

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 игры в жанре «Визуальная новелла»
на платформе Ren'Py**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.04.2024.308-366.ВКР

Научный руководитель,
доцент кафедры СП, к.т.н.
_____ М.В. Сухов

Автор работы,
студент группы КЭ-404
_____ Л.А. Сухоруков

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-404
Сухорукову Леониду Андреевичу,
обучающемуся по направлению
09.03.04 «Программная инженерия»

- 1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)
Разработка игры в жанре «Визуальная новелла» на платформе Ren'Py.
- 2. Срок сдачи студентом законченной работы:** 03.06.2024 г.
- 3. Исходные данные к работе**
 - 3.1. Руководство по движку Ren'Py. [Электронный ресурс]
URL: <https://www.renpy.org/doc/html/> (дата обращения: 09.02.2024 г.).
 - 3.2. Руководство по языку программирования Python. [Электронный ресурс]
URL: <https://metanit.com/python/tutorial/> (дата обращения: 09.02.2024 г.).
 - 3.3. Шелл, Д. Геймдизайн. Как создать игру, в которую будут играть все. / Д. Шелл; переводчик А. Лысенко. // Москва.: Альпина Паблицер, 2019. – 640 с.
 - 3.4. Андрианова Н., Яковлева С. Как создавать истории. Основы игровой сценаристики и нарративного дизайна за 12 шагов. – Москва: Бомбора, 2023. – 298 с.
- 4. Перечень подлежащих разработке вопросов**
 - 4.1. Выполнить анализ предметной области.

4.2. Провести анализ методов разработки компьютерных игр и игровых движков.

4.3. Создать концепцию и описать сюжетную линию игры.

4.4. Разработать и реализовать компьютерную игру.

4.5. Осуществить тестирование разработанного приложения.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
доцент кафедры СП, к.т.н.

М.В. Сухов

Задание принял к исполнению

Л.А. Сухоруков

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Описание предметной области проекта	7
1.2. Обзор аналогичных проектов	10
1.3. Анализ существующих решений для реализации проекта.....	12
2. КОНЦЕПЦИЯ ИГРЫ И СЮЖЕТНАЯ ЛИНИЯ.....	15
2.1. Основная концепция игры	15
2.2. Сюжетная линия и ее развитие.....	16
3. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ.....	18
3.1. Функциональные требования к проектируемой системе	18
3.2. Диаграмма вариантов использования.....	19
4. АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ	22
4.1. Компоненты приложения.....	22
4.2. Проектирование интерфейса	23
5. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ СИСТЕМЫ	28
5.1. Реализация компонентов системы	28
5.2. Реализация интерфейса	31
5.3. Тестирование системы.....	35
ЗАКЛЮЧЕНИЕ	38
ЛИТЕРАТУРА.....	39
ПРИЛОЖЕНИЯ.....	41
Приложение А. Спецификация вариантов использования.....	41
Приложение Б. Листинг мини-игры	45

ВВЕДЕНИЕ

Актуальность проблемы

В современном мире практически не осталось людей, которые никогда не играли в видеоигры, будь то сложная компьютерная стратегия или простейший «три в ряд» для телефонов. Игры постепенно проникают во многие сферы нашей жизни. В такое время как никогда актуальна сфера игровой индустрии. Разработчики соревнуются в создании игры, которая сможет завоевать как можно большую аудиторию. Даже не имея миллионных бюджетов, люди могут создавать чудесные игры, покоряющие сердца игроков по всему миру, в особенности в жанре визуальная новелла. Данный жанр игр позволяет рассказать игрокам историю и дать им возможность погрузиться в этот мир от лица главного героя. Актуальность сферы разработки игр сложно переоценить, игровая индустрия ежегодно приносит миллиардные доходы разработчикам и издателям. Данный проект дает отличную возможность для оценки своих навыков программиста и геймдизайнера.

Постановка цели и задач

Основной целью выпускной квалификационной работы является разработка компьютерной игры в жанре «визуальная новелла» на платформе Ren'Py.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выполнить анализ предметной области;
- 2) изучить особенности и возможности движка Ren'Py;
- 3) написать сценарий и создать визуальное оформление игры;
- 4) спроектировать архитектуру приложения, включая проектирование файловой системы и интерфейса;
- 5) произвести программную реализацию проекта;
- 6) протестировать реализованное приложение.

Структура и содержание работы

Работа состоит из введения, пяти разделов, заключения, списка литературы и двух приложений. Объем работы составляет 48 страниц, объем списка литературы – 23 источника.

В первом разделе «Анализ предметной области» приведен анализ аналогичных проектов и анализ существующих решений для реализации проекта.

Второй раздел «Концепция игры и сюжетная линия» содержит описание сюжета и хода повествования.

Третий раздел «Анализ требований к программной системе» содержит функциональные и нефункциональные требования к проектируемой системе, диаграмму вариантов использования и спецификацию основных вариантов использования.

В четвертом разделе «Архитектура программной системы» описаны компоненты проектируемого приложения, а также представлено проектирование интерфейса.

В пятом разделе «Реализация и тестирование системы» приведены листинги реализации компонентов системы, а также структура файловой системы, скриншоты пользовательского интерфейса и результаты функционального тестирования.

В приложении А содержится спецификация вариантов использования готового игрового приложения.

