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

КУРСОВАЯ РАБОТА

по дисциплине «WEB-технологии»

Тема: «Разработка игры на языке JavaScript»

Студент гр. 0303 _____ Мыратгелдиев А.М.

Преподаватель _____ Беляев С.А.

Санкт-Петербург

2022

ЗАДАНИЕ
НА КУРСОВУЮ РАБОТУ

Студент Мыратгелдиев А.М.

Группа 0303

Тема работы: Разработка игры на языке JavaScript

Исходные данные:
Необходимо написать на ES6 JavaScript игру в соответствии с учебным пособием Беляев С.А. «Разработка игр на языке JavaScript».

Содержание пояснительной записки:
«Содержание», «Введение», «Менеджер карты», «Отображение объектов», «Взаимодействие с пользователем», «Реализация логики поведения объектов», «Управление звуками», «Управляющий элемент игры» «Заключение», «Список использованных источников».

Предполагаемый объем пояснительной записки:
Не менее 12 страниц.

Дата выдачи задания: 02.09.2022

Дата сдачи реферата: 02.12.2022

Дата защиты реферата: 02.12.2022

Студент _____ Мыратгелдиев А.М.

Преподаватель _____ Беляев С.А.

АННОТАЦИЯ

В курсовой работе необходимо создать игру на языке программирования JavaScript (ES6) в соответствии с ученым пособием Беляев С.А. «Разработка игр на языке JavaScript».

СОДЕРЖАНИЕ

	Введение	5
1.	Менеджер карты	6
1.1.	Сохранение карты	6
2.	Отображение объектов	8
2.1.	Создание объектов игры	8
2.2.	Загрузка изображений для объектов	8
3.	Взаимодействие с пользователем	10
3.1.	Реализация менеджера событий	10
4.	Реализация логики поведения объектов	11
4.1.	Менеджер физики объектов	11
5.	Управление звуками	12
5.1.	Менеджер звука	12
6.	Управляющий элемент игры	13
6.1.	Менеджер игры	13
	Заключение	14
	Список использованных источников	15
	Приложение А. Скриншоты работы программы	16

ВВЕДЕНИЕ

Цель работы:

Научится создавать интернет-приложение - игру на языке программирования JavaScript (стандарт ES6).

Основные задачи:

В игре необходимо реализовать:

1. Минимум 2 уровня игры.
2. Все менеджеры в соответствии с учебным пособием.
3. Таблицу рекордов.
4. Препятствия.
5. «Интеллектуальных» противников и «бонусы».
6. Используются tiles с редактором Tiled в соответствии с учебным пособием.

1. МЕНЕДЖЕР КАРТЫ

1.1. Сохранение карты

Для сохранения карты игры, объектов на карте и отображения карты был написан объект *mapManager*. В нем хранятся данные о карте, такие как размер карты в тайлах, размер тайлов в пикселях, количество слоев, сами слои.

```
let mapManager = {
    xCount: 0,
    yCount: 0,
    tSize: {x: 32, y: 32},
    mapSize: {x: 32, y: 32},
    tileSets: []
}

mapManager.parseImageObj = function (image, t) {...}  // парсинг тайлсетов

mapManager.loadImage = function (t) {...}  // загрузка тайлсетов

mapManager.parseMap = function (tilesJSON)  {...}  // парсинг карты

mapManager.loadMap = function (path) {...}  // загрузка карты

mapManager.getTileset = function (tileIndex) {...}

mapManager.getTile = function (tileIndex) {...}

mapManager.drawMap = function (ctx) {...}  // отрисовка карты

mapManager.parseEntities = function () {...}  // парсинг сущностей (объектов)

mapManager.getDataIdx = function (x, y) {...}

mapManager.getTilesetIdx = function (x, y) {...}
```

Рис. 1. Поля и методы объекта mapManager.

Созданные карты представлены на рис. 2 и рис. 3.

