

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
профессор

\_\_\_\_\_ Л.Б. Соколинский

«\_\_\_»\_\_\_\_\_ 2024 г.

**Разработка компьютерной игры в жанре «RPG»  
на платформе Ren'Py**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 02.03.02.2024.308-545.ВКР**

Научные руководители:  
ст. преподаватель кафедры СП  
\_\_\_\_\_ Н.С. Силкина,

преподаватель кафедры СП  
\_\_\_\_\_ М.Н. Глизница

Автор работы,  
студент группы КЭ-402  
\_\_\_\_\_ Н.С. Ермолин

Ученый секретарь  
(нормоконтролер)  
\_\_\_\_\_ И.Д. Володченко  
«\_\_\_»\_\_\_\_\_ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_ Л.Б. Соколинский

29.01.2024 г.

### **ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**

студенту группы КЭ-402

Ермолину Николаю Сергеевичу,

обучающемуся по направлению

02.03.02 «Фундаментальная информатика и информационные технологии»

**1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка компьютерной игры в жанре «RPG» на платформе Ren'Py.

**2. Срок сдачи студентом законченной работы:** 03.06.2024 г.

**3. Исходные данные к работе**

3.1. Ren'Py's documentation. [Электронный ресурс] URL:

<https://www.renpy.org/doc/html/index.html>. (дата обращения: 01.02.2024 г.).

3.2. Ren'Py документация на русском. [Электронный ресурс] URL:

[http://ru.renpyedia.shoutwiki.com/wiki/Renpy\\_для\\_чайников](http://ru.renpyedia.shoutwiki.com/wiki/Renpy_для_чайников). (дата обращения: 01.02.2024 г.).

3.3. Сайт движка Ren'Py. [Электронный ресурс] URL: <https://www.renpy.org>

(дата обращения: 01.02.2024 г.).

3.4. Онлайн сервис инди игр. [Электронный ресурс] URL: <https://itch.io> (дата

обращения: 01.02.2024 г.).

**4. Перечень подлежащих разработке вопросов**

4.1. Провести обзор существующих игр в похожем жанре.

4.2. Проанализировать и решить какие идеи будут присутствовать в игре.

4.3. Реализовать игру до состояния Демо версии.

4.4. Провести тестирование.

**5. Дата выдачи задания: 29.01.2024 г.**

**Научные руководители:**

ст. преподаватель кафедры СП

Н.С. Силкина,

преподаватель кафедры СП

М.Н. Глизница

**Задание принял к исполнению**

Н.С. Ермолин

## ГЛОССАРИЙ

1. *Визуальная новелла* – это жанр компьютерной игры, подвид текстового квеста, в котором зрителю демонстрируется история при помощи вывода на экран текста, изображений, а также звукового сопровождения [1].

2. *Ассет* – это может быть набор готовых ресурсов, которые можно использовать при создании игр [2].

3. *NPC / НПС* – персонаж в играх, который не находится под контролем игрока [3].

4. *Scripts/Скрипты* – небольшая программа, которая выполняет конкретную задачу [4].

5. *Геймплей* – компонент игры, отвечающий за взаимодействие игры и игрока. Геймплей описывает, как игрок взаимодействует с игровым миром, как игровой мир реагирует на действия игрока и какой есть набор действий, который предлагает игроку игра [5].

6. *Fantasy / Фэнтези* – это жанр, который обычно основан на мифологии, магии, существах и сюжетах, несуществующих в реальном мире. Он часто включает в себя элементы волшебства, приключений, битв с добром и злом, а также исследование фантастических миров и персонажей [6].

7. *Dark Fantasy / Темное фэнтези* – поджанр Fantasy, включающий в себя элементы ужасов, готики или проще говоря когда окружающий мир не очень дружелюбен к персонажу [7].

8. *RPG* – это может быть жанр компьютерных игр, в которых игроки принимают на себя роль вымышленного персонажа и взаимодействуют с виртуальным миром, выполняя различные задания, развивая своего персонажа и принимая решения, которые влияют на ход игры [8].

9. *AAA проект / игра* – это термин обозначающий класс высокобюджетных компьютерных игр [9].

## **ОГЛАВЛЕНИЕ**

ГЛОССАРИЙ.....	4
ВВЕДЕНИЕ.....	6
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	8
1.1. Предметная область проекта .....	8
1.2. Анализ проектов с похожими жанрами.....	8
1.3. Обоснование выбора.....	12
2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ.....	14
2.1. Функциональные и нефункциональные требования.....	14
2.2. Диаграмма вариантов использования .....	14
3. АРХИТЕКТУРА СИСТЕМЫ.....	16
3.1. Общее описание архитектуры системы.....	16
3.2. Диаграмма компонентов и их описание .....	17
4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ СИСТЕМЫ .....	19
4.1. Реализация компонентов и интерфейсов системы .....	19
4.2. Тестирование игрового приложения.....	23
ЗАКЛЮЧЕНИЕ .....	25
ЛИТЕРАТУРА.....	26
ПРИЛОЖЕНИЯ.....	28
Приложение А. Спецификация вариантов использования.....	28
Приложение Б. Скриншоты итоговой версии игры .....	31

## **ВВЕДЕНИЕ**

### **Актуальность**

В последнее время в мире очень сильно распространяется такой вид развлечения как компьютерные игры. На сегодняшний день есть огромное количество самых различных игр. От обычных небольших игр, до огромных AAA проектов где можно провести огромное количество своего свободного времени. Каждый человек выбирает, то что ему больше всего по душе. Начиная с этапа разработки, при использовании одного и того же движка может выйти огромное количество разнообразных вариаций игр. Как пример можно рассмотреть такой игровой движок как Ren'Py [10].

Движок Ren'Py, это бесплатный, свободный и открытый движок для создания как некоммерческих, так и коммерческих игр. Сам движок был разработан благодаря человеку под псевдонимом Tom Rothamel «PyTom» в 2004 году. На данный момент есть много различных документаций, как на английском, так и на русском [11, 12].

В основном движок Ren'Py используют для разработки визуальных новелл, однако если можно использовать один и тот же движок и делать разные игры, то тоже самое можно сделать и с Ren'Py. В этом и будет заключаться актуальная задача, разработать игру, используя движок в жанре RPG.

### **Постановка задачи**

Целью выпускной квалификационной работы является «Разработка компьютерной игры в жанре RPG на Ren'Py».

Для достижения поставленной цели необходимо решить следующее:

- 1) провести обзор существующих игр в похожем жанре;
- 2) проанализировать и решить какие идеи будут присутствовать в игре;
- 3) реализовать игру до состояния Демо версии;
- 4) провести тестирование.

## **Структура и содержание работы**

Работа состоит из глоссария, введения, четырех глав, заключения, списка литературы и приложений. Объем работы составляет 33 страницы, объемом списка литературы – 19 источников.

В глоссарии даны определения терминам используемые для объяснения тех или иных вещей.

В первой главе предстоит провести «Анализ предметной области». В данной главе предстоит ознакомиться с выбранным жанром, проанализировать проекты с похожими жанрами или тематиками и решить какие средства будут использованы при разработке.

Во второй главе предстоит провести «Определение требований». В данной главе предстоит рассмотреть функциональные и нефункциональные требования, изучить методы описания требований к программным системам на основе диаграмм вариантов использования UML.

