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. 0382 Корсунов А.А.

Преподаватель Беляев С.А.

Санкт-Петербург

2022

Цель работы.

Целью работы является разработки игры на языке JavaScript, отвечающей заданным требованиям.

Основные теоретические сведения.

JavaScript (сокращенно - JS) — это легковесный, интерпретируемый объектно-ориентированный язык. Наиболее широкое применение находит как язык сценариев веб-страниц, но также используется и в других программных продуктах, например, node.js.

Tiled — кроссплатформенный открытый редактор тайловых карт для игр. Он позволяет создавать карты для 2-мерных игр (с видом сбоку, таких, как платформеров, или с видом сверху, к примеру JRPG).

Постановка задачи.

Задачи:

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 с соответствии с УП

Ход работы.

1) С помощью приложения Tiled были созданы две карты для игры:

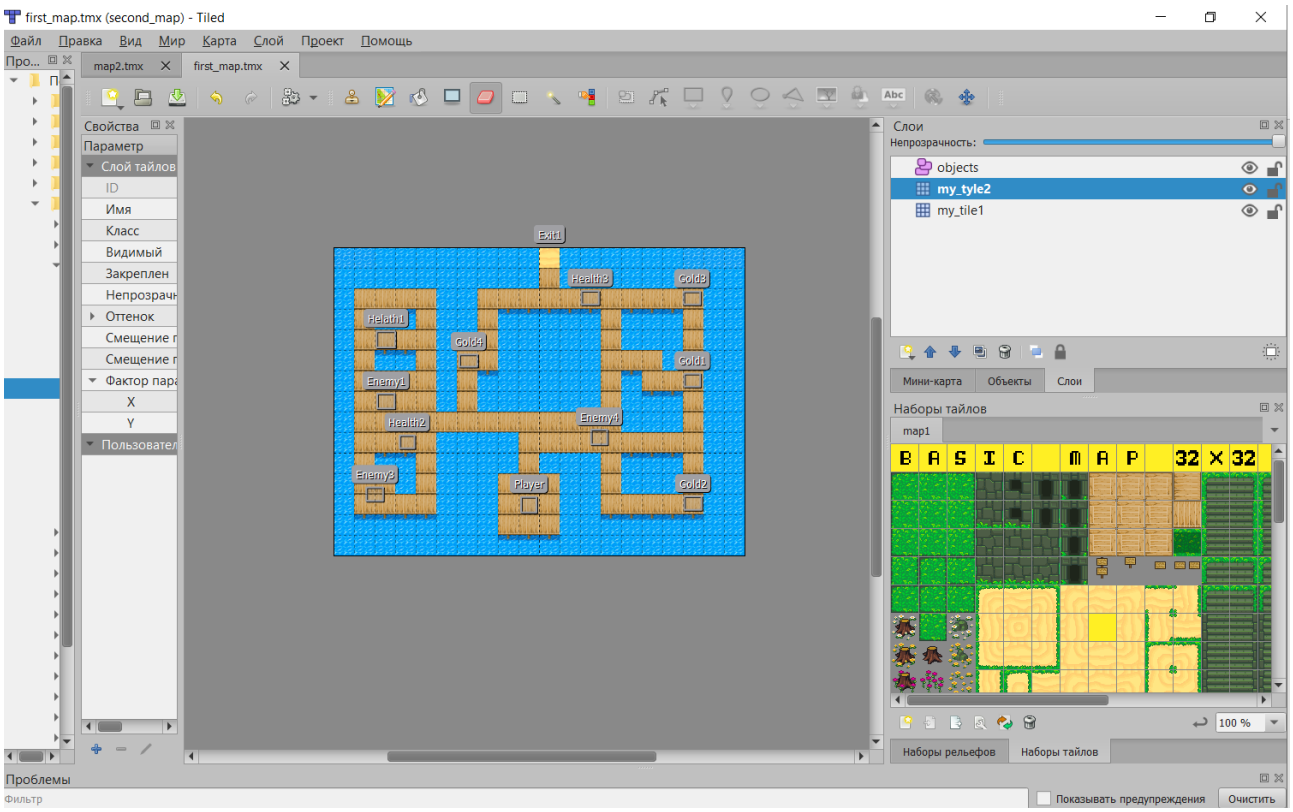

Рисунок 1 — Карта для первого уровня игры

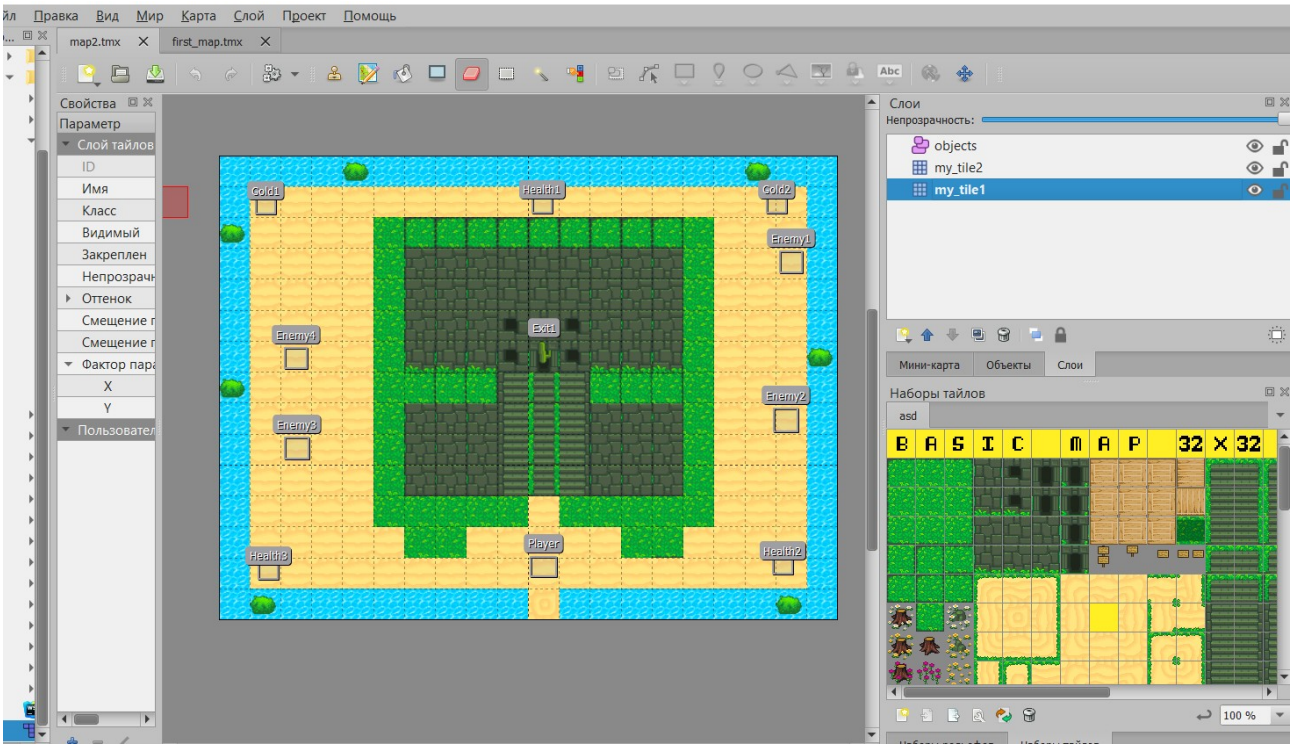

Рисунок 2 — Карты для второго уровня игры

Каждая карта (рис. 1-2) имеет два слоя тайлов и один слой объектов. Обе были сохранены в формате JSON:

Рисунок 3 — Карты, подключенные к коду игры

2) С помощью программы Leshy SpriteSheet Tool был создан атлас спрайтов для объектов игры (рис. 4)

Рисунок 4 — Спрайты игры

«Рыцарь» - спрайта для объекта, управляемого игроком,

«Летучая мышь» - спрайт для объектов-врагов,

«Золото» - спрайт для объектов, при поднятии которых игрок получает «Очки», которые непосредственно влияют на место в таблице рекордов

