
Глава 1. Крестики-нолики

Содержание

| | |
|-----------------------------------|---|
| Краткое описание | 1 |
| Скриншоты работы приложения | 2 |
| Примеры разработанного кода | 3 |

Краткое описание

Программа Tic-tac-toe предназначена для игры в крестики-нолики с ботом

Интерфейс программы представляет собой веб-страницу, на которой можно нажимать на клетки. Бот, следуя своей логике, стремится заполнить ячейки в ряд.

После того, как один из игроков заполнит три значения в ряд или по диагонали, игра заканчивается.

Всего существует 2 игрока, что играют друг с другом:

1. Игрок-человек

Игрок-бот

Скриншоты работы приложения

Рисунок 1.1. Интерфейс пользователя

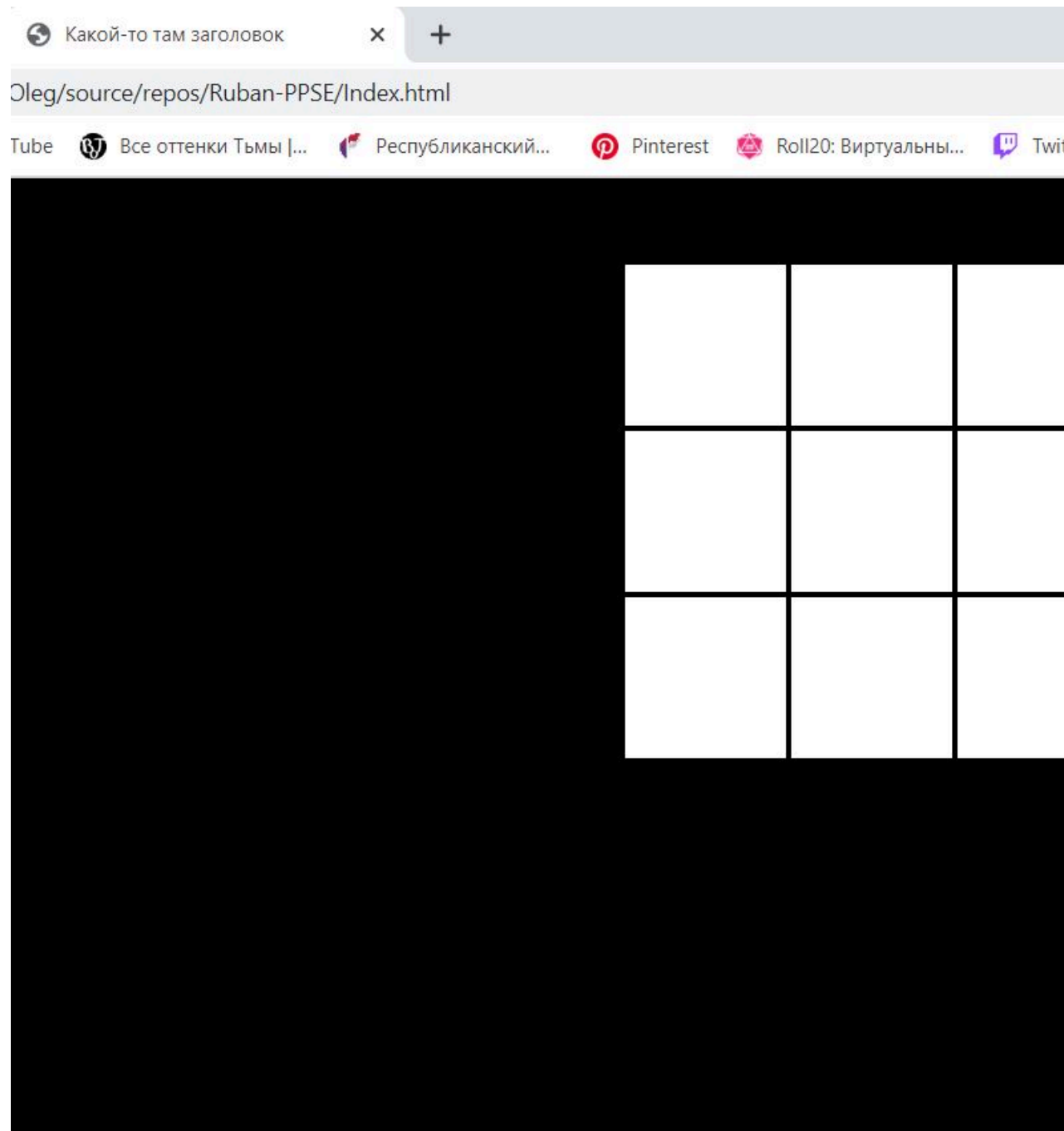
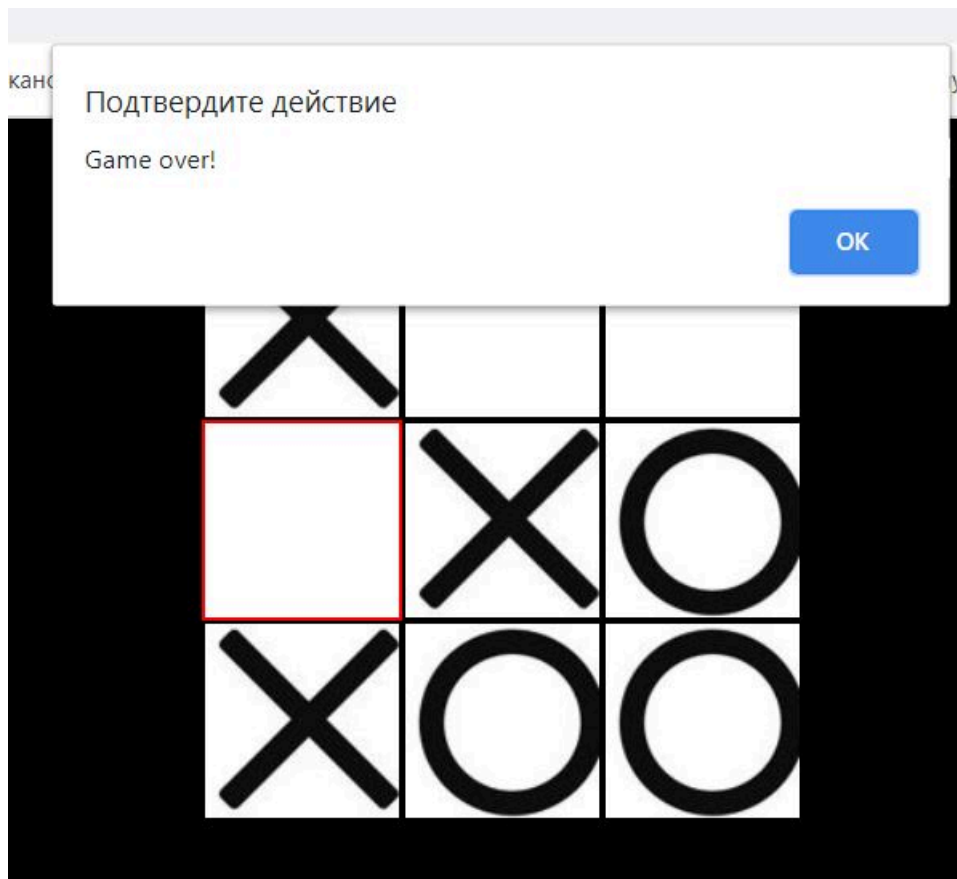