В третьей главе предстоит провести «Архитектура системы». В данной главе предстоит рассмотреть общее описание архитектуры системы, изучить и реализовать диаграмму компонентов составляющих систему.

В четвертой главе предстоит провести «Реализация и тестирование системы» В данной главе показаны примеры из реализованного приложения и проведенные тесты.

В заключение подведены итоги работы и дальнейшие планы.

В списке литературы представлены ссылки на источники используемые для реализации проекта и написании работы.

В приложении А содержится описание спецификаций вариантов использования.

В приложении Б содержатся скриншоты текущей версии игры.

## **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

### **1.1. Предметная область проекта**

Перед началом следует ознакомиться с выбранным жанром. Основной жанр игры – «RPG». Примером жанра «RPG» являются компьютерные игры, в которых игрок управляет одним или несколькими персонажами. Эти персонажи обладают различными характеристиками, которые можно развивать, а также умениями и предметами, влияющими на ход игры.

Кроме жанра «RPG», игра будет использовать тематику «Fantasy». Жанр «Fantasy» характеризуется наличием волшебства, магии, фантастических существ и миров, отличных от реального. Он часто включает эпические приключения, битвы между добром и злом, а также создание уникальных и мистических миров с собственными правилами и законами.

В итоге, игра будет выполнена в жанре «Fantasy RPG».

Перед началом реализации проекта и выбора идей для реализации следует провести анализ и рассмотреть другие игровые проекты, выполненные в схожем жанре.

### **1.2. Анализ проектов с похожими жанрами**

Во время поиска и анализа проектов на движке Ren'Py в жанре «Fantasy RPG» было обнаружено, что таких игр практически не существует или мало. Однако, были найдены игры, созданные на других движках, которые содержат элементы фэнтези и RPG.

Первым примером для анализа будет проект под названием «TES Skyrim» [13].

Игра в жанре Fantasy RPG представляет собой игру от третьего лица, где игроку предоставляется персонаж. У этого персонажа имеется древо навыков, которое можно развивать, выполняя различные действия. На рисунке 1 представлен вид самого древа навыков. В игре также присутствует большое количество NPC и заданий. После выполнения заданий игрок получает награду в виде золота, оружия или других предметов.



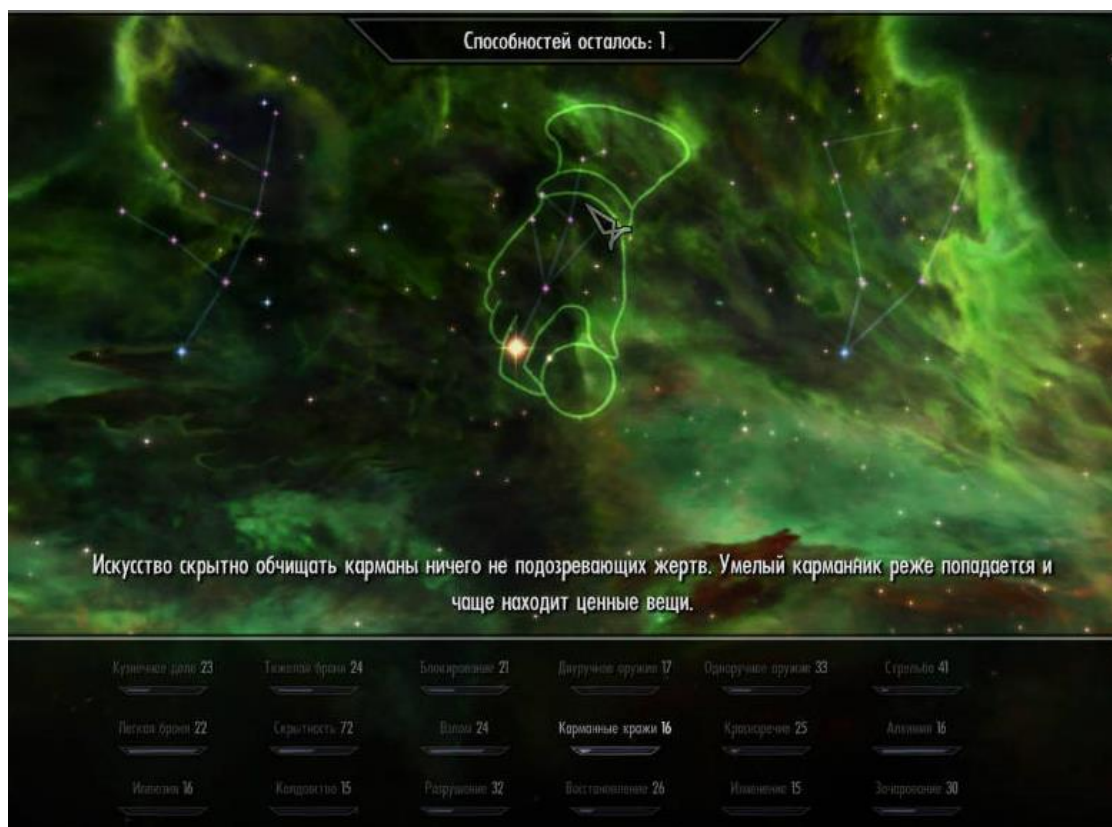


Рисунок 1 – Древо навыков

Вторым примером является игра под названием «Witcher 3» [14]. Эта игра представляет собой игру от третьего лица, включающую прокачку персонажа, взаимодействие и общение с NPC, выполнение заданий. На игровой карте присутствуют места интереса, в которых расположены сокровища, события и монстры. Игра в целом содержит огромное количество разного рода заданий. В игре есть несколько способов как принять задание. При разговоре с NPC, случайно наткнуться на событие которое приведет к заданию или взять задание на доске объявлений.

На доске местные жители оставляют просьбы о помощи. Пример внешнего вида доски объявлений представлен на рисунке 2. Таким образом, игрок может выбрать задание, сорвав объявление, выполнить его и вернуться к заказчику, чтобы сдать задание и получить награду.



Рисунок 2 – Доска объявлений

Третий пример это игра которая разработана на RPG Maker и называется «Vampire life» [15]. Игра в духе «Dark Fantasy». На данный момент игра находится на стадии разработки, однако уже содержит некоторый контент.

В этой игре игрок может путешествовать по глобальной карте, представленной на рисунке 3. Он может сражаться с монстрами и бандитами, выполнять задания, прокачивать свою репутацию (быть хорошим или плохим персонажем), торговаться, создавать предметы, носить оружие, броню и многое другое.



Рисунок 3 – Глобальная карта

Последняя игра для анализа – это игра под названием «Heads Will Roll» [16], созданная на движке Ren'Py. Эта игра представляет собой пошаговую RPG с элементами выживания и нелинейным сюжетом, который меняется в зависимости от решений, принимаемых игроком. Внешний вид самой игры представлен на рисунке 4.



Рисунок 4 – Игра «Heads Will Roll»

В данной игре главный герой является игроком. Игра включает RPG-систему развития персонажа. Возможна экипировка персонажа, при этом каждая вещь имеет специфические эффекты. Игрок может взаимодействовать с персонажами, что влияет на дальнейший геймплей. Также предусмотрены возможности создания и продажи предметов, разборки старых и приобретения новых.

