|  |  |
| --- | --- |
| AI |  |
| var FIELD\_SIZE\_X = 20;  var FIELD\_SIZE\_Y = 20;  var SNAKE\_SPEED = 300;  var snake = [];  var direction = 'y+'; // y+ — вверх (уменьшение Y)  var gameIsRunning = false;  var snake\_timer;  var score = 0;  function init() {  prepareGameField();  var wrap = document.getElementsByClassName('wrap')[0];  wrap.style.width = (FIELD\_SIZE\_Y \* 20) + 'px';  document.getElementById('snake-start').addEventListener('click', startGame);  document.getElementById('snake-renew').addEventListener('click', refreshGame);  addEventListener('keydown', changeDirection);  }  function prepareGameField() {  var game\_table = document.createElement('table');  game\_table.className = 'game-table';  for (var i = 0; i < FIELD\_SIZE\_X; i++) {  var row = document.createElement('tr');  row.className = 'game-table-row row-' + i;  for (var j = 0; j < FIELD\_SIZE\_Y; j++) {  var cell = document.createElement('td');  cell.className = 'cell-' + i + '-' + j; // формат cell-Y-X  row.appendChild(cell);  }  game\_table.appendChild(row);  }  var field = document.getElementById('snake-field');  field.innerHTML = '';  field.appendChild(game\_table);  }  function startGame() {  if (gameIsRunning) return;  gameIsRunning = true;  score = 0;  // очищаем старые состояния, если есть  var prevSnake = document.querySelectorAll('.snake-unit');  prevSnake.forEach(function(c){ c.classList.remove('snake-unit'); });  var prevFood = document.querySelectorAll('.food-unit');  prevFood.forEach(function(c){ c.classList.remove('food-unit'); });  snake = [];  respawn();  createFood();  snake\_timer = setInterval(move, SNAKE\_SPEED);  }  function respawn() {  var startY = Math.floor(FIELD\_SIZE\_X / 2);  var startX = Math.floor(FIELD\_SIZE\_Y / 2);  var head = document.getElementsByClassName('cell-' + startY + '-' + startX)[0];  var tailY = startY + 1;  if (tailY >= FIELD\_SIZE\_X) tailY = startY - 1;  var tail = document.getElementsByClassName('cell-' + tailY + '-' + startX)[0];  tail.classList.add('snake-unit');  head.classList.add('snake-unit');  snake.push(tail); // первый элемент — хвост  snake.push(head); // последний — голова  }  function move() {  if (!gameIsRunning) return;  var head = snake[snake.length - 1];  var classes = head.className.split(' ');  var coordClass = null;  for (var i = 0; i < classes.length; i++) {  if (classes[i].indexOf('cell-') === 0) { coordClass = classes[i]; break; }  }  if (!coordClass) { finishTheGame(); return; }  var parts = coordClass.split('-'); // ["cell", "Y", "X"]  var coordY = parseInt(parts[1], 10);  var coordX = parseInt(parts[2], 10);  var newY = coordY, newX = coordX;  if (direction === 'x-') newX = coordX - 1;  else if (direction === 'x+') newX = coordX + 1;  else if (direction === 'y+') newY = coordY - 1;  else if (direction === 'y-') newY = coordY + 1;  if (newX < 0 || newX >= FIELD\_SIZE\_Y || newY < 0 || newY >= FIELD\_SIZE\_X) {  finishTheGame();  return;  }  var newUnit = document.getElementsByClassName('cell-' + newY + '-' + newX)[0];  if (!newUnit || isSnakeUnit(newUnit)) {  finishTheGame();  return;  }  newUnit.classList.add('snake-unit');  snake.push(newUnit);  if (!haveFood(newUnit)) {  var removed = snake.shift();  removed.classList.remove('snake-unit');  }  }  function isSnakeUnit(unit) {  return snake.indexOf(unit) !== -1;  }  function haveFood(unit) {  if (unit.classList.contains('food-unit')) {  unit.classList.remove('food-unit');  createFood();  score++;  return true;  }  return false;  }  function createFood() {  var foodCreated = false;  while (!foodCreated) {  var foodY = Math.floor(Math.random() \* FIELD\_SIZE\_X);  var foodX = Math.floor(Math.random() \* FIELD\_SIZE\_Y);  var cell = document.getElementsByClassName('cell-' + foodY + '-' + foodX)[0];  if (!cell.classList.contains('snake-unit') && !cell.classList.contains('food-unit')) {  cell.classList.add('food-unit');  foodCreated = true;  }  }  }  function changeDirection(e) {  switch (e.keyCode) {  case 37: if (direction !