**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. 1381 |  | Васильева О. М. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2023

## Цель и задачи работы.

## Целью работы является изучение работы web-сервера nginx со статическими файлами и создание клиентских JavaScript web-приложений.

## Для достижения поставленной цели требуется решить следующие задачи:

## – 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.

**Основные теоретические сведения.**

Асимметричные ключи используются в асимметричных алгоритмахшифрования и являются ключевой парой. Закрытый ключ известен только владельцу. Открытый ключ может быть опубликован и используется для проверки подлинности подписанного документа (сообщения). Открытый ключ вычисляется, как значение некоторой функции от закрытого ключа, но знание открытого ключа не дает возможности определить закрытый ключ.По секретному ключу можно вычислить открытый ключ, но по открытому ключу практически невозможно вычислить закрытый ключ.

nginx (https://nginx.ru/ru/) – веб-сервер, работающий на Unix-подобных операционных системах и в операционной системе Windows.

## JavaScript (https://developer.mozilla.org/ru/docs/Web/JavaScript) – язык

## программирования, он поддерживает объектно-ориентированный и функциональный стили программирования. Является реализацией стандарта

## ECMAScript.

## Общая формулировка задание.

## Необходимо создать web-приложение – игру в тетрис.

## Основные требования:

## – сервер – nginx, протокол взаимодействия – HTTPS версии не ниже 2.0;

## – отображается страница для ввода имени пользователя с использованием HTML-элементов <input>;

## – статическая страница отображает «стакан» для тетриса с использованием HTML-элемента <canvas>, элемент <div> используется для отображения следующей фигуры, отображается имя пользователя;

## – фигуры в игре – классические фигуры тетриса (7 шт. тетрамино);

## – случайным образом генерируется фигура и начинает падать в «стакан» (описание правил см., например, https://ru.wikipedia.org/wiki/Тетрис);

## – пользователь имеет возможность двигать фигуру влево и вправо, повернуть на 90 и «уронить»;

## – если собралась целая «строка», она должна исчезнуть;

## – при наборе некоторого заданного числа очков увеличивается уровень, что заключается в увеличении скорости игры;

## – пользователь проигрывает, когда стакан «заполняется», после чего ему отображается локальная таблица рекордов;

– вся логика приложения написана на JavaScript.

Необязательно: оформление с использованием CSS.

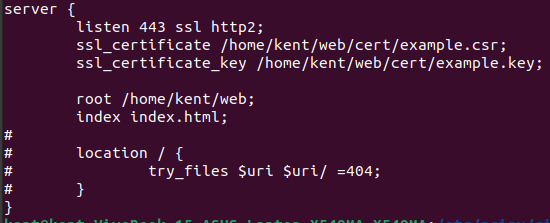
Постарайтесь сделать такую игру, в которую вам будет приятно играть.

Помните, когда-то эта игра была хитом! Преимуществом будет использование звукового сопровождения событий: падение фигуры, исчезновение «строки».

## Выполнение работы.

1. Настройка сервера nginx для работы по протоколу HTTPS.

Настройка файла конфигурации:



1. HTML-страницы.

В проекте имеются два html-файла index.html и start.html, первый представляет собой страницу для ввода имени пользователя, а вторая является самой игрой.

