

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

**ОТЧЕТ**  
**по курсовой работе**  
**по дисциплине «Web-Технологии»**  
**Тема: Разработка игр на языке JavaScript**

Студент гр. 1304

Шаврин А.П.

Преподаватель

Беляев С.А.

Санкт-Петербург

2023

## Цель работы.

Создать игру на языке JavaScript, изучить особенности разработки игр на языке JavaScript, освоить необходимые для решения задачи инструменты.

## Задание.

Необходимо создать игру на языке программирования JavaScript.

Все делают в соответствии с общим заданием в соответствии с учебным пособием на «чистом» JavaScript (ES6). В группах по 1 человеку.

1. Минимум 2 уровня игры
2. Реализованы все менеджеры в соответствии с учебным пособием (УП)
3. Есть таблица рекордов
4. Есть препятствия
5. Есть «интеллектуальные» противники и «бонусы»
6. Используются tiles с редактором Tiled ([www.mapeditor.org](http://www.mapeditor.org)) в соответствии с УП

## Выполнение работы.

1. В качестве сервера был настроен и использован express сервер. Сперва были заданы необходимые настройки (файл server.js), а затем в router.js выполнялась обработка запросов со стороны приложения

```
1  import express from "express"
2  import http from "http"
3
4  import { router } from "../router.js"
5
6  const app = express()
7
8  // Подключение обработчика шаблонов ejs, шаблоны - в папке views
9
10 app.set("view engine", "ejs")
11 app.set("views", `./views`)
12
13
14 //Настройка приложения
15 app.use("/public", express.static("public")) // Указание статической папки public
16 app.use(express.json()) // Обработка параметров в body
17 app.use("/", router) // Использование маршрутизации
18
19 //Настройка http сервера
20 const http_server = http.createServer(app)
21
22 // Задание порта
23 http_server.listen([8080])
```

Рис 1. Server.js.

```
1 import express from "express"
2
3
4 export const router = express.Router();
5
6 router.get("/", (req, res) => {
7   res.redirect("./authorization");
8 })
9
10 router.get("/authorization", (req, res) => {
11   res.render("authorization.ejs");
12 })
13
14 router.get("/game", (req, res) => {
15   res.render("game.ejs");
16 })
17
18 router.get("/game_over", (req, res) => {
19   res.render("game_over.ejs");
20 })
21
```

Рис 2. Router.js.

2. В игре реализовано три страницы:

- Authorization – на ней, происходит ввод и запоминание в localStorage имени текущего игрока и выбранного им уровня.



Рис 3. Страница authorization.

- Game\_over – в конце игры игрока переносит на эту страницу, где он может увидеть таблицу рекордов.



Рис 4. Страница game\_over.

- Game - на этой странице происходит вся игровая логика.