При анализе данных игр было принято решение выбрать следующие моменты, которые будут реализованы в проекте.

1. Сделать возможность перемещаться как по глобальной карте и по разным локациям.

2. Сделать стенд с заданиями. Подойдя к стенду, игрок сам выбирает что ему выполнить.
3. Прокачка основных и пассивных навыков в зависимости от того чем занят игрок. Если игрок часто сражается, то прокачиваются навыки боя, если игрок что-то собирает, то навыки сбора и так далее.
4. Возможность носить и использовать различные вещи.
5. Возможность продавать и покупать разные вещи.
6. Во время вылазок или выполнения заданий, на игрока могут напасть и ему придется защищаться.
7. Возможность взаимодействия с NPC или объектами.

### **1.3. Обоснование выбора**

Для реализации будет использован движок Ren'Py. Обычно этот движок применяется для создания визуальных новелл, но также существует множество проектов в различных жанрах, которые предлагают более сложный геймплей, чем простое нажатие одной кнопки для продолжения диалога.

Для данного движка доступно большое количество информации, которую можно найти как на официальном сайте Ren'Py, так и на других ресурсах. Также существуют видеоуроки по созданию игр на этом движке.

Для создания персонажей будет использоваться программа «Koikatsu Party» и связанная с ней программа «Chara Studio». «Koikatsu Party» будет применяться для создания персонажей, которые будут использоваться в проекте. «Chara Studio» представляет собой редактор, в котором можно создавать различные сцены, расставляя объекты по необходимости. Программа также содержит множество готовых ассетов, что упрощает некоторые аспекты разработки.

Этот выбор сделан из-за простоты использования как самих программ, так и игрового движка.

## **Вывод по первому разделу**

Во время анализа предметной области проекта было проведено ознакомление с основным жанром игры – «RPG». Для уточнения будущей концепции игры был дополнительно рассмотрен жанр «Fantasy».

В ходе поиска и анализа проектов в схожих жанрах были выделены четыре проекта.

Из первого проекта была взята особенность системы прокачки «что используется, то и качается».

Из второго проекта была заимствована идея брать задания с доски объявлений, позволяя игроку самостоятельно выбирать, какие задания выполнять.

Из третьего проекта была взята возможность путешествовать по глобальной карте, чтобы игрок мог исследовать мир и сталкиваться с различными событиями.

Из четвертого проекта была подчеркнута возможность взаимодействия с инвентарем, где игрок может видеть, что у него есть, что он может делать со своими вещами, а также просматривать свои характеристики и другую внутриигровую информацию.

Среди всех проанализированных проектов только один был создан на движке Ren'Py, что свидетельствует о редком использовании этого движка для создания игр в жанрах, не связанных с визуальными новеллами. Это может быть преимуществом для тех, кто хочет занять этот пробел и создавать игры на Ren'Py в различных жанрах с разными особенностями.

Выбор Ren'Py был обоснован его простотой и удобством. Программы «Koikatsu Party» и «Chara Studio» были выбраны из-за их простоты, наличия большого количества ассетов и легкости создания сцен для использования в игровом приложении.

## **2. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ**

### **2.1. Функциональные и нефункциональные требования**

В игровых терминах функциональные требования – это конкретные действия или задачи, которые должна выполнять игра, например, управление персонажами, активация объектов или инициирование событий.

Нефункциональные требования относятся к свойствам или ограничениям, которым должна соответствовать игра, например, насколько быстро она загружается, насколько легко её поддерживать или обновлять.

Функциональные требования к игровому приложению:

- 1) игрок может перемещаться по разным локациям и по глобальной карте;
- 2) игрок может прокачивать своего игрового персонажа;
- 3) игрок может взаимодействовать с NPC или объектами;
- 4) игрок может брать и выполнять задания;
- 5) игрок может взаимодействовать с объектами в своем инвентаре;
- 6) игрок может сражаться или убежать во время стычек с врагами;
- 7) игрок может продавать и покупать вещи в магазине.

Нефункциональные требования к игровому приложению:

- 1) игра должна работать на операционной системе Windows не ниже 10;
- 2) игра должна быть создана при помощи игрового движка Ren'Py;
- 3) игра должна быть написана с использованием языка Python, которые использует Ren'Py.

### **2.2. Диаграмма вариантов использования**

Для проектирования системы был использован язык графического описания UML [18]. В соответствии с требованиями была построена диаграмма вариантов использования, данная диаграмма представлена на ри-

сунке 5. Диаграмма отражает модель взаимодействия актера «Игрок» с разрабатываемым игровым проектом. С системой взаимодействует только один основной актер – «Игрок», использующий игровой проект.

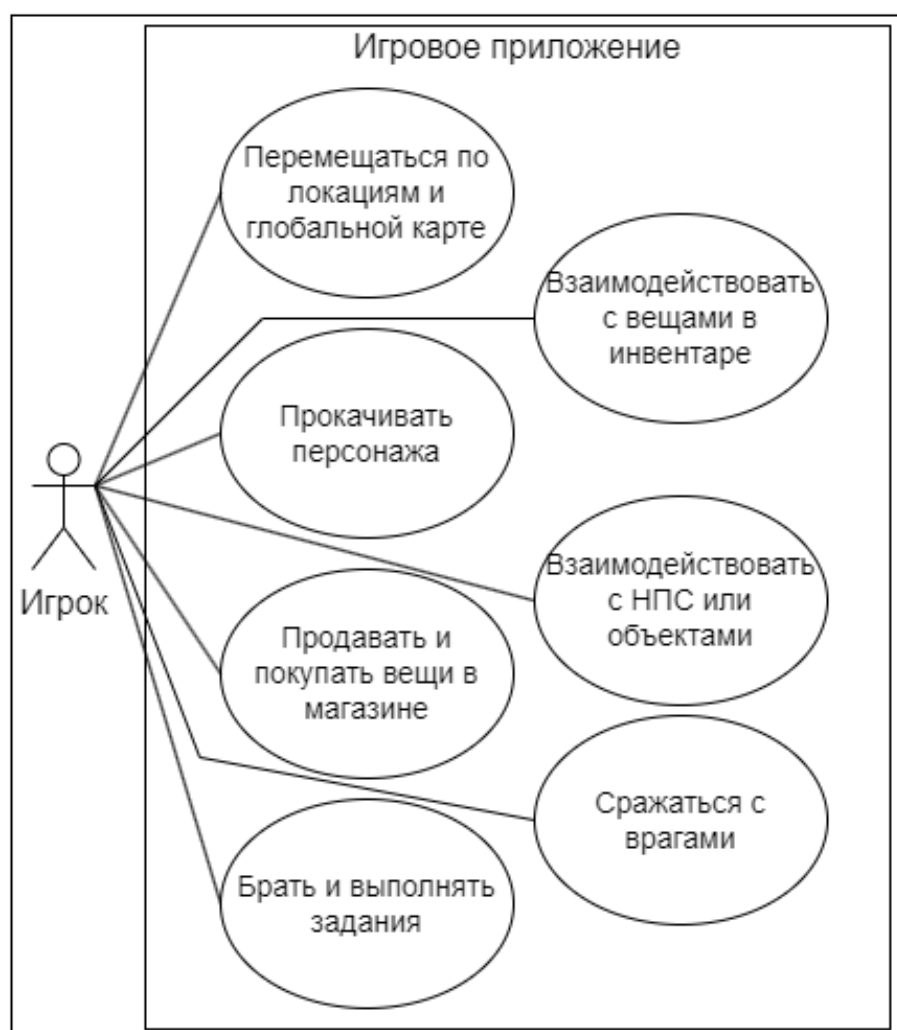


Рисунок 5 – Диаграмма вариантов использования

На самой диаграмме представлен один актер, он же игрок, а также расположены 7 прецедентов.

