

Создайте папку «Zmeika\_game» в вашем проекте

Реализуйте игру «Змейка»

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Snake</title>
  <link rel="stylesheet" href="MyStyles.css">
  <script src="myJS.js"></script>
</head>
<body>
<div class="wrap">
  <div id="snake-field"></div>
  <div class="buttons">
    <div id="snake-start">Старт</div>
    <div id="snake-renew">Новая игра</div>
  </div>
</div>
</body>
</html>
```

Стили

```
#snake-field{
  display: inline-block;
}
#snake-renew,#snake-start{
  width: 150px;
  padding: 15px;
  color: white;
  background-color: green;
  text-align: center;
  cursor: pointer;
  float: left;
  margin-right: 10px;
}
.game-table{
  border-collapse: collapse;
  background-color: #3e3e3e;
}
td.game-table-cell{
  width: 15px;
  height: 15px;
  border: 1px dotted #eee;
}
.snake-unit{
  background-color: white;
}
.food-unit{
  background-color: red;
}
.buttons{
  display: inline-block;
}
.wrap{
  margin: 30px auto 0;
  text-align: center;
}
```

```
// Глобальные переменные:
var FIELD_SIZE_X = 20; // строки
var FIELD_SIZE_Y = 20; // столбцы
var SNAKE_SPEED = 300; // Интервал между перемещениями змейки
var snake = []; // Сама змейка
var direction = 'y+'; // Направление движения змейки
var gameIsRunning = false; // Запущена ли игра
var snake_timer; // Таймер змейки
var food_timer; // Таймер для еды
var score = 0; // Результат

function init() {
    prepareGameField(); // Генерация поля

    var wrap = document.getElementsByClassName('wrap')[0];

    wrap.style.width = '400px';
    // События кнопок Старт и Новая игра
    document.getElementById('snake-start').addEventListener('click', startGame);
    document.getElementById('snake-renew').addEventListener('click', refreshGame);

    // Отслеживание клавиш клавиатуры
    addEventListener('keydown', changeDirection);
}
```

```
/**
 * Функция генерации игрового поля
 */
function prepareGameField() {
    // Создаём таблицу
    var game_table = document.createElement('table');
    game_table.setAttribute('class', 'game-table');

    // Генерация ячеек игровой таблицы
    for (var i = 0; i < FIELD_SIZE_X; i++) {
        // Создание строки
        var row = document.createElement('tr');
        row.className = 'game-table-row row-' + i;

        for (var j = 0; j < FIELD_SIZE_Y; j++) {
            // Создание ячейки
            var cell = document.createElement('td');
            cell.className = 'game-table-cell cell-' + i + '-' + j;

            row.appendChild(cell); // Добавление ячейки
        }
        game_table.appendChild(row); // Добавление строки
    }

    document.getElementById('snake-field').appendChild(game_table); // Добавление таблицы
}
```

```

/**
 * Старт игры
 */
function startGame() {
    gameIsRunning = true;
    respawn(); // создали змейку

    snake_timer = setInterval(move, SNAKE_SPEED);
    setTimeout(createFood, 5000);
}

/**
 * Функция расположения змейки на игровом поле
 */
function respawn() {
    // Змейка - массив td
    // Стартовая длина змейки = 2

    // Respawn змейки из центра
    var start_coord_x = Math.floor(FIELD_SIZE_X / 2);
    var start_coord_y = Math.floor(FIELD_SIZE_Y / 2);

    // Голова змейки
    var snake_head = document.getElementsByClassName('cell-' + start_coord_y + '-' + start_coord_x)[0];
    snake_head.setAttribute('class', snake_head.getAttribute('class') + ' snake-unit');
    // Тело змейки
    var snake_tail = document.getElementsByClassName('cell-' + (start_coord_y - 1) + '-' + start_coord_x)[0];
    snake_tail.setAttribute('class', snake_tail.getAttribute('class') + ' snake-unit');

    snake.push(snake_head);
    snake.push(snake_tail);
}

```

```

/**
 * Движение змейки
 */
function move() {
    // console.log('move', direction);
    // Сборка классов
    var snake_head_classes = snake[snake.length - 1].getAttribute('class').split(' ');