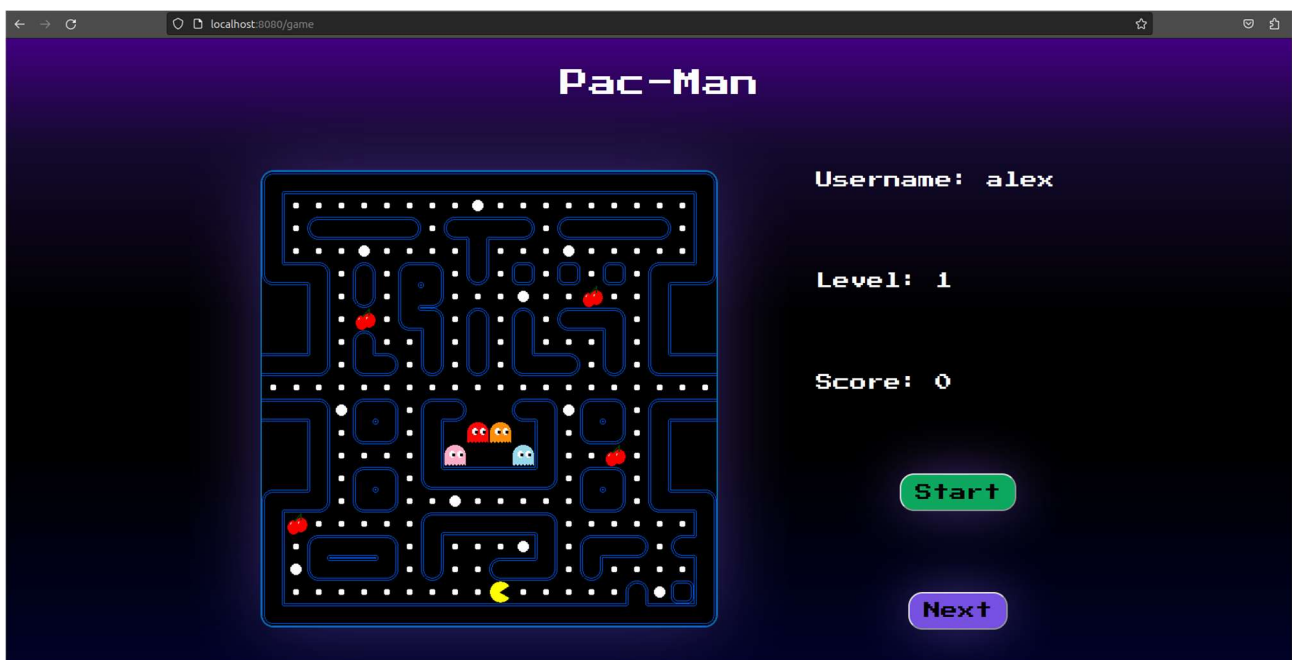


Рис 5. Страница game.

3. Карты уровней и начальное расположение объектов игры на них создано с помощью редактора карт Tiled. Были загружены необходимые изображения и из них составлено 2 игровые карты, на которые затем были размещены все объекты в их стартовом положении.

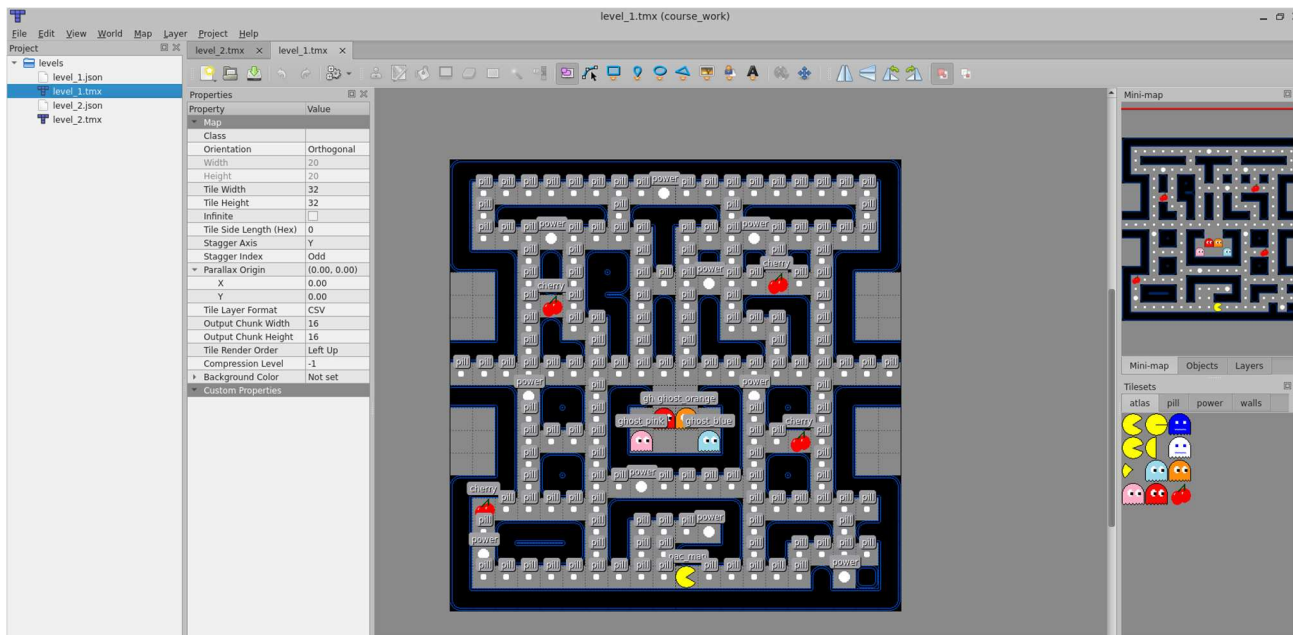


Рис 6. Уровень 1.