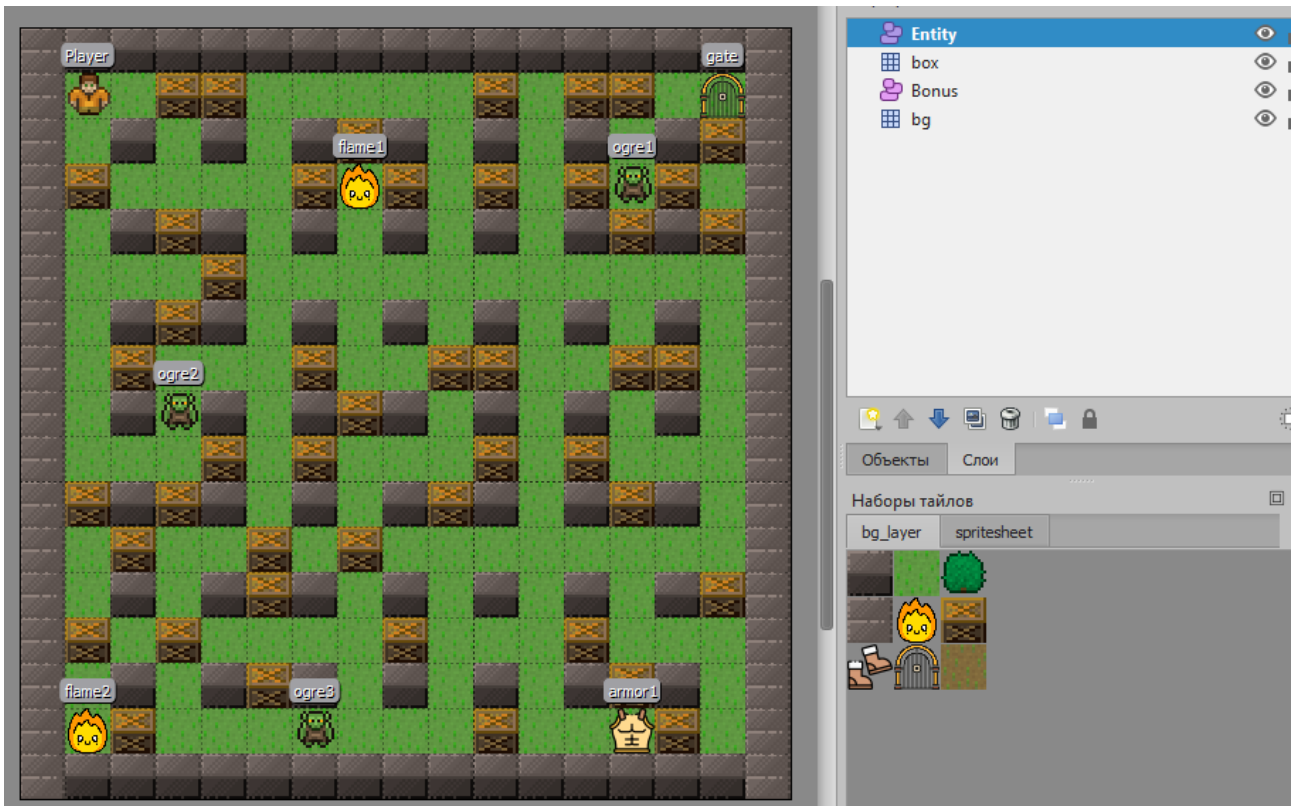

Рис. 2 — Карта первого уровня игры

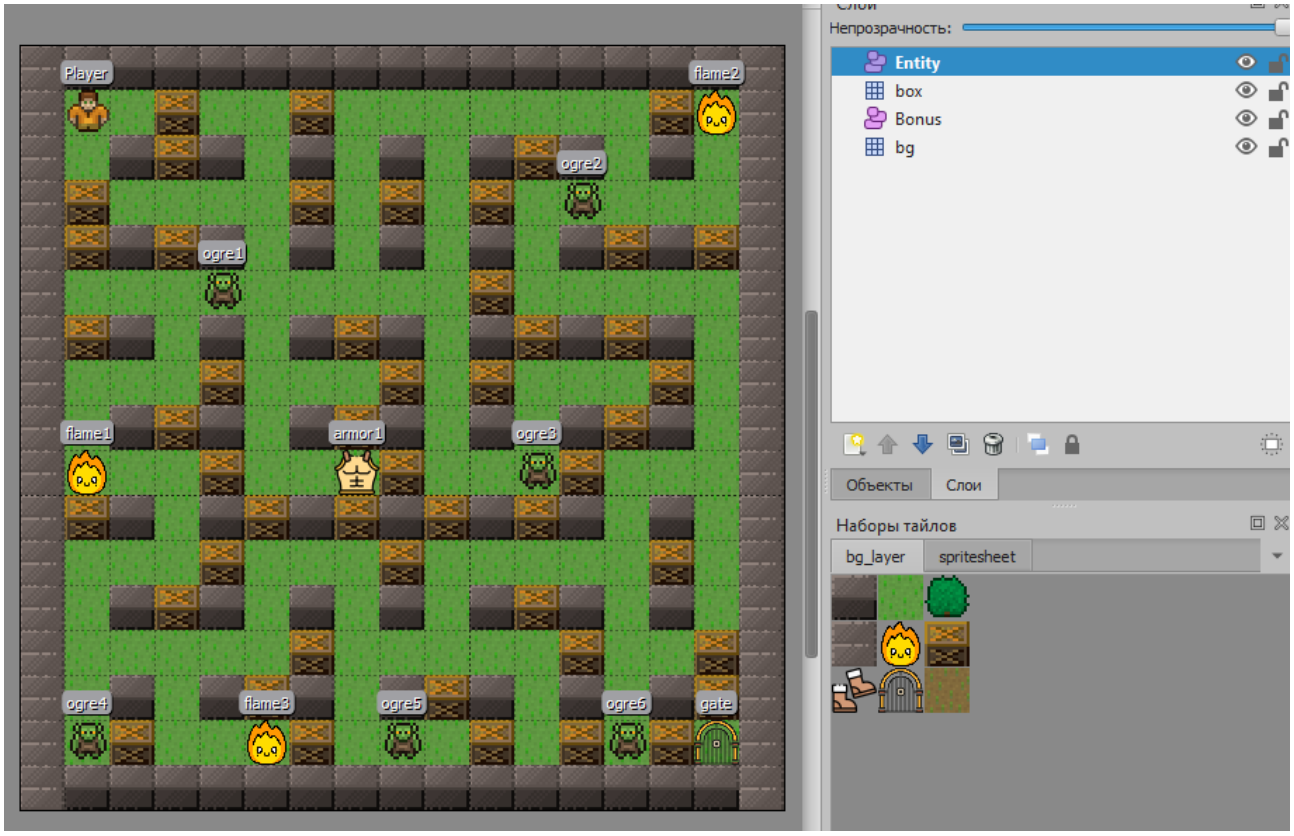

Рис. 3 — Карта второго уровня игры

2. ОТОБРАЖЕНИЕ ОБЪЕКТОВ

2.1. Создание объектов игры

Был написан класс Entity, который имеет поля для хранения координат и размера объекта. Далее, от этого класса наследуются классы Player, Enemy, Bomb, Bonus.

```
class Entity {...}

class Player extends Entity {...}  // игрок

class Enemy extends Entity {...}  // враги

class Bonus extends Entity {...}  // Бонусы

class Bomb extends Entity {...}  // бомбы
```

Рис. 4. Объекты игры

2.2. Загрузка изображений для объектов

Для корректного изображения игровых объектов на поле был разработан специальный менеджер spriteManager. Он обрабатывает значения, хранящиеся в json-файле, который представляет собой атлас спрайтов. В самом начале менеджер загружает атлас и соответствующую картинку через AJAX-технологию. Когда необходимо отобразить объект на поле происходит обращение к менеджеру, и тот находит в атласе нужную запись и рисует необходимую часть, описанную согласно всей хранящейся информации.

