Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ**

**УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

**РАЗРАБОТКА 2D ПЛАТФОРМЕРА НА GODOT**

Курсовой проект по дисциплине

«Компьютерная графика»

Студент гр. 421-2

\_\_\_\_\_\_\_\_\_\_\_\_\_ И. В. Пшеничникова

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Руководитель

доцент кафедры АОИ, канд. техн. наук., доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.О. Перемитина

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Томск 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ**

**УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

**ЗАДАНИЕ**

на курсовой проект по дисциплине «Компьютерная графика»

студенту Пшеничниковой Ирине Владимировне

группа 421-2, факультет систем управления

1. Тема проекта: «Разработка 2D платформера на GODOT»
2. Срок сдачи студентом законченного проекта: *7 июня 2024 г.*
3. Исходные данные к проекту: Разработать 2D платформер на игровом движке Godot Engine.
4. Содержание пояснительной записки: введение, анализ задачи, обзор аналогов, алгоритм программы, тестирование и отладка, руководство пользователя, заключение.
5. Дата выдачи задания: *15 марта 2024 г.*

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доцент кафедры АОИ, канд. техн. наук, доцент, Перемитина Т.О.

Задание принял к исполнению 15 марта 2024 г.

(подпись студента)

Оглавление

[Введение 4](#_Toc165507637)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc165507638)

[2 ОБЗОР АНАЛОГОВ И ВЫБОР ИНСТРУМЕНТОВ 6](#_Toc165507639)

[2.1 Обзор движков 6](#_Toc165507640)

[2.2 Сравнение аналогов и выбор инструмента разработки 8](#_Toc165507641)

[3. ПРОЕКТИРОВАНИЕ ИГРЫ 10](#_Toc165507642)

[3.1 Логика игры 10](#_Toc165507643)

[3.2 Взаимодействующие объекты 10](#_Toc165507644)

[3.3 Окружение 12](#_Toc165507645)

[4. СОЗДАНИЕ РАБОЧЕГО ПРОЕКТА 15](#_Toc165507646)

[4.1 Сцены 15](#_Toc165507647)

[4.2 Персонажи 18](#_Toc165507648)

[Приложение А 22](#_Toc165507649)

[Приложение Б 23](#_Toc165507650)

[Приложение В 24](#_Toc165507651)

[Список используемых источников 25](#_Toc165507652)

# Введение

В современном мире разработка 2D компьютерных игр становится всё более доступной и популярной благодаря развитию игровых движков. Одним из популярных жанров 2D игр является игра-платформер[1].

Целью данного курсового проекта является реализация 2D платформера на игровом движке Godot Engine с использованием встроенного языка программирования GDScript.

Платформер – это жанр видеоигр, отличительной чертой которого является передвижение по поверхностям, расположенным на различной высоте – платформам. Игры этого жанра отличаются многообразием препятствий, с которыми сталкиваются пользователи в ходе перемещения по многоуровневым локациям[2].

Разработка 2D платформера позволит понять основные принципы создания игр в данном жанре, включая управление персонажем, физику движения, уровни и обстановку игрового мира.

Для разработки компьютерной игры в жанре «платформер» были поставлены следующие задачи:

* Провести анализ предметной области;
* Спроектировать будущую игру;
* Изучить выбранный стек технологий;
* Разработать 2D игру в жанре «платформер».

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Цель игры в жанре «платформер» обычно заключается в прохождении уровней, победе над врагом или достижении конечной точки. В основе данного жанра лежит преодоление препятствий, взаимодействие с окружающим миром и бои с врагами, управляемыми компьютером.

Целевая аудитория 2D игр в жанре «платформер» обычно включает в себя широкий диапазон возрастных групп, начиная от взрослых, заканчивая детьми. Данный жанр видеоигр может совместить в себе как и консервативные, устоявшиеся решения в игровом дизайне, так и привнести новые уникальные механики в игровой процесс, завлекая этим новых игроков.

Технические аспекты представляют собой стартовое меню с началом игры и магазином, спроектированную 2D местность с платформами для перемещения. Также используются готовые ассеты из свободного доступа для представления текстур и анимаций, использование которых значительно упрощает процесс разработки видеоигры. По окончанию прохождения уровней, пользователь получит сообщение об успешном прохождении игры и награду за это.

# ОБЗОР АНАЛОГОВ И ВЫБОР ИНСТРУМЕНТОВ

## Обзор движков

Перед тем, как начать сравнение движков, следует кратко описать каждый вариант:

**Unity** – это игровой движок, на котором разрабатывают мобильные игры, проекты для ПК (Windows, iOS, Linux) и консолей, например для Xbox или PlayStation. В нем есть разные компоненты для работы с графикой, анимацией, физикой объектов, звуком, шаблонами и скриптами[3].

Преимущества Unity:

* встроенный магазин ассетов;
* конструктор шейдеров;
* собрать игру на Unity получится, даже если вы не умеете писать код. В библиотеке Asset Store есть бесплатные шаблоны персонажей, звуков и фонов, которые можно использовать в первых проектах;
* начать разработку и выпускать свои первые проекты можно бесплатно с тарифом Personal, он предназначен для частных лиц и небольших организаций с доходом менее $100 тыс. за 12 месяцев. Когда этих возможностей будет недостаточно, можно переключиться на более продвинутые тарифы Plus или Pro[4].

**Godot Engine –** открытый кроссплатформенный игровой движок для разработки 2D/3D-видеоигр и приложений для ПК, мобильных устройств, веб-платформ. Адаптирован ко всем распространенным операционным системам, включая Linux, macOS, Windows, Android и iOS [5].

Godot имеет свой собственный высокоуровневый язык программирования GDScript, также можно использовать С# и C++.

Преимущества Godot:

* сравнительно простой в освоении и использовании игровой движок;
* встроенный язык программирования GDScript имеет почти такой же синтаксис, что и Python, и прост для изучения;
* интуитивно понятный редактор, возможность интеграции со сторонними плагинами и приложениями также снижают порог вхождения;
* созданные игры и приложения являются собственностью разработчика. Кроме того, открытость движка способствует появлению множества расширений и инструментов[6].

**Unreal Engine (UE) –** это движок для создания игр. Его основное отличие — хорошая оптимизация: Unreal Engine создавался не как отдельный коммерческий продукт, а как рабочий инструмент, и ориентирован он на 3D-игры, Epic Games (компания-разработчик движка) начали добавлять больше возможностей для двумерных игр относительно недавно.

В качестве языка программирования для Unreal Engine используется C++. Это мощный, быстрый, но довольно сложный язык, который непросто изучить с нуля. Тем не менее его применение позволяет хорошо оптимизировать игры [7].

