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

КУРСОВАЯ РАБОТА

по дисциплине WEB-технологии

Тема: Разработка игр на языке JavaScript

Студент гр. 1384 _____ Кондратенко К.Е.

Преподаватель _____ Беляев С.А.

Санкт-Петербург

2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Необходимо реализовать игру в браузере, на «чистом» JavaScript (ES6). В группах по 1 человеку.

1. ПОСТАНОВКА ЗАДАЧИ

Необходимо реализовать игру, которая будет отвечать следующим требованиям:

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 в соответствии с УП

2. ОПИСАНИЕ РЕШЕНИЯ

2.1. Описание концепции игры

Поскольку необходимо реализовать 2d браузерную игру, было принято решение взять за основу идею преследования главного героем абстрактным врагом. В качестве сеттинга используется следующая ситуация: английский певец Sting решил посетить США (Нью Йорк) как турист, однако его преследует фанат (коренной житель Нью Йорка) – задача игрока как можно дальше убегать от преследователя.

2.2. Описание сущности игрока, противника и бонуса

Для взаимодействия игрока и противника с объектами карты были реализованы классы GG и Enemy. Полями класса являются положение на карте, положение на канвасе, файл визуального отображения и позиция визуальной модели объекта в файле визуального отображения. Для обоих классов реализованы методы передвижения, которые меня x или y позицию, а также метод, обновляющий положение объекта в канвасе. У класса GG имеется поле hp, которое отвечает за текущее количество очков жизни (если значение меньше 0 – игра проиграна).

Класс Heal также имеет положение на карте, на канвасе и в сете тайлов, однако, поскольку бонус статичный, для него нет методов передвижения, но он оснащен полем heal (хранит значение размера бонуса - лечения), healed (булевое

значение, ответственное за факт использования конкретного объекта), а также методом used, который вызывается при использовании бонуса, во время вызова этого метода, поля отображения и heal зануляются.

Описанные классы – менеджеры игрока, противника и бонуса.

2.3. Описание менеджера звука

Для работы со звуком реализован класс SoundManager. Это класс – хранилище, в нем нет реализации специальной логики, он служит для работы со звуком. Его конструктор - создает поля, в которые помещаются файлы звука, полям устанавливается громкость. Можно сказать, что тот класс – реализация паттернов хранитель и singleton.

2.4. Описание менеджера карты

Класс MapManager состоит из конструктора и двух методов – парсера карты и метода создания стен. Конструктор состоит из полей, ответственных за хранения информации о карте, отображение карты в канвасе, а также поля, хранящего состояние карты – матрицу. Парсер получает из карты значения полей, ответственных за размер карты в тайлах, размер тайла, размер карты в пикселях, слои карты. Методы построения стен, строит из карты матрицу (дело в том, что Tiled сохраняет карту в виду одномерного массива, тогда как работать удобнее с двумерным массивом - матрицей) по правилу – если встретилась стена – помечаем клетку значением -1, пустые же клетки помечены нулем.

2.5. Описание менеджера игры

Класс Engine реализует основную игровую логику. Конструктор класса создает объекты героя, карты, врага, бонусов, менеджера звука, создает поля, хранящие счет игрока, информацию о его состоянии, информацию о текущем уровне, время начала игры и обработчик прерываний клавиатуры и канвасы вывода счета и здоровья игрока.

Метода init запускает фоновую музыку, в зависимости от того, какой уровень текущий, парсит карту, рисует карту, игрока и противника в полжении инициализации. Запускает бесконечный цикл, который отвечает за движение противника.

Описание движения игрока – реализованы 4 метода (движения влево, вверх, вправо и вниз), в каждом из которых проверяется, является ли целевая клетка стеной или нет. Если клетка стена, то игроку наносится урон, вызвыается звуковой эффект и обновляется отображение здоровья. Если же следующая клетка не стена, меняется положение игрока, проверяется, не получил ли игрок бонус, а также отрисовывается карта с новым положением игрока.

Описание движения противника – реализованы 4 метода (движения влево, вверх, вправо и вниз), в каждом из которых проверяется, является ли целевая клетка стеной или нет. Если следующая клетка не стена, меняется положение игрока, отрисовывается карта с новым положением игрока и проверяется, не встретился ли противник с игроком, если встретился, то запускается звуковой эффект, игроку наносится урон.

