

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Web-технологии»
Тема: ТЕТРИС НА JAVASCRIPT

Студентка гр. 2384

Соц Е.А.

Преподаватель

Беляев С.А.

Санкт-Петербург

2024

Цель работы

Изучение работы web-сервера nginx со статическими файлами и создание клиентских JavaScript web-приложений.

Задание

- генерация открытого и закрытого ключей для использования шифрования (<https://www.openssl.org/>);
- настройка сервера nginx для работы по протоколу HTTPS;
- разработка интерфейса web-приложения;
- обеспечение ввода имени пользователя;
- обеспечение создания новой фигуры для тетриса по таймеру и ее движение;
- обеспечение управления пользователем падающей фигурой;
- обеспечение исчезновения ряда, если он заполнен;
- по окончании игры – отображение таблицы рекордов, которая хранится в браузере пользователя

Выполнение работы

Ввод имени пользователя осуществляется благодаря `let name = prompt("Ваше имя", "Безымянный");` `prompt` отобразит модальное окно с текстом “Ваше имя”, полем для ввода текста и кнопками ОК/Отмена, “Безымянный” – значение по умолчанию в поле ввода.

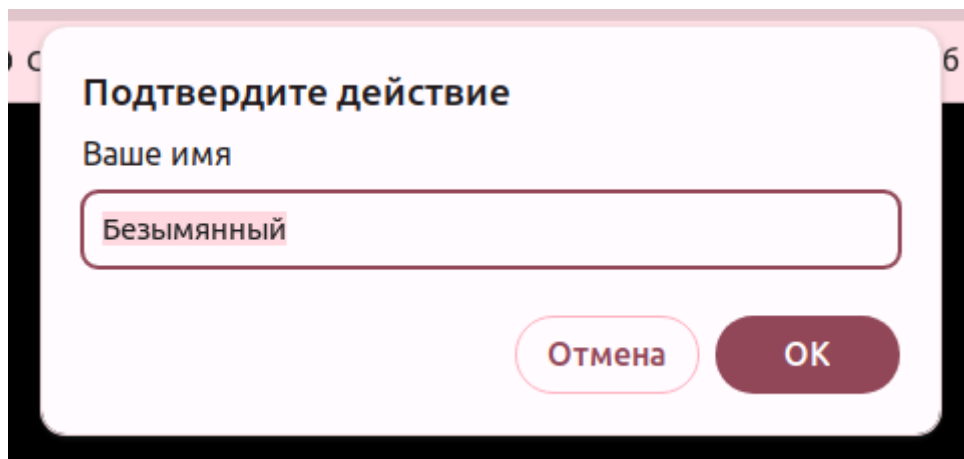


Рисунок 1 – Модальное окно для ввода имени

1. Холсты

- Основной холст (canvas): Используется для отрисовки игрового поля и движущихся фигур.
- Холст для отображения счета (canvasScore): Отображает текущий счет, уровень игры и другие статистические данные.
- Холст для отображения следующей фигуры (canvasNext): Показывает следующую фигуру, которая будет появляться на игровом поле.

2. Игровое поле

Игровое поле представлено двумерным массивом `playfield`, где каждый элемент массива соответствует клетке на игровом поле. Значение 0 означает пустую клетку, а другие значения соответствуют типам фигур.

3. Фигуры

Фигуры (тетромино) представлены в виде матриц, где каждый элемент матрицы равен 1, если клетка занята, и 0, если клетка пуста.

Доступные фигуры:

4. Цвета фигур

Каждая фигура имеет свой цвет, который хранится в объекте colors.

5. Счет и уровень

Счет (score): Текущий счет игрока.

Уровень (level): Уровень игры, который увеличивается каждые 100 очков.