Преимущества UE:

* собрать прототип или быстро прописать логику можно даже без навыков программирования — поможет визуальный язык (Blueprints);
* широкие возможности для 3D-графики, гибкая настройка материалов, освещения и эффектов — все это позволяет создавать в том числе фотореалистичные игры[8];
* Unreal Engine позволяет создать игру под любую операционную систему для компьютеров, а также для всех видов популярных консолей и для мобильных устройств.

## Сравнение аналогов и выбор инструмента разработки

Опираясь на информацию, доступную в различных источниках, был проведен анализ таких характеристик, как стоимость пакета, наличие документации и обучающих пособий по работе в программе, оценка сложности овладения навыками работы в программе, функциональные возможности и системные требования. Данный анализ был проведен с целью определения наиболее подходящего инструмента для разработки 2D платформера. Результаты данного анализа приведены в таблице 2.1.

Таблица 2.1 - Сравнение аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Движок | | |
| Unity | Godot Engine | Unreal Engine |
| Стоимость пакета | Бесплатно, если доходом менее $100 тыс. за 12 месяцев | Бесплатно | Бесплатно, если доход менее $1 млн. за 12 месяцев. Если размещать игру, то Epic Games всё равно забирает 5% от выручки |
| Наличие документации | Официальная документация есть | Официальная документация есть | Официальная документация есть |
| Сложность овладения навыками работы в программе | Если нет трудностей с C#, то сложностей не должно быть | Нет | Да |
| Системные требования | многоядерный Intel или AMD с поддержкой набора инструкций SSE;  видеокарта с поддержкой DX10, DX11, или DX12;  Windows 7 (SP1+) / 10 64-разрядная | Windows 7+;  Оперативная память от 4ГБ;  Процессор - Intel Core i3-7100 или выше | Windows 10 64-разрядная версия;  четырехъядерный процессор Intel или AMD с тактовой частотой 2,5 ГГц или выше;  8ГБ оперативной памяти;  видеокарта, совместимая с DirectX 11 или DirectX 12 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Движок | | |
| Unity | Godot Engine | Unreal Engine |
| Преимущества | Поддержка сообщества;  Низкий порог вхождения в разработку;  Доступность | Простота;  Поддержка сторонних языков;  Гибкость;  Кроссплатформенность;  Обширная документация | Высокая производительность;  Наличие Blueprints;  Кроссплатформенность; |
| Недостатки | Производительность;  Оптимизация;  Нет шаблонов | Сложности с разработкой консольных приложений | Нацеленность на 3D;  Сложность C++;  Высокие системные требования;  Дороговизна дополнительных модулей |

По результатам проведенного анализа видно, что для разработки небольшой игры, например 2D платформера, имеется смысл взять более простой движок, а именно Godot Engine.

# 3. ПРОЕКТИРОВАНИЕ ИГРЫ

### Логика игры

Цель игры «2D платформер» – прыгать по платформам, убивать монстров и набирать как можно больше очков за счет собранных денег, на которые потом можно купить улучшения для главного персонажа.

Игровой процесс будет построен на цикличности: игрок будет проходить уровни, собирать деньги и уничтожать монстров до тех пор, пока не решит завершить игру или не проиграет из-за потери всех жизней.

Как уже говорилось ранее, деньги, собранные во время прохождения уровней, могут быть потрачены на покупку улучшений для персонажа, таких как увеличение здоровья, силы атаки или скорости перемещения. Данные улучшения помогут игроку стать сильнее и эффективнее в борьбе с монстрами и прохождении сложных уровней.

### Взаимодействующие объекты

Игрок управляет главным персонажем – собакой с мечом, изображенном на рисунке 3.1. Игрок имеет возможность передвигать персонажа влево и вправо, кроме этого, персонаж может прыгать и бить монстров мечом. Все движения персонажа анимированы. В приложении А представлена раскадровка анимаций персонажа.



Рисунок 3.**Ошибка! Текст указанного стиля в документе отсутствует.**.1 - Главный персонаж

Противниками главного персонажа являются «слаймы», изображенные на рисунке 3.2. Монстры расставлены по всей карте, они могут атаковать игрока и преследовать его. Каждый монстр уничтожается не с первого удара, то есть у них есть определенное количество здоровья, как и у главного персонажа.

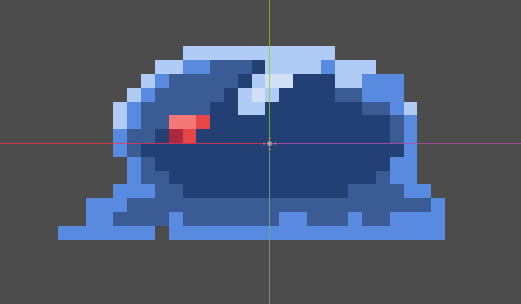


Рисунок 3. 2 – Изображение монстра

Кроме «мелких» противников, игроку нужно будет убить «босса», изображенного на рисунке 3.3. Это монстр, который умеет атаковать и двигаться за персонажем. Здоровья у главного противника будет больше, чем у противников поменьше.



Рисунок 3. 3 - Изображение "Босса"

Все действия монстров, также как и действия главного персонажа, сопровождаются анимацией, раскадровку которой можно увидеть в приложении Б.

### Окружение

Как только игрок открывает игру, на экране появляется главное меню, на котором выбрать действие с помощью нажатия на кнопку. Кнопка «Играть» открывает уровень, и игрок может начать свою игру. Кнопка «Магазин» открывает страницу магазина, на которой можно приобрести улучшения для персонажа. И наконец кнопка «Выход» закрывает окно с игрой. Главный экран игры представлен на рисунке 3.4.