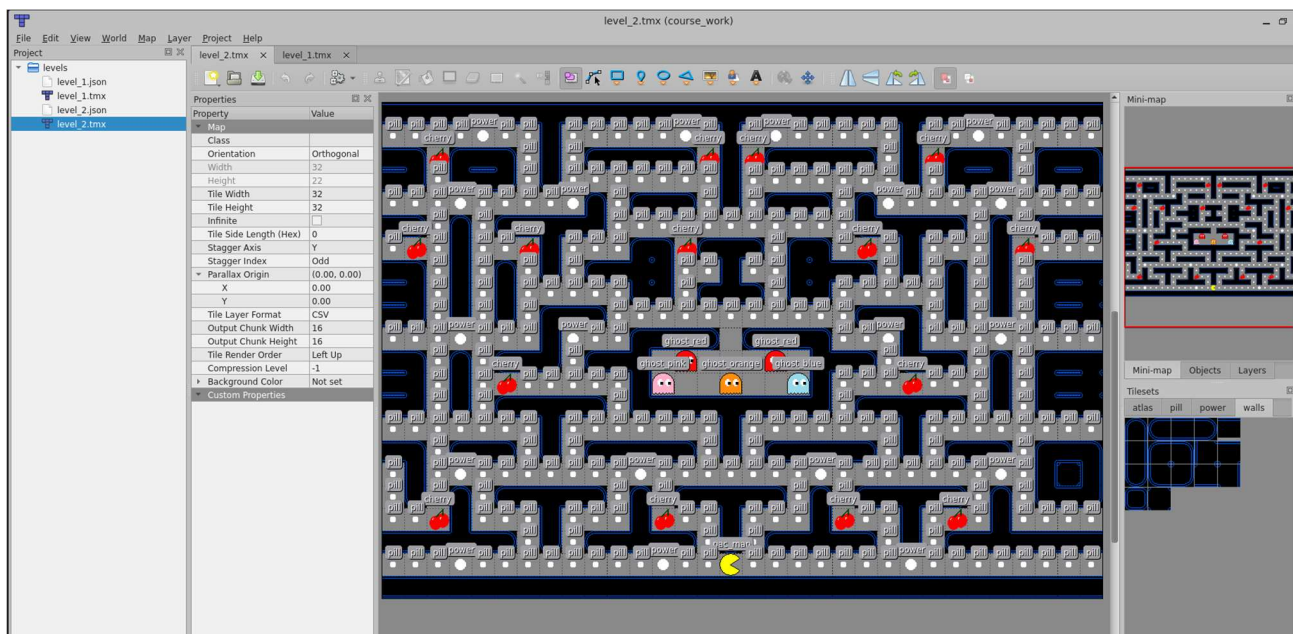


Рис 7. Уровень 2.

4. Все объекты в игре – наследники базового класса объектов игры `GameObject`, а все сущности наследники базового класса `Entity` (`Entity` – наследник от `GameObject`).

Реализован следующий игровой объект:

- `Bonus` – объект бонусов

Реализованы следующие сущности-игровые объекты:

- `Player` – сущность игрока

- Ghost – сущность врага

5. В игре существуют проходимые клетки и непроходимые (клетки стен). Так как реализованная в PhysicManager логика перемещения объектов работает со всеми сущностями, непроходимые клетки непроходимы как для игрока, так и для врагов.

6. Враги умеют преследовать игрока при помощи алгоритма поиска пути A\*, когда тот подходит достаточно близко к ним. Пока игрок находится достаточно далеко от врага или же игрок может убить врагов, те перемещаются случайным образом.

7. Скриншоты игрового процесса:

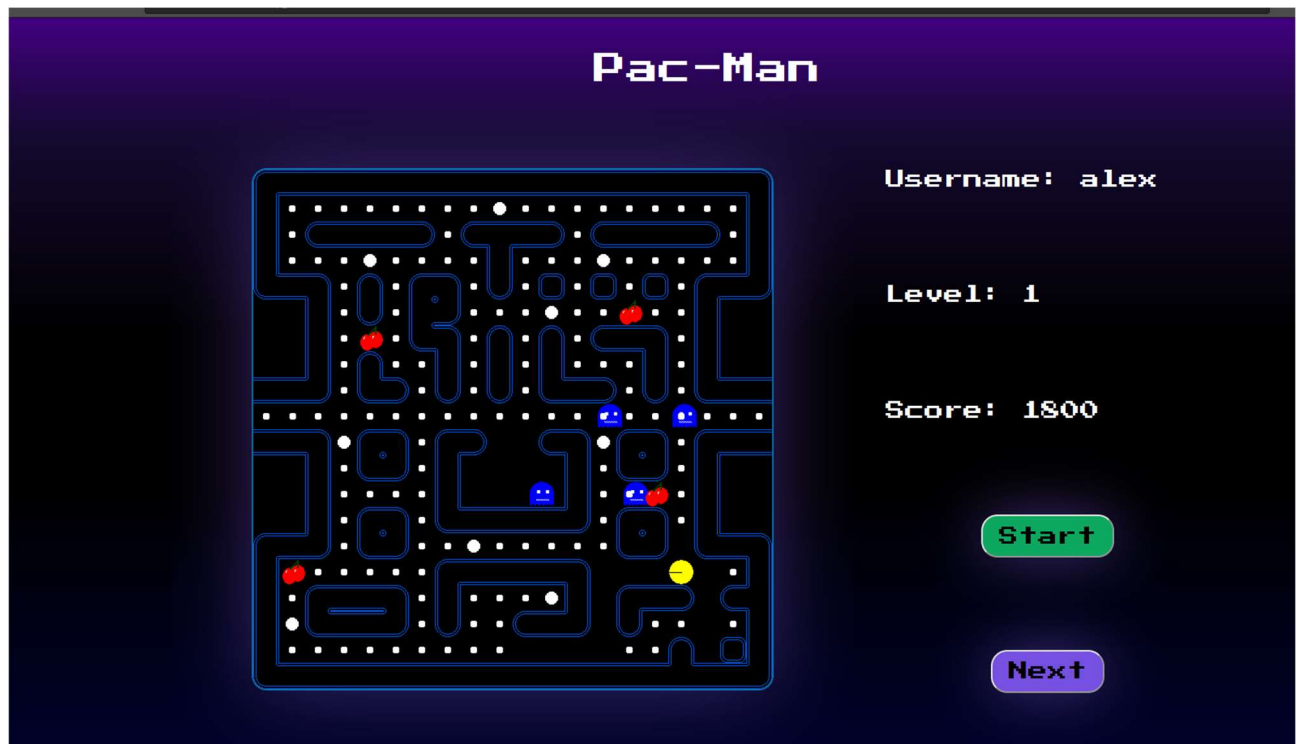


Рис 8. Игровой процесс 1й уровень.

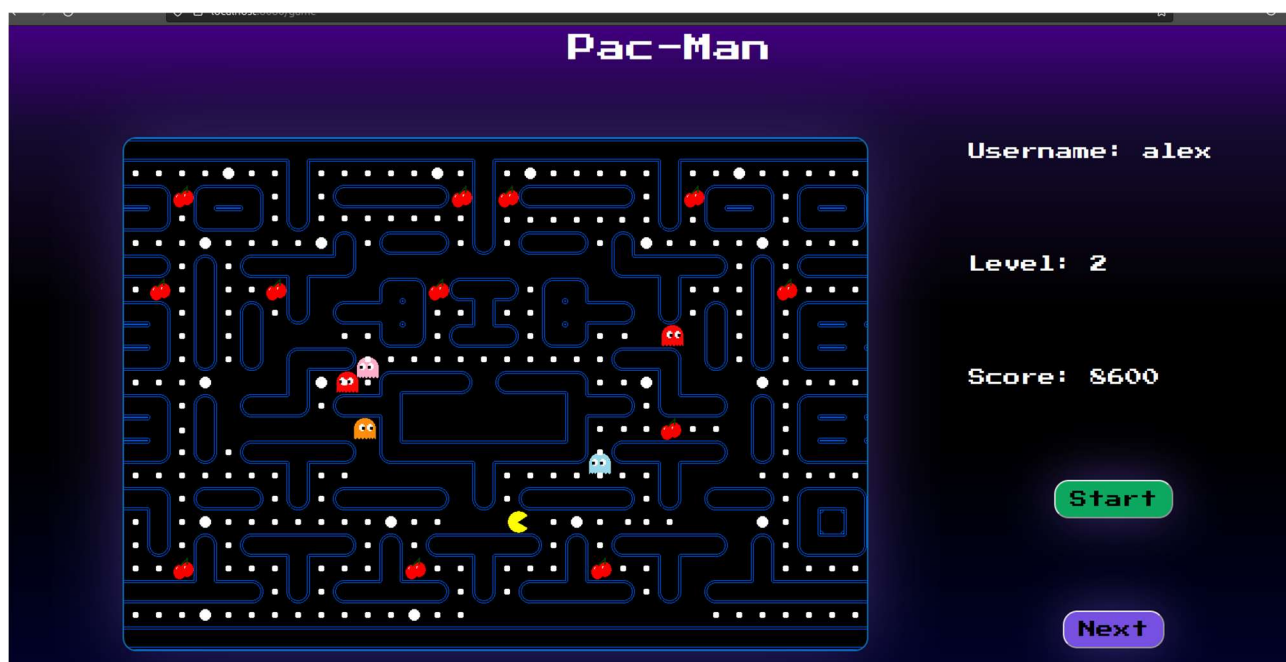


Рис 9. Игровой процесс 2й уровень.

### **Выводы.**

Создана, запускающаяся в браузере игра на языке JavaScript. Реализованы все менеджеры из методических указаний. Освоена работа с такими инструментами, как Tiled и Leshy SpriteSheet, а также изучены новые возможности применения языка JavaScript.