== 'x+') direction = 'x-'; break; // left  case 38: if (direction !== 'y-') direction = 'y+'; break; // up  case 39: if (direction !== 'x-') direction = 'x+'; break; // right  case 40: if (direction !== 'y+') direction = 'y-'; break; // down  }  }  function finishTheGame() {  gameIsRunning = false;  clearInterval(snake\_timer);  alert('Вы проиграли! Ваш результат: ' + score.toString());  }  function refreshGame() {  location.reload();  }  window.onload = init; | var FIELD\_SIZE\_X = 20;//lines  var FIELD\_SIZE\_Y = 20; //columns  var SNAKE\_SPEED = 300; //time between snake's movements  var snake = []; // Snake itself  var direction = 'y+'; // Direction of snake's move  var gameIsRunning = false; // Is game running  var snake\_timer; //Snake's timer  var food\_timer; //timer for eating  var score = 0; // Result  function init() {  prepareGameField(); //Generation of field  var wrap = document.getElementsByClassName ('wrap')[0];  wrap.style.width = '400px';  //Button's events and new game  document.getElementById('snake-start').addEventListener('click',startGame);  document.getElementById('snake-renew').addEventListener('click', refreshGame);  //Keys tracing  addEventListener('keydown', changeDirection);  }  /\*\*  \* Playfield's generation function  \*/  function prepareGameField(){  //table creation  var game\_table = document.createElement('table');  game\_table.setAttribute('class', 'game-table');  //Game-table cells generation  for (var i = 0; i< FIELD\_SIZE\_X; i++) {  //Line's creation  var row = document.createElement ('tr');  row.className = 'game-table-row row-' + i;  for (var j = 0; j < FIELD\_SIZE\_Y;j++){  //Cell's creation  var Cell = document.createElement('td');  cell.className = 'cell-' + i + '-' + j;  row.appendChild (cell);//Cell adding    }  game\_table.appendChild (row); //Line adding  }  document.getElementById('snake-field').appendChild(game\_table); //Table adding  }  /\*\*  \* Start of the game  \*/  function startGame () {  gameIsRunning = true;  respawn();//Create the snake  snake\_timer = setInterval (move, SNAKE\_SPEED);  setTimeout (createFood, 5000);  }  /\*\*  \* Function of snake location on the game-field  \*/  function respawn() {  //Snake - massive td  //Snake's beginning length = 2  //Snake's respawn from center  var start\_coord\_x = Math.floor (FIELD\_SIZE\_X/2);  var start\_coord\_y = Math.floor (FIELD\_SIZE\_Y/2);  //Snake's head  var snake\_head = document.getElementsByClassName('cell-' + start\_coord\_y + '-' + start\_coord\_x)[0];  //Snake's body  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);  }  /\*\*  \* Snake's movement  \*/  function move (){  //console.log('move',direction);  //Classes assembling  var snake\_head\_classes = snake[snake.length - 1].getAttribute ('class').split(' ');  //Head's share  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]);  //Define the new point  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];  }  //Testing  //1)new\_unit isnt part of snake  //2)snake doesnt go through the border of field  //console.log (new\_unit);  if(!isSnakeUnit(new\_unit)&& new\_unit!==undefined){  //New part of snake adding  new\_unit.setAttribute('class', new\_unit.getAttribute('class')+'snake-unit');  snake.push(new\_unit);  //Checking. if we should took off the teil  if(!haveFood(new\_unit)){  //Find the teil  var removed=snake.splice(0,1)[0];  var classes=removed.getAttribute('class').split('');  //remove the teil  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) {  switch (e.keyCode) {  case 37: // влево  if (direction != 'x+') direction = 'x-';  break;  case 38: // вверх  if (direction != 'y-') direction = 'y+';  break;  case 39: // вправо  if (direction != 'x-') direction = 'x+';  break;  case 40: // 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; |
|  |  |
|  |  |