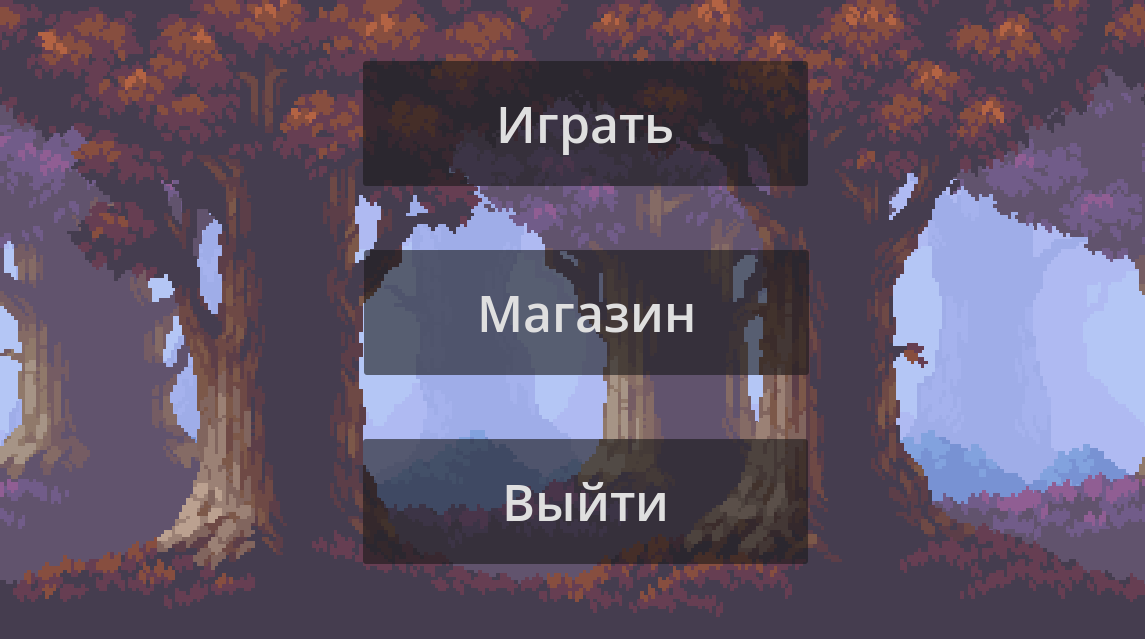


Рисунок 3. 4 - Главный экран

Фон игры представляет собой изображение леса, изображенного на рисунке 3.5. Кроме того, на главном экране фон сделан с использованием паралакс, то есть он двигается. На уровнях фон статичен.



Рисунок 3. 5 - Фон игры

Главный персонаж на всех уровнях собирает игровую валюту – звезда, которая изображена на рисунке 3.6.

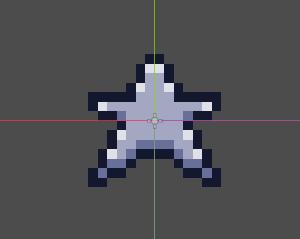


Рисунок 3. 6 - Изображение игровой валюты

Что у игрока, что у монстров есть жизни, которые представляют из себя красную полоску. Когда здоровье уменьшается, то и полоска уменьшается. В случае игрока, здоровье будет уменьшаться, если на него нападет противник, а здоровье противника будет уменьшаться, если его ударит игрок. На рисунке 3.7 представлена линия, изображающая здоровье игрока, а на рисунке 3.8 здоровье монстра.

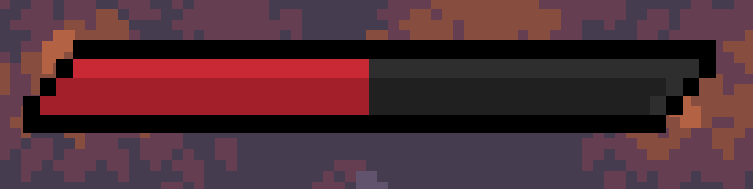


Рисунок 3. 7 - Полоса здоровья главного персонажа

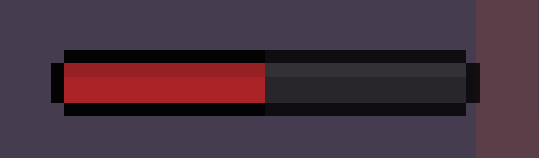


Рисунок 3. 8 - Полоса здоровья монстров

# 4. СОЗДАНИЕ РАБОЧЕГО ПРОЕКТА

### Сцены

Игры в GODOT представляют собой дерево сцен, а каждая сцена представляет собой дерево узлов.

Для создания рабочего проекта были разработаны основные сцены – главное меню, два уровня (один из которых обычный, а второй с боссом) и магазин. Рассмотрим каждую более подробно.

На рисунке 4.1 изображена готовая сцена главного меню. В её основу вошли такие узлы, как кнопки (Button) и фон с эффектом параллакса (ParallaxBackground). Взаимодействие пользователя с кнопками происходит при помощи сигналов, которые «навешиваются» на кнопки. При нажатии на кнопку «Магазин» пользователю открывается сцена магазина, при нажатии на кнопку «Играть» пользователю открывается сцена первого уровня, при нажатии на кнопку «Выйти» игра закрывается.



Рисунок 4. 1 - Сцена "Меню"

На рисунке 4.2 представлена сцена магазина. В основу данной сцены вошли в основном кнопки и надписи. При нажатии на кнопку «Назад», пользователь возвращается в главное меню. При нажатии кнопки «Купить», с пользователя списывается 5 игровых валют (звезды). Подразумевалось, что зелья при приобретении будут давать тот эффект, что написан в их названии (зелье «Ускорение» - повышенная скорость, зелье «Полет» - повышенная высота прыжка, зелье «Здоровье» - увеличенное здоровье), но в силу сложности реализации и ограниченности во времени, данная функция не была осуществлена.

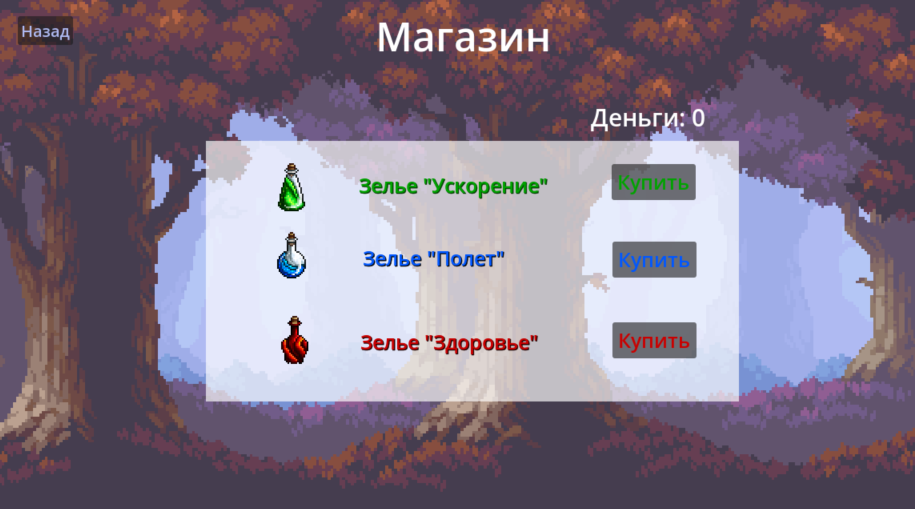


Рисунок 4. 2 - Сцена "Магазин"

На рисунке 4.3 представлена самая крупная сцена – первый уровень. В данную сцену вошли как небольшие узлы (надписи, кнопки, фон), так и узлы в виде других сцен, например, игрок или моб, но об этих сценах речь пойдёт в следующем пункте.