Реализован метод проверки состояния игрока (is_alive), в случае, если здоровье игрок меньше нуля, выключаются все звуковые эффекты, сохраняется результат, выводится сообщение о проигрыше и происходит переход на страницу регистрации.

Метод компановки результат. Поскольку значением имени и результата получаются в разных местах и в разное время, текущее имя пользователя и его результат хранятся в локальном хранилище в виде значение по специальным ключам. Когда получено значение счета, из локального хранилища получаются значения счета и имени и формируются в таблицу рекордом, таблица рекордом помещается в локальное хранилище.

Методы отображения информации об игроке и карты с игроком и противником очищают канвасы и рисуют целевые объекты. Методы отображения информации при достижении 30 очков принудительно сменяет уровень с первого на второй.

Метод построения оптимального пути от противника к игроку называется Lee_alg и реализуется волновой алгоритм Ли.

Метод interrupt_keyboard проверяет нажатие клавиш и если это стрелки управления – вызывает методы перемещения игрока, если это «2», то принудительно включается второй уровень.

2.6. Описание работы программы

Реализовано 3 HTML страницы. Index содержит краткое погружение в контекст игры, поле ввода имени, кнопки Start и Records – к нему подключается файл menu.js, который производит валидацию имени (спец символы ;%:?* считаются непригодными в использовании в качестве ника), если валидация не прошла, высвечивается сообщение об ошибки ввода имени. При нажатии на кнопку Start происходит переход на страницу HTML game, при нажатии на кнопку происходит переход на страницу HTML leader. Страница leader подключается файл table.js, в котором производится экстракция таблицы рекордов и локального хранилища, сортировка по убыванию и вывод записей на экран. Страница game подключает файл game_manager.js и запускается игра.

2.7. Примеры работы программы

Примеры запусков программы в виде фотографий экрана расположены в приложении А.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы был разработан программный код браузерной игры. Были изучены методы написания игры в ООП парадигме на языке JavaScript с использование html и css, методы работы с локальным хранилищем, методы отображение карты в браузере, а также методы построения карт с использованием Tiled.