Более подробная спецификация основных вариантов использования представлены в таблицах 1–7 в приложении А. Спецификация вариантов использования.

### **Вывод по второму разделу**

Были описаны функциональные и нефункциональные требования к системе и построена диаграмма вариантов использования.

### 3. АРХИТЕКТУРА СИСТЕМЫ

#### 3.1. Общее описание архитектуры системы

Архитектурно система состоит из сцен главного меню и игровых уровней/локаций. При запуске игры, по умолчанию запускается главное меню игры, оно уже готовое и настроенное автоматически благодаря Ren'Py.

В меню находятся уже работающие кнопки. «Начать» – запускает сцену самой игры (запуск самой игры). «Загрузить» – загружать ранее сохраненные сцены игры, на которой остановился игрок. «Настройки» – позволяет настраивать игру. «Об игре» – выводит информацию об игре, кто разработал, версия и прочее. «Помощь» – выводит информацию по управлению. «Выход» – закрытие игрового приложения.

Меню так же можно настроить по своему усмотрению и добавить или убрать кнопки или изображения.

Файловая система игры состоит из следующих элементов:

- 1) scripts – содержит в себе скрипты, для работы приложения;
- 2) images – содержит в себе изображения, используемые в игре;
- 3) saves – содержит сохранения пользователя;
- 4) audio – содержит в себе аудиофайлы, используемые в игре;
- 5) gui – содержит элементы интерфейса игры;
- 6) tl – хранит файлы для перевода на другие языки;
- 7) cache – используется для хранения скомпилированных файлов сценариев Ren'Py.

Когда запускается проект, Ren'Py сканирует каталоги game и tl, а затем компилирует скрипты и сохраняет их в каталоге cache, чтобы ускорить процесс запуска в будущем.

С двумя последними элементами во время разработки не было никаких прямых взаимодействий.



### 3.2. Диаграмма компонентов и их описание

Диаграмма компонентов [19], описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный и исполняемый код. Пример диаграммы изображен на рисунке 6.

Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

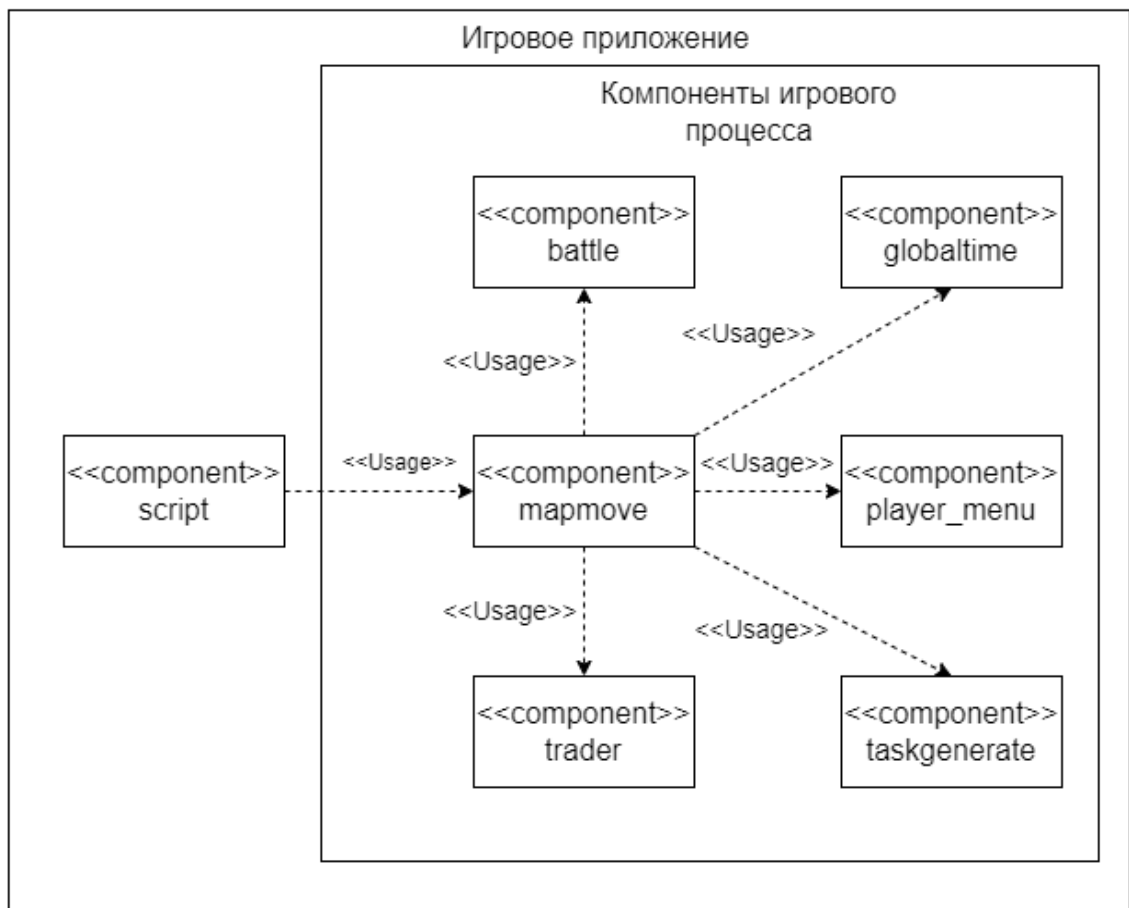


Рисунок 6 – Диаграмма компонентов

Во время разработки были использованы следующие компоненты.

1. Компонент «mapmove» – содержит код который, отвечает за перемещение игрока между локациями и перемещениями по глобальной карте.

2. Компонент «player\_menu» – содержит код в котором прописана логика пользовательского интерфейса. Показ инвентаря, характеристик, заданий.

3. Компонент «taskgenerate» – содержит код для показа и принятия заданий.

4. Компонент «trader» – содержит код отвечающий за возможность торговли. Позволяет игроку продавать и покупать вещи у торговца за внутреигровую валюту.

5. Компонент «battle» – содержит код боя. Когда происходят сражения, здесь обрабатывается процесс боя. Как атакует враг или игрок и возможность побега из боя.

6. Компонент «globaltime» – содержит код для вычисления даты в игровом приложении.

7. Компонент «script» – это основной файл сценария игры, который запускается в первую очередь при запуске игры. Используется для перехода в другие компоненты и может хранить данные которые так же используются в других компонентах.

### **Вывод по третьему разделу**

Была рассмотрена файловая система игры, построена диаграмма компонентов и их взаимодействие между друг другом. Созданы и использованы свои компоненты для работы игрового приложения.

## 4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ СИСТЕМЫ

### 4.1. Реализация компонентов и интерфейсов системы

Для получения примерного представления о том, как работает код в Ren'Py, следует рассмотреть работу некоторых компонентов.