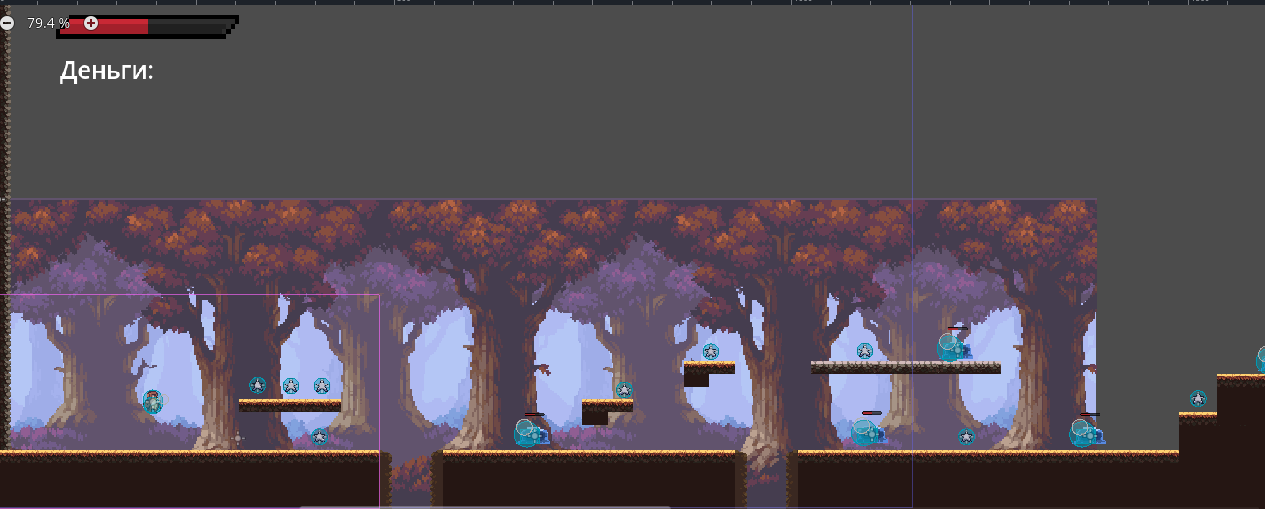


Рисунок 4. 3 - Сцена "Уровень 1"

Стоит обратить внимание на рисовку платформ, и пола. Всё это осуществлялось за счет узла TileMap. Для того, чтобы начать заполнять уровень различными рисовками, следует к узлу TileMap привязать TileSet. TileSet – это набор элементов, из которых будет состоять уровень. В данной работе взят готовый ассет для заполнения уровня. На рисунке 4.4 представлен заполненный готовым ассетом TileMap для дальнейшей прорисовки различных поверхностей.

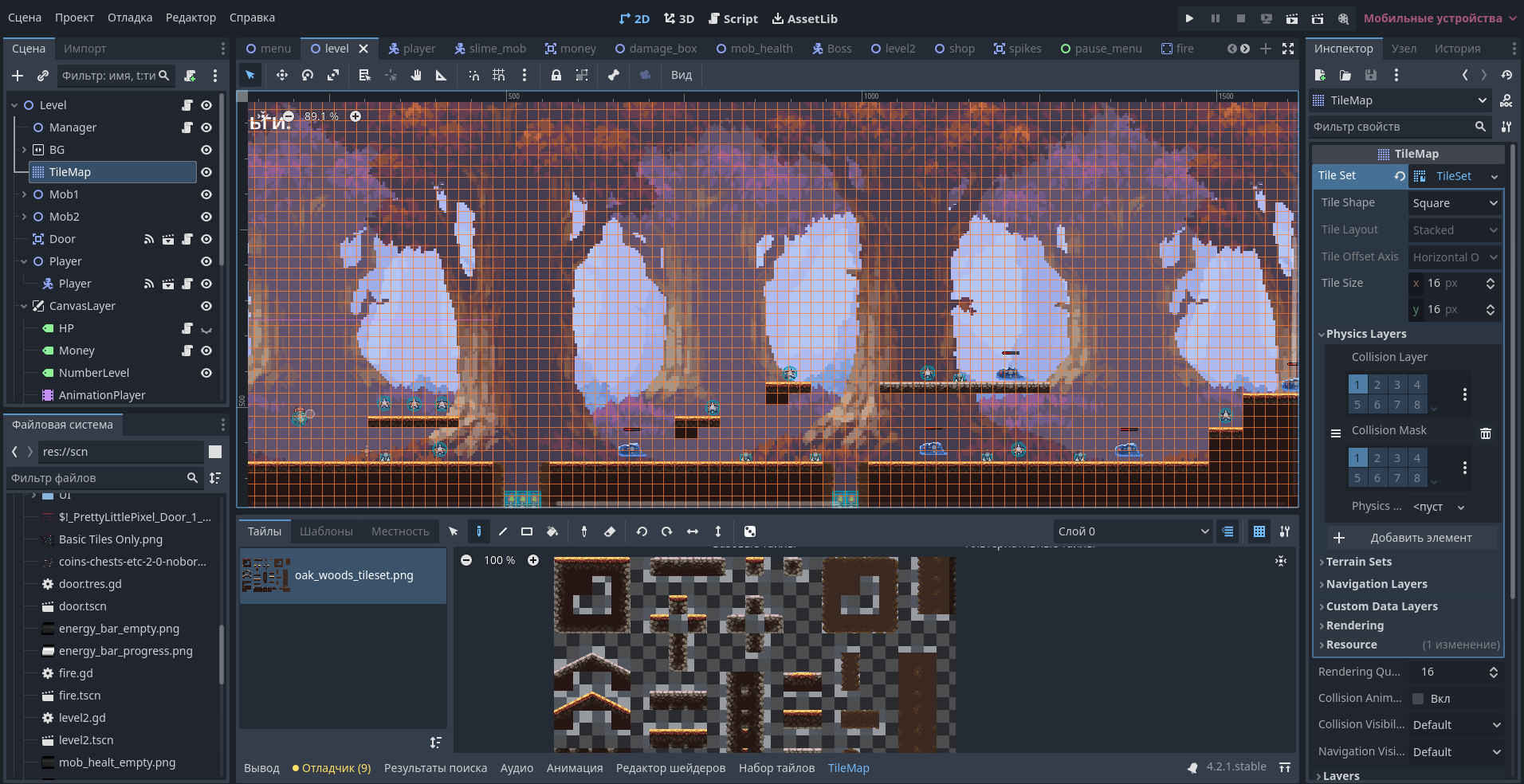


Рисунок 4. 4 - Пример работы с TileMap

Кроме этого, при нажатии кнопки «Esc» игра ставится на паузу, и пользователь может перейти в магазин, продолжить игру или выйти в главное меню. На рисунке 4.5 представлено представление данной функции.