ПРИЛОЖЕНИЕ А

ПРИМЕРЫ ЗАПУСКА ПРОГРАММЫ

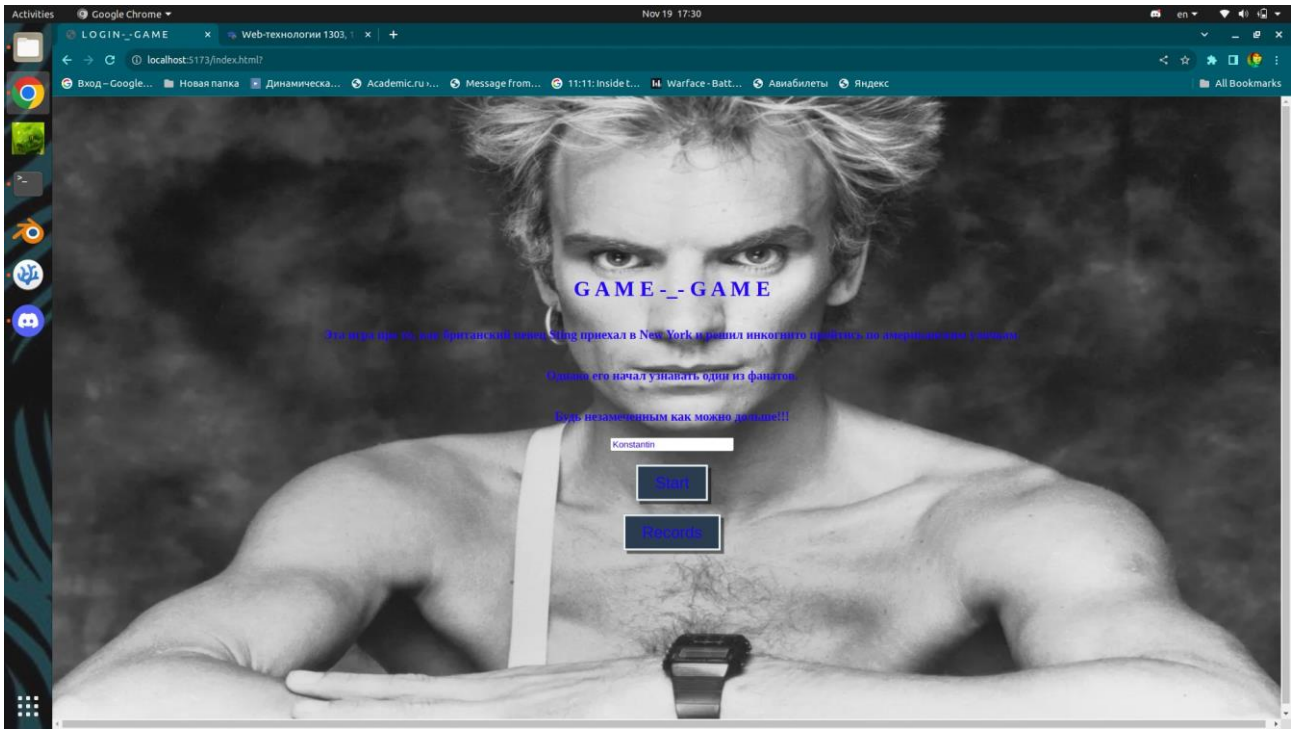

Рисунок 1 – Окно регистрации

Рисунок 2 – Окно рекордов

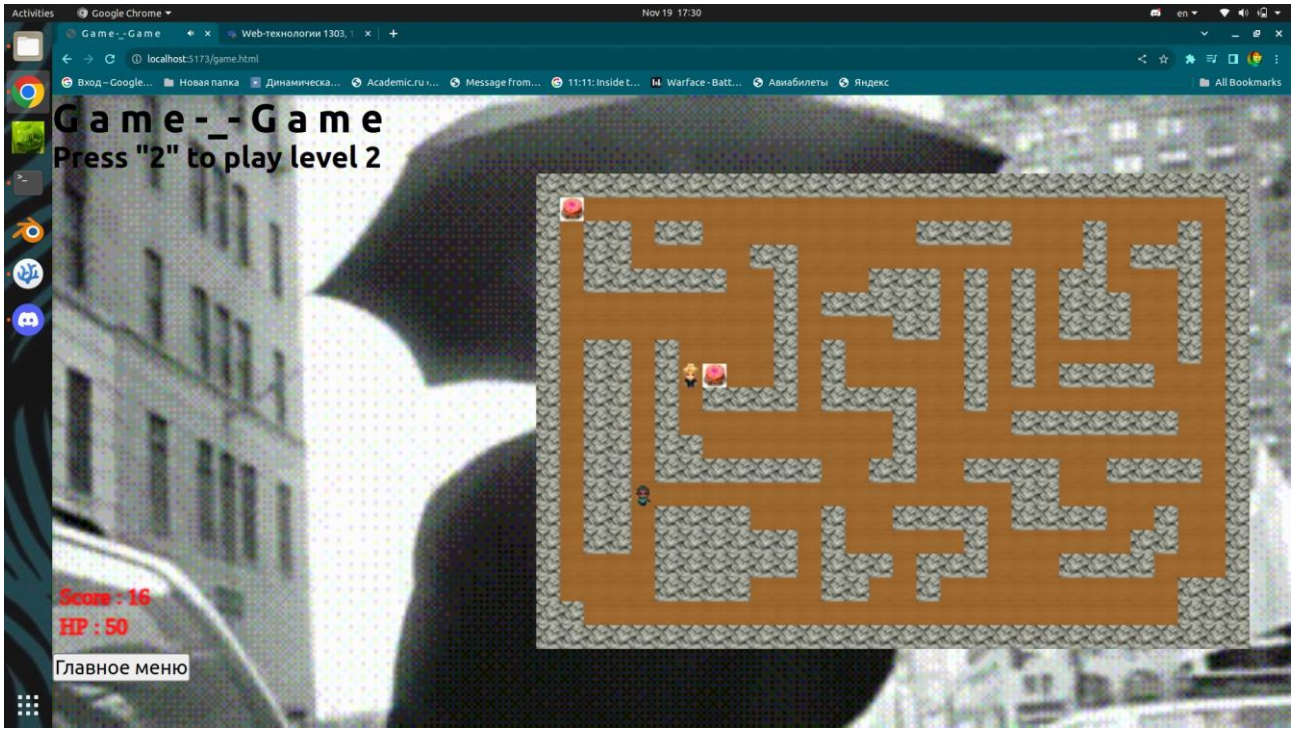

Рисунок 3 – первый уровень (перед бонусом)