Первым примером будет работа пользовательского интерфейса, содержащегося в «player\_menu».

На рисунке 7 представлен код хранения вещей.

```
default items = [  
{"id": 1,      "name": "Топор",          "quest_item": False,  
"quantity": 1, "description": "Предмет для рубки дерева",  
"cbeqiped": False, "slot": "none", "dmg": 0, "arm":0,  
"may_be_eaten": False, "may_be_drink":False, "food_restoration": 0,  
"image": "icon/axe.png",              "weight": 1.5,  
"sell_price": 1, "buy_price": 2  },  
{"id": 2,      "name": "Кирка",          "quest_item": False,  
"quantity": 1, "description": "Предмет для добычи руды",  
"cbeqiped": False, "slot": "none", "dmg": 0, "arm":0,  
"may_be_eaten": False, "may_be_drink":False, "food_restoration": 0,  
"image": "icon/pick.png",              "weight": 1.5,  
"sell_price": 1, "buy_price": 2  },  
]
```

Рисунок 7 – Код хранения вещей

Можно увидеть, что у предмета есть такие характеристики, как id, название, количество, описание, изображение, вес, цена, возможность экипировки и т.д.

После добавления предмета в инвентарь, игрок, зайдя в инвентарь, видит сам предмет, его вес и текущее количество. Пример кода отображения представлен на рисунке 8.

На рисунке 9 показан сам интерфейс и текущее его состояние. У предмета в верхнем левом углу его количество, а внизу – его вес в единичном экземпляре. Текущий и максимальный вес показаны сверху, рядом с кнопкой выхода из инвентаря. Если предмет игроку не нужен, он может от него избавиться. Перегрузка произойдет, если у игрока будет слишком много предметов. Также, если предмет можно экипировать или использовать, будет предусмотрена специальная кнопка для этих действий.

```

screen backpack_screen:
imagemap:
    idle "backpack_bg"
    if current_screen == "backpack_screen":
        imagemap:
            idle 'backpack_bg'
            if Player_weight[0] >= Player_weight[1]:
                text "Текущий вес \n{color=#910725}{size=30}%.1f / %i
{/size}{/color} перевес" %(Player_weight[0], Player_weight[1]) xpos 25
ypos 110
            else:
                text "Текущий вес \n{color=#262829}{size=30}%.1f / %i
{/size}{/color} " %(Player_weight[0], Player_weight[1]) xpos 25 ypos 110
                #text "{size=40}{color=#0f1b4d} %s {/color}{/size}" %
tt.value xpos 500 ypos 500

        imagebutton:
            idle im.Scale('icon/go_back.png', 100, 100)
            hover im.Alpha(im.Scale('icon/go_back.png', 100, 100), 0.5)
            action Hide("backpack_screen"), Hide("ac-
tion_item_menu_screen"), Jump('screen_in_city_jump_from_back') #hovered
tt.action("Назад")
            vbox xalign 0.15 yalign 0.3 spacing 5:
                for i in range(0, len(items), 8):
                    hbox:
                        for item in items[i:i+8]:
                            imagebutton:
                                idle im.Scale(item["image"], 100, 100)
                                hover im.Alpha(im.Scale(item["image"], 100,
100),0.5)
                                action SetVariable("SetVariableItemId",
item["id"]), Jump("action_item_menu_screen_label") #hovered tt.ac-
tion(SetVariableItemId)
                                text "{color=#262829}{size=24}%i {/size}{/color}"
%item['quantity'] xpos -100 ypos 0
                                text "{color=#262829}{size=24}%.1f
{/size}{/color}" %item["weight"] xpos -50 ypos 80
                                vbox xalign 0.1 yalign 0.28 spacing 5:
                                    for slot_info in display_of_equipped_items:
                                        if slot_info["item_equped_now"] == True:
                                            for item in player_equiped_items:
                                                if slot_info["slot"] == item["slot"]:
                                                    imagebutton:
                                                        idle im.Scale(item["image"], 100, 100)
xpos 1500 ypos 200
                                                        hover im.Alpha(im.Scale(item["image"],
100, 100),0.5)
                                                        action SetVariable("SetVariableItemId",
item["id"]), Jump("action_remove_equiped_item_menu_screen_label") #hov-
ered tt.action(SetVariableItemId)
                                                    else:
                                                        add im.Alpha(im.Scale(slot_info["image"], 100, 100),
0.5) xpos 1500 ypos 200
                                                        use coin_screen

```

Рисунок 8 – Показ инвентаря

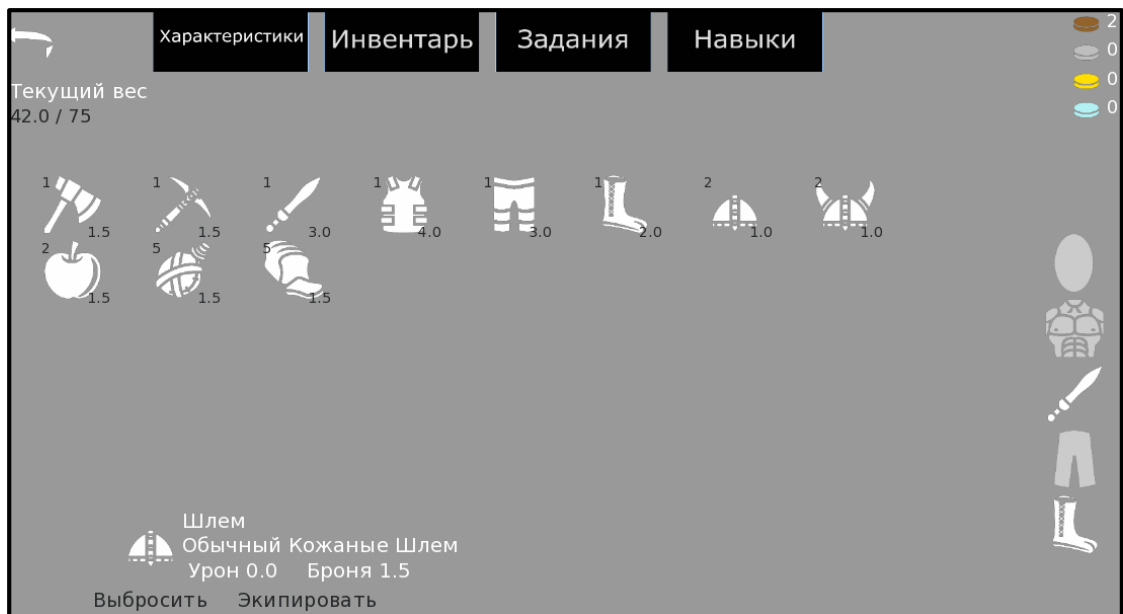


Рисунок 9 – Вид инвентаря и взаимодействие с ним

Вторым примером будет код работы магазина, который содержится в «trader».

На рисунке 10 показан способ хранения вещей у торговца.