«Снаряды» - спрайты для атаки объектов

«Бутыль» - спрайт для объектов, увеличивающих количество очков здоровья игрока на 1

3) Были реализованы все менеджеры в соответствии с учебным пособием (УП):

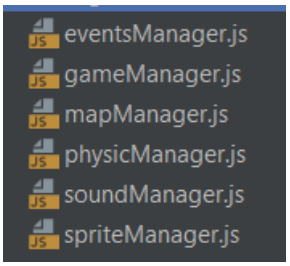

Рисунок 5 — Менеджеры игры

Каждый менеджер представляет из себя класс с соответствующем названием.

3.1) mapManager

Данный менеджер отвечает за обработку карт:

- В конструкторе происходит «разбор» json-файла, отвечающего за карту уровня.
- С помощью метода draw() можно нарисовать карту в контексте.
- С помощью метода getTile() можно получить блок по его индексу для его отображения
- Метод getTileset — вспомогательная функция для получения блока по индексу — возвращает найденный tileset, который разбирается в getTile()
- С помощью метода parseEntities() можно разобрать слой объектов карты, здесь же происходит создания объектов игры
- С помощью метода getTilesetIdx() - можно получить блок по координам
- CenterAt() - метод для центрирования области относительно положения игрока

3.2) spriteManager

Данный менеджер отвечает за обработку атласа спрайтов

- С помощью метода loadAtlas() можно загрузить атлас изображений
- С помощью метода loadImg() можно загрузить изображения менеджера спрайтов
- С помощью метода parseAtlas() можно разобрать атлас с объектами
- С помощью метода drawSprite() можно отобразить спрайт на карте
- С помощью метода gerSprite() можно получить спрайт по его имени