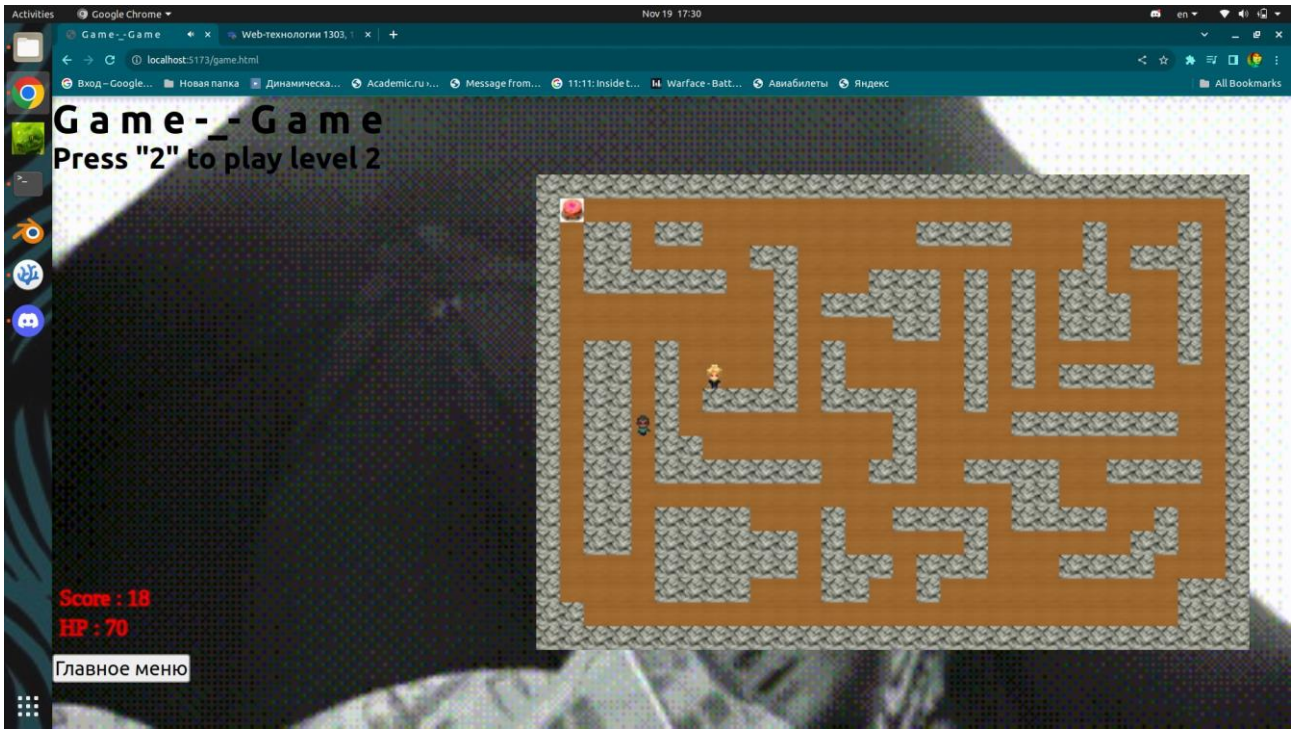

Рисунок 4 – Использование бонуса

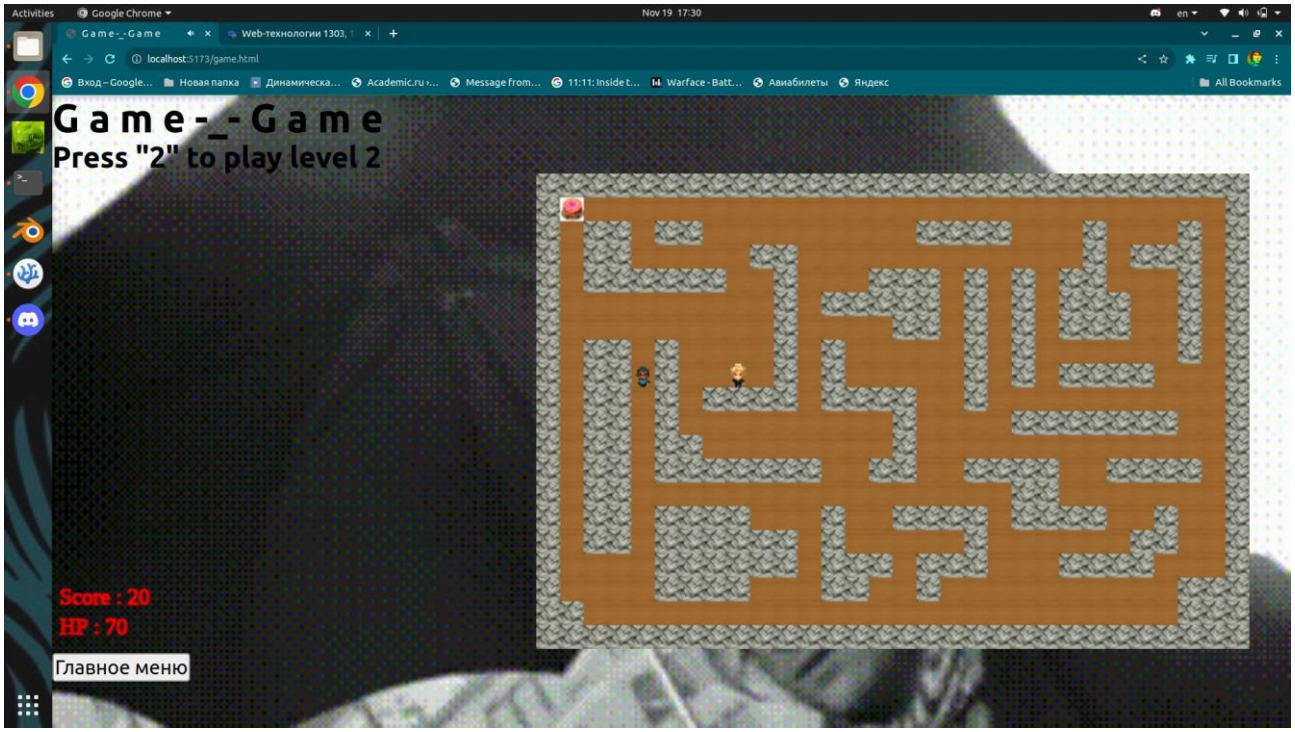

