УДК 004

Создание игры «Судоку» на JavaScript

Болтовский Гавриил Александрович
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Целью данной статьи является создание игры судоку. Игра написана на языке программирования JavaScript, также используется HTML. Результатом исследования станет игра с подробным описанием её реализации.
Ключевые слова: JavaScript, HTML, веб-разработка

Creating a Sudoku Game in JavaScript

Boltovsky Gavriil Alexandrovich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to create a Sudoku game. The game is written in the JavaScript programming language, HTML is also used. The result of the study will be a game with a detailed description of its implementation.
Keywords: JavaScript, HTML, web development

1. Введение

1.1 Актуальность исследования

Существует множество игр, направленных на развитие логического мышления. Одной из наиболее популярных в данной категории является Судоку. Это математическая головоломка, представляющая собой таблицу размером 9x9 клеток. Задача игрока - заполнение этой таблицы числами от 1 до 9 в соответствии с определенными правилами. Судоку требует применения логического мышления и поиска правильных последовательностей с учетом уже имеющихся чисел в таблице.

Данная игра предоставляет возможность развивать логическое мышление, поскольку игроку необходимо анализировать имеющуюся информацию, выявлять закономерности и применять стратегии для достижения правильного решения.

1.2 Обзор исследований

В исследовании В. Красноухова и Д. Кавтарадзе [1] рассматривается вопрос использования игр и головоломок в обучении процессу мышления. Ими приведены различные задачи, оценивается их сложность и подходы в

решении. В статье У. В. Меркуловой [2] обобщается опыт использования компьютерных игр при обучении младших школьников. Отмечается, что включение в урок игр и игровых моментов делает процесс обучения интересным и занимательным.

Много игр уже были созданы с использованием JavaScript. Поварницын Е.Н. в своём исследовании [3] описывает процесс создания «Тетриса». В статье [4] описывается создание «FlappyBird».

Математическая база Судоку рассмотрена в исследовании [5].

1.3 Цель исследования

Целью исследования является создание игры Судоку.

1.4 Постановка задачи

В рамках разработки игры Судоку используются языки программирования JavaScript и HTML. Для обеспечения формализованного процесса генерации игровых полей и проверки их решаемости необходимо определить правила, которым должны соответствовать сгенерированные поля.

2. Методы исследования

В рамках проекта предлагается разработка игры, состоящей из четырех файлов: index.html, sudoku.html, result.html и sudoku.js. Главная страница (index.html) содержит правила игры и кнопку, позволяющую перейти к игровому процессу на страницу sudoku.html.

Страница sudoku.html реализует игровой процесс и включает в себя таблицу, которая генерируется с использованием JavaScript. Генерация таблицы происходит при загрузке страницы с помощью функции generateTable (рис. 1). В этой функции используется переменная sudokuTable, которая представляет собой массив чисел, определяющих игровую таблицу.

```
function generateTable() {
        sudokuTable = generateSudoku();
        const table = document.getElementById("sudoku");
        for (let i = 0; i < 9; i++) {
                let Tr = document.createElement("tr");
                for (let j = 0; j < 9; j++) {
                        let inputId = "cell-" + i + "-" + j;
                        let numGenerate = Math.round(Math.random());
                        let num = Math.floor(Math.random() * 9) + 1;
                        let Td = document.createElement("td");
                        let inputCell = document.createElement("input");
                        inputCell.setAttribute("type", "text");
                        inputCell.setAttribute("maxlength", "1");
                        inputCell.setAttribute("id", inputId);
                        if (numGenerate == 1 && sudokuTable[i][j] != "") {
                                inputCell.setAttribute("value", sudokuTable[i][j]);
                                inputCell.setAttribute("readonly", "readonly");
                        }
                        Td.appendChild(inputCell);
                        Tr.appendChild(Td);
                }
                table.appendChild(Tr);
        }
}
```

Рисунок 1. Код функции “generateTable”

Функция generateTable начинается с вызова функции generateSudoku (рис. 2), которая генерирует таблицу чисел в случайном порядке и возвращает массив чисел. В функции generateSudoku присутствуют две вспомогательные функции: solveSudoku, отвечающая за генерацию таблицы и isValid, выполняющая проверку правильности расположения чисел в таблице.

```
function generateSudoku() {
        const board = Array.from({ length: 9 }, () => Array.from({ length: 9 }, () => 0));
        solveSudoku(board);
        return board;
}
```

Рисунок 2. Код функции “generateSudoku

Функция solveSudoku (рис. 3) проходит по всей таблице, перебирая строки и столбцы. Если значение в ячейке равно 0, функция выбирает одно из чисел от 1 до 9. Если число подходит для данной ячейки согласно функции isValid, оно сохраняется в таблице, иначе значение ячейки остается равным 0.

```
function solveSudoku(board) {
        for (let row = 0; row < 9; row++) {
                for (let col = 0; col < 9; col++) {
                        if (board[row][col] === 0) {
                                for (let num = 1; num <= 9; num++) {
                                        if (isValid(board, row, col, num)) {
                                                board[row][col] = num;
                                                if (solveSudoku(board)) {
                                                        return true;
                                                } else {
                                                        board[row][col] = 0;
                                                }
                                        }
                                }
                                return false;
                        }
                }
        }
        return true;
}
```