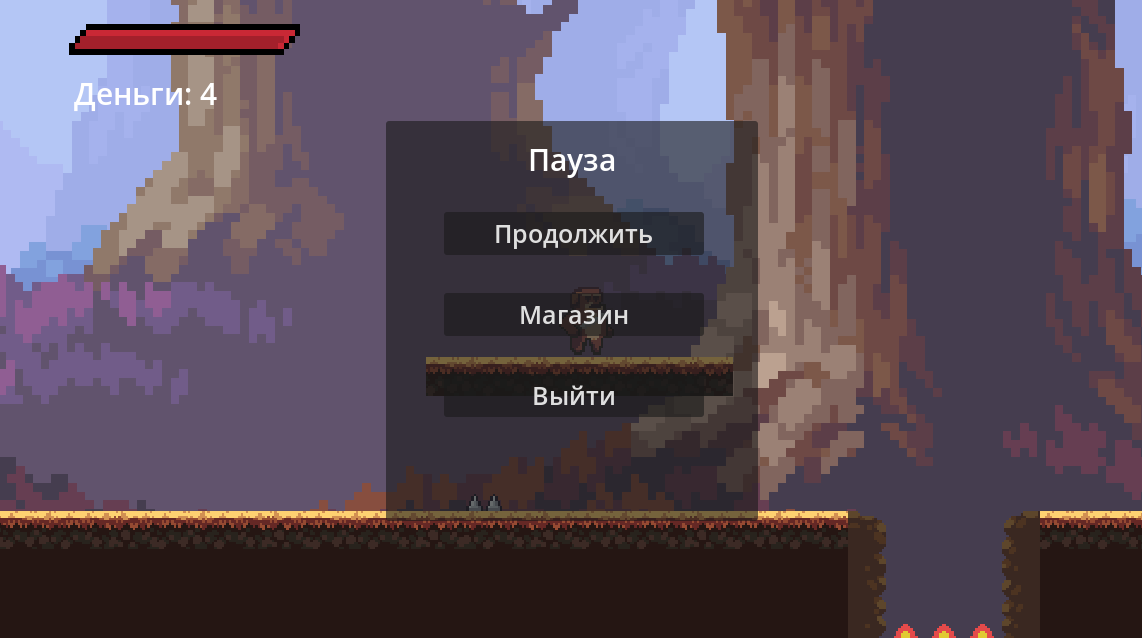


Рисунок 4. 5 - Экран паузы

Последняя сцена из основных сцен – это второй уровень, на котором подразумевается бой с боссом. Данная сцена представлена на рисунке 4.6. Основными узлами здесь являются сцены босса и персонажа.



Рисунок 4. 6 - Сцена "Уровень 2"

Рассмотрим более подробно сцены и взаимодействие между сценами главных персонажей.

### Персонажи

На рисунке 4.7 представлена сцена главного персонажа. Для создания динамичного персонажа требуется анимация, которая добавляется с помощью узла AnimationPlayer или AnimatedSprite2D.