В приложении Б содержится полный листинг кода мини-игры.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области проекта

Конечной целью данной работы является разработка компьютерной игры в жанре «Визуальная новелла». Этот жанр компьютерных игр позволяет игроку погрузиться в повествуемую историю от первого лица, ассоциируя себя с главным героем. Геймплей сопровождается визуальным оформлением, текстовым повествованием, а также возможными дополнительными механиками, такими как взаимодействие с объектами и другие.

Основной упор в данных проектах делается на визуальную и сюжетную составляющую, чтобы, прежде всего, как-то выделить свой продукт среди остальных. Ввиду такой особенности, для разработки и создания подобного проекта необходимы навыки дизайна, рисования и продюсирования, так как самому процессу кодирования и сборке уделяется чуть меньше времени.

Изначально жанр визуальная новелла включал в себя идею симулятора свиданий, где игроки непринужденно общались с персонажами и наслаждались спокойным повествованием. Но с течением времени разработчикам захотелось привнести в свой личный проект какую-то изюминку или отличие, которое сделало бы их игру более привлекательной на фоне конкурентов. Для начала стоит рассмотреть различные аспекты, которые, так или иначе, влияют на весь проект.

Идея

Идея [1] – является отправной точкой ко всей разработке, так как от концепции и истории зависит то, как и какие элементы будут расположены на экране, и как это будет вписываться в игру.

В большинстве современных (и не только) визуальных новелл присутствует механика «судьбы», которая позволяет игроку своим решением или действием кардинально повлиять на происходящее в дальнейшем, либо же мгновенно повлиять на события истории. Стоит понимать, что в зависимости от конкретного проекта, данная механика может быть как осно-

вополагающей, так и просто существовать для иллюзии выбора когда не важно, что выберет или сделает игрок, события все равно будут идти по одному и тому же сценарию. К примеру, в визуальной новелле «Fate: Stay Night» [2] (рисунок 1) существует такая иллюзия выбора. Нам дают возможность «решить» судьбу главного героя и событий вокруг него, но, сделав неправильный выбор, игрок сразу же заканчивает прохождение с плохой концовкой, где говорится, что он сделал неправильный выбор (рисунок 2). В выпускной квалификационной работе за основу будет взята именно эта концепция.



Рисунок 1 – Скриншот из визуальной новеллы «Fate: Stay Night»



Рисунок 2 – Пример неправильно сделанного выбора в «Fate: Stay Night»

Текст

Текст [1] – это основной способ передачи информации игроку, подаваемый посредством диалогов между персонажами. Из него узнается вся информация о героях, окружающем мире и сеттинге.

Сеттинг

Сеттинг [1] (от английского setting, дословно окружение) – среда, в которой происходят основные действия. Так же как и с литературными жанрами, визуальные новеллы могут быть совершенно разными по общему настроению и повествованию. Начиная от повседневных школьных дней и заканчивая космическими приключениями по галактике.

Персонаж

Персонаж [1] – один из главных элементов всей игры. Зачастую именно от его лица ведется повествование, от которого напрямую зависит восприятие истории игроком. Возможно и такое, что роль главного героя будет отводиться самому игроку, либо же, как зачастую происходит, на месте главного героя стоит уже прописанный персонаж.

Визуальный стиль

Визуальный стиль [1] – это то, что зачастую и придает всему проекту какой-то особый шарм. Людям интереснее играть в то, что не только ощущается интересно, но и выглядит соответствующе своему жанру и сеттингу. Общий визуальный стиль большинства визуальных новелл происходит из японской мультипликации.

Механика

Основной механикой [1] любой визуальной новеллы является система диалогов/выборов. Разработчики могут старательно передавать описание деталей объектов через текст либо же сделать упор на визуальную составляющую, выделив текст в основном на диалоги. Также могут присутствовать и другие расширяющие игровой процесс механики.

1.2. Обзор аналогичных проектов

При рассмотрении аналогов стоит учитывать, что каждая визуальная новелла по-своему уникальна, будь то выдающийся сюжет, красочная картинка или особенная механика. Все игры данного жанра невозможно сравнивать на равных условиях. Зачастую успех того или иного проекта зависит от наполнения игры и интересов сообщества. Поэтому стоит рассмотреть несколько популярных проектов, их плюсы и минусы.

«Бесконечное лето»

«Бесконечное лето» [3] (рисунок 3) – игра в жанре визуальная новелла, разработанная российской командой Soviet Games в 2013 году. Стала очень популярной, прежде всего у СНГ сообщества благодаря сеттингу Советского Союза, простому и приятному повествованию, а также идее симулятора свиданий в основе.

Из плюсов можно выделить:

- 1) сеттинг, близкий по духу к СНГ сообществу;
- 2) игровой процесс, не отличающийся высокой сложностью.

Из минусов можно выделить:

- 1) сюжет, который слишком прост в понимании;
- 2) систему выбора диалогов, в которой сложно разобраться.



Рисунок 3 – Скриншот из визуальной новеллы «Бесконечное лето»

«Врата Штейна»

«Врата Штейна» [4] (рисунок 4) – визуальная новелла за авторством японских компаний 5pb. и Nitroplus, выпущенная в 2009 году. Снискала большую популярность как среди японской, так и мировой аудитории. В основе игры лежат sci-fi идеи о перемещении во времени и мультивселенных.

Из плюсов можно выделить:

- 1) очень интересный сюжет;
- 2) реализм;
- 3) большой выбор концовок.