Рисунок 5 – после использования бонуса

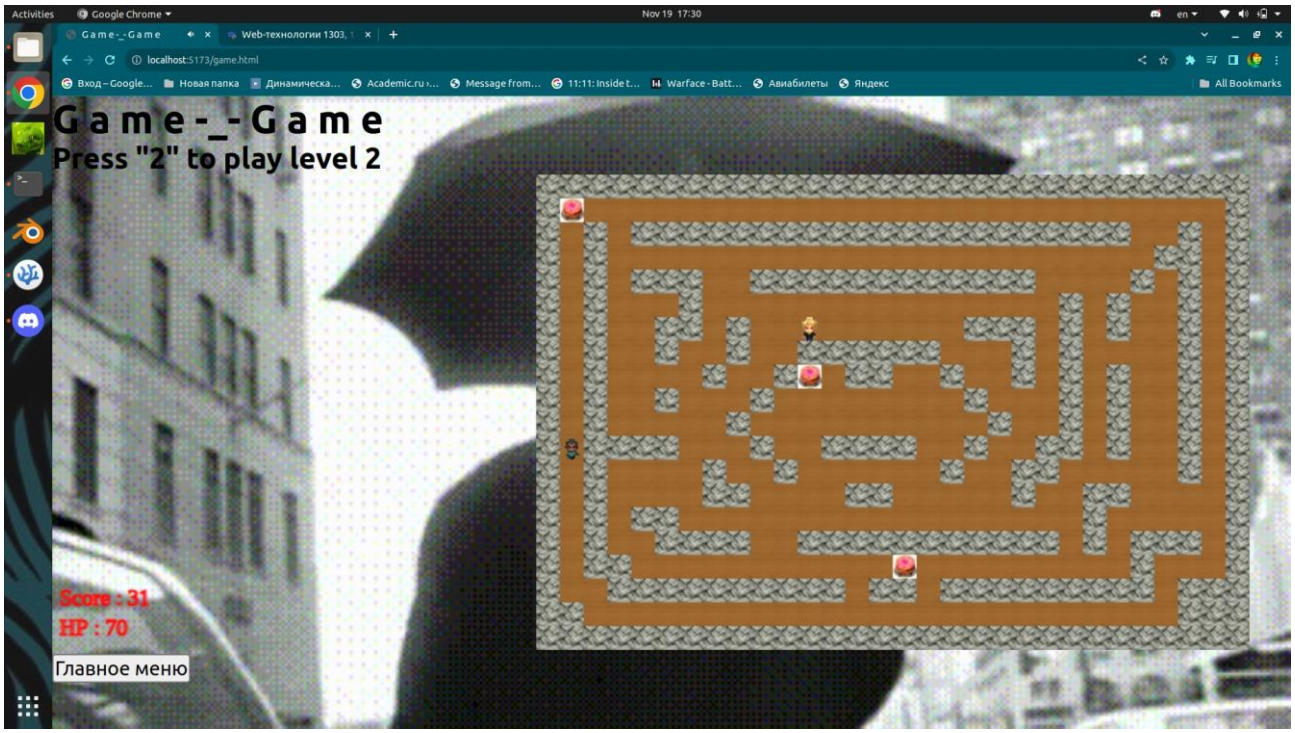

Рисунок 6 – уровень 2

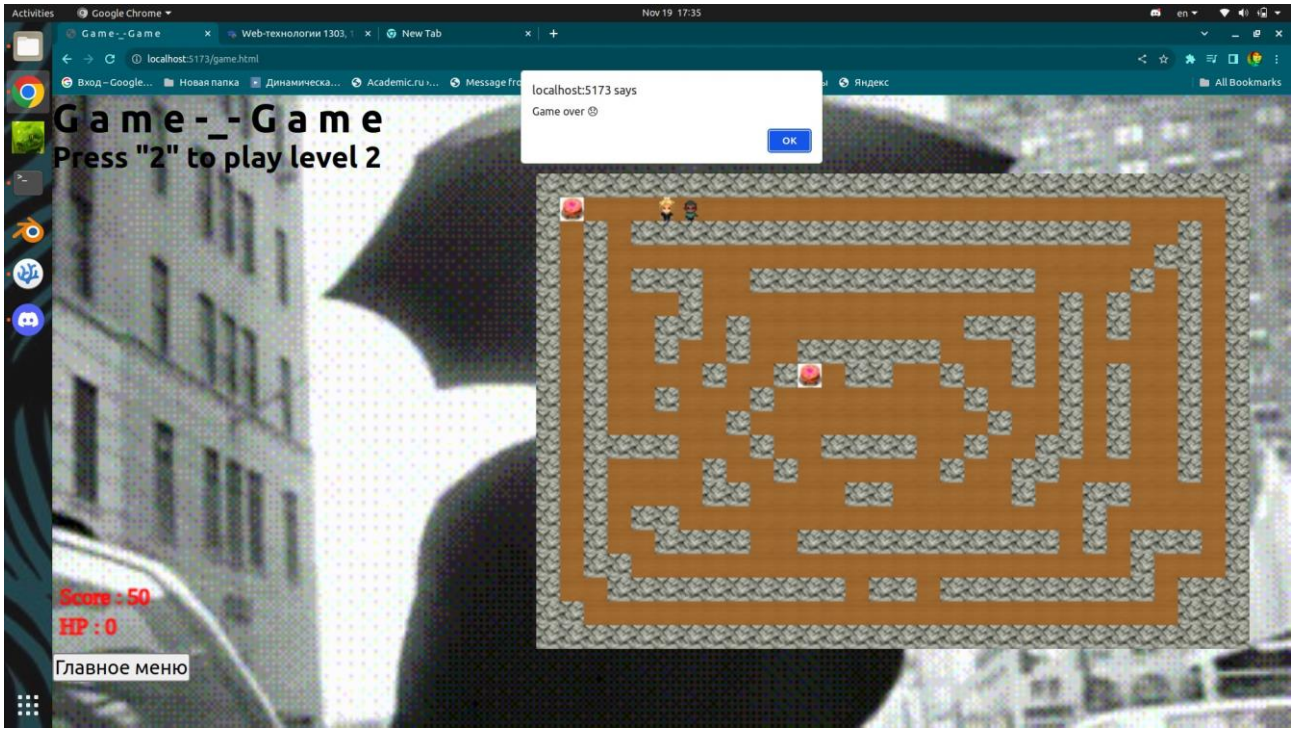