3.3) eventsManager

Данный менеджер отвечает за обработку событий на карте (т. е. за взаимодействие с пользователем)

- С помощью метода setup() сопоставляются события с клавишами на клавиатуре и мыши
- Метод kill() - отключает прослушивание других событий

- Другие методы сопоставляют события и действия, так, на клавиши «wasd» игрок может перемещаться в четырех направлениях, а на клавишу «пробел» и кнопки мыши — метать снаряды

3.4) physicManager

Данный менеджер отвечает за логику взаимодействия объектов на карте в каждый тик времени

- С помощью метода enityAtXY() можно определить объект по его координатам (используется для определения взаимодействия объектов при столкновении)
- Метод update() определяет взаимодействие объекта с другими элементами уровня, такие как: на какие блоки объект может наступать, а на какие нет, условия уничтожения объекта, взаимодействие объекта с другими объектами

3.5) gameManager

Данный менеджер отвечает за инициализацию, загрузку всех необходимых ресурсов, хранение и управление всеми объектами игры, обновление уровня игры

- В конструкторе происходит загрузка карты, атласа, инициализация объектов, событий, отрисовка уровня игры;
- Метод initPlayer() инициализирует объект игрока
- Метод kill() заносит объекты, в специальный массив, эти объекты позже будут уничтожены
- Метод update() обновляет уровень игры
- Метод play() начинает исполнение игры

3.6) soundManager

Данный метод отвечает за проигрывание звука в игре

- С помощью метода load() можно загрузить один аудиофайл

- С помощью метода loadArray() можно загрузить массив аудиофайлов
- С помощью метода play() можно проиграть загруженный аудиофайл
- С помощью метода stopAll() можно отключить все звуки в игре

4) Таблица рекордов выводиться пользователю при успешном завершении игры — убийстве всех врагов на втором уровне и достижения определенного блока — входа в «храм» (рис. 6-8)

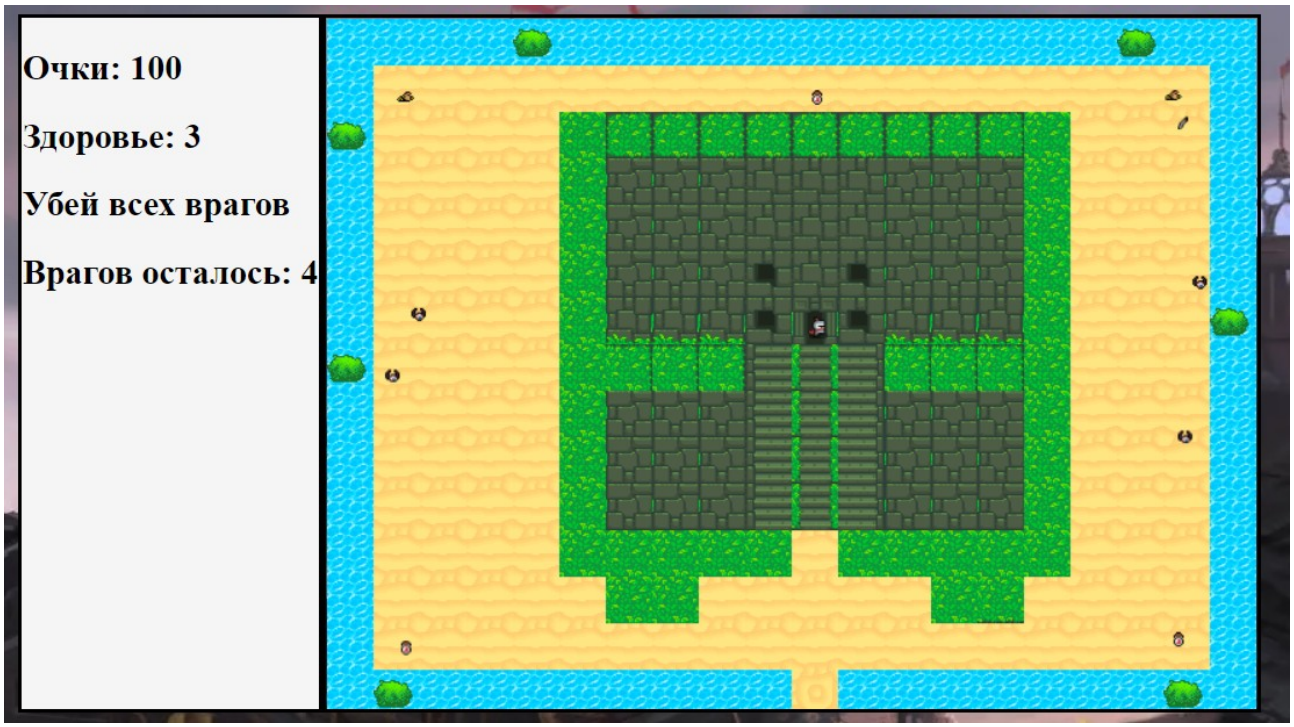

Рисунок 6 — Невозможность завершить игру, пока не будут убиты все враги на втором уровне игры