Из минусов можно выделить:

- 1) затянутое повествование;
- 2) сложность игрового процесса;
- 3) странный визуальный стиль.



Рисунок 4 – Скриншот из визуальной новеллы «Врата Штейна»

«Данганронпа»

«Данганронпа» [5] (рисунок 5) – японская франшиза визуальных новелл, разработанная студиями Spike Chunsoft, Spike, Abstraction Games V.V. Стала популярна за счет интересного сюжета и геймплея с элемента-

ми RPG. В основе сюжета всей франшизы берется идея противостояния надежды и отчаяния, но в концепции отдельных игр происходит смертельная игра между персонажами.

Из плюсов можно выделить:

- 1) интересный игровой процесс;
- 2) интригующий сюжет.

Из минусов можно выделить:

- 1) обилие жестокости;
- 2) сложность восприятия сюжета.

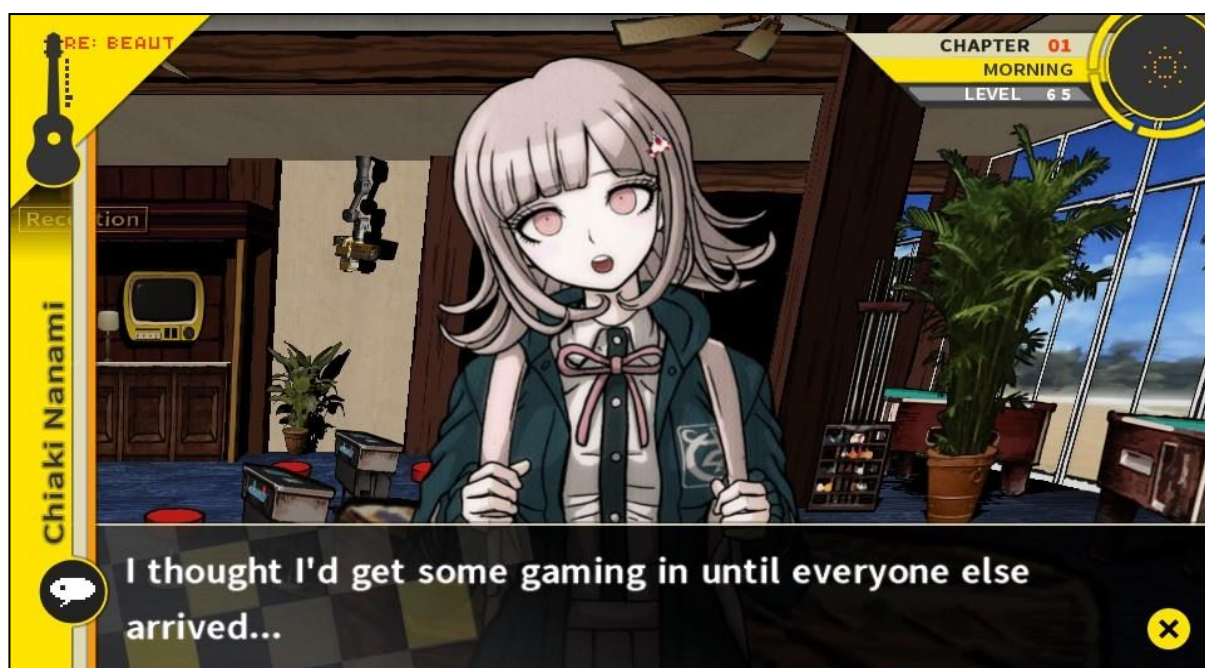


Рисунок 5 – Скриншот из визуальной новеллы «Данганронпа»

1.3. Анализ существующих решений для реализации проекта

Теперь, когда есть понимание того, что лежит в концептуальной основе визуальной новеллы, следует разобраться, каким образом проект данного типа может быть успешно разработан и на какой основе.

Игровой движок – базовое программное обеспечение, созданное для разработки и создания видеоигр. Необходим движок, на котором существует возможность разработки визуальной новеллы. Стоит рассмотреть все возможные варианты.

Unity

Unity [6] – универсальный движок, на котором можно создать любую игру, в том числе визуальную новеллу. Является бесплатным и при этом имеет самый обширный функционал, однако требует значительной степени познания ввиду собственной сложности.

К достоинствам Unity относятся:

- 1) универсальность;
- 2) продвинутость в сравнении с другими движками.

К недостаткам Unity относятся:

- 1) длительный срок реализации проектов;
- 2) сложность в освоении;
- 3) требовательность к мощности компьютера.

RPG Maker

RPG Maker [7] – изначально движок для создания игр в жанре JRPG, но также подходит для разработки визуальных новелл. Является платным, очень прост в освоении, так как почти не требует знаний программирования, но при этом имеет очень ограниченный функционал. Существует возможность дополнения движка с помощью расширений, написанных на JavaScript.

К достоинствам RPG Maker относятся:

- 1) простота освоения;
- 2) отсутствие необходимости знания программирования.

К недостаткам RPG Maker относятся:

- 1) необходимость приобретения движка;
- 2) ограниченность функционала;
- 3) сложность освоения языка JavaScript.

TyranoBuilder

TyranoBuilder [8] – движок для создания визуальных новелл без программирования. Также как и RPG Maker не бесплатный, прост в освоении, но при этом имеет ограниченный функционал.