Рисунок 7- проигрыш

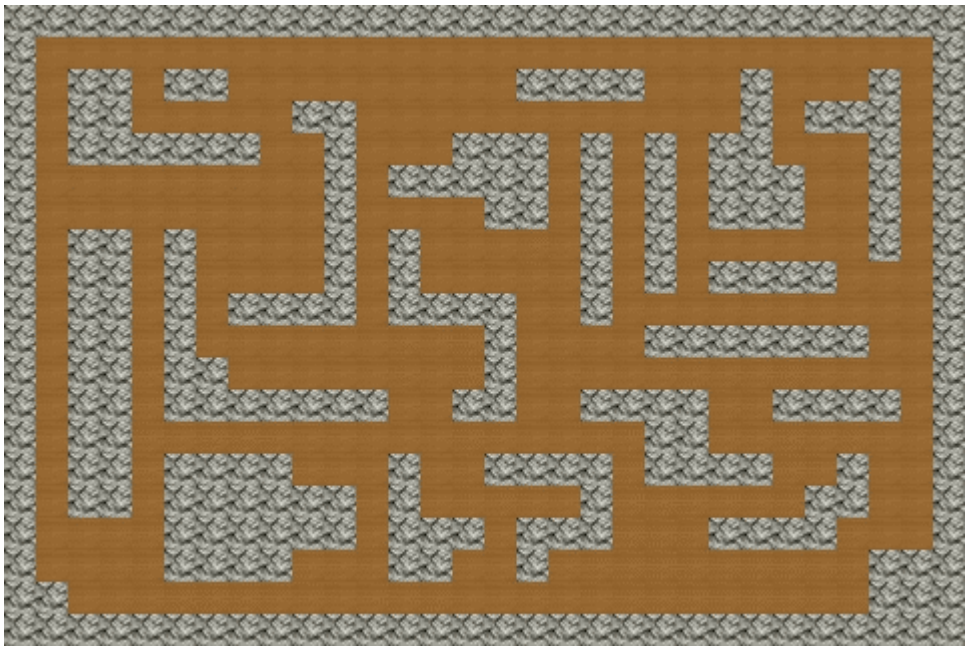

Рисунок 8 – карта первого уровня

