**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. 0304 |  | Асташёнок М.С. |
| --- | --- | --- |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2022

**ЗАДАНИЕ**

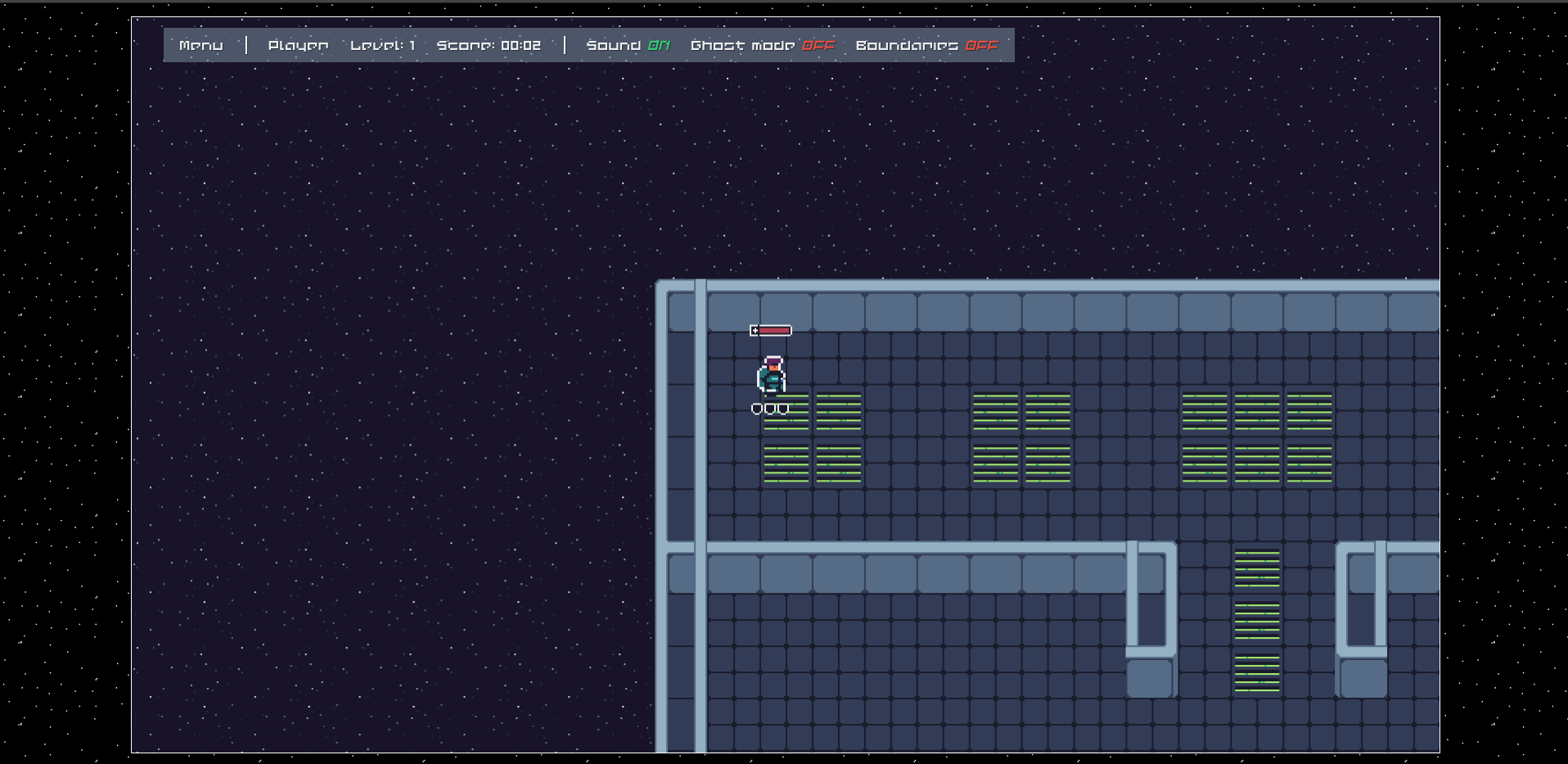
**НА КУРСОВУЮ РАБОТУ**

| Студент Асташёнок М.С. | | |
| --- | --- | --- |
| Группа 0304 | | |
| Тема работы: Разработка игры на языке JavaScript | | |
| Исходные данные:  Необходимо выполнить курсовую работу в соответствии с учебным пособием Беляев С.А. «Разработка игр на языке JavaScript». Курсовая работа выполняется на «чистом» JavaScript (ES6).   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) в соответствии с УП | | |
| Содержание пояснительной записки:  «Содержание», «Введение», «Карта игры», «Менеджеры», «Заключение», «Список использованных источников» | | |
| Предполагаемый объем пояснительной записки:  Не менее 10 страниц. | | |
| Дата выдачи задания: 01.09.2022 | | |
| Дата сдачи работы: 02.12.2022 | | |
| Дата защиты работы: 02.12.2022 | | |
| Студент |  | Асташёнок М.С. |
| Преподаватель |  | Беляев С.А. |

**АННОТАЦИЯ**

Была поставлена задача написать браузерную 2D игру на языке JavaScript. В игре должны быть реализованы такие менеджеры, как менеджер карты, менеджер спрайтов, менеджер событий, менеджер звука, менеджер физики, менеджер “интеллекта” противников, а также менеджер игры. Уровни игры должны быть построены с использованием редактора “тайлов” Tiled. В игре должно быть реализовано не менее двух уровней.

В ходе выполнения курсовой работы была разработана игра в жанре 2D-шутера с видом сверху. Скриншот игры см. на рис. 1.

Рисунок 1. Скриншот игры

**СОДЕРЖАНИЕ**

|  | Содержание | 5 |
| --- | --- | --- |
| 1. | Введение | 6 |
| 1.1. | Цель курсовой работы и исходные условия | 6 |
| 1.2. | Задачи | 6 |
| 2. | Карта игры | 7 |
| 2.1. | Ассеты | 7 |
| 2.2. | Уровни игры | 7 |
| 3. | Менеджеры | 10 |
