

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

Максименко Е.М.

Преподаватель

Беляев С.А.

Санкт-Петербург

2022

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Максименко Е.М.

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», «Серверная часть»,
«Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи работы: 14.10.2022

Дата защиты работы: 14.10.2022

Студент _____ Максименко Е.М.

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.

Реализованное приложение состоит из двух частей: серверной и клиентской. Серверная часть отвечает за хранение результатов игроков, а также загрузку всех ресурсов игры. Клиентская часть содержит всю логику взаимодействия с пользователем, а также логику игры. Скриншот игры см. на рис. 1.

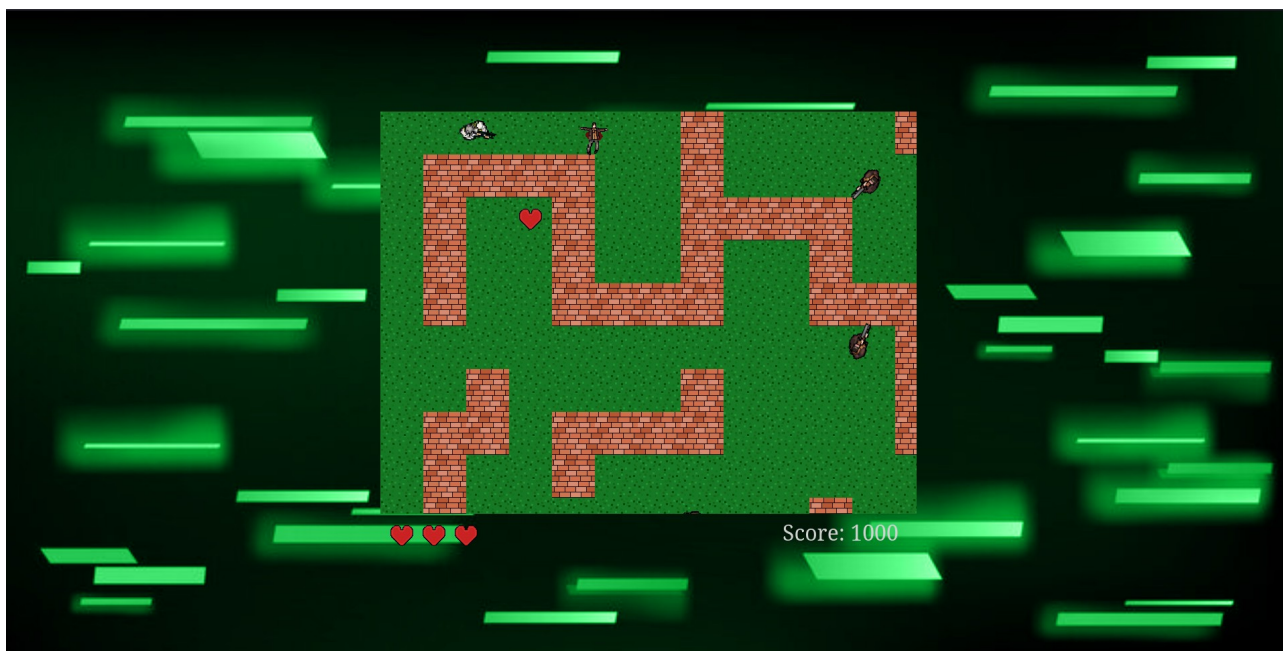


Рисунок 1. Скриншот игры

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	9
3.1. Менеджер карты	9
3.2. Менеджер спрайтов	9
3.3. Менеджер событий	9
3.4. Менеджер звука	9
3.5. Менеджер физики	10
3.6. Менеджер «интеллекта» противников	10
3.7. Менеджер игры	11
4. Серверная часть	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 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) 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
- Разработка серверной части игры
- Тестирование приложения

2. КАРТА ИГРЫ

2.1. Ассеты

Для игры были разработаны ассеты, содержащие текстуры клетки и стены, а также спрайты игрока, противника, пули, бонуса и финишной клетки. Объекты игрока и противника содержат различные спрайты для анимации ходьбы и стрельбы, а также смерти персонажа. Спрайты и текстуры были объединены в атласы, представленные на рис. 2.