К достоинствам TyranoBuilder относятся:

- 1) простота освоения;
- 2) отсутствие необходимости знания программирования.

К недостаткам TyranoBuilder относятся:

- 1) необходимость приобретения движка;
- 2) ограниченность функционала.

Ren'Py

Ren'Py [9–12] – самый популярный движок для разработки визуальных новелл. Он бесплатный и не имеет такой высокой сложности как Unity. Вся разработка осуществляется на языке программирования Python [13–14]. Также игры, созданные на этом движке, имеют возможность портирования на платформы Android и Linux.

К достоинствам Ren'Py относятся:

- 1) бесплатность;
- 2) простота освоения;
- 3) возможность портирования на системы Android и Linux.

К недостаткам Ren'Py относятся:

- 1) ограниченность функционала;
- 2) сложность при разработке масштабных проектов.

Вывод по первому разделу

Из всего вышесказанного можно сделать определенные выводы по поводу выбора программной среды для разработки и методов ускорения разработки, учитывая сроки. За основу игры был выбран движок Ren'Py на базе языка программирования Python. В игре будут присутствовать выборы, которые могут повлиять на дальнейший ход событий сюжетной линии.

2. КОНЦЕПЦИЯ ИГРЫ И СЮЖЕТНАЯ ЛИНИЯ

Целью данного раздела является описание концепции игры и ее сюжетного повествования [17].

2.1. Основная концепция игры

За основу мира игры было взято такое направление научной фантастики как Стиппанк, вдохновленное технологиями XIX века, в первую очередь паровыми машинами. Действия игры происходят в годах схожих с 1912–1920 годами реального мира. Два государства Иприт и Болимвер существовали в мире и согласии до тех пор, пока в империи Болимвер не было обнаружено необычное растение, произрастающее на его территории. По результатам научных исследований были выявлены удивительные свойства цветка при его сжигании и испарении. Таким образом, при смешивании его экстракта с другими реагентами получившаяся жидкость при испарении обладала уникальными свойствами различной степени. Данный вид растений был впоследствии назван учеными как Вapорис Амарилло-вый, но в народе, благодаря церкви, получил название «Эдем».

Ввиду такого научного открытия и ожидаемого скачка в технологиях империи Болимвер, правительством Иприт было принято решение о начале военного наступления на земли массового произрастания Эдема в Болимвере. Болимвер, имея неоспоримое преимущество в технологиях на основе пара, одерживал сокрушительные победы на полях сражений. Ключевая битва состоялась у крепости Асловец. Армия Иприта несла крупные потери, ввиду чего незамедлительно отступила к крепости. Неожиданное появление фиолетового облака переменяло ход битвы. С невероятной скоростью пар распространился по всему полю сражения. При контакте с фиолетовым облаком солдаты тут же падали замертво.

Паника быстро охватила солдат Болимвера. Несмотря на попытки бегства, любой из солдат, вдохнувший хотя бы малую часть пара, погибал. Единственным спасением были костюмы паровой защиты имеющиеся у

паромедиков. Гарретт Морган был одним из таких. Ему и его товарищам медикам удалось обезопасить себя от фиолетового облака. Они бежали с поля боя обратно в Болимвер чтобы успеть сообщить о надвигающейся угрозе.

Солдаты Иприт также незамедлительно ретировались, пользуясь тем, что ветер дул на восток в сторону Болимвера. Они сообщили правительству о необычном фиолетовом облаке пара, что убивает все на своем пути. В сжатые сроки гениальным ученым удалось разработать паробарьер – пар высокой плотности, который не пропускал сквозь себя любые другие испарения, и успешно накрыть им столицу Иприта.

Ввиду сложившихся обстоятельств, Болимвер не имел достаточно времени для разработки похожего пара. Вся империя была окутана фиолетовым облаком, из-за этого большая часть населения Болимвера погибла.

Пользуясь моментом, армия Иприта, облаченная в парокостюмы, продолжила наступление. Империя Болимвер пала под натиском остатков армии Иприта.

Спустя 10 лет фиолетовое облако заняло большую часть атмосферы обоих государств, снизив при этом свою концентрацию. Несмотря на это, оно все также несет медленный губительный эффект на организм человека. Болимвер стал полноценной территорией Иприта. Вся ответственность за распространение фиолетового пара была наложена на ныне несуществующую империю Болимвер. Оставшиеся в живых разделились на два лагеря: одни перешли на сторону Иприта, другие же стали вольными кочевниками. Гарретт Морган, как один из выживших в войне, не примкнул к Иприту, а стал проводить свои собственные исследования пара, будучи парохимиком. С данного момента начинается игровое повествование.

2.2. Сюжетная линия и ее развитие

Основная сюжетная линия представляет собой линейное повествование истории главного героя – Гаррета Моргана. Ему на пути будут встре-

чаться различные персонажи по-разному к нему настроенные, как негативные, так положительные и нейтральные. Игроку дается возможность в определенные моменты игры сделать выбор (чаще всего данные выборы присутствуют в диалогах с персонажами), который в дальнейшем может повлиять на исход событий (пропуск части сюжета или выход на плохую концовку).

Для лучшего понимания сюжета была составлена схема основных событий игры, представленная на рисунке 6.

