

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Алтайский государственный технический университет  
им. И. И. Ползунова»  
Факультет информационных технологий  
Кафедра информатики, вычислительной техники и информационной  
безопасности

Отчет защищен с оценкой «\_\_»

Преподаватель С. В. Умбетов

«\_\_» «\_\_\_\_\_» 2023 г.

Отчёт по лабораторной работе №7 по дисциплине  
«Алгоритмизация и программирование»  
«Разработка игры Крестики-нолики»

ЛР 09.03.01.14.002

Студент группы: ПИЭ-22 Е. А. Сафронова

Преподаватель: ассистент, к. т. н. С. В. Умбетов

БАРНАУЛ 2023

## **Оглавление**

Введение:.....	3
Ход работы:.....	4
Вывод: .....	13

**Введение:**

Цели и задачи:

Продолжение работы с функциями и тегами, знакомство с canvas.

Разработка игры «Крестики –нолики»

Задание принял:  Сафронова Е. А.

## Ход работы:

### Вариант №1

Написать игру «Крестики-нолики» с полем 3 на 3 клетки, используя Canvas.

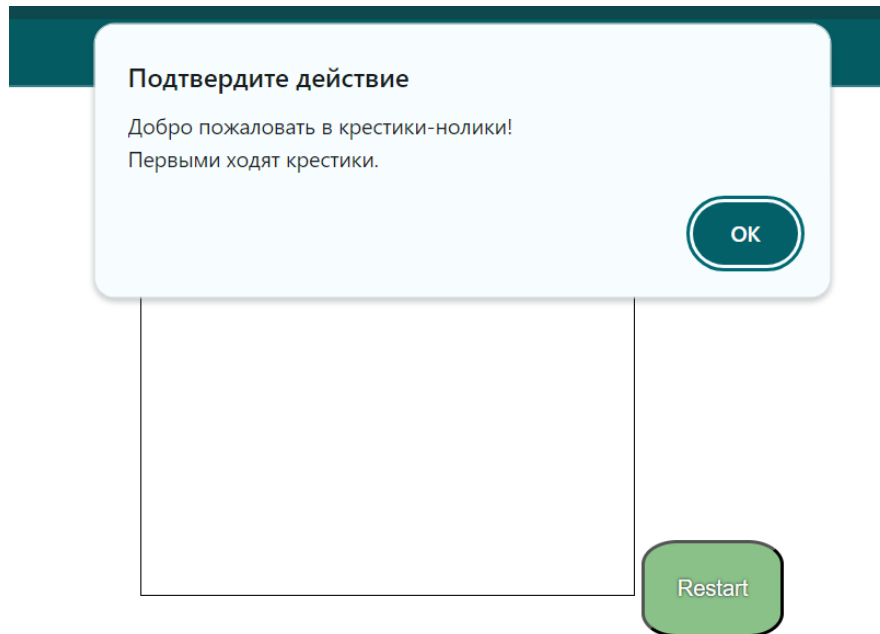


Рисунок 1 – Приветственный экран, игровое поле еще не заполнено.

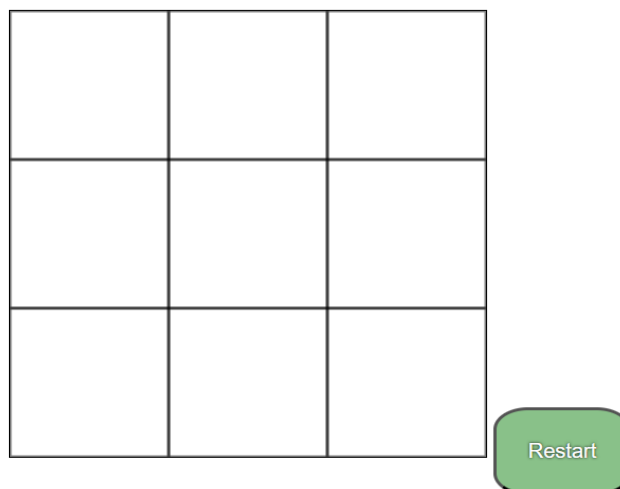


Рисунок 2 – Начало игры

Результат игры выводится сразу, не рисуя последний ход участника. При нажатии кнопки «Restart» поле canvas очищается, и снова начинают ходить крестики.