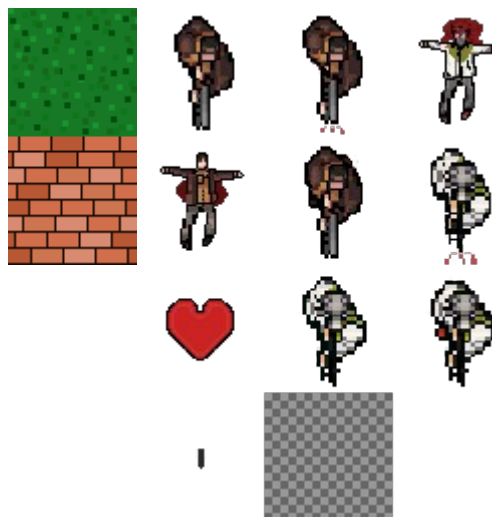


Рисунок 2. Атласы с текстурами и спрайтами

2.2. Уровни игры

Уровни игры были спроектированы с помощью утилиты Tiled. Было построено 2 игровых уровня, на которых расположен игрок, несколько противников, несколько бонусов, восстанавливающих здоровье игрока при подборе, а также финиш, до которого игроку предстоит добраться. Первый уровень состоит из 300 клеток: 20 в ширину и 15 в высоту. Второй уровень состоит из 1200 клеток: 40 в ширину и 30 в высоту. Конфигурацию первого уровня игры см. на рис. 3, второго уровня — на рис. 4.



Рисунок 3. Конфигурация первого уровня игры

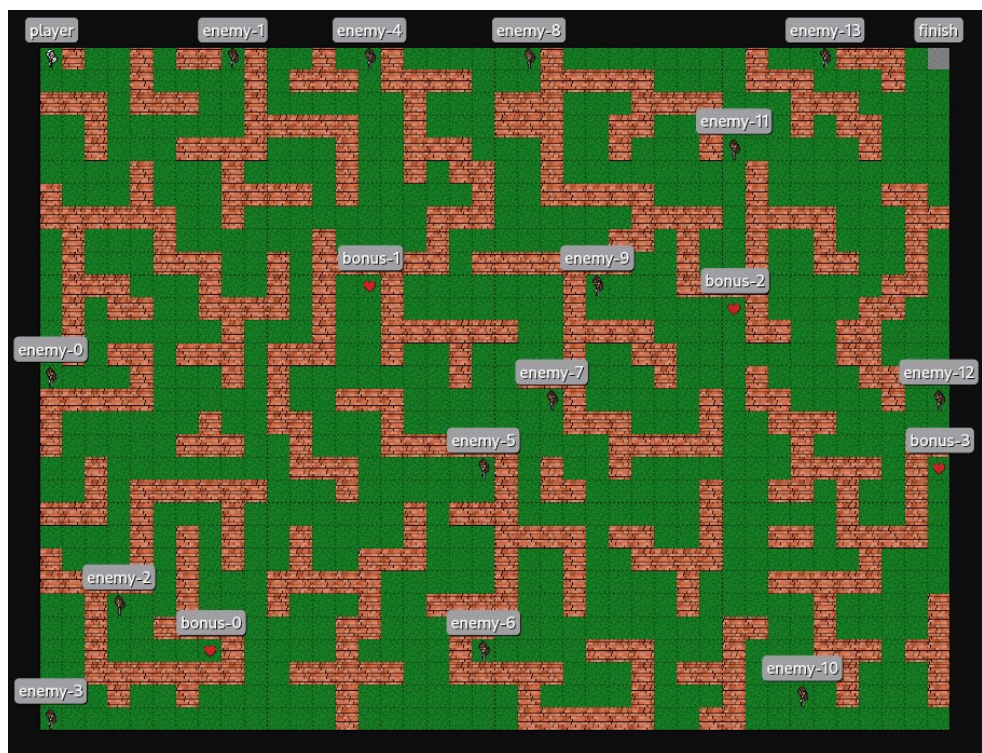


Рисунок 4. Конфигурация второго уровня игры

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.