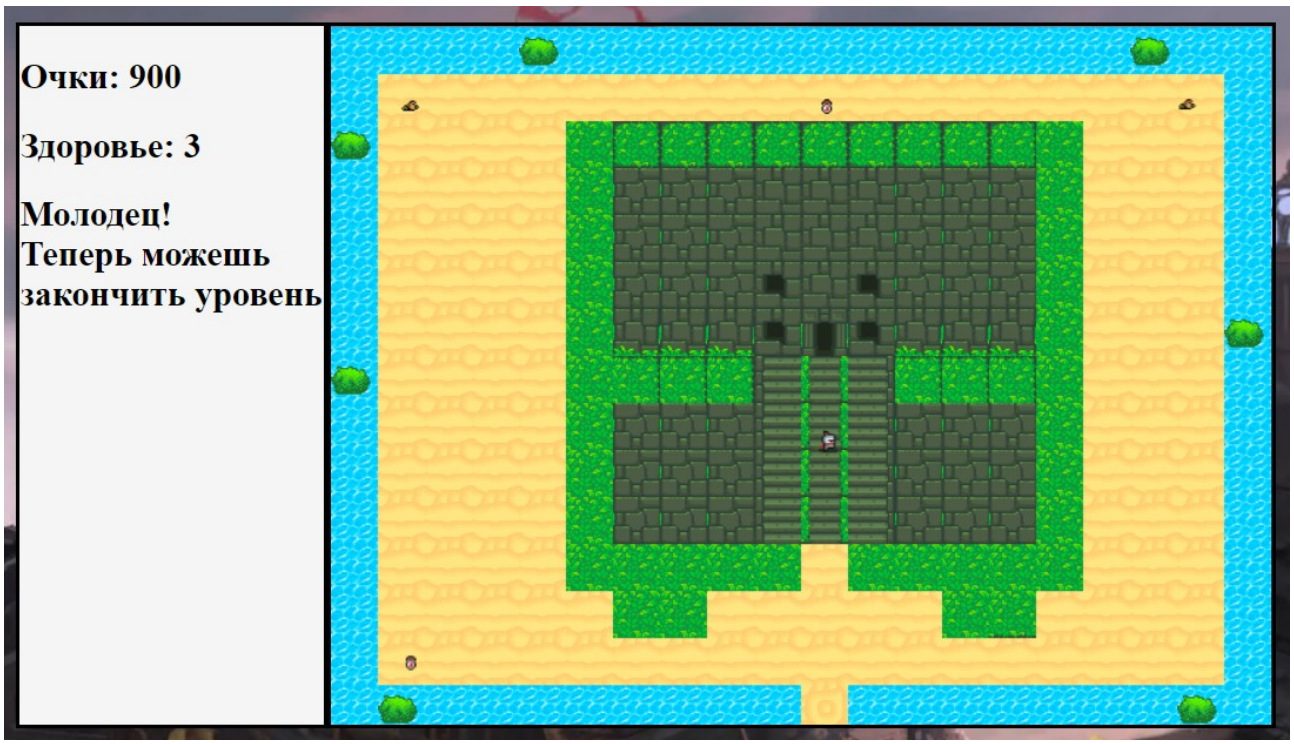

Рисунок 7 — Все враги на уровне убиты — можно завершить игру

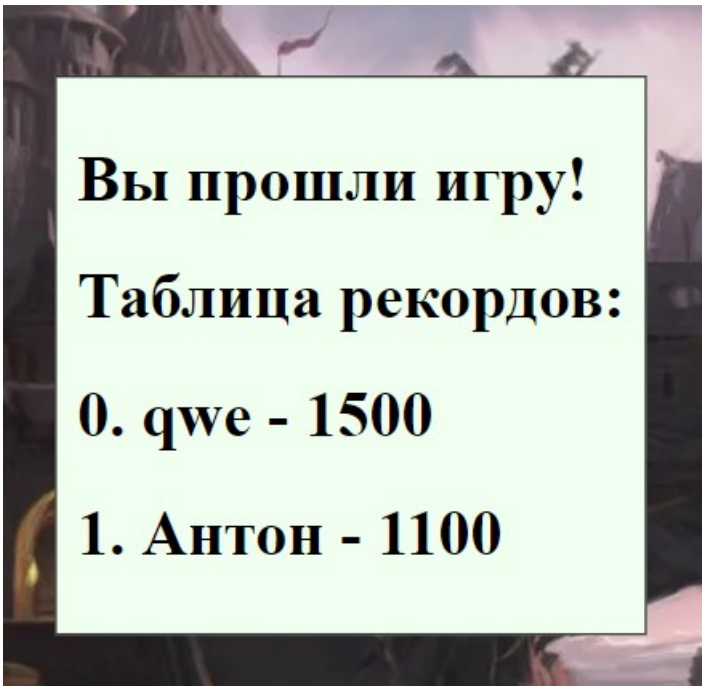

Рисунок 8 — Таблица рекордов при достижении условий прохождения игры

Таблица рекордов хранить в локальном хранилище, функция для взаимодействия с ним — update_records()