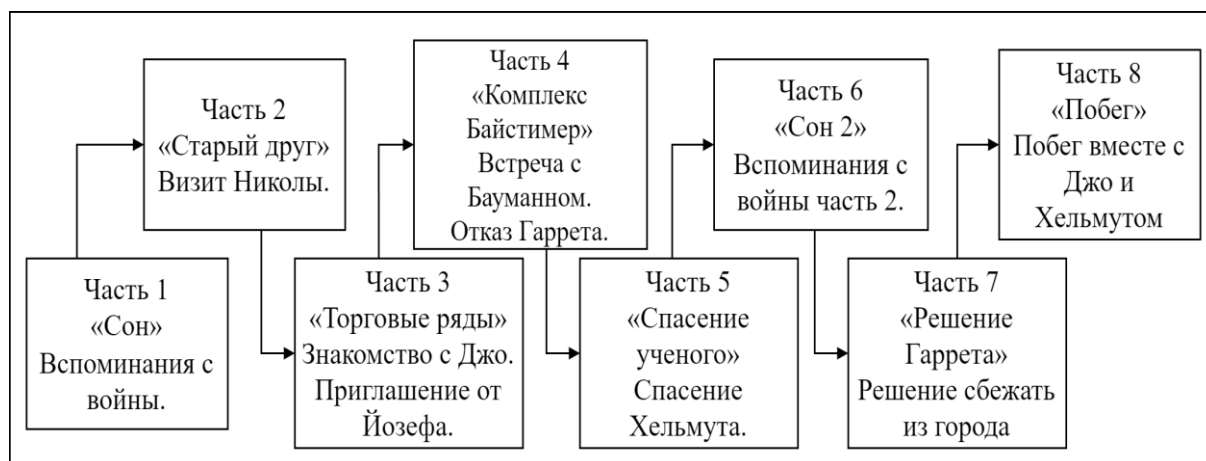


Рисунок 6 – Схема основного сюжета игры

Вывод по второму разделу

Была определена основная концепция мира игры, основанная на жанре стимпанк, а также разработана сюжетная линия игры, состоящая из 8 частей с возможностью нелинейного повествования (выход на плохую концовку).

3. АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОЙ СИСТЕМЕ

Целью данного раздела является определение основных характеристик и функциональных возможностей игрового приложения, обеспечивающих наилучший пользовательский опыт.

3.1. Функциональные требования к проектируемой системе

Проектируемое приложение должно предоставлять пользователю следующие возможности.

1. Пользователь должен иметь возможность начать новую игру, нажав соответствующую кнопку «Начать» в главном меню.

2. Пользователь должен иметь возможность загрузить раннее сохраненную игру, нажав соответствующую кнопку «Загрузить» в главном меню.

3. Пользователь должен иметь возможность узнать информацию об игре, нажав соответствующую кнопку «Об игре» в главном меню.

4. Пользователь должен иметь возможность узнать о горячих клавишах управления посредством нажатия соответствующей кнопки «Помощь» в главном меню.

5. Пользователь должен иметь возможность выхода из приложения посредством нажатия соответствующей кнопки «Выход» в главном меню.

6. Пользователь должен иметь возможность видеть подробности сохраненной игры, такие как дата сохранения и время сохранения.

7. Пользователь должен иметь возможность настраивать режим отображения (в окне/полноэкранный), менять скорость появления текста диалогов на экране, менять громкость звуков и музыки, менять режим пропуска диалогов (всего текста, после выборов или пропуск переходов) в соответствующем меню, доступ к которому осуществляется из главного меню по кнопке «Настройки».

8. В ходе игры пользователь должен иметь возможность пропускать диалоги, использовать авто-режим (автоматическое чтение диалогов), читать историю диалогов, сохранять прогресс.

9. В ходе игры пользователь должен иметь возможность попасть в мини-игру посредством диалога.

10. В ходе игры пользователь должен иметь возможность пользоваться контекстным меню, которое включает в себя все возможности главного меню: сохранение текущего прогресса, загрузка сохранения, настройки игры, выход в главное меню, выход из игры, раздел об игре, раздел помощь.

11. В ходе игры пользователь должен иметь возможность делать выборы в диалоге посредством выбора одного варианта ответа или действия из нескольких, что может повлиять на дальнейший диалог в целом.

Нефункциональные требования к проектируемой системе

Проектируемое приложение имеет следующие нефункциональные требования:

1) приложение должно поддерживать ОС Windows версий 8, 10 и 11;

2) приложение должно поддерживать управление с помощью клавиатуры, манипулятора мышь или тачпада.

3.2. Диаграмма вариантов использования

На основе всех функциональных и нефункциональных требований была разработана диаграмма вариантов использования игрового приложения, приведенная на рисунке 7. Также на основе данной диаграммы была разработана спецификация основных вариантов использования, приведенная в приложении А.