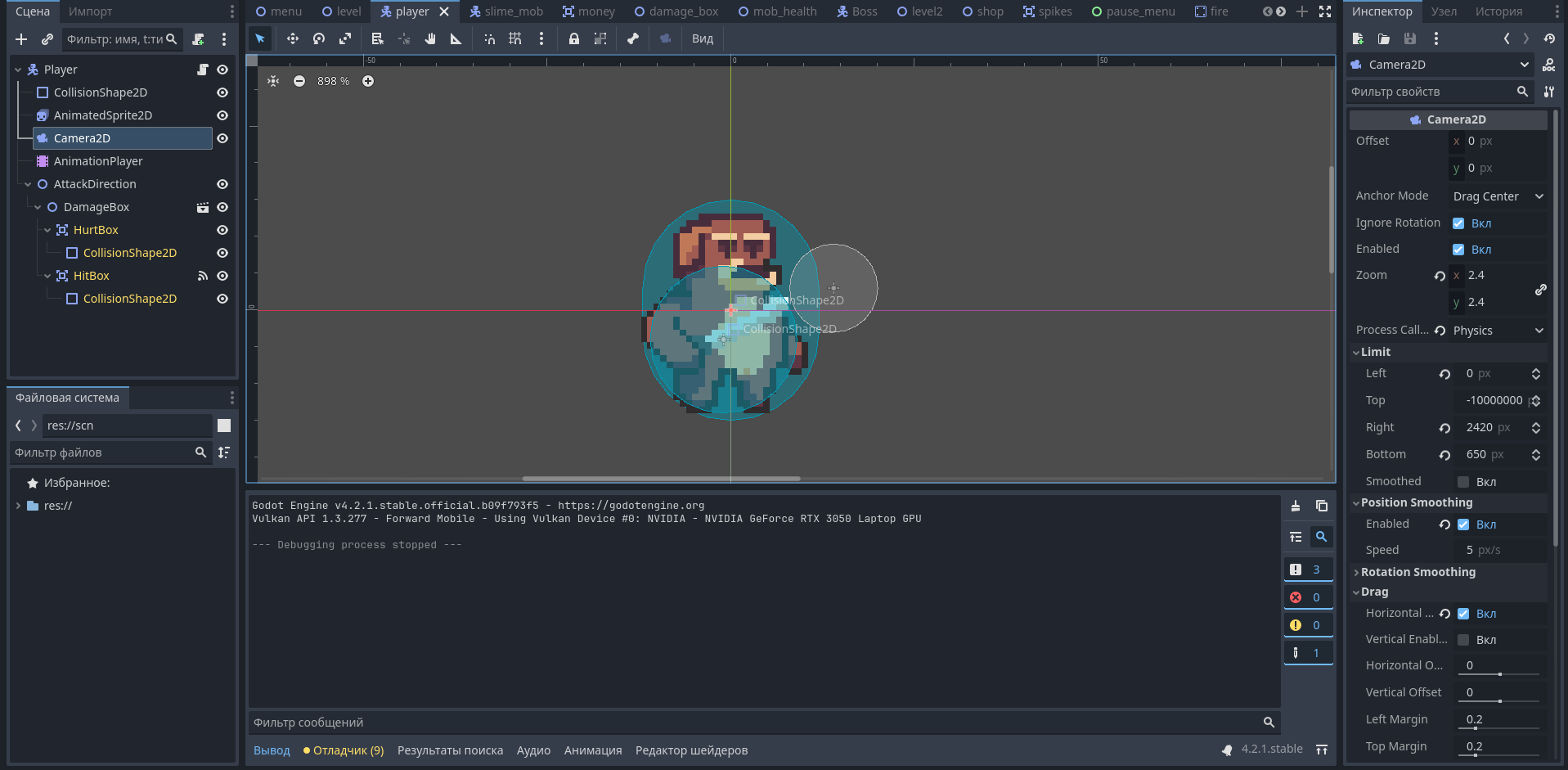


Рисунок 4.7 - Сцена главного персонажа

Так как главный герой двигается, пользователь должен отслеживать это движение. Для этого нужна камера, которая добавляется отдельным узлом к сцене персонажа. Настройка камеры осуществлялась с помощью графического интерфейса, а не кодом.

На рисунке 4.8 и 4.9 представлена сцена моба и босса.