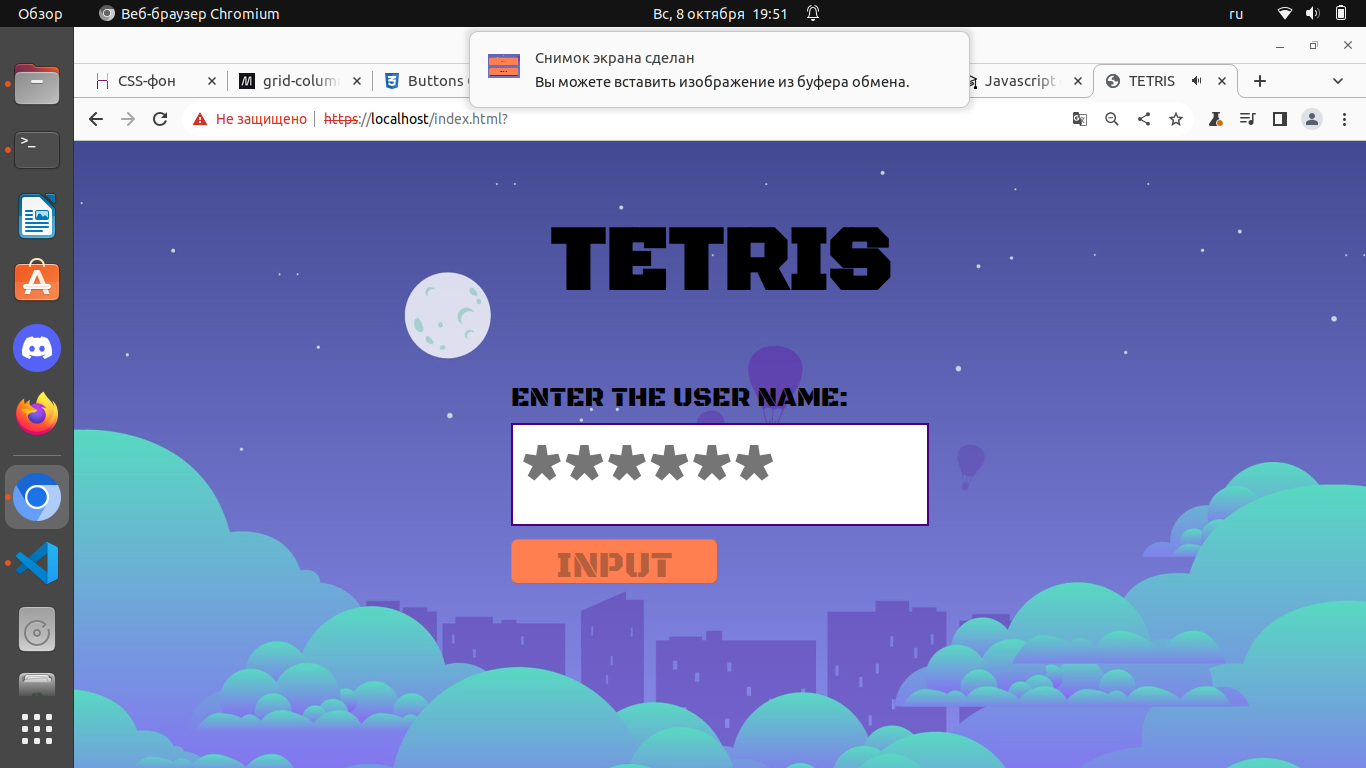


Рис. 1 – демонстрация вида страницы index.html

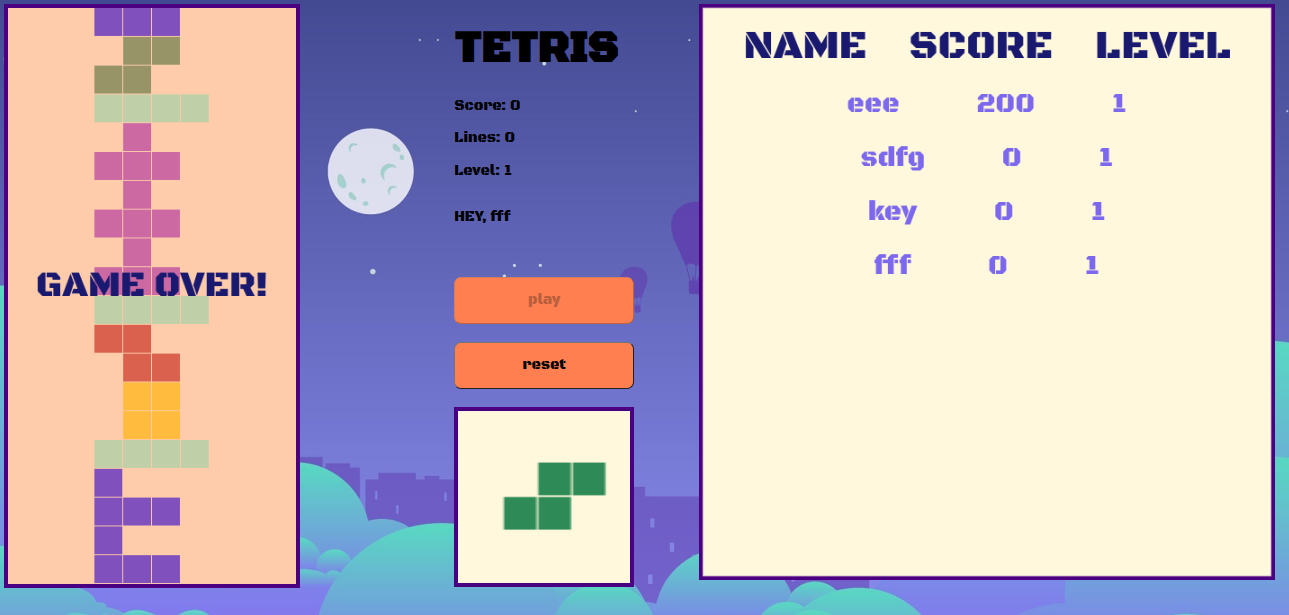


Рис. 2- демонстрация вида страницы start.html

На первой страницы (рис.1) расположено окошко для ввода имени пользователя и кнопка ввода INPUT. Кнопка является неактивной (см. рис. 3) до тех пор, пока в поле ввода не будет введен хотя бы один символ. После этого кнопка станет активной для действия.



Рис. 3 – Кнопка INPUT неактивная для нажатия.

На основной странице игры (рис. 2) расположено стакан, окно для отображения следующей фигуры, кнопка PLAY для начала игры, кнопка RESET для перехода на первую страницу и повторную регистрацию и информация об уровне, количестве собранных линий и очках. См. рис. 4.

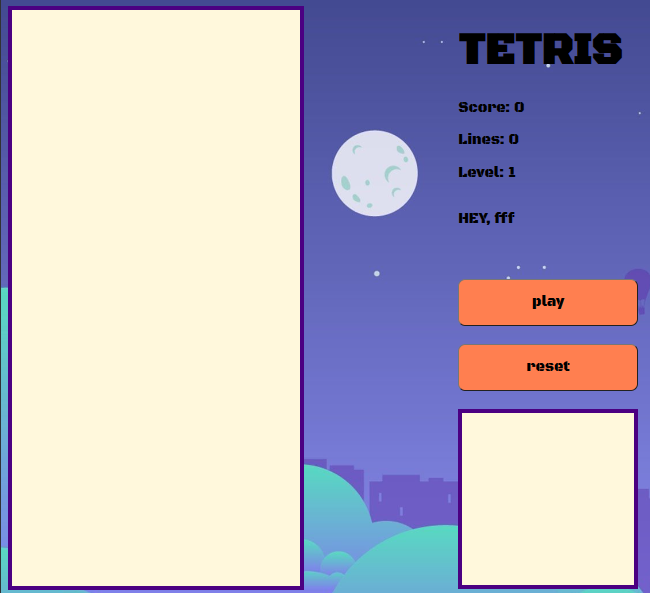


Рис. 4 – Информационный вид страницы до начала игры.

При нажатии на кнопку PLAY она становится неактивной (см. рис. 5).

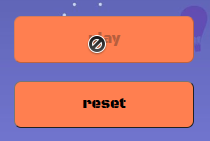


Рис. 5 – кнопка PLAY неактивна для нажатия.

Таблица рекордов появляется на странице в конце игры при проигрыше. См. рис. 6.



Рис. 6 – таблица рекордов.

1. CSS-файл.

Все основные описания для объектов страницы прописаны в файле style.css. В написании работы подключался шрифт ‘Black Ops One’ c [https://fonts.googlepis.com](https://fonts.googlepis.com/). Для фона страниц было использовано изображение. Для позиционирования объектов на странице использовался flexbox и css Grid.

1. Js – файлы.

Было написано два файла с кодом app1.js и app.js.

Первый файл отвечает за действия на первой странице: проверка наличия символов в форме (для активации кнопки INPUT) и сохранение имени пользователя в localStorage. Сохранение имени пользователя происходит следующим образом – если введенное имя уже есть на сайте, то оно помещается в конец списка, а элементы после него сдвигаются вверх по списку; если введенного имени не было раньше, то оно просто добавляется в конец.