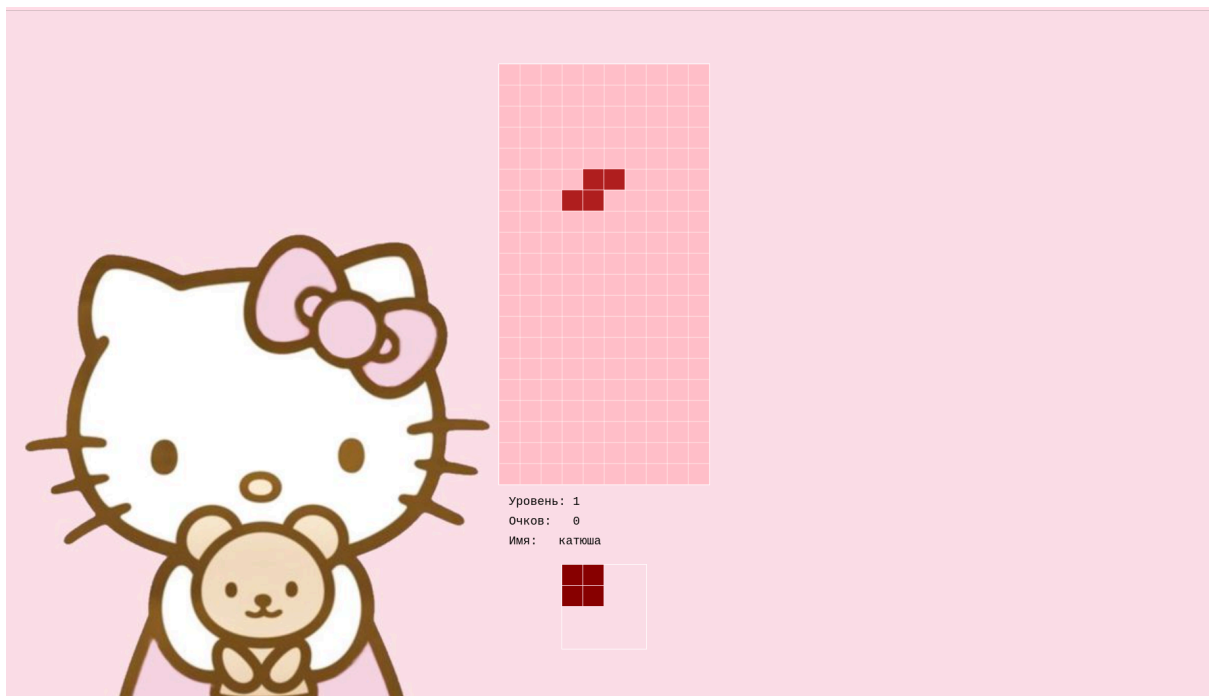


Рисунок 2 – Полотно игры

6. Рекорды

Рекорды игроков хранятся в localStorage и отображаются в контекстном меню при окончании игры.

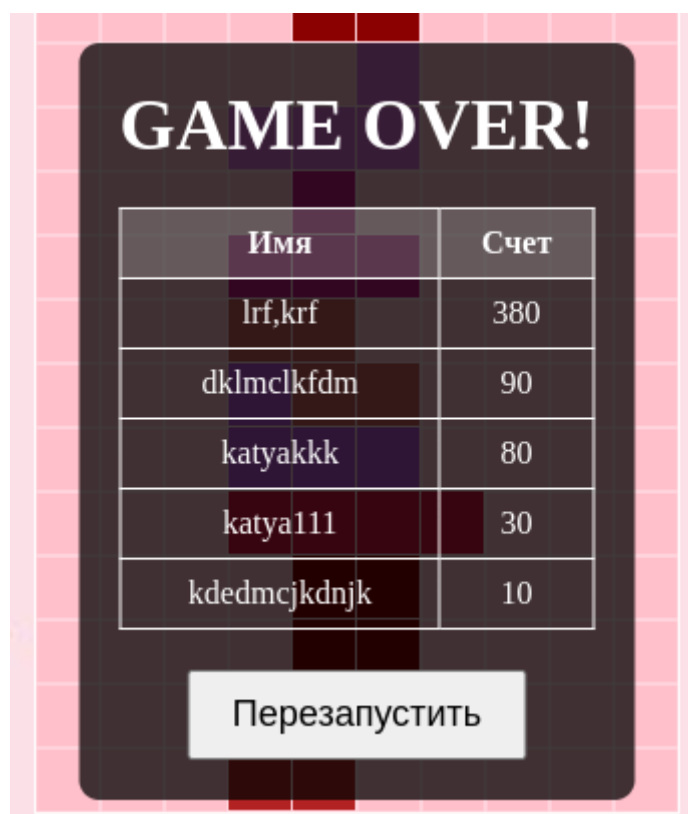


Рисунок 3 – Таблица рекордов и кнопка “Перезапустить” после окончания игры

Основные функции игры

1. Генерация последовательности фигур

Функция `generateSequence()` генерирует случайную последовательность фигур и добавляет их в массив `tetrominoSequence`. Последовательность создается путем случайного выбора фигур из массива `sequence`, который содержит все возможные фигуры.

2. Получение следующей фигуры

Функция `getNextTetromino()` возвращает следующую фигуру из последовательности. Если последовательность пуста, она генерируется заново.

3. Проверка возможности движения фигуры

Функция `isValidMove(matrix, cellRow, cellCol)` проверяет, может ли фигура быть перемещена в указанную позицию на игровом поле.

4. Установка фигуры на игровое поле

Функция `placeTetromino()` устанавливает текущую фигуру на игровое поле. Если фигура не может быть установлена (например, если она выходит за границы поля), игра заканчивается.

5. Очистка заполненных линий

Если линия на игровом поле полностью заполнена фигурами, она очищается, а все линии выше нее опускаются на одну клетку вниз.

6. Отображение счета и уровня

Функция `showScore()` отображает текущий счет и уровень игры на холсте `canvasScore`.

7. Отображение следующей фигуры

Функция `drawNextTetromino()` отображает следующую фигуру на холсте `canvasNext`.

8. Обработка нажатий клавиш

Функция `document.addEventListener('keydown', ...)` обрабатывает нажатия клавиш для управления фигурами:

Стрелка влево/вправо: Перемещение фигуры влево/вправо.

Стрелка вверх: Вращение фигуры.

Стрелка вниз: Ускорение падения фигуры.

Пробел: Мгновенное падение фигуры до самого низа.

9. Главный цикл игры

Функция `loop()` является основным циклом игры, который обновляет состояние игры и отрисовывает его на холсте. Цикл вызывается с помощью `requestAnimationFrame()`.

Вывод

В ходе лабораторной работы была создана игра тетрис с использованием HTML5 Canvas и JavaScript. Игра включает в себя основные механики, такие как управление фигурами, поворот, падение, очистка заполненных рядов и отображение статистики. Также реализована система рекордов, позволяющая игрокам сохранять свои результаты и сравнивать их с предыдущими играми.