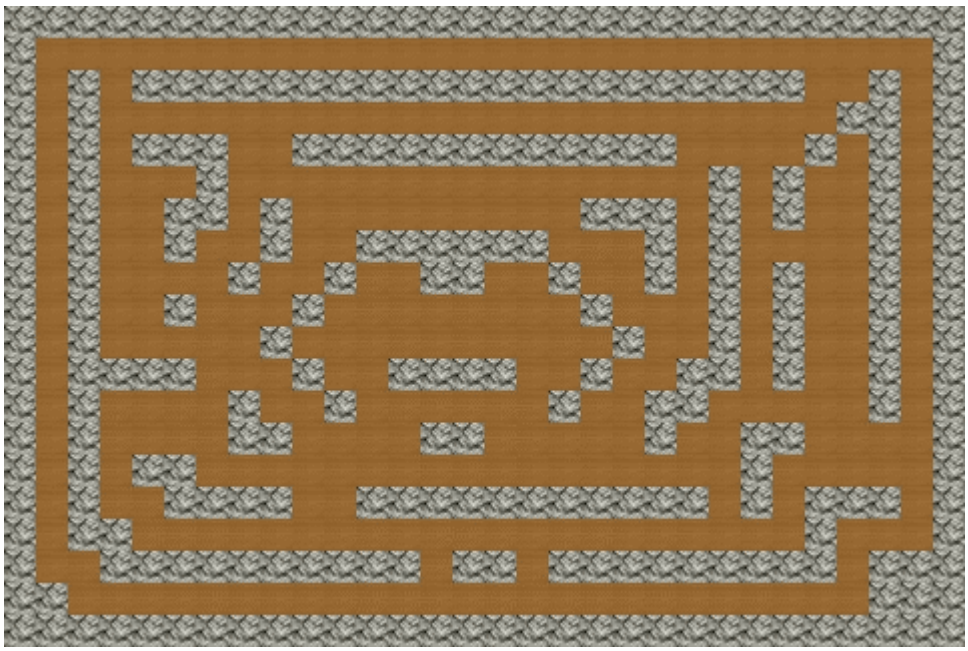

Рисунок 9 – карта второго уровня

Рисунок 10 – набор персонажей

Рисунок 11 - Tileset