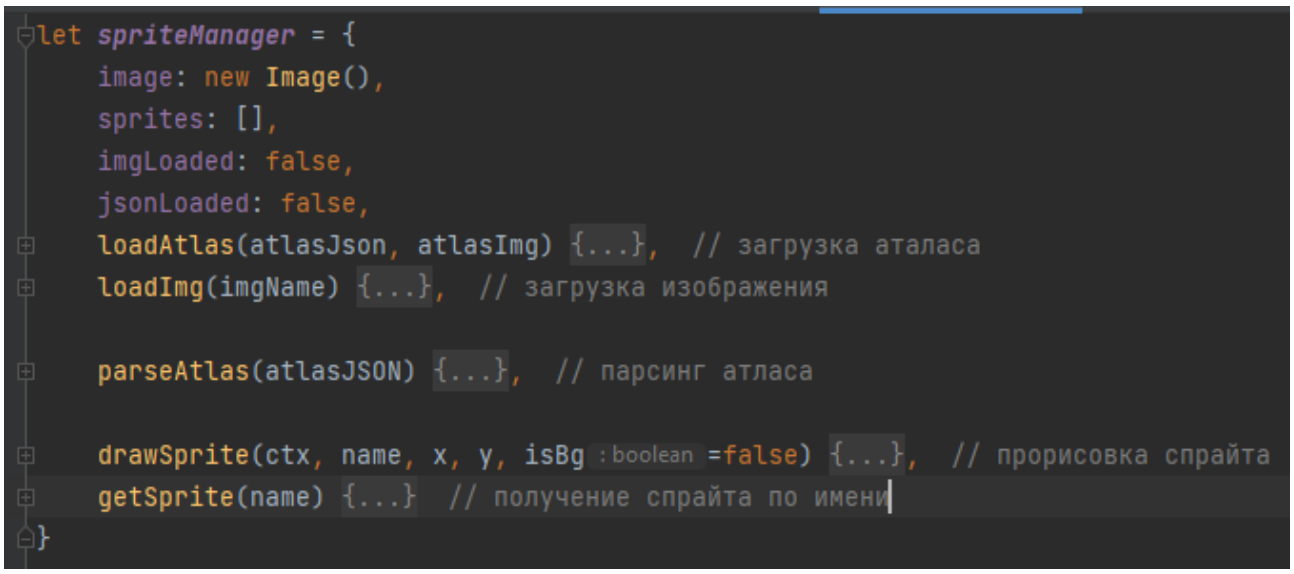

Рис. 5. Поля и методы объекта spriteManager.

3. ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ

3.1. Реализация менеджера событий

Для взаимодействия с пользователем был создан менеджер событий. С его помощью происходит обработка определённых клавиш. Из-за его реализации имеется возможность в любой момент поменять клавиши для удобства пользователя.

```
let eventsManager = {
    bind: [],
    action: [],
    setup() {
        this.bind[87] = 'up'  // W
        this.bind[65] = 'left'  // A
        this.bind[83] = 'down'  // S
        this.bind[68] = 'right'  // D
        this.bind[32] = 'plant'
        document.body.addEventListener( type: "keydown", this.onKeyDown)
        document.body.addEventListener( type: "keyup", this.onKeyUp)
    },
    onKeyDown(event) {
        let action = eventsManager.bind[event.keyCode]
        if (action)
            eventsManager.action[action] = true
    },
    onKeyUp(event) {
        let action = eventsManager.bind[event.keyCode]
        if (action)
            eventsManager.action[action] = false
    }
}
```

Рис. 6. Поля и методы объекты eventsManager.

4. РЕАЛИЗАЦИЯ ЛОГИКИ ПОВЕДЕНИЯ ОБЪЕКТОВ

4.1. Менеджер физики объектов

При создании объектов у каждого из них могут быть свои особенности отображения, изменения состояния, влияния на другие объекты, но все они имеют общую работу перемещения. В связи с этим общая логика по обновлению была вынесена в специальный объект — менеджер физики объектов (physicManager).

```
let physicsManager = {
    tileIntersection(x, y, dirX, dirY) {...},


    update(obj) {...},
    entityAtXY(obj, x, y) {...}
}
```

Рис. 7. Методы объекта physicsManager.

5. УПРАВЛЕНИЕ ЗВУКАМИ

5.1. Менеджер звука

Для управления звуком в игре был создан менеджер звука (soundManager), который в самом начале загружает все звуки, а после проигрывает их при клике пользователем на кнопку проигрывания звука.

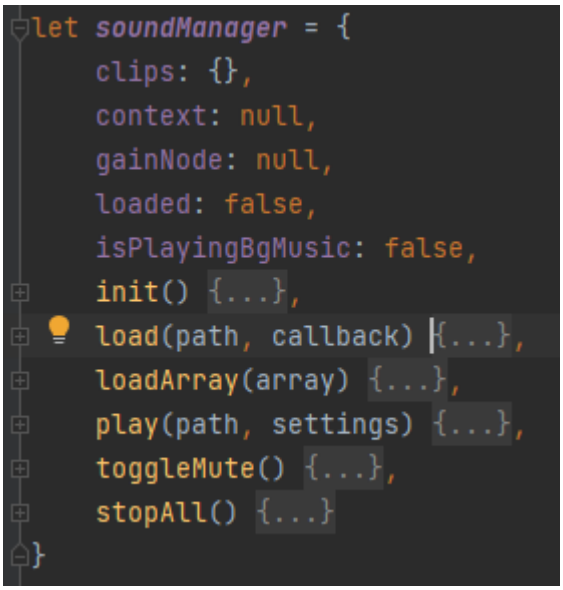

Рис. 8. Поля и методы объекта soundManager.