5) В игре есть препятствия — игрок не может наступить на блоки воды, храма или кусты, так же игрок не может наступить на врага или бонусы (от врага он получит урон, а бонус он «поднимет»)

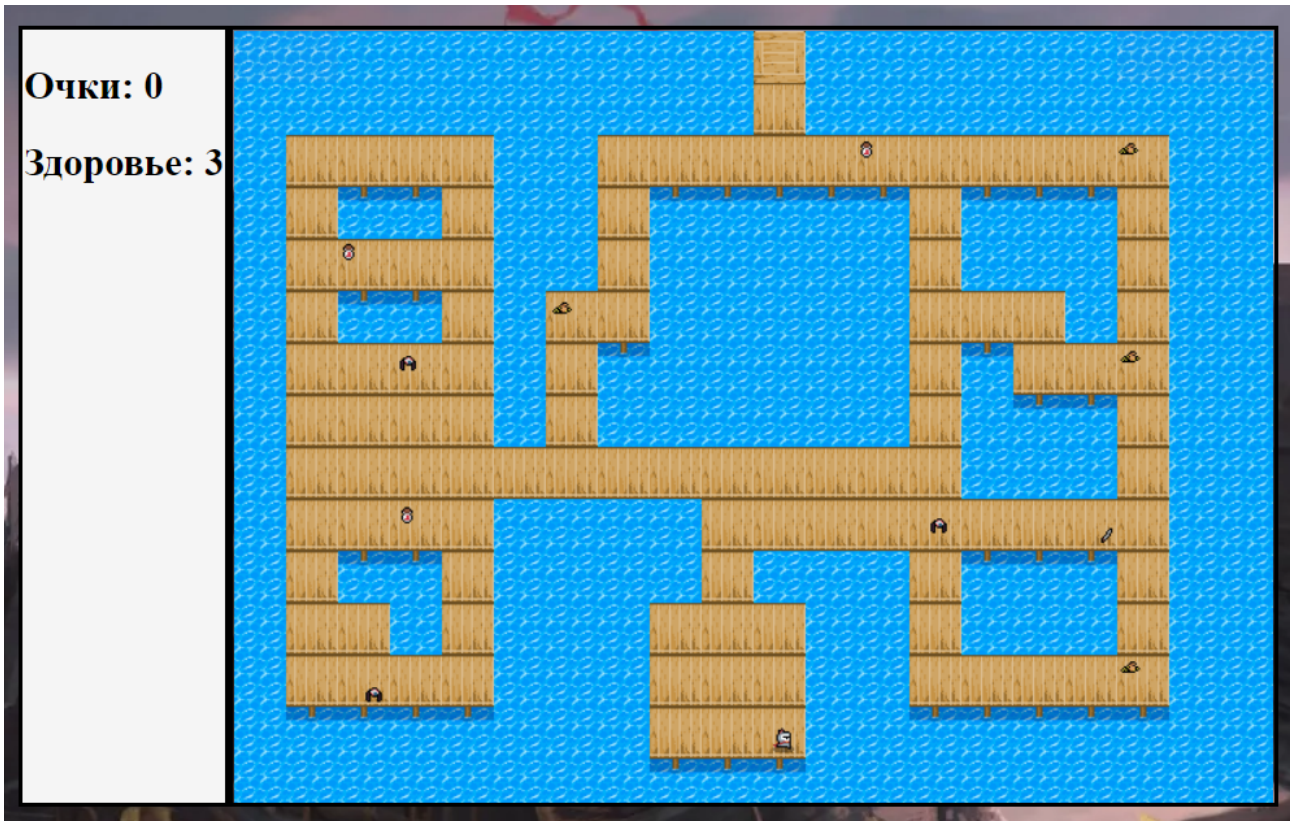

Рисунок 9 — Невозможность перейти препятствия — блоки воды

6) В игре есть интеллектуальные противники и бонусы:

Противники (рис. 9) изображены в виде летучих мышей: они умеют двигаться в четырех направлениях и в определенные интервалы кидать снаряды, в случае если они упрутся в препятствие — они сразу поменяют свое направление. Противник имеет 3 очка здоровья, после его уничтожения игрок получает 200 очков

Бонус (монеты) — представляют собой небольшую горсть монет, при столкновении объекта игрока с этим бонус последний уничтожается, а игрок получает 100 очков (рис. 10-11)

