```
/* jshint esversion: 6 */

// global constants
// modals
const RULES_MODAL = document.getElementById("modal-rules");
const RULES_MODAL_BTN = document.getElementById("btn-rules");
const RULES_MODAL_CLOSE = document.getElementById("close-rules");
const START_GAME_MODAL = document.getElementById("modal-start-game");
const START_GAME_MODAL_BTN = document.getElementById("btn-start-game");
const START_GAME_MODAL_CLOSE = document.getElementById("close-start-game");
const CONGRATULATIONS_MODAL = document.getElementById("modal-congratulations");
const CONGRATULATIONS_TEXT = document.getElementById("congratulations-text");
const CONGRATULATIONS_DOG = document.getElementById("congratulations-dog");
const CONGRATULATIONS_BOWL = document.getElementById("congratulations-bowl");
const START_GAME_FORM = document.getElementById("start-game-form");
// dom elements
const MAIN = document.getElementsByTagName("main")[0];
const BODY = document.getElementsByTagName("body")[0];
const MENU_CONTAINER = document.getElementById("menu-container");
const MENU_NAV = document.getElementById("game-menu");
const MENU_NAV_BTN = document.getElementById("btn-menu");
const BACK_NAV_BTN = document.getElementById("btn-back");
const MAZE_CONTAINER = document.getElementById("outer-maze-container");
const TREATS = document.getElementById("treats");
const TIME = document.getElementById("time");
const HIDDEN_TREATS = document.getElementsByClassName("treat");
const SOUND_BTN = document.getElementById('sound');
// control arrows
const CONTROL_UP = document.getElementById("up");
const CONTROL_LEFT = document.getElementById("left");
const CONTROL_DOWN = document.getElementById("down");
const CONTROL_RIGHT = document.getElementById("right");
// highscore table
const TABLE = document.getElementById("highscore");
const TABLE_BODY = TABLE.getElementsByTagName("tbody")[0];
const RESET_HIGHSCORE = document.getElementById("reset-highscore");
const RESTART = document.getElementById("restart");

// global variables needed during the whole game
let isPlaying = false;
let treatsCollected = 0;
let treatsAvailable;
let timePassed;
let seconds;
let timer = false;
let timerInterval;
// sounds
let sound = false;
let soundTreat = new Audio("assets/sounds/treat.mp3");
let soundEating = new Audio("assets/sounds/eating.mp3");
let soundPanting = new Audio("assets/sounds/panting.mp3");
soundEating.loop = true;
soundPanting.loop = true;

// necessary to move the dog character
// adapted from https://www.the-art-of-web.com/mazing.js
let Position = function (x, y) {
    this.x = x;
    this.y = y;
};
Position.prototype.toString = function () {
    return this.x + ":" + this.y;
};

document.addEventListener("DOMContentLoaded", function () {
    /**
     * Open the modal according to the given event.
     * @param {event} e
     */
    function openModal(e) {
```

CONFIGURE

version 2.13.6
(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function
Function with the larg
median is 0.5.