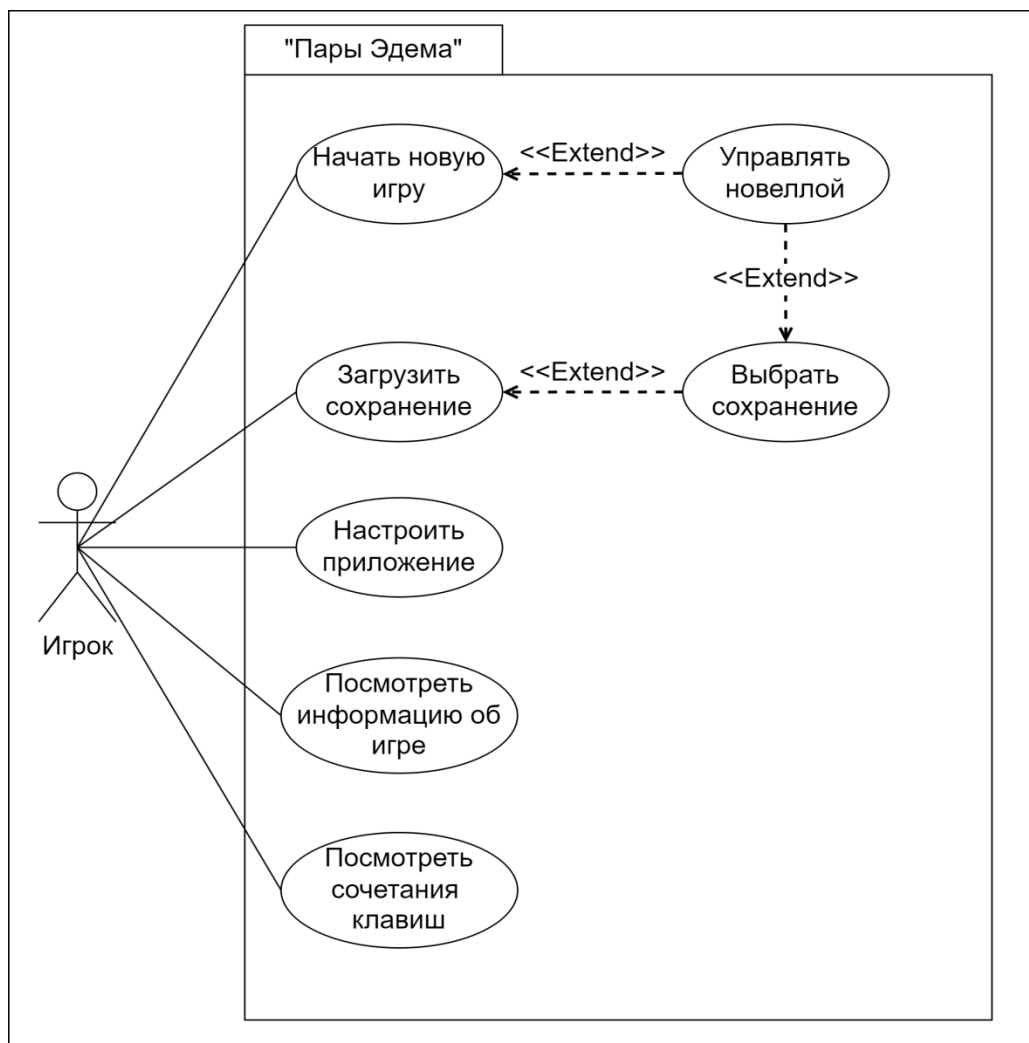


Рисунок 7 – Диаграмма вариантов использования

Основные актеры, взаимодействующие с системой

Игрок – пользователь игры, которому доступны возможности взаимодействия с приложением. Является единственным актером.

Краткое описание вариантов использования

«Начать новую игру» – пользователь приступает к игровому процессу, системой автоматически создается точка сохранения игры посредством нажатия кнопки «Начать» в главном меню.

«Управлять новеллой» – пользователь получает возможность проигрывать сюжет и влиять на игровой процесс посредством пролистывания текста новеллы (нажатие на диалоговое окно), осуществления сюжетных выборов (кнопки, появляющиеся в определенный момент) или прохождения мини-игр (сборание пазлов).

«Загрузить сохранение» – пользователь попадает в список сохранений, где ему предлагается выбрать одно из сохранений для загрузки. Реализовано с помощью кнопки «Загрузить» в главном меню.

«Выбрать сохранение» – пользователь выбирает сохранение из списка и загружает его, тем самым приступая к игровому процессу. Сохранение может быть сделано как самим игроком во время игры, так и системой автоматически. Сохранения, созданные автоматически, находятся в отдельном разделе автосохранений. Реализовано с помощью картинки сохранения в меню загрузки игры.

«Настроить приложение» – пользователь попадает в меню, где может настроить отображение экрана, изменить скорость появления текста диалогов на экране, изменить громкость звуков и музыки, изменить режим пропуска диалогов (всего текста, после выборов или пропуск переходов). Реализовано с помощью соответствующей кнопки «Настройки» в главном меню.

«Посмотреть информацию об игре» – пользователь попадает в меню, где может увидеть список людей тем или иным образом причастных к разработке приложения. Помимо этого, здесь указаны лицензии Ren'py и ПО, с помощью которого сделано приложение. Реализовано с помощью кнопки «Об игре» в главном меню.

«Посмотреть сочетания клавиш» – пользователь попадает в меню, где указаны все сочетания горячих клавиш для взаимодействия с приложением. Данные комбинации клавиш встроены в движок игры по умолчанию. Реализовано с помощью кнопки «Помощь» в главном меню.

Вывод по третьему разделу

Были определены основные характеристики и функциональные возможности игрового приложения, обеспечивающие наилучший пользовательский опыт, сформированы функциональные и нефункциональные требования, а также описаны основные варианты использования.

4. АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ

Целью данного раздела является проектирование архитектуры разрабатываемого приложения. Помимо этого, в данном разделе описано проектирование максимально простого и удобного графического пользовательского интерфейса.