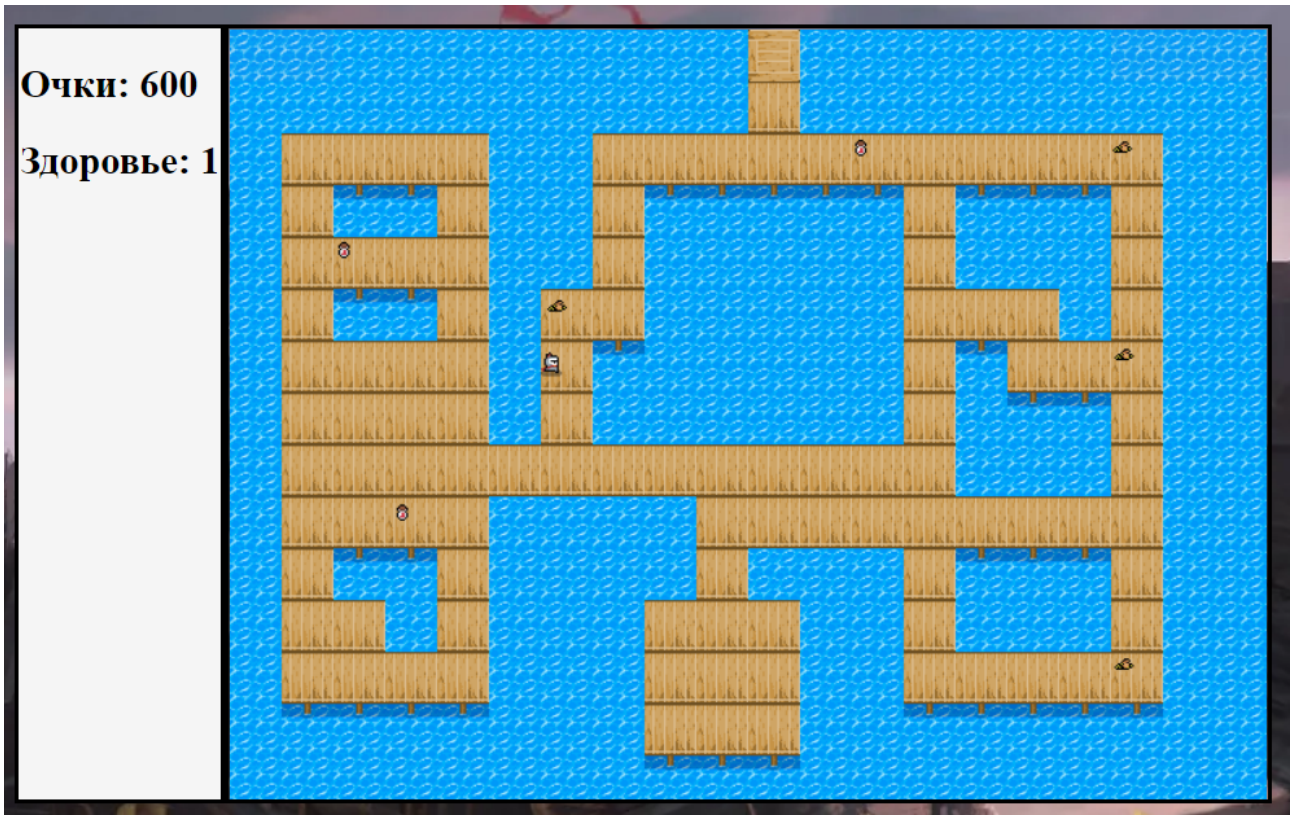

Рисунок 10 — Приближение к бонусу «Монеты»