Largest function has

The most complex fu
of 15 while the media

One undefined va

150	MazeBuilder
153	MazeBuilder
156	MazeBuilder
159	MazeBuilder

```
            isPlaying = false;
            // set aria-hidden for the main content hidden behind the modal for accessibility
            MAIN.setAttribute('aria-hidden', 'true');
            // set overflow: hidden for the body to prevent scrolling the hidden content
            BODY.style.overflow = "hidden";
            // open the model according to the clicked button
            switch (e.target) {
                // open rules modal
                case RULES_MODAL_BTN:
                    RULES_MODAL.style.display = "block";
                    RULES_MODAL.setAttribute('aria-hidden', 'false');
                    break;
                // open start game modal
                case START_GAME_MODAL_BTN:
                    START_GAME_MODAL.style.display = "block";
                    START_GAME_MODAL.setAttribute('aria-hidden', 'false');
                    handleRadioButtons();
                    break;
                default:
                    break;
            }
        }

        /**
         * Close the modal according to the given event.
         * @param {event} e
         */
        function closeModal(e) {
            MAIN.setAttribute('aria-hidden', 'false');
            BODY.style.overflow = "unset";
            // close the model according to the clicked button
            switch (e.target) {
                // close rules modal
                case RULES_MODAL_CLOSE:
                    RULES_MODAL.style.display = "none";
                    RULES_MODAL.setAttribute('aria-hidden', 'true');
                    break;
                // close start game modal
                case START_GAME_MODAL_CLOSE:
                    START_GAME_MODAL.style.display = "none";
                    START_GAME_MODAL.setAttribute('aria-hidden', 'true');
                    break;
                // game started - close modal
                case START_GAME_FORM:
                    START_GAME_MODAL.style.display = "none";
                    START_GAME_MODAL.setAttribute('aria-hidden', 'true');
                    break;
                // game restarted - close congratulations modal
                case RESTART:
                    CONGRATULATIONS_MODAL.style.display = "none";
                    CONGRATULATIONS_MODAL.setAttribute('aria-hidden', 'true');
                    break;
                default:
                    break;
            }
        }

        /**
         * Start the game with the selected options from the form.
         * @param {event} e
         */
        function startGame(e) {
            if (timer) {
                clearInterval(timerInterval);
            }
            isPlaying = true;
            if (sound) {
                soundPanting.play();
            }

            e.preventDefault();
            closeModal(e);
```

JSHint

version 2.13.6

(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function

Function with the larg

median is 0.5.

Largest function has

The most complex fu

of 15 while the media

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder

```
143             MENU_NAV.style.display = "flex";
144             MENU_NAV.setAttribute('aria-hidden', 'false');
145             closeMenu();
146             // initialize the maze based on the selected difficulty
147             let Maze;
148             switch (START_GAME_FORM.difficulty.value) {
149                 case "easy":
150                     Maze = new MazeBuilder(8, 4);
151                     break;
152                 case "medium":
153                     Maze = new MazeBuilder(10, 5);
154                     break;
155                 case "hard":
156                     Maze = new MazeBuilder(14, 8);
157                     break;
158                 case "mobile":
159                     Maze = new MazeBuilder(5, 6);
160                     break;
161                 default:
162                     break;
163             }
164             Maze.display("maze-container");
165             // place treats inside the maze
166             checkTreats();
167             treatsAvailable = HIDDEN_TREATS.length;
168             // place the selected dog character
169             placeDog(START_GAME_FORM.dog.value);
170             startTimer();
171         }
172 
173         /**
174          * Start the timer and update the value every second.
175          */
176         function startTimer() {
177             timer = true;
178             let start = Date.now();
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);
185             }, 1000);
186         }
187 
188         /**
189          * Check which key was pressed on the keyboard.
190          * Return without action if the value of isPlaying is false.
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:
201                         e.preventDefault();
202                         moveDog("up");
203                         break;
204                     // down arrow
205                     case 40:
206                         e.preventDefault();
207                         moveDog("down");
208                         break;
209                     // left arrow
210                     case 37:
211                         e.preventDefault();
212                         moveDog("left");
213                         break;
214                     // right arrow
```

version 2.13.6

(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function

Function with the larg

median is 0.5.