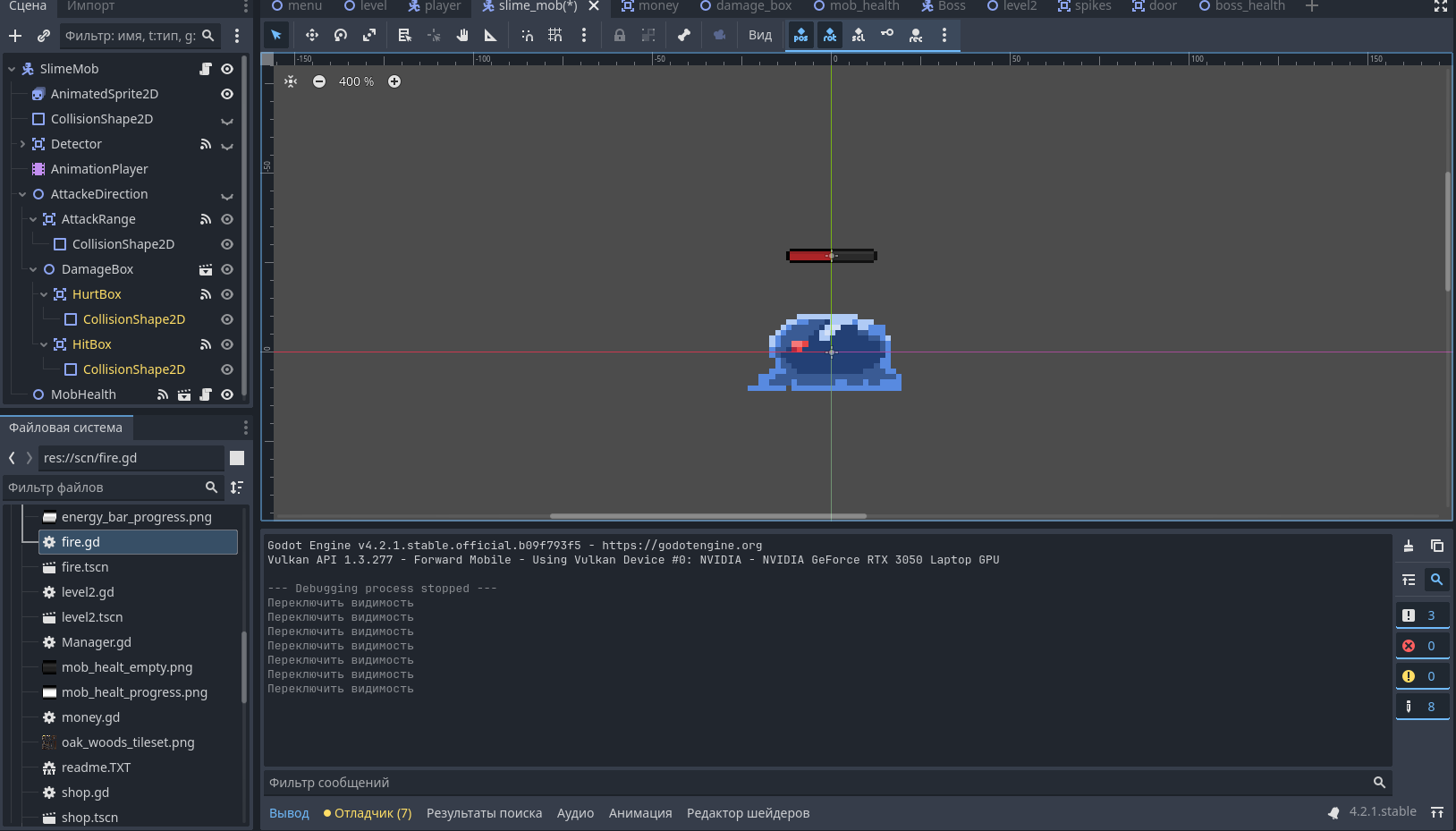


Рисунок 4. 8 - Сцена моба

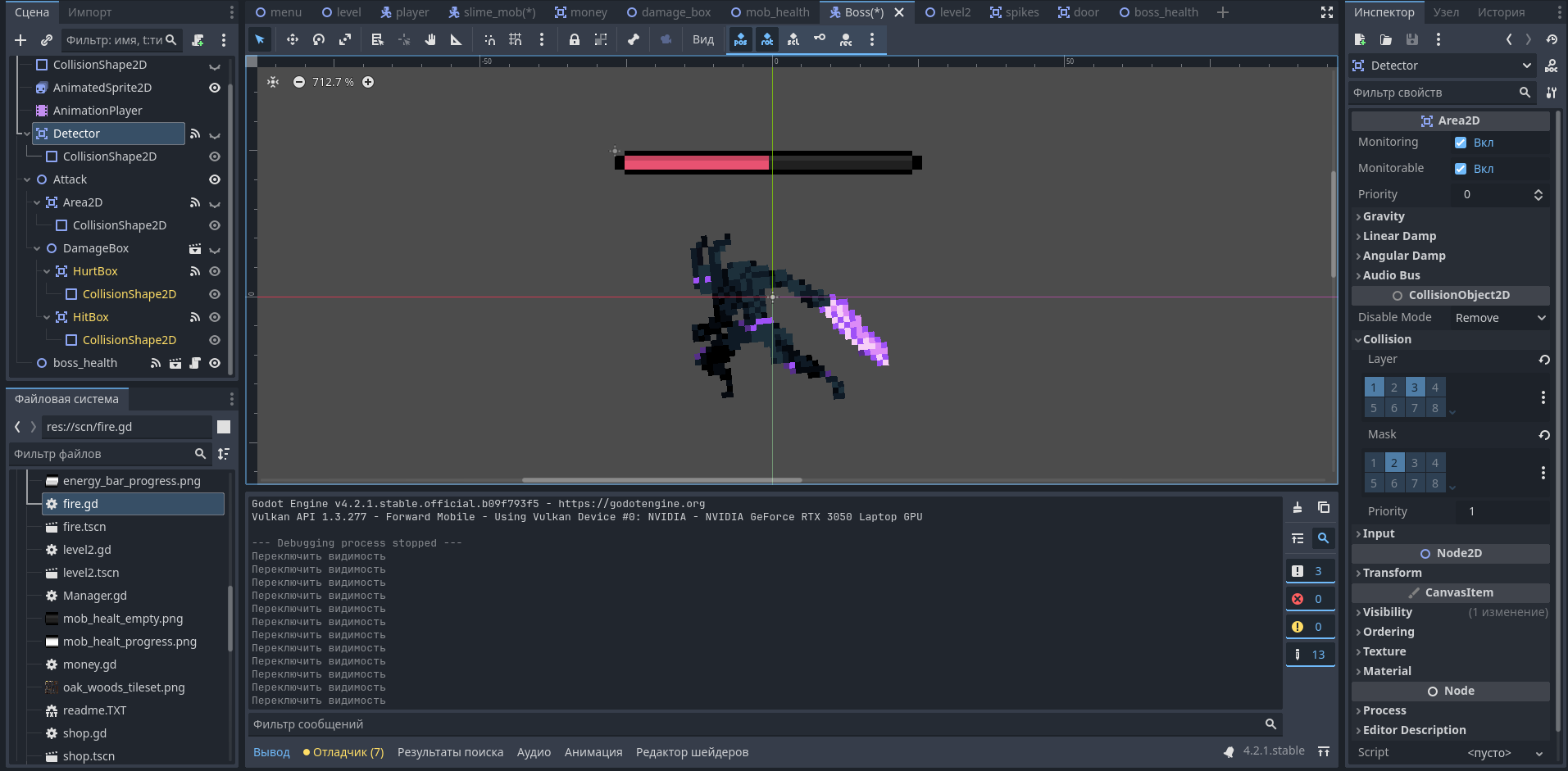


Рисунок 4. 9 - Сцена босса

Взаимодействие между персонажами в GODOT происходит за счет Area2D, CollisionShape2D и сигналов. Например, для того, чтобы босс погнался за главным персонажем, ему был создан узел Area2D, а к нему уже узел CollisionShape2D, который изображен на рисунке 4.10. Для данной области было создано два сигнала. Один из них срабатывает, когда персонаж заходит в зону (начать погоню), а другой, когда персонаж выходит из зоны (остановить погоню). Сигналы прикрепляются к главному скрипту как обычные функции. Примеры некоторых сигналов представлены в приложении В.

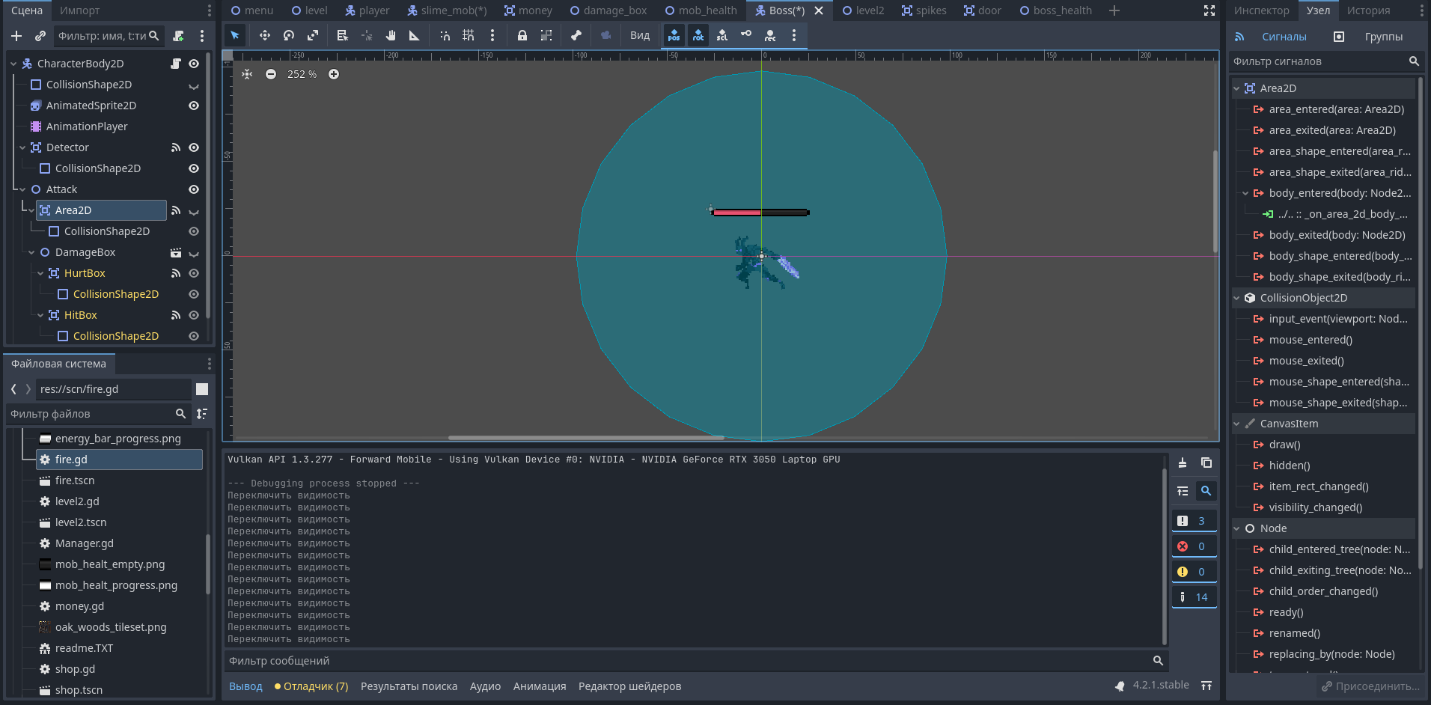


Рисунок 4. 10 - Пример выделения области

Кроме того, были созданы сигналы для нанесения удара и дальнейшего получения урона, как для персонажа от моба, так и для моба от главного персонажа, сигналы для сбора игровой валюты и перехода на следующий уровень (так как переход осуществляется с помощью «двери»).