У торговца два хранилища. Первое служит главным хранилищем откуда торговец берет вещи во временное хранилище. Второе это временное хранилище. Временное используется для торговли. Когда в игре проходят сутки, то предметы у торговца обновляются и в магазине происходит смена товаров. А так же игрок может вернуть проданный ранее предмет если он его продал случайно в тот же день.

```
default trader_items_default = []
default trader_items = []
```

Рисунок 10 – Хранилище торговца

Для покупки и продажи вещей был написан следующий код, который представлен на рисунке 11.

```

label action_buy_item_menu:
    show screen shop
    python:
        for item in trader_items:
            if item["id"] == SetVariableItemId:
                if item["buy_price"] > Player_money[0]:
                    renpy.jump(label='if_less_money')
                item_exists = False
                for inv_item in items:
                    if inv_item["id"] == SetVariableItemId:
                        # Увеличиваем количество предмета на 1
                        inv_item["quantity"] += 1
                        item_exists = True
                        break
                if not item_exists:
                    new_item = {
                        "id": item["id"], "name": item["name"],
                        "quest_item": item["quest_item"],
                        "quantity": 1,
                        "description": item["description"],
                        "cbeqiped": item["cbeqiped"],
                        "slot": item["slot"],
                        "dmg": item["dmg"], "arm": item["arm"],
                        "may_be_eaten": item["may_be_eaten"],
                        "may_be_drink": item["may_be_drink"],
                        "food_restoration": item["food_restoration"],
                        "image": item["image"], "weight": item["weight"],
                        "sell_price": item["sell_price"],
                        "buy_price": item["buy_price"]
                    }
                    items.append(new_item)
                if item["quantity"] > 1:
                    item["quantity"] -= 1
                else:
                    trader_items.remove(item)
                player_passive_skills[1]["current_level_point"] += 0.5
                Player_stats[1] += 0.1
                Player_money[0] -= item["buy_price"]
        call update_skills_label
        call update_inventory_weight
        call screen shop

```

Рисунок 11 – Пример покупки и продажи

В данном коде показана логика торговли, а именно покупка вещей.

Код для покупки и продажи похож только отличие в том что при покупке игрок добавляет предмет к себе, а при продаже продавец к себе.

При покупке игра смотрит на предмет, если такой предмет уже есть, то он увеличивает количество этого предмета в инвентаре в размере 1. Если такого предмета нет, то игра создает такой предмет. И в конце покупки снимаются деньги. Если же у игрока денег нет или просто недостаточно для свершения покупки, то сделки не будет.

На рисунке 12 показано как выглядит интерфейс торговли на данный момент.



Рисунок 12 – Интерфейс торговли

На интерфейсе показаны 2 инвентаря. Слева инвентарь игрока, а справа торговца. Когда игрок выбирает предмет то снизу показывается этот предмет, его цена и возможность покупки или продажи.

#### 4.2. Тестирование игрового приложения

В ходе разработки, были проведены тесты для проверки работоспособности игрового приложения. В таблице 1 представлены результаты тестов. Данные тесты проверяют работоспособность основных механик игры.

Таблица 1 – Функциональное тестирование игрового приложения

№	Название теста	Действия	Ожидаемый результат	Тест пройден?
1	Переход между локациями	Игрок нажимает на место куда он хочет перейти	Игрок попадает в другую локация	Да
2	Передвижение по глобальной карте	Игрок нажимает на клетку на которую он хочет переместиться	Игрок перемещается на соседнюю клетку	Да

№	Название теста	Действия	Ожидаемый результат	Тест пройден?
3	Открытие инвентаря	Игрок нажимает на кнопку в виде рюкзака и открывает инвентарь	Открывается инвентарь и показывается что у игрока есть и в какие вкладки он может еще перейти	Да
4	Переход между вкладками в инвентаре	Игрок нажимает на другие вкладки для перехода между ними	Игрок может переходить между вкладками	Да
5	Прокачка навыков	При получении очка навыков, игрок может прокачать любой свой навык на выбор	Игрок прокачал навык	Да
6	Торговля, продажа	Игрок продает предмет в магазине	Игрок продает товар, получает деньги и этот товар попадает к торговцу	Да
7	Торговля, покупка	Игрок покупает предмет в магазине	Игрок покупает предмет с него снимаются деньги и этот предмет появляется в инвентаре	Да
8	Выполнение заданий	Игрок принимает задание	Игрок выполнил, сдал задание и получил за это награду	Да
9	Взаимодействие. Сбор предметов	Игрок может взаимодействовать с объектами	Игрок собрал предмет при нажатии на него	Да
10	Сражение с противниками	Игрок во время заданий или путешествия может наткнуться на врага и начнется бой	Игрок или побеждает или проигрывает	Да
11	Взаимодействовать с NPC	Игрок нажимает на силуэт NPC для взаимодействия с ним.	Игрок взаимодействовал с NPC. Он с ним пообщался или поторговал.	Да

### Вывод по четвертому разделу

Был представлен код и пример его работы, а также проведены тесты для проверки работоспособности приложения.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были сделаны следующие шаги для разработки и реализации игрового приложения в жанре RPG на платформе Ren'Py.

1. Проведен обзор существующих игр в схожих жанров или тематике. Выбрано в каком направлении будет реализовано приложение.

2. Были проанализированы игры и выбраны идеи которые будут присутствовать в самой игре. Произведен анализ предметной области для понимания что будет сделано на какой платформе и что будет реализовано.

3. Реализована игра до состояния Демо версии. Были описаны функциональные и не функциональные требования к системе. Как система будет реагировать на действия игрока и что будет происходить при помощи диаграммы вариантов использования. Была рассмотрена архитектура системы. Были созданы свои компоненты для реализации приложения и построена диаграмма компонентов для наглядного понимания как компоненты работают и взаимодействуют друг с другом. Была проведена реализация компонентов, небольшая демонстрация их на примере небольшого представленного участка кода и примерный результат этих действий.

4. Было проведено тестирование. После реализации игрового приложения было проведено тестирование на поиск и ликвидации возможных ошибок при выполнении разных действий игроком.

В будущем планируется и дальше дорабатывать и видоизменять данное приложение с возможностью выпуска в массовое пользование.

## ЛИТЕРАТУРА

1. Определение «Визуальная новелла». [Электронный ресурс] URL: [https://stopgame.ru/blogs/topic/58568/vizualnye\\_novelly\\_chno\\_eto\\_i\\_s\\_chem\\_e\\_dyat](https://stopgame.ru/blogs/topic/58568/vizualnye_novelly_chno_eto_i_s_chem_e_dyat) (дата обращения: 01.02.2024 г.).
2. Определение «Ассет». [Электронный ресурс] URL: <https://unity.com/ru/pages/introduction-to-asset-store> (дата обращения: 01.02.2024 г.).
3. Определение «NPC». [Электронный ресурс] URL: <https://tesonline.org/NPC> (дата обращения: 01.02.2024 г.).
4. Определение «Script». [Электронный ресурс] URL: [https://www.nic.ru/help/skripty-chno-eto-vidy-i-kak-polzovatbsya\\_11103.html](https://www.nic.ru/help/skripty-chno-eto-vidy-i-kak-polzovatbsya_11103.html) (дата обращения: 01.02.2024 г.).
5. Определение «Геймплей». [Электронный ресурс] URL: <https://plarium.com/ru/glossary/gameplay-meaning/> (дата обращения: 01.02.2024 г.).
6. Определение «Фентези». [Электронный ресурс] URL: <https://ast.ru/news/razbor-zhanra-fentezi/> (дата обращения: 01.02.2024 г.).
7. Определение «Темное фентези». [Электронный ресурс] URL: <https://zhiltsova.su/zhanr-temnoe-fentezi/> (дата обращения: 01.02.2024 г.).
8. Определение для «RPG». [Электронный ресурс] URL: <https://dtf.ru/games/1380701-chno-zhe-takoe-rpg> (дата обращения: 01.02.2024 г.).
9. Определение для «AAA игр». [Электронный ресурс] URL: [https://stopgame.ru/show/138007/chno\\_takoe\\_aaa\\_igra\\_chno\\_znachat\\_eti\\_bukvy](https://stopgame.ru/show/138007/chno_takoe_aaa_igra_chno_znachat_eti_bukvy) (дата обращения: 01.02.2024 г.).
10. Сайт движка «Ren'Py». [Электронный ресурс] URL: <https://www.renpy.org> (дата обращения: 01.02.2024 г.).
11. Документация для «Ren'Py». [Электронный ресурс] URL: <https://www.renpy.org/doc/html/index.html#> (дата обращения: 01.02.2024 г.).