Largest function has

The most complex fu

of 15 while the media

One undefined va

150	MazeBuilder
153	MazeBuilder
156	MazeBuilder
159	MazeBuilder

```
                case 39:
                    e.preventDefault();
                    moveDog("right");
                    break;
                default:
                    break;
            }
        }
    }

    /**
     * Check which arrow button was clicked.
     * Return without action if the value of isPlaying is false.
     * @param {event} e
     */
    function checkArrow(e) {
        if (!isPlaying) {
            return;
        } else {
            switch (e.target.id) {
                // up arrow
                case "up":
                    moveDog("up");
                    break;
                // down arrow
                case "down":
                    moveDog("down");
                    break;
                // left arrow
                case "left":
                    moveDog("left");
                    break;
                // right arrow
                case "right":
                    moveDog("right");
                    break;
                default:
                    break;
            }
        }
    }

    /**
     * Move the dog in the direction given as a parameter.
     * Return without action if the next position is a wall.
     * (Function adapted from https://www.the-art-of-web.com/mazing.js)
     * @param {string} direction
     */
    function moveDog(direction) {
        let dogPos = findDog();
        let nextPos = new Position(dogPos.x, dogPos.y);
        let dir;
        let nextClass;

        if (dogPos) {
            // set the value of dir to the current value, only change it if the dog is mov
            if (this.maze[dogPos].classList.contains("right")) {
                dir = "right";
            } else if (this.maze[dogPos].classList.contains("left")) {
                dir = "left";
            }
            switch (direction) {
                case "up":
                    nextPos.x--;
                    break;
                case "down":
                    nextPos.x++;
                    break;
                case "left":
                    nextPos.y--;
                    dir = "left";
                    break;
```

version 2.13.6 (https://github.com/jshint/j

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function

Function with the larg

median is 0.5.

Largest function has

The most complex fu

of 15 while the media

One undefined va

150	MazeBuilder
153	MazeBuilder
156	MazeBuilder
159	MazeBuilder

```
                case "right":
                    nextPos.y++;
                    dir = "right";
                    break;
                default:
                    break;
            }
        }
        if (this.maze[nextPos]) {
            nextClass = this.maze[nextPos].className;
            if (nextClass == "wall") {
                return;
            }

            if (dir === "right") {
                this.maze[nextPos].classList.remove("left");
                this.maze[nextPos].classList.add("right");
            } else if (dir === "left") {
                this.maze[nextPos].classList.remove("right");
                this.maze[nextPos].classList.add("left");
            }

            // remove treat from maze if it was collected
            if (nextClass.includes("treat")) {
                if (sound) {
                    soundTreat.play();
                }
                this.maze[nextPos].classList.remove("treat");
                treatsCollected++;
                TREATS.innerText = treatsCollected;
            }

            this.maze[dogPos].classList.remove("dog");
            this.maze[nextPos].classList.add("dog", START_GAME_FORM.dog.value);

            // end the game
            if (nextClass.includes("exit")) {
                endGame();
            }
        }
    }

    /**
     * The user has reached the exit.
     * Stop the timer, open the congratulations modal.
     */
    function endGame() {
        isPlaying = false;
        if (sound) {
            soundPanting.pause();
            soundEating.play();
        }
        clearInterval(timerInterval);
        timer = false;
        // set aria-hidden for the main content hidden behind the modal for accessibility
        MAIN.setAttribute('aria-hidden', 'true');
        // set overflow: hidden for the body to prevent scrolling the hidden content
        BODY.style.overflow = "hidden";
        // calculate and display the score
        let score = calculateScore();
        updateHighscore(score);
        // open congratulations modal
        CONGRATULATIONS_MODAL.style.display = "block";
        CONGRATULATIONS_MODAL.setAttribute('aria-hidden', 'false');
        CONGRATULATIONS_TEXT.innerText = "You collected " + treatsCollected + " out of " +
        CONGRATULATIONS_DOG.innerHTML = '<img src="assets/images/character_' + START_GAME_
        // calculate treat amount
        let bowl;
        let points = treatsAvailable / treatsCollected;
        if (points === 1) {
            bowl = "extra_full";
        } else if (points == "Infinity") {
```

JSHint

version 2.13.6 (https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function

Function with the larg

median is 0.5.

Largest function has :