4.1. Компоненты приложения

На рисунке 8 представлена диаграмма компонентов приложения, использующего игровой движок Ren'py. Игровое приложение состоит из четырех основных компонентов: Меню, Скрипты, Мини-игра и Ресурсы. Основным компонентом системы являются Скрипты, содержащие в себе исходные коды, написанные на языке Python, которые осуществляют взаимодействие системы с игровыми ресурсами.

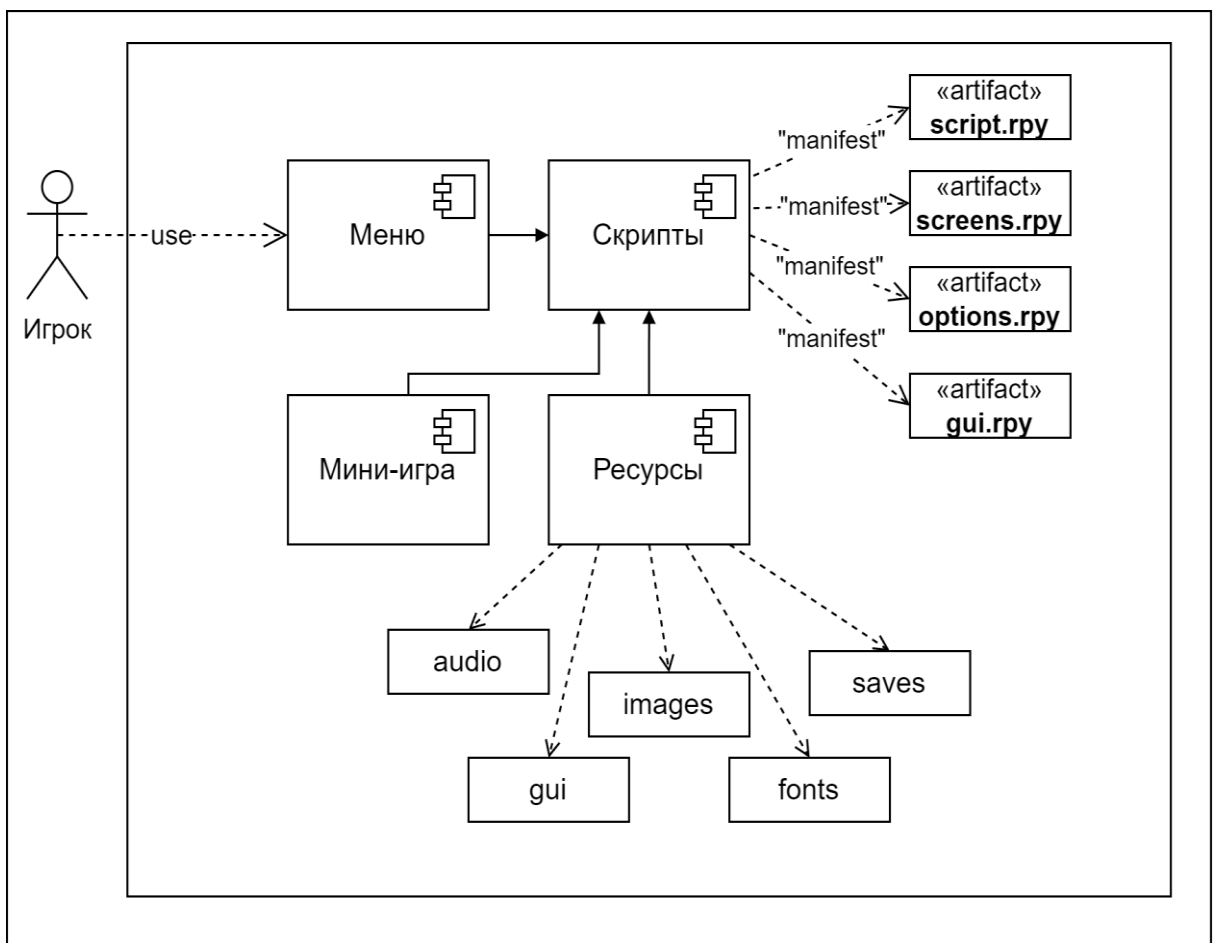


Рисунок 8 – Компоненты игрового приложения

Рассмотрим каждый компонент по отдельности более подробно.

Меню – компонент, включающий в себя внешний интерфейс всей системы для осуществления взаимодействия игрока с приложением.

Мини-игра – компонент, отвечающий за работу мини-игры.

Ресурсы – компонент, содержащий в себе файлы ресурсов необходимые для игрового приложения, такие как: аудио, фоны, спрайты персонажей, шрифты, картинки для интерфейса.

Скрипты – компонент, содержащий в себе артефакты, отвечающие за различные составляющие системы. Все они имеют расширение .gru, используемое движком Ren'py.

Все компоненты отвечают за реализацию различного функционала игрового приложения, но компонент «Скрипты», являясь основным, связывает их всех в одну систему.

4.2. Проектирование интерфейса

Графический интерфейс пользователя или graphical user interface (сокращенно GUI) – графический интерфейс, с помощью которого происходит взаимодействие человека с различным программным обеспечением. Данный интерфейс должен быть максимально понятен и интуитивен для обеспечения наилучшего опыта использования.

При запуске игрового приложения пользователь попадает в главное меню (рисунок 9), где ему доступно несколько различных опций взаимодействия с приложением: «Начать», «Загрузка», «Об игре», «Помощь», «Настройки», «Выход».