| 3.1. | Менеджер карты | 10 |
| 3.2. | Менеджер спрайтов | 10 |
| 3.3. | Менеджер событий | 10 |
| 3.4. | Менеджер звука | 11 |
| 3.5. | Менеджер физики | 11 |
| 3.6. | Менеджер «интеллекта» противников | 12 |
| 3.7. | Менеджер игры | 12 |
|  | Заключение | 13 |
|  | Список использованных источников | 14 |

**1. ВВЕДЕНИЕ**

**1.1. Цель курсовой работы и исходные условия.**

Необходимо выполнить курсовую работу в соответствии с учебным пособием Беляев С.А. «Разработка игр на языке JavaScript». Учебное пособие можно взять в библиотеке.

Все делается 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) в соответствии с УП

Отчёт о выполнении курсовой работы включает:

* постановку задачи;
* описание решения (программы);
* скриншот программы.

**1.2. Задачи**

Для достижения поставленной цели требуется решить следующие задачи:

* Разработка ассетов и карты игры
* Разработка всех менеджеров игры
* Тестирование приложения

**2. КАРТА ИГРЫ**

**2.1. Ассеты**

Для игры были разработаны ассеты, содержащие следующие текстуры и анимации:

* Анимации игрока, врагов и «босса»: покой, бег, стрельба, смерть
* Тектуры окружения: стены, пол, фон, клетка-телепорт
* Тектуры пуль
* Тектуры для коллизий.
* Тектуры для индикаторов: полосы здоровья, полоса перезарядки, индикатор количества щитов у игрока.

Объекты игрока, противника и босса игры содержат различные спрайты для анимации ходьбы, стрельбы, а также смерти персонажа и нахождения персонажа в состоянии покоя. Все спрайты и текстуры игры представлены в папке с проектом по ссылке public/assets.

**2.2. Уровни игры**

Уровни игры были спроектированы с помощью утилиты Tiled. Было построено 2 игровых уровня, на которых расположен игрок и несколько противников.

Также, на первом уровне расположен телепорт для перехода на следующий уровень, а на втором уровне в конце карты расположена арена с главным босом игры.

