

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

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Web-технологии»**  
**Тема: Разработка игр на языке JavaScript**

Студентка гр. 1304

\_\_\_\_\_

Ярусова Т.В.

Преподаватель

\_\_\_\_\_

Беляев С.А.

Санкт-Петербург

2023

## **ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Студентка Ярусова Т.В.

Группа 1304

Тема работы : разработка игр на языке JavaScript

Технические требования:

- Все делают в соответствии с общим заданием в соответствии с учебным пособием на «чистом» JavaScript (ES6). В группах по 1 человеку.
  - Минимум 2 уровня игры
  - Реализованы все менеджеры в соответствии с учебным пособием
- (УП)
- Есть таблица рекордов
  - Есть препятствия
  - Есть «интеллектуальные» противники и «бонусы»
  - Используются tiles с редактором Tiled в соответствии с УП

Содержание пояснительной записки:

«Содержание», «Введение», «Основные теоретические положения»,  
«Реализация программы», «Тестирование», «Пользовательская инструкция»,  
«Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 4.09.2023

Дата сдачи работы: 20.11.2023

Дата защиты работы: 20.11.2023

Студентка

\_\_\_\_\_

Ярусова Т.В.

Преподаватель

\_\_\_\_\_

Беляев С.А.

## АННОТАЦИЯ

В курсовой работе реализована игра, которая содержит несколько html страниц, их оформление css и саму игровую логику в js файлах. Реализована страница авторизации, страница, на которой повествуется об истории игры, страница игрового процесса и страница окончания игры с таблицей рекордов. Реализовано минимум 2 уровня, все менеджеры в соответствии с УП, есть таблица рекордов, препятствия, а также “интеллектуальные ” противники и “бонусы”. Используются tiles с редактором Tield в соответствии с УП.

## **SUMMARY**

The course work implements a game that contains several html pages, their css design and the game logic itself in js files. An authorization page has been implemented, a page that tells about the history of the game, a gameplay page and a game ending page with a high score table. At least 2 levels have been implemented, all managers are in accordance with the UP, there is a table of records, obstacles, as well as “intelligent” opponents and “bonuses”. Tiles with the Tile editor are used in accordance with the UP.

## СОДЕРЖАНИЕ

Введение	7
1. Реализация программы	8
1.1. Страница авторизации	8
1.2. Страница истории игры	8
1.3. Страница игрового действия	8
1.4. Страница окончания игры	13
2. Тестирование	12
3. Пользовательская инструкция	17
Заключение	18
Список использованных источников	19

## ВВЕДЕНИЕ

Целью данной работы является разработка игры на языке JavaScript. Необходимо реализовать минимум 2 уровня, реализовать менеджеры в соответствии с УП, реализовать таблицу рекордов, создать препятствия, а также «интеллектуальных» противников и «бонусов», использовать tiles с редакторов Tiled в соответствии с УП.

На основе выше изложенной цели, были сформулированы следующие задачи:

1. Изучение архитектуры игр на языке JavaScript
2. Изучение работы с localStorage
3. Изучение работы с редактором Tiled

## 1. РЕАЛИЗАЦИЯ ПРОГРАММЫ

### 1.1. Страница авторизации

html: init.html

css: init.css

В данных файлах разметка и стиль страницы, которая выглядит как блок, в котором есть поле ввода имени пользователя и кнопка для считывания информации из поля и перехода на новую страницу.

js: init.js

В данном модуле создается одноименный объект класса *Initializer()* и вызывается метод *initButton()*, который инициализирует кнопку, которая отвечает за переход на новую страницу и инициализацию пользователя. Если кнопка была нажата, то срабатывает метод *start()* и совершается переход на страницу *historyPage.html*

### 1.2. Страница истории игры

html: histryPage.html

css: historyPage.css

В данных файлах разметка и стиль страницы, которая выглядит как блок, в котором содержится текст и кнопка «Далее», которая является ссылкой, для перехода на страницу игрового действия.