Рисунок 3. Код функции “solveSudoku”

Функция isValid (рис. 4) проверяет правильность расположения числа в таблице, проверяя его уникальность в строке, столбце и квадрате 3x3, в котором оно находится. Результатом функции является логическое значение true или false.

```
function isValid(arr, row, col, num) {
        for (let i = 0; i < 9; i++) {
                if (arr[row][i] === num) {
                        return false;
                }
        }
        for (let i = 0; i < 9; i++) {
                if (arr[i][col] === num) {
                        return false;
                }
        }
        const subgridRow = Math.floor(row / 3);
        const subgridCol = Math.floor(col / 3);
        for (let i = subgridRow * 3; i < subgridRow * 3 + 3; i++) {
                for (let j = subgridCol * 3; j < subgridCol * 3 + 3; j++) {
                        if (arr[i][j] === num) {
                                return false;
                        }
                }
        }
        return true;
}
```

Рисунок 4. Код функции “isValid”

После создания массива чисел функция generateTable создает таблицу на странице. JavaScript позволяет добавлять элементы на страницу и задавать им атрибуты. В данном случае создаются ячейки таблицы (81 в общей сложности), которые имеют уникальный идентификатор в формате "cell-row-col", где row и col - номера строки и столбца соответственно.

Таблица должна быть частично заполнена, чтобы предоставить пользователю возможность решить головоломку. Для этого используется переменная numGenerate, которая случайным образом принимает значения 0 или 1. Если numGenerate равно 1, число добавляется в таблицу и ячейка становится доступной только для чтения (атрибут readonly).

Для улучшения визуальной навигации игрока таблица разделяется на квадраты 3x3 путем добавления обводки ячеек с помощью CSS-стилей (рис. 5).

			4		6	7		9
4	5		7	8			2	3
7			1		3		5	6
	1				5			
		5			7	2		
8			2	1		3		
		1				9		8
6	4		9	7			3	
9	7		5				4	2

Рисунок 5. Улучшенная таблица Судоку на странице sudoku.html

Последним этапом разработки является реализация проверки правильности заполнения таблицы. Для этого используется функция checkSudoku (рис. 6), которая собирает значения из всех ячеек таблицы и сравнивает их с исходным массивом чисел. Если значения совпадают, игра завершается, в противном случае выводится сообщение об ошибке.

```
function checkSudoku() {
        let finishGame = true;
        const table = document.getElementById("sudoku");
        for (let i=0;i<9;i++) {
                for (let j=0;j<9;j++) {
                        let elemId = "cell-" + i + "-" + j;
                        let userNum = document.getElementById(elemId).value;
                        if (sudokuTable[i][j] != userNum) {
                                finishGame = false;
                                console.log(i,j,sudokuTable[i][j],userNum,elemId);
                                break;
                        }
                }
        }
        if (finishGame == true) {
                alert("В решении была допущена ошибка!");
        }
        else {
                location.href = "result.html";
        }
}
```

Рисунок 6. Код функции checkSudoku

При успешном решении задачи игрок перенаправляется на страницу result.html (рис. 7).

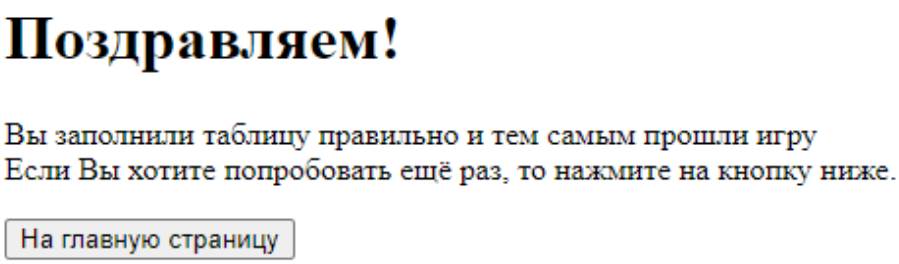

Рисунок 7. Страница result.html

Таким образом, разработанная игра Судоку позволяет пользователю развивать логическое мышление при решении головоломки и предоставляет возможность проверки правильности решения.

3. Выводы

Таким образом, была создана игра Судоку с использованием JavaScript.

Библиографический список

1. Красноухов В., Кавтарадзе Д. Игры и головоломки в обучении мышлению // Образовательная политика. 2012. №1 (57).

2. Меркулова У. В. Использование компьютерных игр при обучении младших школьников // Если. 2013. Т. 16. С. 17.
3. Поварницын Е.Н. СОЗДАНИЕ БРАУЗЕРНОЙ ИГРЫ «ТЕТРИС» // Форум молодых ученых. 2020. №2 (42).
4. Семченко Р. В., Еровлев П. А. Создание игры «FlappyBird» на JavaScript //Постулат. 2019. №. 8.
5. Felgenhauer B., Jarvis F. Mathematics of sudoku I //Mathematical Spectrum. 2006. Т. 39. №. 1. С. 15-22.