6. УПРАВЛЯЮЩИЙ ЭЛЕМЕНТ ИГРЫ

6.1. Менеджер игры

На данном этапе необходимо объединить все элементы в единое целое и запустить исполнение. В качестве инструмента объединения используется менеджер игры. Менеджер игры обеспечивает инициализацию, загрузку всех необходимых ресурсов, хранение и управление всеми объектами игры, регулярное обновление и отображение пользователю игрового мира.

Рис. 9. Поля и методы объекта gameManager.

В приложении А представлены скриншоты работы программы.

ЗАКЛЮЧЕНИЕ

Результатом курсовой работы стал полноценное интернет-приложение – игра на языке JavaScript (стандарт ES6) состоящая из двух уровней.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Беляев С.А. “Разработка игр на языке JavaScript”. Издательство «Лань», 2016.
2. Редактор карты // Tiled URL: https://www.mapeditor.org/
3. Создание атласа спрайтов // Leshy SpriteSheet Tool URL: https://www.leshylabs.com/apps/sstool/

ПРИЛОЖЕНИЕ А
СКРИНШОТЫ РАБОТЫ ПРОГРАММЫ

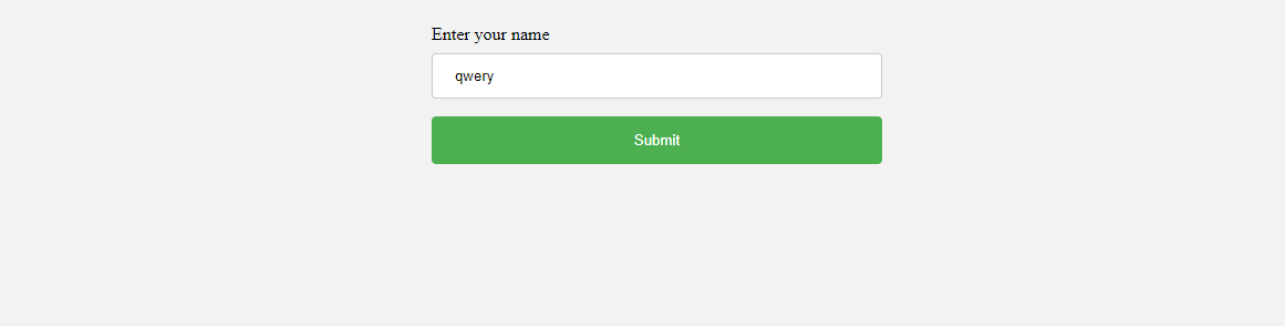

Рис. 10. Стартовое окно игры

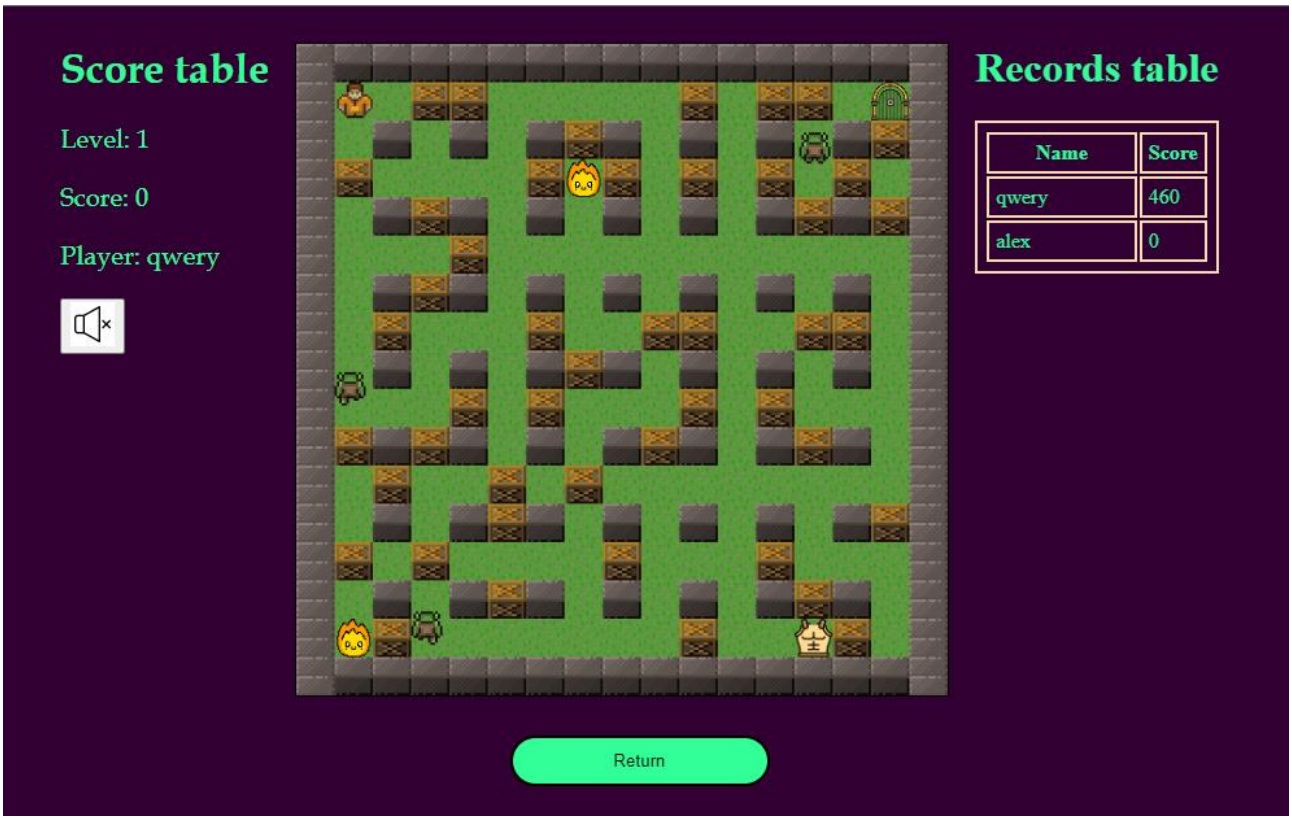

Рис. 11. Первый уровень

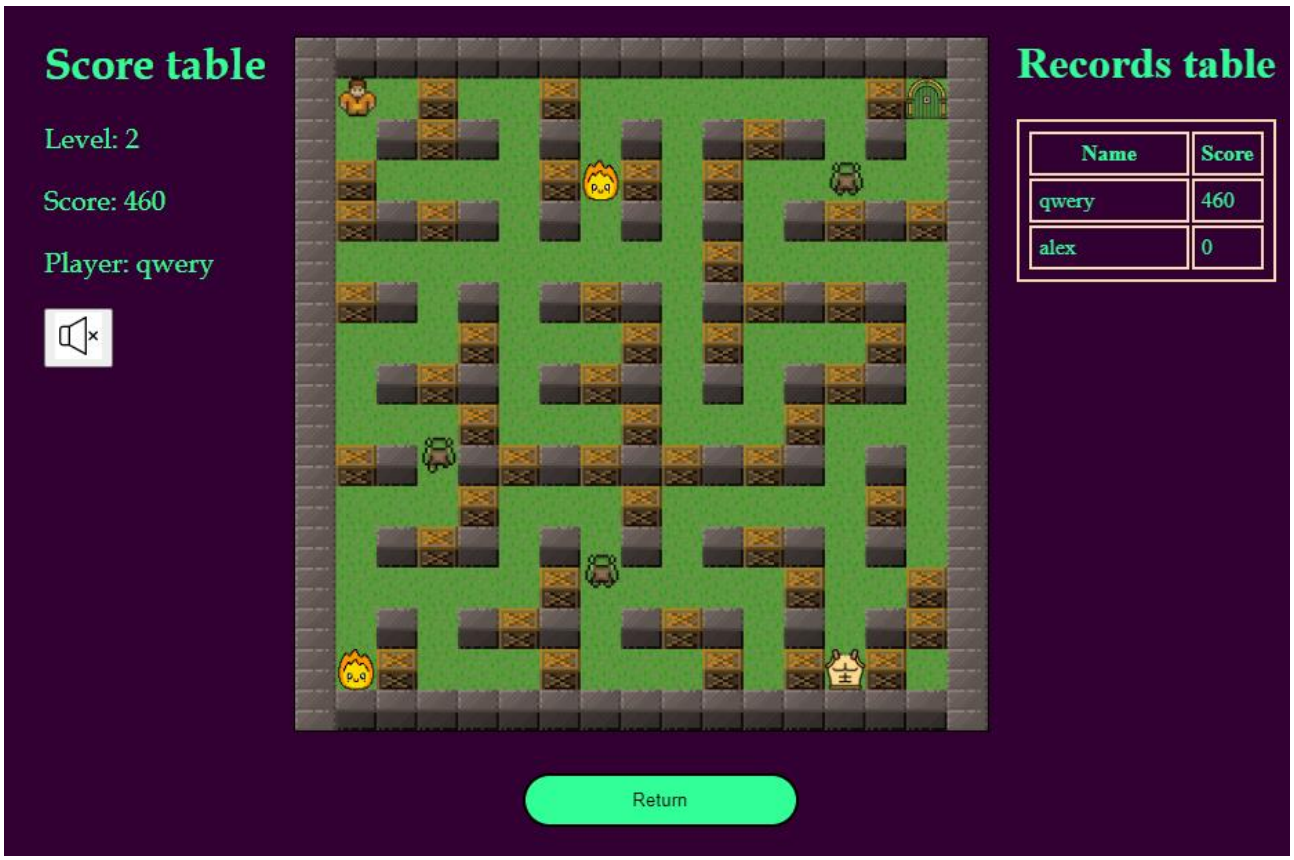

Рис. 12 Второй уровень

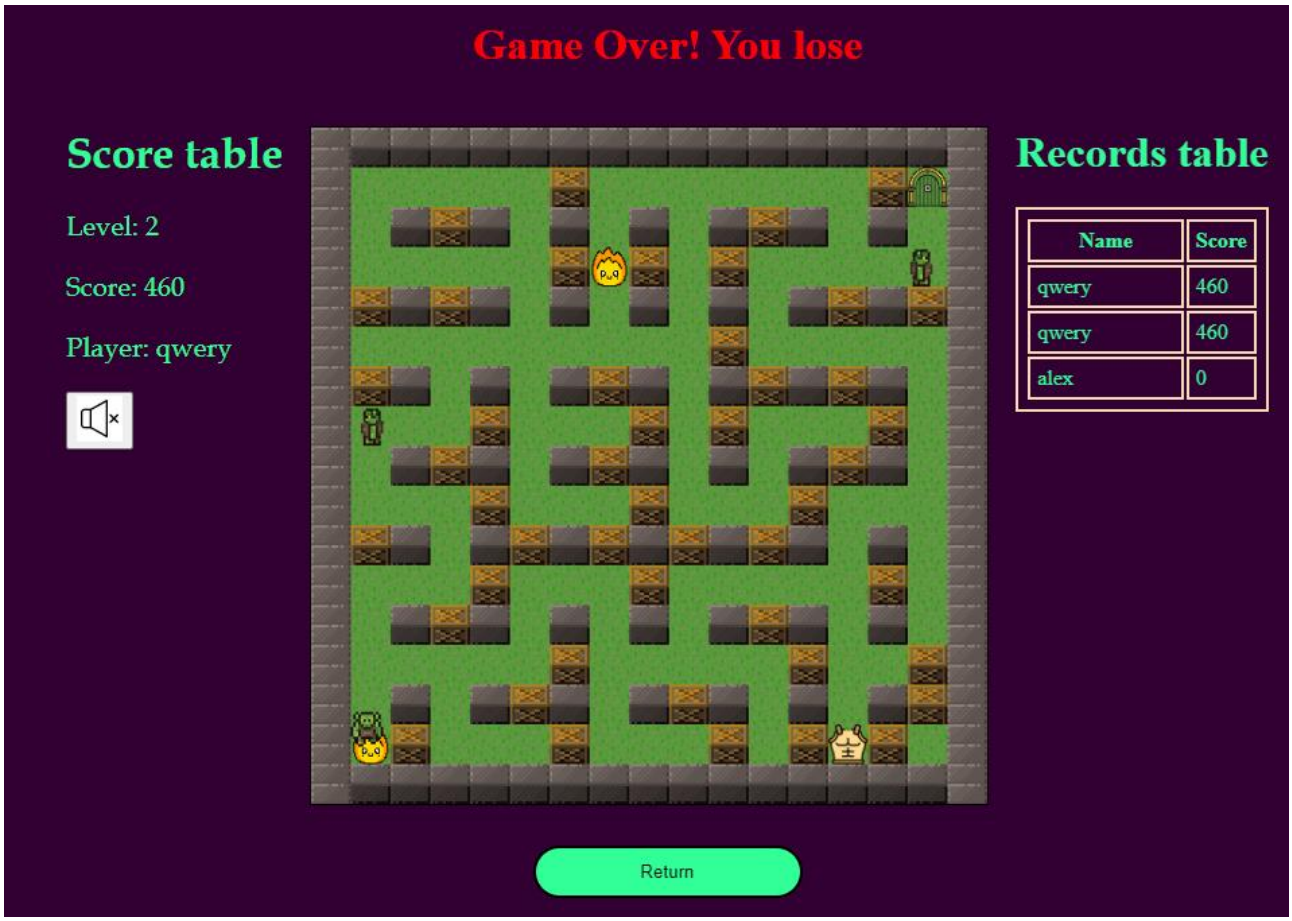

Рис. 13. Проигрыш игрока.