Рисунок 1.2. Интерфейс пользователя по окончании игры

Примеры разработанного кода

Для разработки пользовательского интерфейса использовался язык разметки HTML и таблица стилей CSS. Пример стилей приведен ниже.

Пример 1.1. Пример разметки пользовательского интерфейса

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, block
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
    margin: 0;
    padding: 0
}
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav,
    display: block
}
body {
    line-height: 1
}
ol, ul {
    list-style: none
}
blockquote, q {
    quotes: none
```

```
}

    blockquote:before, blockquote:after, q:before, q:after {
        content: none
    }
table {
    border-collapse: collapse;
    border-spacing: 0
}
body {
    background-color: black;
}
.actions {
    text-align: center;
}
.ai {
    background-image: url('../img/ai.jpg');
}
.container {
    margin: 50px auto;
    width: 300px;
}
.cell {
    background-color: white;
    border: 2px solid black;
    float: left;
    height: 96px;
    width: 96px;
}
    .cell:hover {
        border-color: red;
    }
.player {
    background-image: url('../img/player.jpg');
}
```

Пример 1.2. Часть скриптов работы веб-приложения

```
var t = new Array(9);

/**
 * Работа с функцией {@link $ai}
 *
 * Выбирает
 * ячейку, на которую бот
 * сходит
 *
 * Возвращает функцию
 * move() со сгенерированным
 * номером ячейки
 */
function ai() {
    var id = Math.floor(Math.random() * 9);
    t[id] ? ai() : move(id, 'ai');
}
```

```
/** Определяет, закончилась ли игра */
function checkEnd() {
    if (t[0] === 'ai' and t[1] === 'ai' and t[2] === 'ai' || t[0] === 'player' and t[1] === 'player' and t[2] === 'player') return true;
    if (t[3] === 'ai' and t[4] === 'ai' and t[5] === 'ai' || t[3] === 'player' and t[4] === 'player' and t[5] === 'player') return true;
    if (t[6] === 'ai' and t[7] === 'ai' and t[8] === 'ai' || t[6] === 'player' and t[7] === 'player' and t[8] === 'player') return true;
    if (t[0] === 'ai' and t[3] === 'ai' and t[6] === 'ai' || t[0] === 'player' and t[3] === 'player' and t[6] === 'player') return true;
    if (t[1] === 'ai' and t[4] === 'ai' and t[7] === 'ai' || t[1] === 'player' and t[4] === 'player' and t[7] === 'player') return true;
    if (t[2] === 'ai' and t[5] === 'ai' and t[8] === 'ai' || t[2] === 'player' and t[5] === 'player' and t[8] === 'player') return true;
    if (t[0] === 'ai' and t[4] === 'ai' and t[8] === 'ai' || t[0] === 'player' and t[4] === 'player' and t[8] === 'player') return true;
    if (t[2] === 'ai' and t[4] === 'ai' and t[6] === 'ai' || t[2] === 'player' and t[4] === 'player' and t[6] === 'player') return true;
    if (t[0] and t[1] and t[2] and t[3] and t[4] and t[5] and t[6] and t[7] and t[8]) return true;
}

/**
 * Функция, отвечающая за
 * заполнение ячеек
 * @param {string} id Заполняемая ячейка.
 * @param {string} role Роль - игрок или бот.
 */
function move(id, role) {
    if (t[id]) return false;
    t[id] = role;
    document.getElementById(id).className = 'cell ' + role;
    !checkEnd() ? (role === 'player') ? ai() : null : reset();
}

/**
 * Функция, отвечающая за
 * перезапуск игры
 * после окончания
 */
function reset() {
    alert("Game over!");
    location.reload();
}
```

Примечание

Код приложения находится в свободном доступе и может быть найден в репозитории на гитхабе.

Tic-tac-toe [<https://github.com/pi17v-ppse/Ruban>]