Оба уровня имеют размерность 60\*60 тайлов, каждый тайл размером в 64 пикселя. Карты первого и второго уровня представлены соответственно на рисунках 2 и 3:

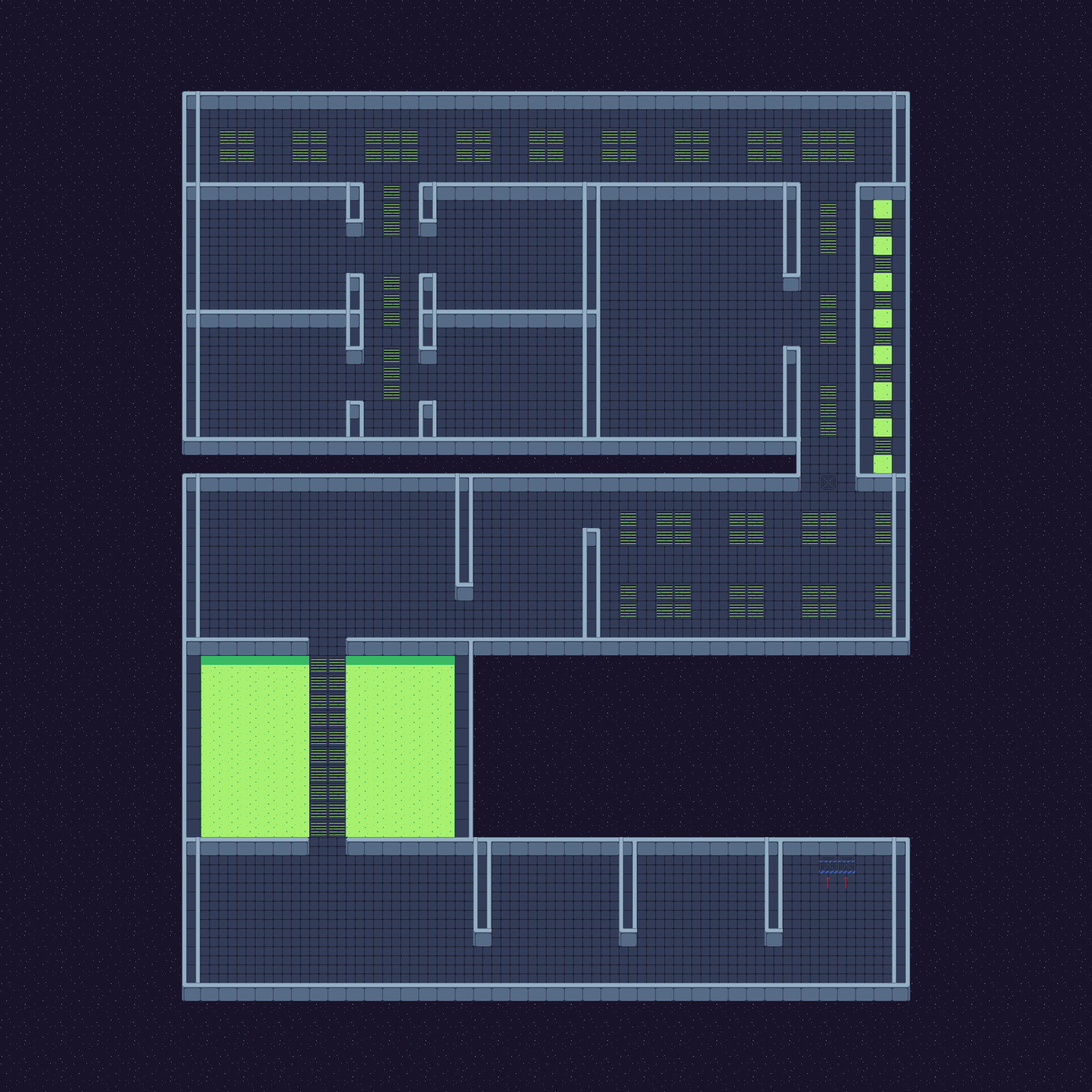


Рисунок 2. Карта первого уровня игры

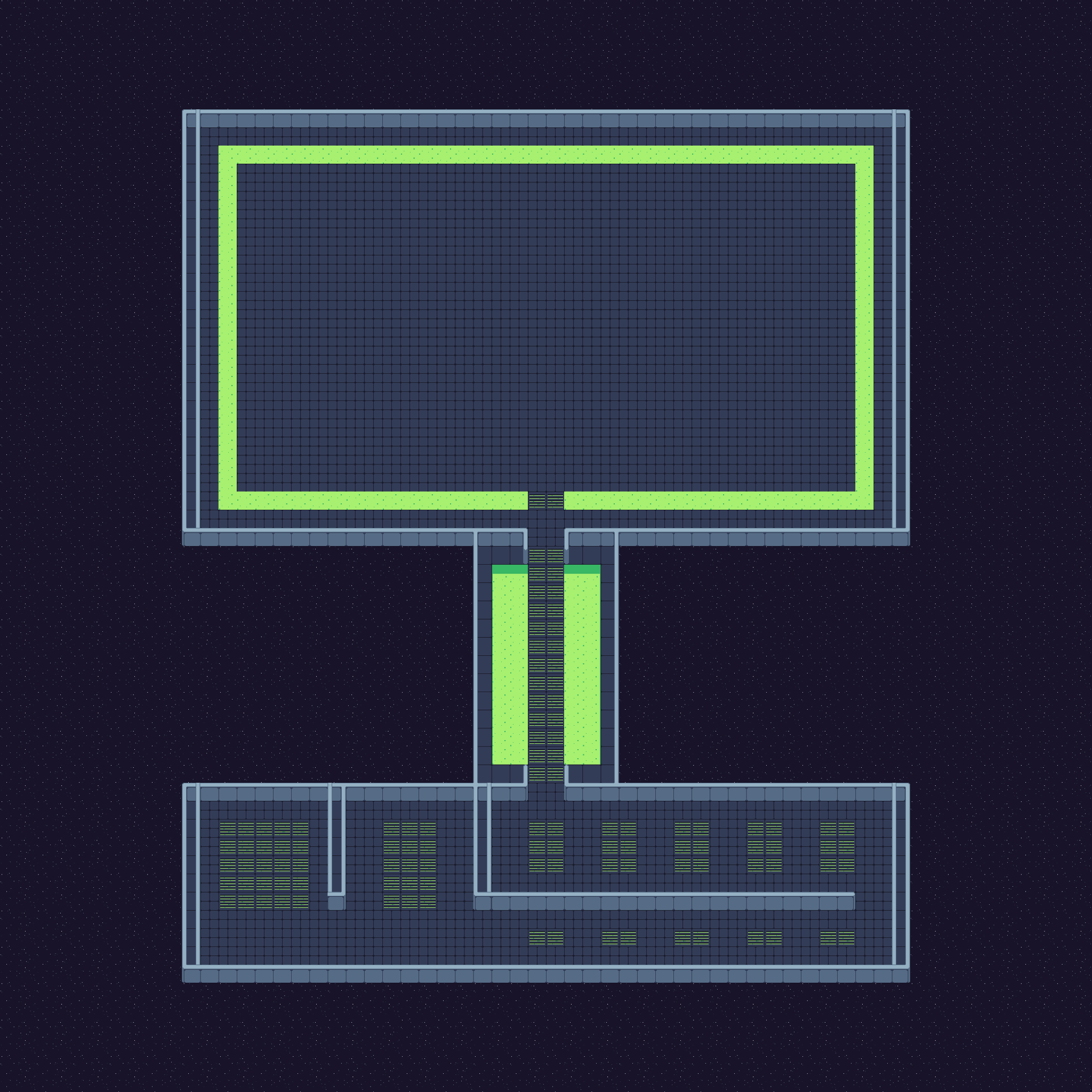


Рисунок 3. Карта второго уровня игры

**3. МЕНЕДЖЕРЫ**

Менеджеры отвечают за различные элементы игровой логики, такие как отрисовка карты и объектов, обработка действий пользователя и т. д. Каждый менеджер обладает собственной определенной задачей, однако все менеджеры связаны друг с другом и с основным менеджером - менеджером игры.

**3.1. Менеджер карты**

Для совершения передвижения по карте используется менеджер MovementEngine. Как можно заметить из скриншотов игры, персонаж не передвигается - он всегда находится на закрепленной точке в <canvas>, примерно в центре (таким образом удалось добиться оцентрирования камеры и игрока). Передвижение игрока - это передвижение всей карты, поэтому данный менеджер отвечает в первую очередь за передвижение карты под игроком: придание ускорения, сопротивления движению, столкновения. Однако т.к. игрок находится не в углу карты, а по центру канваса, все операции взаимодействия с картой происходят с отступом, который описывает центральное положение модельки игрока в канвасе.