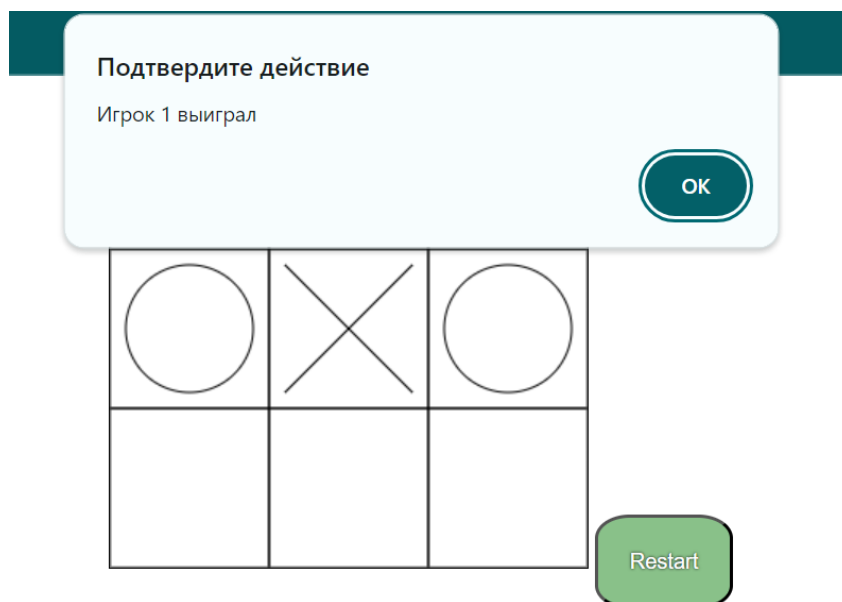


Рисунок 3 – Выигрыш по вертикали

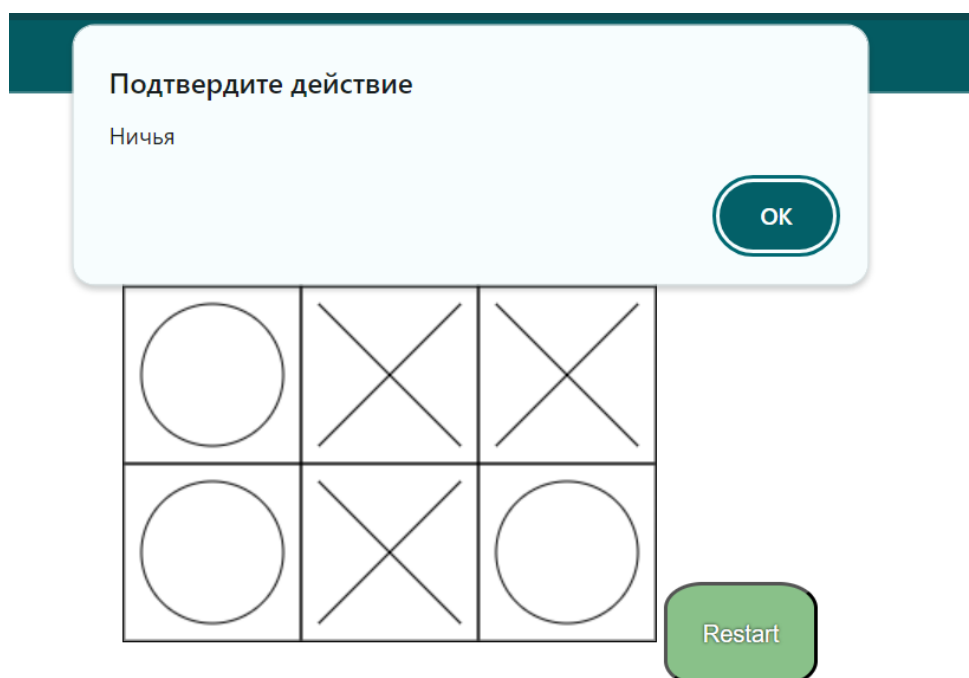


Рисунок 4 – Ничья

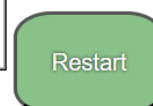
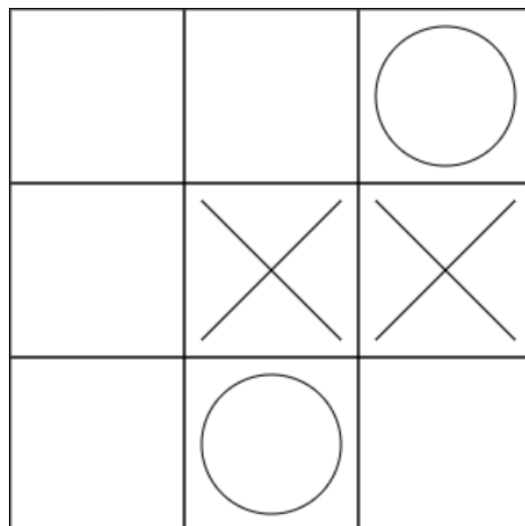


Рисунок 5.1 – До нажатия кнопки

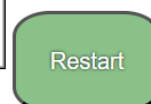
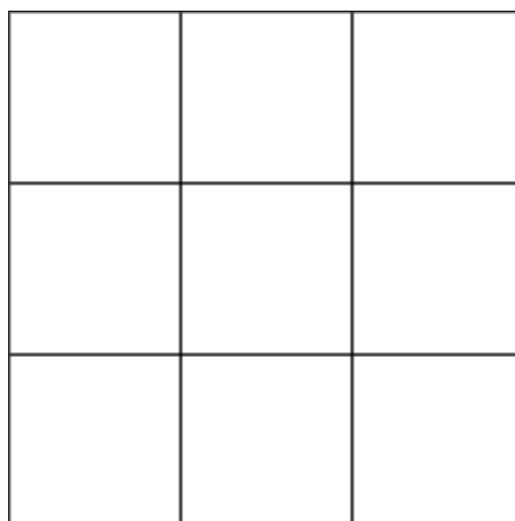


Рисунок 5.2 – После нажатия кнопки

---

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link href="LR72.css" rel="stylesheet" type="text/css">
</head>
<body>
  <div id="game" style="text-align: center;">
    <canvas id="canvas" width="300" height="300"></canvas>
    <button id="restart">Restart</button>
  </div>
  <script src="LR7js2.js"></script>
</body>
</html>

```

---

Рисунок 6 – Код html

---

```

#canvas {
  border: 1px solid black;
}

#restart {
  margin-top: 10px;
}

#div {
  width: 300px;
  height: 300px;
}

button {
  background-color: rgba(17, 129, 17, 0.493);
  color: white;
  padding: 20px;
  margin-bottom: 10px;
  text-align: center;
  text-shadow: black 0 0 2px;
  border-radius: 25%;
}

```

---

Рисунок 7 – Код css

```

// Функция для отрисовки игрового поля
function draw_field() {
    // Очищаем canvas
    context.clearRect(0, 0, width, height);

    // Рисуем таблицу
    for (var i = 0; i < cells_count; i++) {
        for (j = 0; j < 3; j++) {
            context.strokeRect(0 + i * 100, 0 + j * 100, 100, 100);}
        }
}

```

С рисунка 8 по рисунок 13 - Код JavaScript.

---

```

var canvas = document.getElementById('canvas');
var restart_button = document.getElementById('restart');
var context = canvas.getContext('2d');
var width = 300;
var height = 300;

// Размер ячейки
//При доработке пользователю можно будет вводить свои значения
var cell_size = 100;
var cells_count = width/cell_size;
var field = [];
alert("Добро пожаловать в крестики-нолики!\nПервыми ходят крестики.")