# Приложение А

(справочное)

**Раскадровка анимаций главного персонажа**



Рисунок А.**Ошибка! Текст указанного стиля в документе отсутствует.**.1 – Анимация «Без движения»



Рисунок А. 2 - Анимация «Бег»

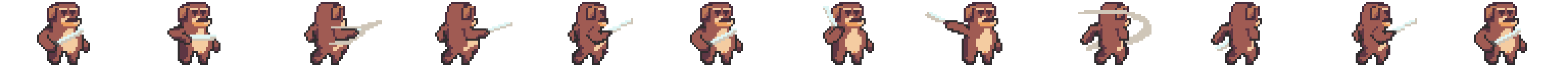


Рисунок А. 3 - Анимация «Удар»



Рисунок А. 4 - Анимация получения урона



Рисунок А. 5 - Анимация «Смерть»



Рисунок А. 6 - Анимация «Прыжок»

# Приложение Б

(справочное)

**Раскадровка анимаций монстров**



Рисунок Б. 1 - Анимация «Без движения»



Рисунок Б. 2 - Анимация атаки

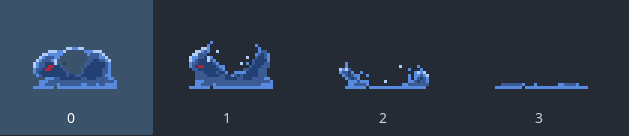


Рисунок Б. 3 - Анимация смерти

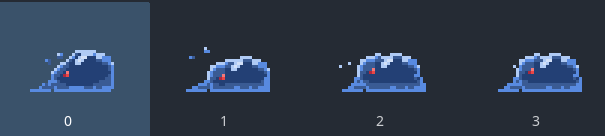


Рисунок Б. 4 - Анимация получения урона

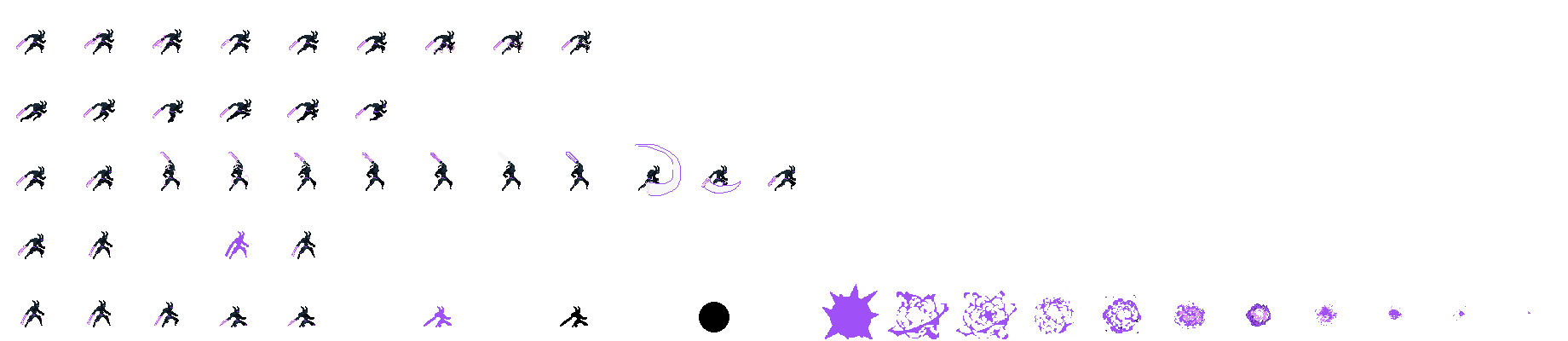


Рисунок Б. 5 - Раскадровка анимации главного злодея

# Приложение В

(справочное)

**Пример сигналов**

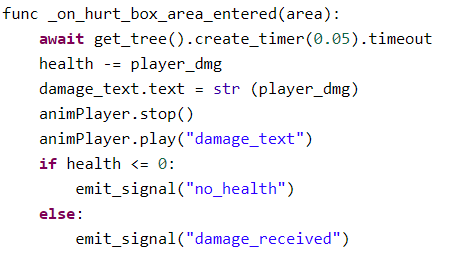


Рисунок В. 1 - Сигнал для отслеживания здоровья моба

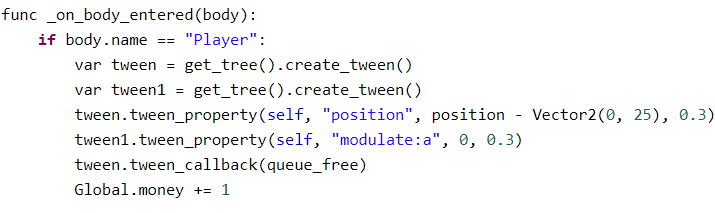


Рисунок В. 2 - Сигнал для сбора игровой валюты

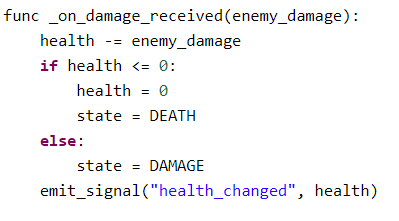


Рисунок В. 3 - Сигнал для отслеживания состояния персонажа после удара моба

# Список используемых источников

1. Руководство по популярным жанрам 2D игр [Электронный ресурс] – URL <https://gamemaker.io/ru/blog/2d-game-genres> (дата обращения: 26.03.2024).
2. Большая Российская энциклопедия [Электронный ресурс] – URL <https://bigenc.ru/c/platformer-b45518> (дата обращения: 26.03.2024).
3. Что такое Unity? [Электронный ресурс] – URL <https://blog.skillfactory.ru/glossary/unity/> (дата обращения: 07.04.2024).
4. Официальный сайт Unity [Электронный ресурс] – URL <https://unity.com/ru> (дата обращения: 07.04.2024).
5. Что такое Godot Engine [Электронный ресурс] – URL <https://blog.skillfactory.ru/glossary/godot-engine/> (дата обращения: 07.04.2024).
6. Официальный сайт Godot Engine [Электронный ресурс] – URL <https://godotengine.org/> (дата обращения 07.04.2024).
7. Что такое Unreal Engine? [Электронный ресурс] – URL <https://blog.skillfactory.ru/glossary/unreal-engine/> (дата обращения: 08.04.2024).
8. Официальный сайт Unreal Engine [Электронный ресурс] – URL <https://www.unrealengine.com/en-US> (дата обращения: 08.04.2024).