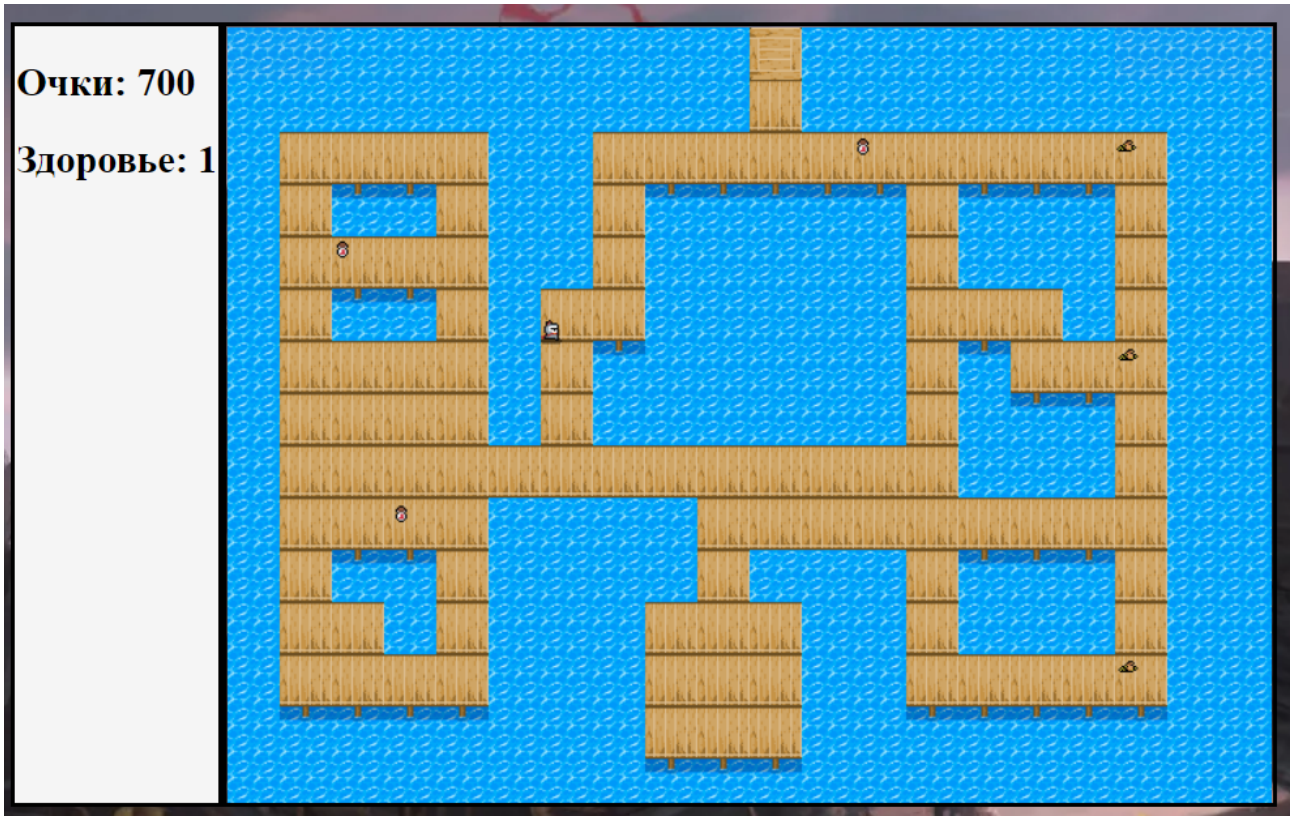

Рисунок 11 — Столкновение игрока с бонусом «Монеты», последний уничтожился, а игрок получил 100 очков

Бонус (бутыль) — представляет собой небольшое красное зелье, при столкновении объекта игрока с этим бонус последний уничтожается, а игрок получает 1 очко здоровья (рис. 12-13)