12. Краткая документация на русском для «Ren'Py». [Электронный ресурс] URL: [http://ru.renpyedia.shoutwiki.com/wiki/Renpy\\_для\\_чайников](http://ru.renpyedia.shoutwiki.com/wiki/Renpy_для_чайников). (дата обращения: 01.02.2024 г.).
13. Сайт игры «Skyrim». [Электронный ресурс] URL: <https://elderscrolls.bethesda.net/en> (дата обращения: 01.02.2024 г.).
14. Сайт игры «Witcher 3». [Электронный ресурс] URL: <https://www.thewitcher.com/us/ru/witcher3#dlc> (дата обращения: 01.02.2024 г.).
15. Сайт игры «Vampire life». [Электронный ресурс] URL: <https://v1vampirelife.wordpress.com/основная-информация/> (дата обращения: 01.02.2024 г.).
16. Сайт на игру «Heads Will Roll». [Электронный ресурс] URL: [https://store.steampowered.com/app/1525420/Heads\\_Will\\_Roll/](https://store.steampowered.com/app/1525420/Heads_Will_Roll/) (дата обращения: 01.02.2024 г.).
17. Сайт на игру «Koikatsu Party». [Электронный ресурс] URL: [https://store.steampowered.com/app/1073440/\\_Koikatsu\\_Party/](https://store.steampowered.com/app/1073440/_Koikatsu_Party/) (дата обращения: 01.02.2024 г.).
18. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно–ориентированный анализ и проектирование, 2е издание. – Пер. с англ. – СПб: Символ Плюс, 2007. – С. 89–138.
19. Объяснение про диаграммы компонентов. [Электронный ресурс] URL: <https://skarlupka.ru/articles.php?id=72> (дата обращения: 01.02.2024 г.).

## ПРИЛОЖЕНИЯ

### Приложение А. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) системы приведена в таблицах 1–7.

Таблица 1 – Спецификация ВИ «Перемещаться по локациям и глобальной карте»

Прецедент: Перемещаться по локациям и глобальной карте
ID: 1
Краткое описание: Позволяет игроку перемещаться по локациям в игре
Главные актеры: Игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок дошел до момента, когда он может путешествовать.
Основной поток: Вариант использования начинается, когда игрок перемещается по локациям или по глобальной карте.
Постусловия: Игрок переместился в выбранную локацию или по глобальной карте.

Таблица 2 – Спецификация ВИ «Прокачивать персонажа»

Прецедент: Прокачивать персонажа
ID: 2
Краткое описание: Игрок выбирает навыки для прокачки
Главные актеры: Игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок получил навык для прокачки.
Основной поток: Вариант использования начинается, когда игрок тратит полученное очко навыка для прокачки навыков.
Постусловия: Игрок прокачал навык.

Таблица 3 – Спецификация ВИ «Продавать и покупать вещи в магазине»

Прецедент: Продавать и покупать вещи в магазине
ID: 3
Краткое описание: Игрок продает и покупает вещи в магазине
Главные актеры: Игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок зашел в магазин.
Основной поток: Вариант использования начинается, когда игрок продает или покупает вещи у торговца.
Постусловия: Игрок купил или продал вещи.

Таблица 4 – Спецификация ВИ «Брать и выполнять задания»

Прецедент: Брать и выполнять задания
ID: 4
Краткое описание: Игрок может брать задания и выполнять их
Главные актеры: Игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок зашел в гильдию где берут задания.
Основной поток: Вариант использования начинается, когда игрок берет задания и пока у игрока активно задание, он может его выполнить и сдать.
Постусловия: Игрок выполнил задание и получил награду.

Таблица 5 – Спецификация ВИ «Взаимодействовать с вещами в инвентаре»

Прецедент: Взаимодействовать с вещами в инвентаре
ID: 5
Краткое описание: Позволяет игроку взаимодействовать с вещами в инвентаре
Главные актеры: Игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок открыл свой инвентарь.
Основной поток: Вариант использования начинается, когда игрок открывает инвентарь и может выкидывать, использовать или одевать предметы в инвентаре.
Постусловия: Игрок надел вещь или выбросил или использовал ее.

Таблица 6 – Спецификация ВИ «Взаимодействовать с NPC или объектами»

Прецедент: Взаимодействовать с NPC или объектами
ID: 6
Краткое описание: Позволяет игроку взаимодействовать с NPC или объектами
Главные актеры: игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок наткнулся на объект или встретился с NPC
Основной поток: Вариант использования начинается, когда игрок нажимает на NPC или объект для взаимодействия с ним.
Постусловия: Игрок взаимодействовал с NPC или объектом.

Таблица 7 – Спецификация ВИ «Сражаться с врагами»

Прецедент: Сражаться с врагами
ID: 7
Краткое описание: позволяет игроку сражаться с врагами
Главные актеры: игрок
Предусловия: 1. Запуск игрового приложения 2. Игрок во время вылазок или путешествуя по миру наткнулся на врага
Основной поток: Вариант использования начинается, когда игрок встречается в битву в которой он может победить или проиграть
Постусловия: Если игрок победил, то он получает награду и идет дальше по своим делам. Если игрок проиграл, то игра для него окончена.

## Приложение Б. Скриншоты итоговой версии игры

На рисунках 1–6 представлены скриншоты из самой игры.