Второй файл отвечает за игру. В начале объявляются константные величины (ширина и высота стакана, размер фигуры, количество фигур, объект с тетраминами, объект с цветами фигур, объект правил начисления очков), отрисовка элементов canvas. При нажатии на кнопку PLAYвызываются две функции start() и playgame(). Первая функция создает двумерный массив поля, а вторая функция осуществляет основной цикл игры.

Функция generate() создает создает рандомно набор фигур. В этой функции вызывается функция для генерации числа от 1 до 7.

Функция nextTetramino() берет из набора фигур последнюю и определяет для нее основные характеристики: номер фигуры, матрица фигуры, начальное положения для появления фигуры на поле.

Функция для поворота фигуры figure\_rotate() поворачивает фигуру по часовой стрелке.

Функция для постановки фигуры в поле setting\_the\_shape(). Проходимся по всем строкам и столбцам, если фигура сверху вылезает за пределы поля, то конец игры (вызывается функция окончания игры), если нет то фиксируем фигуру в поле. Функция также проверяет заполненность одного ряда – если ряд заполнен то убираем его и сдвигам все вниз на одну клетку. После окончания всех действий берем следующую фигуру из очереди и также вызываем функцию начисления очков, если был убран ряд.

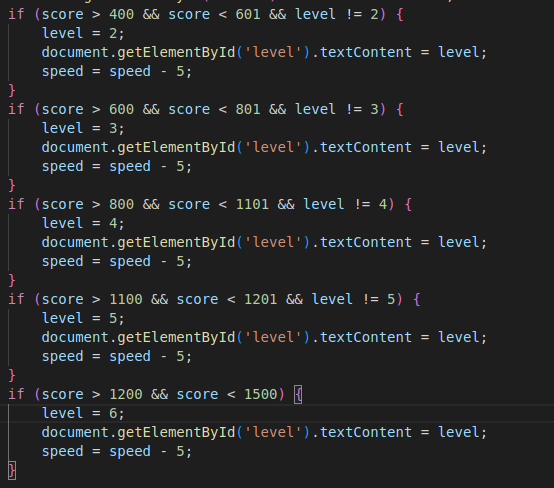
В основной функции игры playgame() запускаем анимацию с помощью requestAnimationFrame(), далее на каждом кадре очищаем игровое поле и отрисовываем его заново вместе с фигурами, находящимися на поле. Далее берем фигуру из очереди (сгенерированный набор) – на первом уровне она сдвигается вниз каждые 30 кадров – когда фигура закончила двигаться фиксируем ее на поле с помощью setting\_the\_shape().

Функция проверки движения фигуры checking\_after\_rotation() проверяет вылезает ли фигура за пределы поля по ширине и высоте и пересекается ли она с другими фигурами.

Функция окончания игры gameover() вызывается, если фигура вылезла за пределы поля. В теле функции останавливается анимация с помощью cancelAnimationFrame(), затем записываются очки, уровень и количество линий в информацию о пользователе, далее отрисовывается на игровом поле прямоугольная область с надписью «GAME OVER!» (проигрыш сигнализирует музыка) и вызывается функция для вывода таблицы рекордов.

Функция вывода таблицы рекордов table\_record() вырисовывает справа на странице квадратную область с информацией обо всех пользователях. Максимально в таблице выводятся 10 позиций, также все рекорды сортируются по убыванию и получается что выводятся 10 лучших результатов.

Функция начисления очков, количества линий и повышения уровня accrual() вызывается в случаи, если убрался ряд. Очки начисляются в соответствии с тем сколько рядов за раз было убрано. Уровни повышаются по правилам приведенным в коде:



1 уровень – у игрока изначально;

2 уровень – если очков больше 400 и меньше 601;

3 уровень – если очков больше 600 и меньше 801;

4 уровень - если очков больше 800 и меньше 1101;

5 уровень – если очков больше 1100 и меньше 1201;

6 уровень – если очков больше 1200 и меньше 1500.

При каждом повышение уровня увеличивается скорость падения фигур.

## Вывод.

В ходе лабораторной работы были изучены основы языка javascript, html и css, настроен сервер nginx для работы по протоколу HTTPS и написана создано веб-приложение – игра в тетрис.