### 1.3. Страница игрового действия

html: level\_1.html, level\_2.html

css: level.css



В данных файлах разметка и стиль страницы, которая выглядит как три блока. В первом блоке правила игры и инструкция по управлению персонажа, второй – канва, на которой с помощью скрипта отрисовывается игровое поле, третий – информации о количестве жизней и очках текущего игрока.

js:game.js

В данном модуле навешиваются слушатели событий на объекты страницы. С помощью метода *loadAll()* одноименного объекта класса *GameManager* происходит инициализация игрового поля, а с помощью метода *draw(ctx)* и *play()* происходит отрисовка игры и ее действие.

Реализованы функции *gameOver()* и *checkUsers()*, которые вызываются в момент окончания игры и перенаправляют на страницу [gameOverPage.html](#).

## **Managers**

- **GameManager**

Данный менеджер отвечает за загрузку данных о поле, хранение сущностей и их время жизни. В данном менеджере происходит основной цикл игры, в котором происходит реакция на перемещения игрока и отрисовка изменении состояний всех сущностей.

- **MapManager**

Данный менеджер отвечает за инициализацию игрового поля и за его отрисовку.

- **SpriteManager**

Данный менеджер отвечает за инициализацию спрайтов сущностей и за их отрисовку.

- **EventManager**

Данный менеджер отвечает за реакцию на пользователя. В частности на нажатие клавиш клавиатуры.

- **PhysicManager**

Данный менеджер отвечает за перемещение сущностей по игровому полю. За изменение их координат на поле.

- **SoundManager**

Данный менеджер отвечает за музыкальное сопровождение во время игры. Играет фоновая музыка на протяжении всей игры и дополняется звуками при взаимодействии сущностей между собой.

## **Entities**

Класс сущностей, который имеет базовые поля для каждой сущности, которая расширяет данный класс.

- **Player** – класс сущности игрока и соответствующие ему реакции на объекты поля и другие сущности на поле.

- **Enemy** - класс сущности врага и соответствующие ему реакции на объекты поля и другие сущности на поле. Также прописана функция *checkPlayer()*, с помощью которой отслеживается позиция игрока на поле. Если игрок находится в поле видимости врага, то вызывается функция преследования игрока *A\_star()*.

- **Bonus** – класс сущности бонуса, за который игроку начисляются очки.

- **Weapon** – класс сущности оружия, который создается в классе *Player* при нажатии определенной клавиши. При взаимодействии с полем или другой сущностью уничтожается. Если сущность, с которой столкнулась данная сущность, враг, то данная сущность уничтожает еще и врага.

#### **1.4.Страница окончания игры**

html: gameOverPage.html

css: gameOverPage.css

В данных файлах разметка и стиль страницы, которая выглядит как блоки, в которых содержится информация об окончании игры, информация о текущем игроке и таблица рекордов. Также содержит кнопку «Начать сначала», которая перенаправляет на страницу с первым уровнем игры, и кнопку «Меню», которая перенаправляет на страницу авторизации.

js: gameOver.js

В данном файле идет заполнение информации для отображения на странице. В модуль информация приходит из localStorage. Также реализована функция компаратора recordSort(), с помощью которой сортируется таблица рекордов по убыванию очков.