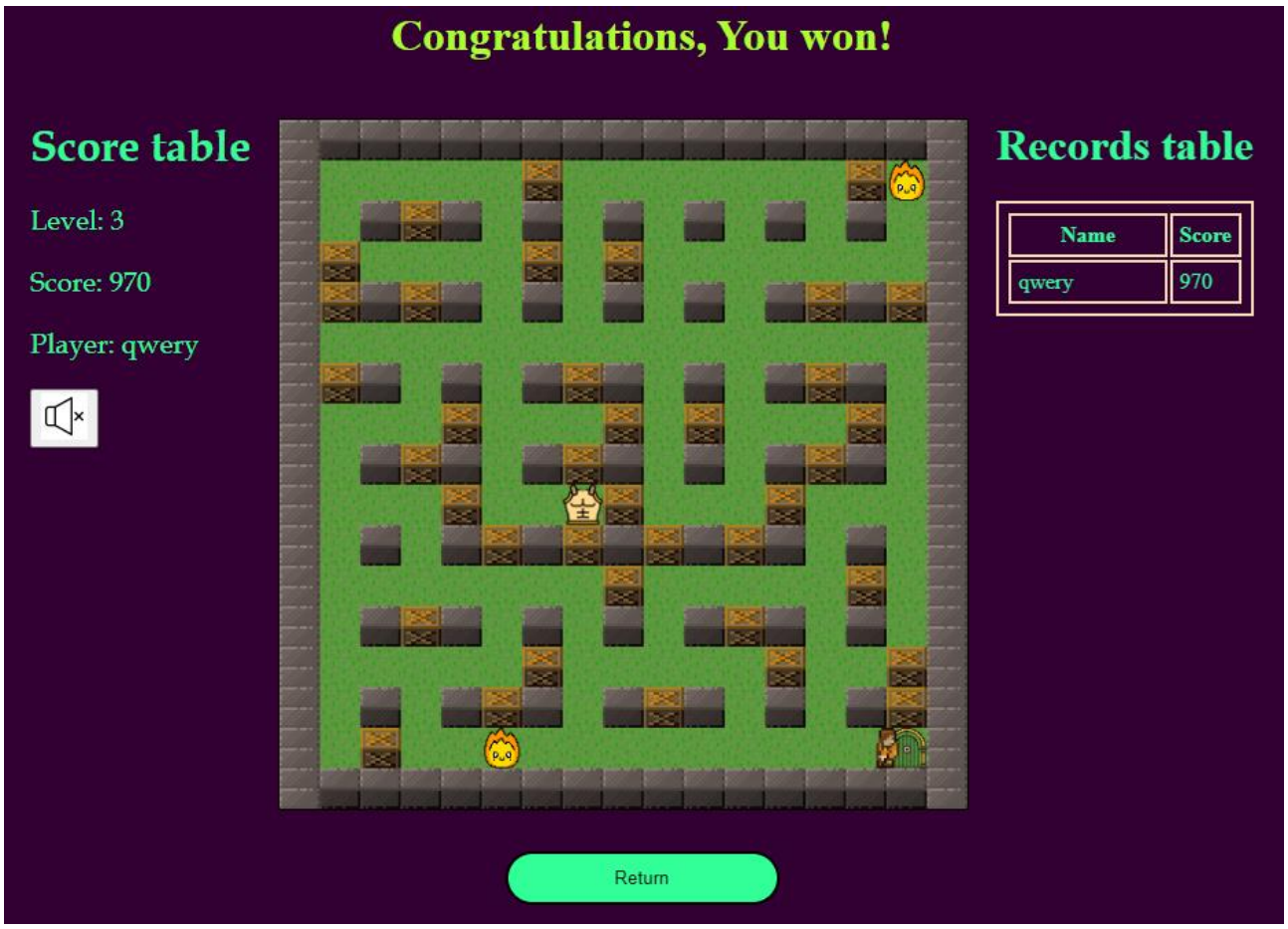

Рис. 14. Победа игрока

Рис. 7 — Победа игрока