazeBuilder

https://jshint.com 9/9