    // Сдвиг головы
    var new_unit;
    var snake_coords = snake_head_classes[1].split('-');
    var coord_y = parseInt(snake_coords[1]);
    var coord_x = parseInt(snake_coords[2]);

    // Определяем новую точку
    if (direction == 'x-') {
        new_unit = document.getElementsByClassName('cell-' + (coord_y) + '-' + (coord_x - 1))[0];
    }
    else if (direction == 'x+') {
        new_unit = document.getElementsByClassName('cell-' + (coord_y) + '-' + (coord_x + 1))[0];
    }
    else if (direction == 'y+') {
        new_unit = document.getElementsByClassName('cell-' + (coord_y - 1) + '-' + (coord_x))[0];
    }
    else if (direction == 'y-') {
        new_unit = document.getElementsByClassName('cell-' + (coord_y + 1) + '-' + (coord_x))[0];
    }
}

```

```

// Проверки
// 1) new_unit не часть змейки
// 2) Змейка не ушла за границу поля
//console.log(new_unit);
if (!isSnakeUnit(new_unit) && new_unit !== undefined) {
    // Добавление новой части змейки
    new_unit.setAttribute('class', new_unit.getAttribute('class') + ' snake-unit');
    snake.push(new_unit);

    // Проверяем, надо ли убрать хвост
    if (!haveFood(new_unit)) {
        // Находим хвост
        var removed = snake.splice(0, 1)[0];
        var classes = removed.getAttribute('class').split(' ');

        // удаляем хвост
        removed.setAttribute('class', classes[0] + ' ' + classes[1]);
    }
}
else {
    finishTheGame();
}
}

```

```

/**
 * Проверка на змейку
 * @param unit
 * @returns {boolean}
 */
function isSnakeUnit(unit) {
    var check = false;

    if (snake.includes(unit)) {
        check = true;
    }
    return check;
}

/**
 * проверка на еду
 * @param unit
 * @returns {boolean}
 */
function haveFood(unit) {
    var check = false;

    var unit_classes = unit.getAttribute('class').split(' ');

    // Если еда
    if (unit_classes.includes('food-unit')) {
        check = true;
        createFood();

        score++;
    }
    return check;
}

```

```

/**
 * Создание еды
 */
function createFood() {
    var foodCreated = false;

    while (!foodCreated) {
        // рандом
        var food_x = Math.floor(Math.random() * FIELD_SIZE_X);
        var food_y = Math.floor(Math.random() * FIELD_SIZE_Y);

        var food_cell = document.getElementsByClassName('cell-' + food_y + '-' + food_x)[0];
        var food_cell_classes = food_cell.getAttribute('class').split(' ');

        // проверка на змейку
        if (!food_cell_classes.includes('snake-unit')) {
            var classes = '';
            for (var i = 0; i < food_cell_classes.length; i++) {
                classes += food_cell_classes[i] + ' ';
            }

            food_cell.setAttribute('class', classes + 'food-unit');
            foodCreated = true;
        }
    }
}

```

```

/**
 * Изменение направления движения змейки
 * @param e - событие
 */
function changeDirection(e) {
    console.log(e);
    switch (e.keyCode) {
        case 37: // Клавиша влево
            if (direction !== 'x+') {
                direction = 'x-'
            }
            break;
        case 38: // Клавиша вверх
            if (direction !== 'y-') {
                direction = 'y+'
            }
            break;
        case 39: // Клавиша вправо
            if (direction !== 'x-') {
                direction = 'x+'
            }
            break;
        case 40: // Клавиша вниз
            if (direction !== 'y+') {
                direction = 'y-'
            }
            break;
    }
}

```

```
/**
 * Функция завершения игры
 */
function finishTheGame() {
    gameIsRunning = false;
    clearInterval(snake_timer);
    alert('Вы проиграли! Ваш результат: ' + score.toString());
}

/**
 * Новая игра
 */
function refreshGame() {
    location.reload();
}

// Инициализация
window.onload = init;
```

Задание. Преобразуйте игру, увеличьте игральное поле, измените цвета, скорость игры

Выложите а Github и портфолио