3.1. Менеджер карты

Для загрузки и отображения карты игры используется менеджер карты. Он отвечает за загрузку текстур клеток, конфигурации уровня, отображение уровня. Также менеджер карты при разборе конфигурации уровня строит граф на основе карты для построения путей врагов с помощью алгоритма A^* (см. п. 3.6).

3.2. Менеджер спрайтов

Для загрузки и отображения на карте объектов используется менеджер спрайтов. Он отвечает за загрузку атласа спрайтов, а также за отрисовку объектов. Также в менеджере спрайтов осуществляется поворот спрайта на определенный угол при отрисовке.

3.3. Менеджер событий

Менеджер событий отвечает за обработку действий пользователя. В менеджере обрабатываются нажатия пользователя на клавиши клавиатуры (WASD для движения и Space для стрельбы), а также движение мыши и нажатие на левую клавишу мыши. С помощью мыши пользователь может управлять поворотом игрока и прицеливанием. События сохраняются в менеджере, после чего на очередной итерации игрового цикла обрабатываются в менеджере игры.

3.4. Менеджер звука

В игре реализованы различные звуковые эффекты, такие как звук ходьбы, выстрелов и смерти противника. За загрузку и воспроизведение этих звуков отвечает менеджер звука. Есть возможность воспроизведения звука с учетом отдаленности источника.

3.5. Менеджер физики

Для реализации коллизии одних объектов с другими либо объектов с внешним окружением (стенами) был реализован менеджер физики. При обновлении физики для определенного объекта проверяется, что у него есть скорость (в противном случае проверять физику нет смысла). У каждого объекта имеется область (прямоугольной), для которой проверяются коллизии, - bounding box. Если bounding box пересекается со стеной, у объекта вызывается метод взаимодействия с данной стеной, при этом движение объекта в данном направлении не возможно. Если препятствия на пути нет, проверяется коллизия с другими объектами. Если два объекта взаимодействуют друг с другом, то у обоих вызывается функция, отвечающая за действие при взаимодействии. Если объекты не могут взаимодействовать (например, противник и бонус), то объект продолжит движение в данном направлении, иначе — нет.

Финиш в игре реализован как отдельный объект, при взаимодействии игрока с которым уровень завершается.

3.6. Менеджер «интеллекта» противников

В игре противники обладают подобием «интеллекта». Противники начинают стрелять в игрока, если он находится в их зоне видимости, они начинают двигаться на место громкого звука, который издал игрок. За реализацию этого поведения отвечает данный менеджер.

Для противников в игре характерно три поведения: противник «потревожен» и не «потревожен», а также противник стреляет в игрока. Если противник не «потревожен», его движение будет абсолютно случайным и скорость перемещения его будет невысокой. Если же противник был

«потревожен» звуком выстрела либо шагами игрока (радиус обнаружения будет разный), или противник увидел игрока, но не может стрелять по нему, тогда он следует на место положения игрока, в котором он был в момент обнаружения противником. Путь до игрока просчитывается с помощью алгоритма A^* на основании графа, построенного в менеджере карты (см. п. 3.1). Если игрок находится на сравнительно небольшом расстоянии от противника, то противник перестает куда-либо идти и начинает по нему стрелять, не зависимо от наличия стен на пути выстрела.

3.7. Менеджер игры

Менеджер игры является по сути своей связующим звеном между всеми остальными менеджерами. Основные функции данного менеджера состоят в загрузке и выгрузке ресурсов уровня, а также запуске игрового цикла. Итерация игрового цикла состоит из получения событий из менеджера событий, их обработки (задание вектора движения игроку, задание направления его взгляда, стрельба), обновления состояния противника, обновления физики всех объектов и отрисовки уровня и объектов. Кроме этого менеджер игры инициализирует остальные менеджеры, связывает их вместе. Также менеджер игры отправляет на сервер результаты игрока.

4. СЕРВЕРНАЯ ЧАСТЬ

Серверная часть приложения отвечает за загрузку ресурсов игры, за выдачу и сохранение результатов игры пользователей. Серверная часть реализована с помощью Node.js и библиотеки express. Результаты игрока хранятся на сервере в виде JSON-файла. При добавлении новой записи о результатах игрока она добавляется в массив результатов, который записывается в JSON-файл. При получении результатов отдается содержимое переменной на сервере.

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 разработаны 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/>