The most complex fu

of 15 while the media

One undefined va

150	MazeBuilder
153	MazeBuilder
156	MazeBuilder
159	MazeBuilder

```
            bowl = "empty";
        } else if (points < 2) {
            bowl = "full";
        } else {
            bowl = "half_full";
        }
        CONGRATULATIONS_BOWL.innerHTML = '<img src="assets/images/bowl_' + bowl + '.svg" a
    }

    /**
     * Calculate the score of the current attempt.
     * @returns {number} score
     */
    function calculateScore() {
        let treatsWeight = 0.6;
        let timeWeight = 0.4;
        let difficultyWeight = {
            easy: 0,
            medium: 0.1,
            hard: 0.2,
            // same weight as hard for mobile maze as navigating with the buttons takes mo
            mobile: 0.2
        };

        let score = ((treatsCollected / treatsAvailable) * treatsWeight) + (1 / seconds) *
        return Math.round(score * 100);
    }

    /**
     * Save the current highscore values to localStorage.
     * Display the highscore in the modal.
     * @param {number} score
     */
    function updateHighscore(score) {
        TABLE_BODY.innerHTML = "";
        let highscoreData = [];
        let timestamp = new Date().getTime();
        let currentAttempt;

        let storedHighscoreData = localStorage.getItem("highscoreData");

        if (storedHighscoreData) {
            highscoreData = JSON.parse(storedHighscoreData);
        }

        highscoreData.push({
            dog: START_GAME_FORM.dog.value,
            difficulty: START_GAME_FORM.difficulty.value,
            score: score,
            timestamp: timestamp
        });

        highscoreData.sort(function (a, b) {
            return b.score - a.score;
        });

        if (highscoreData.length > 10) {
            highscoreData.pop();
        }

        for (let i = 0; i < highscoreData.length; i++) {
            let newRow = TABLE_BODY.insertRow(TABLE_BODY.rows.length);
            let posCell = newRow.insertCell(0);
            let dogCell = newRow.insertCell(1);
            let difficultyCell = newRow.insertCell(2);
            let scoreCell = newRow.insertCell(3);

            if (highscoreData[i].dog === "retriever") {
                highscoreData[i].dog = "golden Retriever";
            }

            posCell.innerText = i + 1 + ".";
```

version 2.13.6
(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function
Function with the larg
median is 0.5.

Largest function has

The most complex fu
of 15 while the media

One undefined va

150	MazeBuilder
153	MazeBuilder
156	MazeBuilder
159	MazeBuilder

```
            dogCell.innerText = highscoreData[i].dog.charAt(0).toUpperCase() + highscoreDa
            difficultyCell.innerText = highscoreData[i].difficulty.charAt(0).toUpperCase()
            scoreCell.innerText = highscoreData[i].score;

            if (highscoreData[i].timestamp === timestamp) {
                currentAttempt = newRow;
            }
        }

        if (currentAttempt) {
            currentAttempt.classList.add("current-attempt");
        }

        localStorage.setItem("highscoreData", JSON.stringify(highscoreData));
    }

    /**
     * Delete the highscore information in localStorage and empty the table.
     */
    function resetHighscore() {
        localStorage.clear();
        TABLE_BODY.innerHTML = "";
    }

    /**
     * Reset game values and open menu.
     * @param {event} e
     */
    function restartGame(e) {
        closeModal(e);
        soundEating.pause();
        treatsCollected = 0;
        TREATS.innerText = 0;
        TIME.innerText = "00:00";
        openMenu();
        MENU_NAV.style.display = "none";
        MENU_NAV.setAttribute('aria-hidden', 'true');
    }

    /**
     * Loop through the maze and find the current position of the dog character.
     * (Function adapted from https://www.the-art-of-web.com/mazing.js)
     * @returns {object} dog position
     */
    function findDog() {
        this.mazeContainer = document.getElementById("maze");

        for (let i = 0; i < this.mazeContainer.children.length; i++) {
            for (let j = 0; j < this.mazeContainer.children[i].children.length; j++) {
                let el = this.mazeContainer.children[i].children[j];
                this.maze[new Position(i, j)] = el;
                if (el.classList.contains("dog")) {
                    /* get position of the dog */
                    this.dogPos = new Position(i, j);
                }
            }
        }

        return this.dogPos;
    }

    /**
     * Add class names to the treats to use different treat images for each.
     */
    function checkTreats() {
        let treatClasses = ["bone", "can", "cookies", "food", "water"];
        let counter = 0;
        for (let treat of HIDDEN_TREATS) {
            treat.classList.add(treatClasses[counter]);
            counter++;
            if (counter > treatClasses.length - 1) {
                counter = 0;
```

version 2.13.6
(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function
Function with the larg
median is 0.5.