### 3. ТЕСТИРОВАНИЕ

#### 1. Страница авторизации (см. рис. 1)



Рисунок 1 – страница авторизации

#### 2. Страница истории персонажа и цели игры (см. рис. 2)

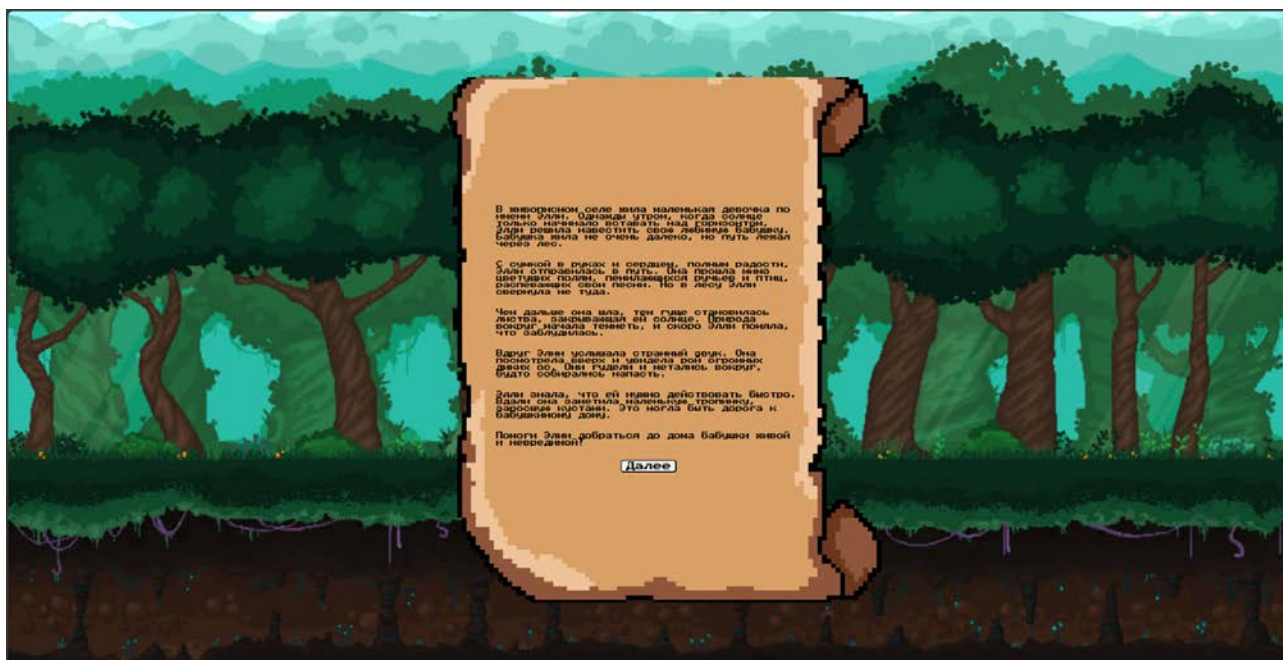


Рисунок 2 – страница истории персонажа и цели игры

### 3. Страница игрового процесса (см. рис. 3 – 4)



Рисунок 3 – Первый уровень

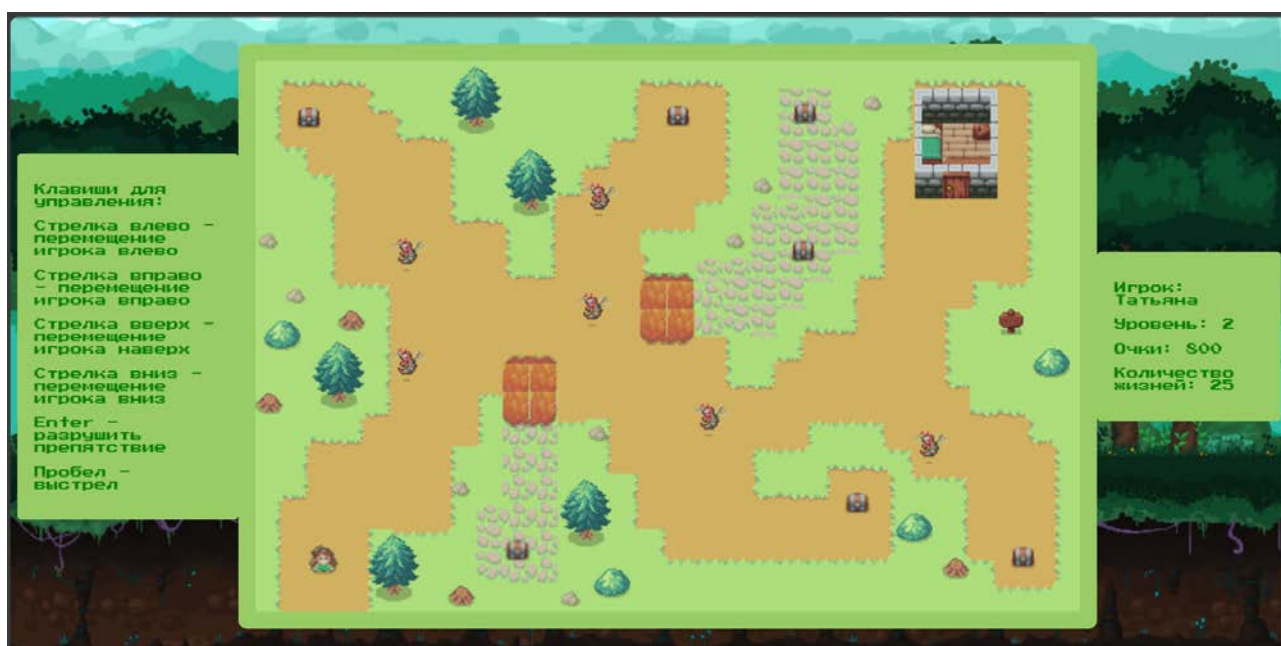


Рисунок 4 – Второй уровень



#### 4. Реакции игрока на нажатие клавишей (см. рис. 5-6)



Рисунок 5 – Реакция на клавишу «Enter». Совершение удаления стены



Рисунок 6 – Реакция на клавишу «space». Совершение выстрела.

#### 5. Увеличение очков в таблице (см. рис. 7)

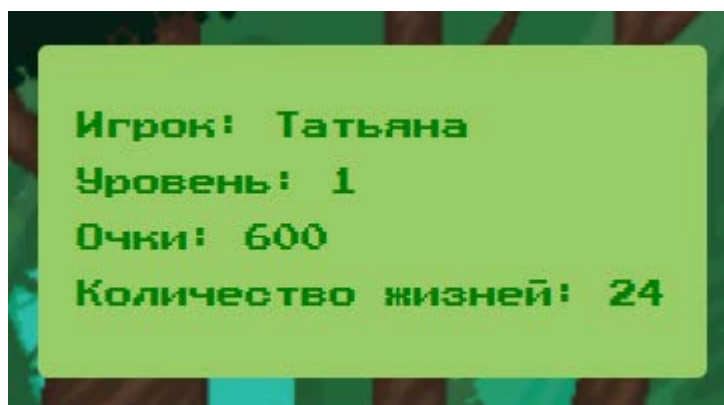


Рисунок 7 – Увеличение очков за бонусы и убийство врага

## 6. Переход на новый уровень (см. рис.8)



Рисунок 8 – Событие, после которого открывается новый уровень

## 7. Страница после окончания игры (см. рис. 9)



Рисунок 9 – Страница после окончания игры



#### **4.ПОЛЬЗОВАТЕЛЬСКАЯ ИНСТРУКЦИЯ**

1. Откройте файл init.html и запустите его через браузер.
2. Для начала прохождения игры Вам необходимо авторизироваться и нажать кнопку «Войти».
3. Далее откроется страница с историей персонажа и целью игры. Необходимо нажать кнопку «Далее», чтобы перейти к самой игре.
4. На странице игрового процесса описаны правила управления.
5. После поражения или победы откроется страница окончания игры, на которой будет выведена таблица рекордов и будет предложено начать игру сначала или перейти на страницу авторизации.

## **ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы изучены особенности архитектуры игр на языке JavaScript, наработаны навыки работы с localStorage и с редактором Tiled.

Написана игра на языке JavaScript. Реализовано 2 уровня, реализовано менеджеры в соответствии с УП, реализована таблица рекордов, созданы препятствия, а также «интеллектуальные» противники и «бонусы», использованы tiles с редактором Tiled в соответствии с УП.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Правила оформления пояснительной записки к курсовой работе// se.moevm.info URL: <http://se.moevm.info/doku.php/courses:programming:report>
2. Учебное пособие Беляев С. А. Б 43 Разработка игр на языке JavaScript: Учебное пособие. — СПб.: Издательство «Лань», 2016. — 128 с.:ил.— (Учебники для вузов. Специальная литература).
3. Редактор Tiled// URL: [www.mapeditor.org](http://www.mapeditor.org)