// Переменная для хранения текущего игрока (1 - крестик, 2 - нолик)
var player = 1;

```

Рисунок 8



```

// Рисуем крестики и нолики
for (var i = 0; i < cells_count; i++) {
  for (var j = 0; j < cells_count; j++) {
    var cell = field[i][j];
    switch (cell) {
      case 1:
        context.beginPath();
        context.moveTo(i * cell_size + 10, j * cell_size + 10);
        context.lineTo((i + 1) * cell_size - 10, (j + 1) * cell_size - 10);
        context.moveTo((i + 1) * cell_size - 10, j * cell_size + 10);
        context.lineTo(i * cell_size + 10, (j + 1) * cell_size - 10);
        context.stroke();
        break;
      case 2:
        context.beginPath();
        context.arc(i * cell_size + cell_size / 2, j * cell_size + cell_size / 2, cell_size / 2 - 10, 0, Math.PI * 2);
        context.stroke();
        break;
      default:
        break;
    }
  }
}

```

Рисунок 9

```

// Функция для проверки победы
function check_win() {
    for (var i = 0; i < cells_count; i++) {
        var row = field[i];
        if (row[0] !== 0 && row[0] === row[1] && row[1] === row[2]) {
            return row[0];
        }
    }

    for (var i = 0; i < cells_count; i++) {
        if (field[0][i] !== 0 && field[0][i] === field[1][i] && field[1][i] === field[2][i]) {
            return field[0][i];
        }
    }

    if (field[0][0] !== 0 && field[0][0] === field[1][1] && field[1][1] === field[2][2]) {
        return field[0][0];
    }
    if (field[2][0] !== 0 && field[2][0] === field[1][1] && field[1][1] === field[0][2]) {
        return field[2][0];
    }

    // Ничья
    if (full_field()) {
        return -1;
    }

    // Игра продолжается
    return 0;
}

```

Рисунок 10

```

// Проверка заполненности поля
function full_field() {
    for (var i = 0; i < cells_count; i++) {
        for (var j = 0; j < cells_count; j++) {
            if (field[i][j] === 0) {
                return false;
            }
        }
    }
    return true;
}

```

Рисунок 11

```

function cell_click(event) {
    // Координаты клика
    var x = event.pageX - canvas.offsetLeft;
    var y = event.pageY - canvas.offsetTop;
    // Номер ячейки
    var i = Math.floor(x / cell_size);
    var j = Math.floor(y / cell_size);
    if (field[i][j] !== 0) {
        return;
    }

    field[i][j] = player;

    // Переключаем игрока
    //Если значение равно 1, то меняем его на 2, а иначе на 1
    player = player === 1 ? 2 : 1;

    // Проверяем победу
    var winner = check_win();
    if (winner !== 0) {
        let message;
        if (winner === -1) {
            message = 'Ничья';
        } else {
            message = 'Игрок ' + winner + ' выиграл';
        }

        alert(message);

        // Обнуляем игровое поле и перерисовываем его
        field = [];
    }
}

```

Рисунок 12

```

    for (var i = 0; i < cells_count; i++) {
        field.push([]);
        for (var j = 0; j < cells_count; j++) {
            field[i].push(0);
        }
    }
    draw_field();
} else {
    // Перерисовываем игровое поле
    draw_field();
}
}

// кнопка "Restart"
restart_button.addEventListener('click', function() {
    field = [];
    for (var i = 0; i < cells_count; i++) {
        field.push([]);
        for (var j = 0; j < cells_count; j++) {
            field[i].push(0);
        }
    }
    draw_field();
});

```

Рисунок 13

```

// Начало игры
for (var i = 0; i < cells_count; i++) {
    field.push([]);
    for (var j = 0; j < cells_count; j++) {
        field[i].push(0);
    }
}

// Отслеживание клика на canvas
canvas.addEventListener('click', cell_click);

// Первая отрисовка
draw_field();

```

Рисунок 14

Ссылка на github: <https://github.com/KateEas/LR7.git>

## **Вывод:**

В ходе лабораторной работы я продолжила учиться взаимодействовать с элементами html – страницы через JavaScript – код. Познакомилась с Canvas.

Изначальная идея заключалась в создании объектов, являющихся соединением элементов html и canvas, но потом решила пойти по более простому пути.

Данную игру можно доработать, сделав размер квадратной матрицы не 3, а больше (сколько захочет игрок). Но тогда надо будет слегка корректировать условие победы для главной и побочной диагонали, учитывая параллельные им прямые (подобное было в предыдущей лабораторной работе). Единственный вопрос остается в условии выигрыша в таком случае.

Впервые попробовала использовать тернарный оператор «?». Он оказался очень удобным, теперь буду применять его.