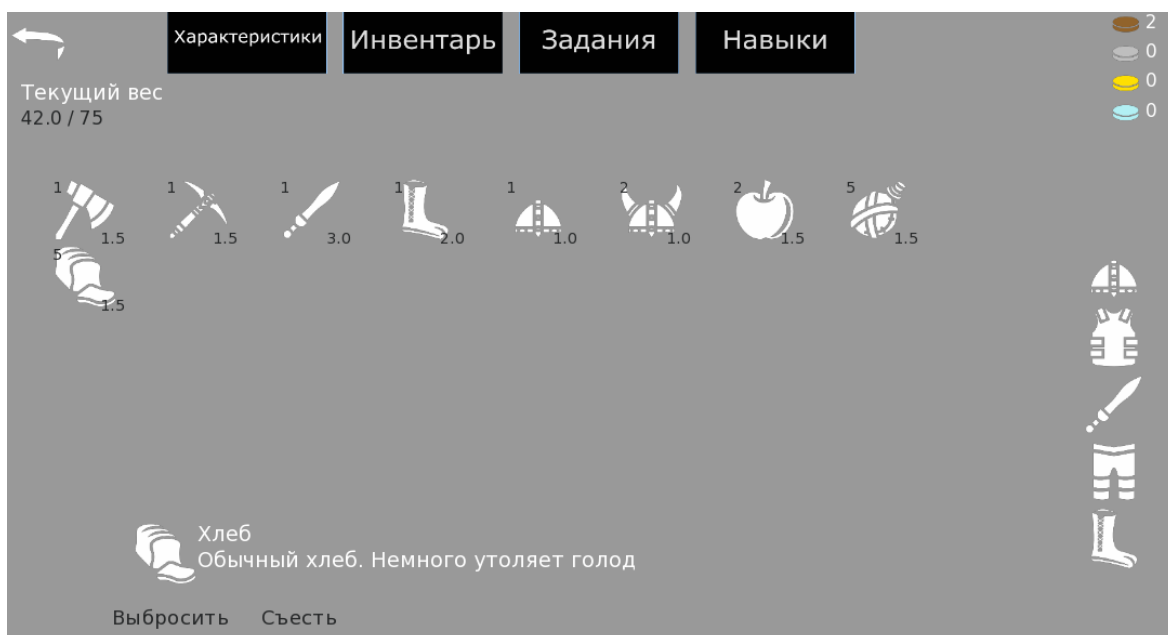


Рисунок 1 – Инвентарь

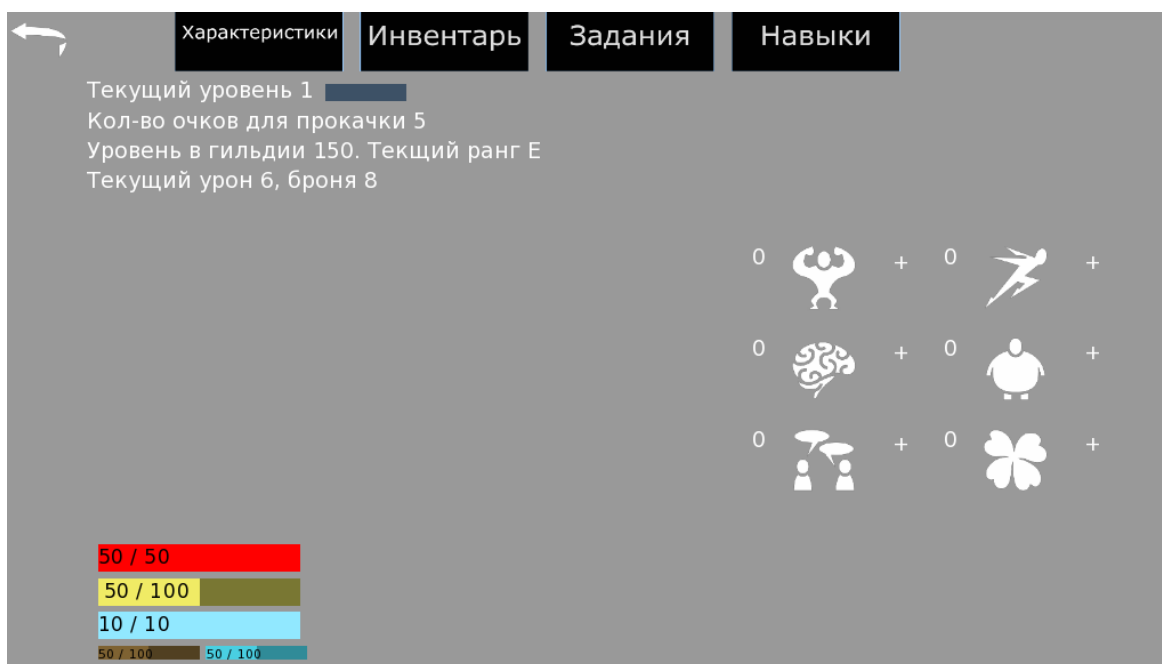


Рисунок 2 – Характеристики игрока

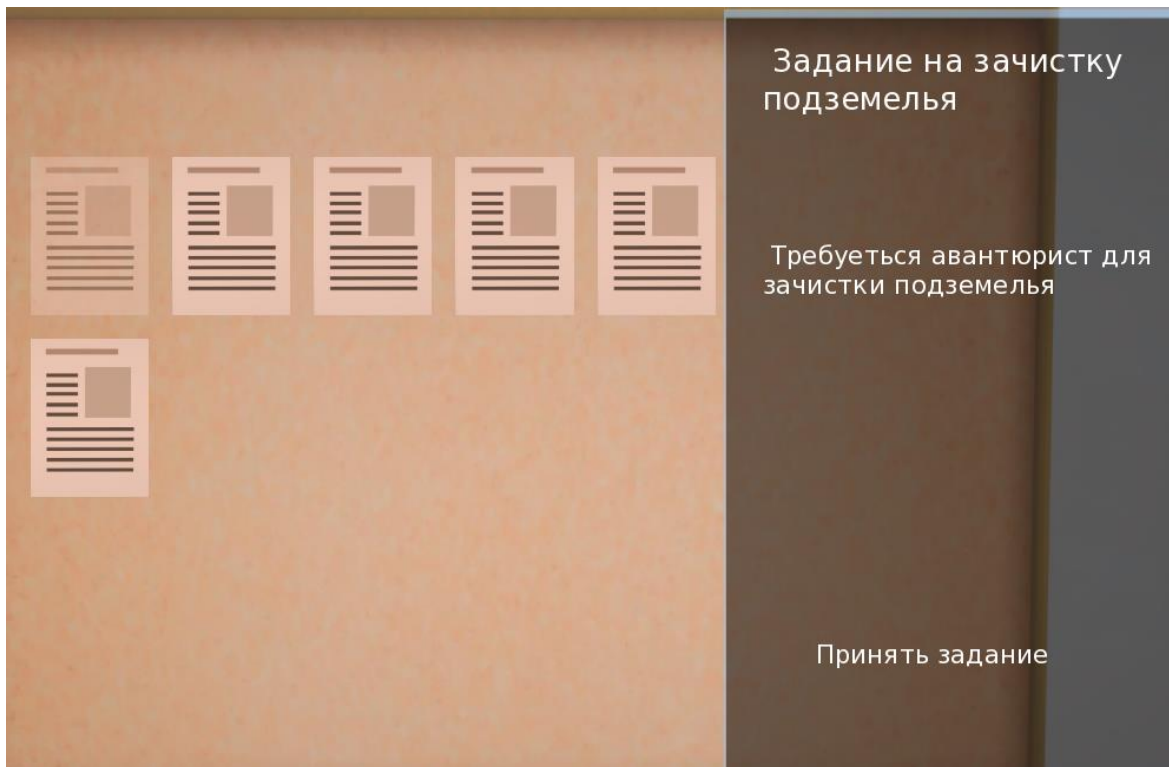


Рисунок 3 – Стенд с заданиями



Рисунок 4 – Глобальная карта мира





Рисунок 5 – Главная площадь



Рисунок 6 – Пример сражений