Largest function has
The most complex fu
of 15 while the media

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder

```
            }
        }
    }

    /**
     * Add the according dog as a class to the entrance div.
     * @param {string} dog The selected value from the start game form.
     */
    function placeDog(dog) {
        let entrance = document.getElementsByClassName("entrance")[0];
        entrance.classList.add(dog, "dog", "right");
    }

    /**
     * Open main menu, show back to game button and hide menu button.
     */
    function openMenu() {
        isPlaying = false;
        MENU_CONTAINER.style.display = "block";
        MENU_CONTAINER.setAttribute('aria-hidden', 'false');
        MENU_NAV_BTN.style.display = "none";
        MENU_NAV_BTN.setAttribute('aria-hidden', 'true');
        BACK_NAV_BTN.style.display = "flex";
        BACK_NAV_BTN.setAttribute('aria-hidden', 'false');
        MAZE_CONTAINER.style.display = "none";
        MAZE_CONTAINER.setAttribute('aria-hidden', 'true');
    }

    /**
     * Close main menu, hide back to game button and show menu button.
     */
    function closeMenu() {
        isPlaying = true;
        MENU_CONTAINER.style.display = "none";
        MENU_CONTAINER.setAttribute('aria-hidden', 'true');
        BACK_NAV_BTN.style.display = "none";
        BACK_NAV_BTN.setAttribute('aria-hidden', 'true');
        MENU_NAV_BTN.style.display = "flex";
        MENU_NAV_BTN.setAttribute('aria-hidden', 'false');
        MAZE_CONTAINER.style.display = "block";
        MAZE_CONTAINER.setAttribute('aria-hidden', 'false');
    }

    /**
     * Only show the mobile option in the start game form if the screen width is too small
     */
    function handleRadioButtons() {
        let screenWidth = window.innerWidth;

        if (screenWidth < 600) {
            document.getElementById('mobile').checked = true;
        } else {
            document.getElementById('medium').checked = true;
        }
    }

    /**
     * Turn the sound of the game on and off.
     * @param {event} e
     */
    function toggleSound(e) {
        let toggle = e.target;
        let state = toggle.getAttribute('aria-checked');

        if (state === 'true') {
            toggle.setAttribute('aria-checked', false);
            sound = false;
            soundPanting.pause();
        } else {
            toggle.setAttribute('aria-checked', true);
            sound = true;
            soundPanting.play();
```

version 2.13.6
(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function
Function with the larg
median is 0.5.

Largest function has

The most complex fu
of 15 while the media

One undefined va

150	MazeBuilder
153	MazeBuilder
156	MazeBuilder
159	MazeBuilder

```
        }
    }

    // event listeners
    // modal listeners
    RULES_MODAL_BTN.addEventListener("click", openModal);
    RULES_MODAL_CLOSE.addEventListener("click", closeModal);
    START_GAME_MODAL_BTN.addEventListener("click", openModal);
    START_GAME_MODAL_CLOSE.addEventListener("click", closeModal);
    // game and navigation listeners
    START_GAME_FORM.addEventListener("submit", startGame);
    MENU_NAV_BTN.addEventListener("click", openMenu);
    BACK_NAV_BTN.addEventListener("click", closeMenu);
    SOUND_BTN.addEventListener("click", toggleSound);
    // key press listener
    document.onkeydown = checkKey;
    // control arrows listeners
    CONTROL_UP.addEventListener("click", checkArrow);
    CONTROL_LEFT.addEventListener("click", checkArrow);
    CONTROL_DOWN.addEventListener("click", checkArrow);
    CONTROL_RIGHT.addEventListener("click", checkArrow);
    // reset and restart
    RESET_HIGHSCORE.addEventListener("click", resetHighscore);
    RESTART.addEventListener("click", restartGame);
});
```

version 2.13.6
(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 24 function
Function with the larg
median is 0.5.

Largest function has

The most complex fu
of 15 while the media

One undefined va

150 MazeBuilder

153 MazeBuilder

156 MazeBuilder

159 MazeBuilder