**3.2. Менеджер спрайтов**

Для рендеринга карты и спрайтов используется класс Render, который производит отрисовку абсолютно всех элементов игры: карты, игрока, врагов, индикаторов. Также есть возможно в данном менеджере включить отображение хитбоксов всех элементов игры - таким образом можно увидеть элементы, которые могут сталкиваться с препятствиями, а также сами препятствия.

**3.3. Менеджер событий**

Менеджер событийотвечает за обработку действий пользователя. В игре этот менеджер представлен классом Controller. Он обрабатывает все клавиши передвижения (WASD для всех сторон света), а также ЛКМ для стрельбы. Также в классе существует надстройка для возможности одновременно нажимать несколько клавиш передвижения, что позволяет персонажу передвигаться одновременно по обеим осям.

**3.4. Менеджер звука**

В игре реализованы различные звуковые эффекты, такие как:

* Передвижение игрока и врагов
* Звук бластера (оружия игрока и врагов)
* Звук получения урона игроком (он же является звуком смерти для врагов)
* Звук смерти игрока
* Музыка для начала схватки с боссом игры.

Менеджером звука в игре выступает AudioManager. Данный класс позволяет воспроизводить перечисленные выше звуки при возникновении той или иной игровой ситуации. Также интерфейс игры наделен возможностью отключать звук при необходимости - для этого достаточно нажать на соответствующую кнопку в верхней панели. Звуки игры останавливаются при окончании игры, проигрыше или выходе в главное меню.

**3.5. Менеджер физики**

Также был реализован менеджер столкновения одних объектов с другими под названием BoundaryEngine. Главной задачей такого менеджера выступала обработка следующих игровых ситуаций:

* Столкновение игрока/врага со стеной.
* Столкновение пули со стеной.
* Столкновение пули с игроком/врагом.

При столкновении игрока/врага со стеной, первый будет сбрасывать свою скорость передвижения и пытаться продвинуться дальше, пока окончательно не остановится.

Пули же исчезают при столкновении со стеной. При столкновении с игроком или врагом, в зависимости от их источника, пуля нанесет тому персонажу заданное число урона и исчезнет.

**3.6. Менеджер «интеллекта» противников**

В игре также представлен менеджер для “интеллекта” противников - EnemyMovementManager. Данный менеджер представляет собой поведение врагов в соответствии с той или иной игровой ситуацией. Они могут быть следующими:

* **покой** - персонаж игрока находится на достаточно большом расстоянии от врага, из-за чего последний все еще не засек игрока. В этом состоянии враг либо стоит на месте, либо хаотично передвигается, однако на небольшое расстояние.
* **преследование** - враг засек игрока, однако он стоит слишком далеко, чтобы начать атаковать игрока. Алгоритмом преследования игрока врагами выступает жадный алгоритм: для каждого из 8 направлений вокруг врага в пределах 1 тайла (4 для сторон света и 4 для диагоналей) рассчитывается расстояние до игрока, а затем выбирается лучшее положение, самое близкое к игроку. По итогу, враг совершает передвижение в указанную сторону. Однако данный подход не всегда приводит к фактическому столкновению игрока с врагом, т.к. алгоритм не позволяет врагу обходить препятствия.
* **атака** - игрок достаточно близко к врагу, чтобы второму начать атаку. Враги останавливаются и начинают стрелять в игрока. В случае, если игрок отошел, враги могут снова начать преследовать его, либо вовсе отстанут.

**3.7. Менеджер игры**

Менеджер игры, представленный классом Game, объединяет в себе все менеджеры, что позволяет управлять всем игровым процессом прямо из данного класса. В Game происходит инициализация всех менеджеров и необходимых объектов, а также запуск всех необходимых для игры процессов.

**ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы была разработана браузерная игра на языке JavaScript. Цель игры состоит в том, чтобы, пробиваясь через противников, добраться до конца уровня.

Для реализации логики игры были разработана различные менеджеры, отвечающие за различные области, такие как карта, объекты, звуки и т. д. Игра представляет собой шутер с «интеллектуальными» противниками.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Беляев С. А. Разработка игр на языке JavaScript: Лань, 2020. — 128 с.

2. Учебный ресурс для веб-разработчиков // MDN Web Docs. URL: https://developer.mozilla.org/ru/