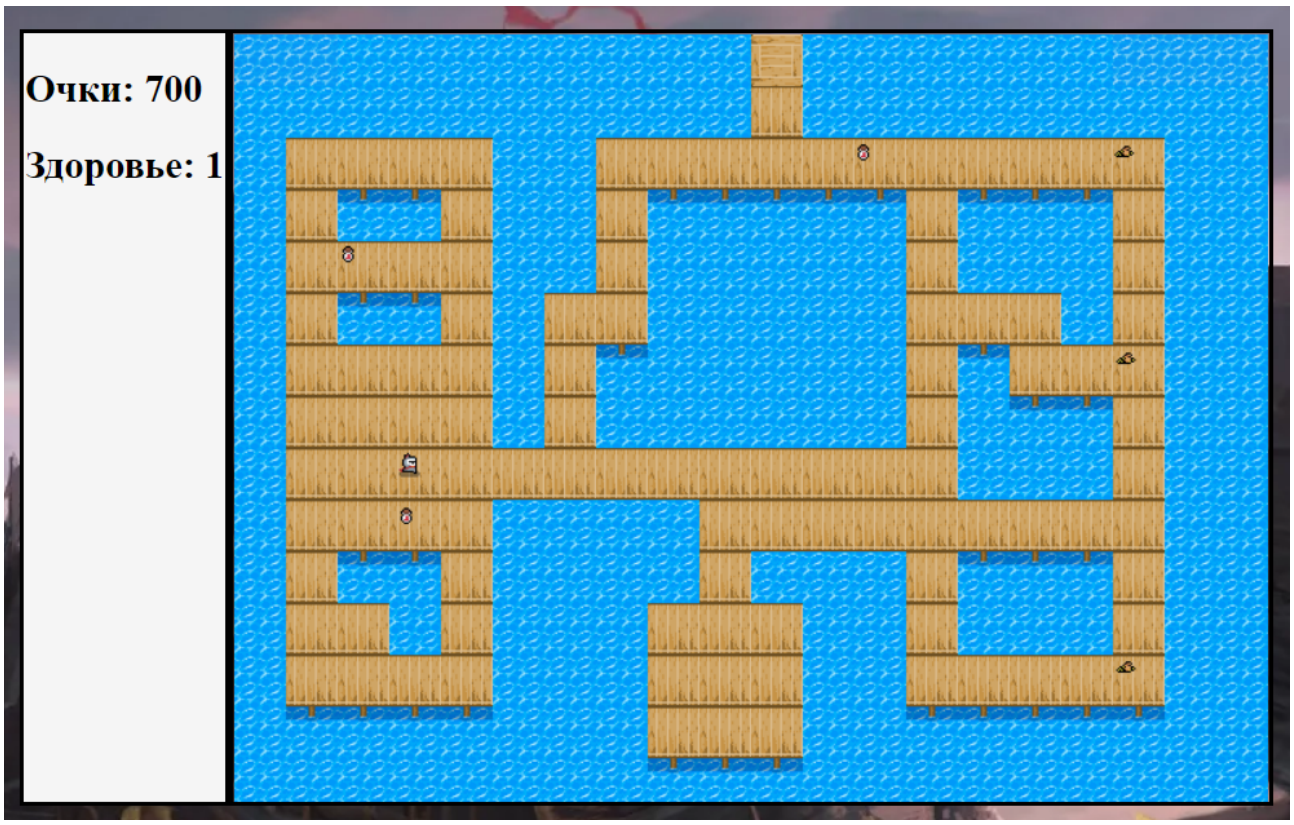

Рисунок 12 — Приближение к бонусу «Бутыль»

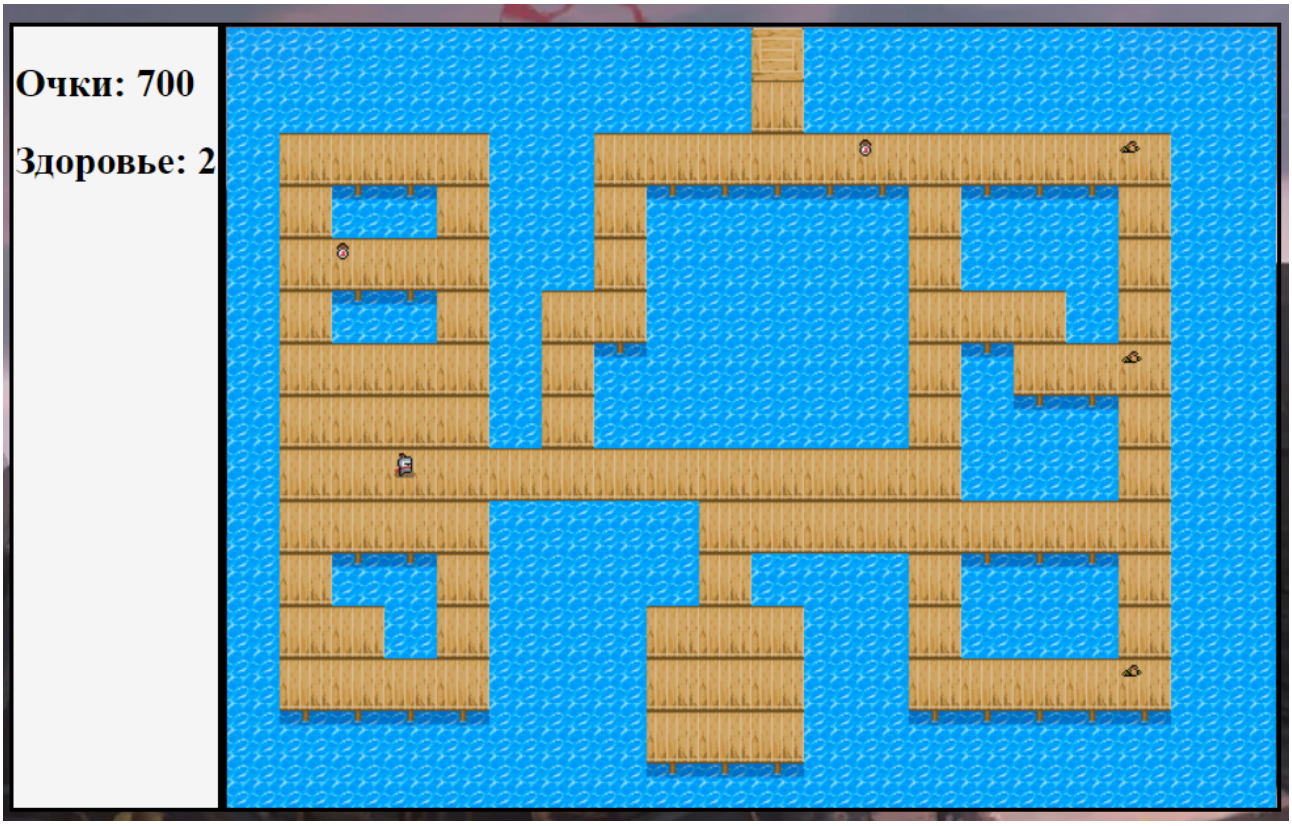

Рисунок 13 — Столкновение игрока с бонусом «Бутыль», последний уничтожился, а игрок получил 1 очко здоровья

Выводы.

Была разработана игра на языке JavaScript, отвечающая заданным требованиям: два уровня игры, есть враги и бонусы, есть препятствия, карты создавались с помощью редактора карт Tiled.