При нажатии кнопки «Начать» пользователь попадает на экран игрового процесса (рисунок 10), на котором он может видеть «диалоговое окно», «портрет персонажа» при определенных условиях и «панель взаимодействия» внизу экрана с различными опциями.



Рисунок 9 – Макет главного меню

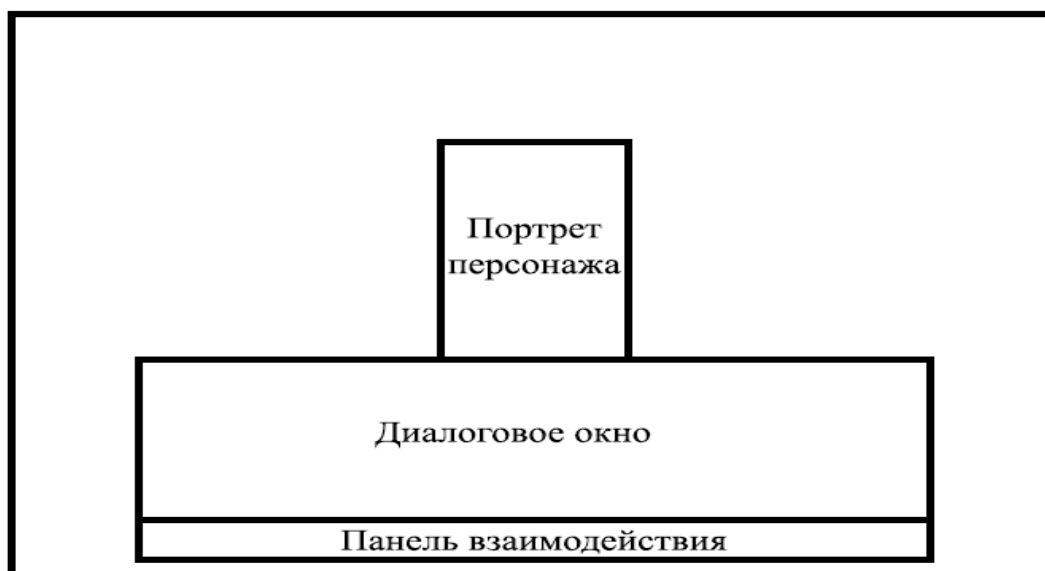


Рисунок 10 – Макет игрового процесса

Во время игрового процесса пользователь может нажать на кнопку «Опции» на панели взаимодействия или на кнопку «Esc» на клавиатуре для того, чтобы попасть в контекстное меню (рисунок 11).

Контекстное меню включает в себя все опции главного меню. Помимо этого пользователь может сохранить игровой процесс в одну из 6 ячеек слотов сохранения представленных на экране. Аналогичным образом пользователь может загрузить ранее сохраненный прогресс. Также пользователь может прочитать «Историю» диалогов.

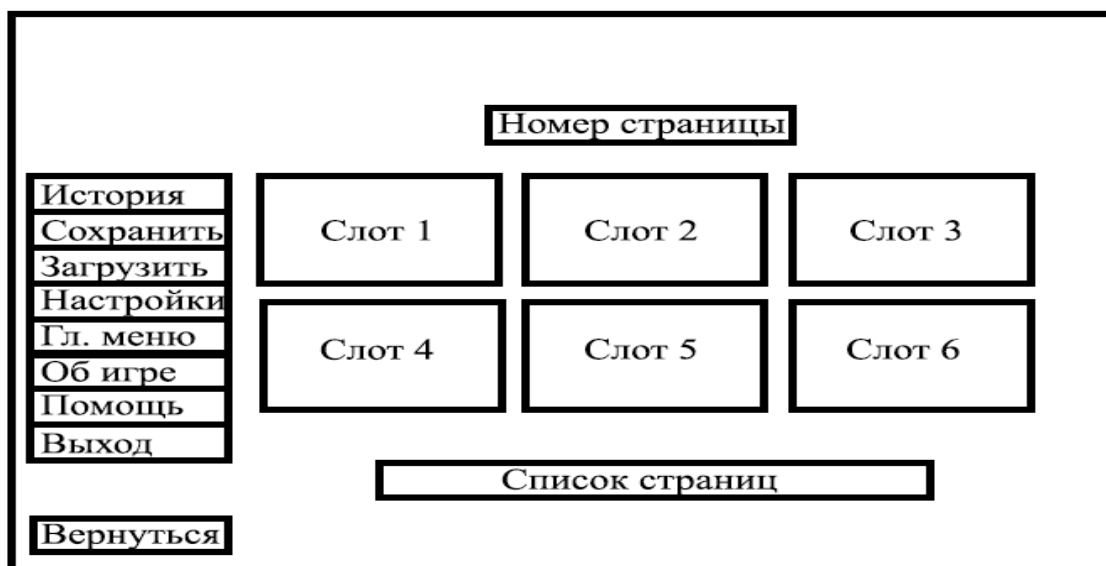


Рисунок 11 – Макет контекстного меню

Вернемся к главному меню и разберем оставшиеся экраны доступные пользователю помимо «Начать». При нажатии кнопки «Загрузить» пользователь попадает на экран загрузки сохранения (рисунок 12). Так же как и контекстное меню, этот экран содержит 6 ячеек для сохранений, которые можно загрузить.

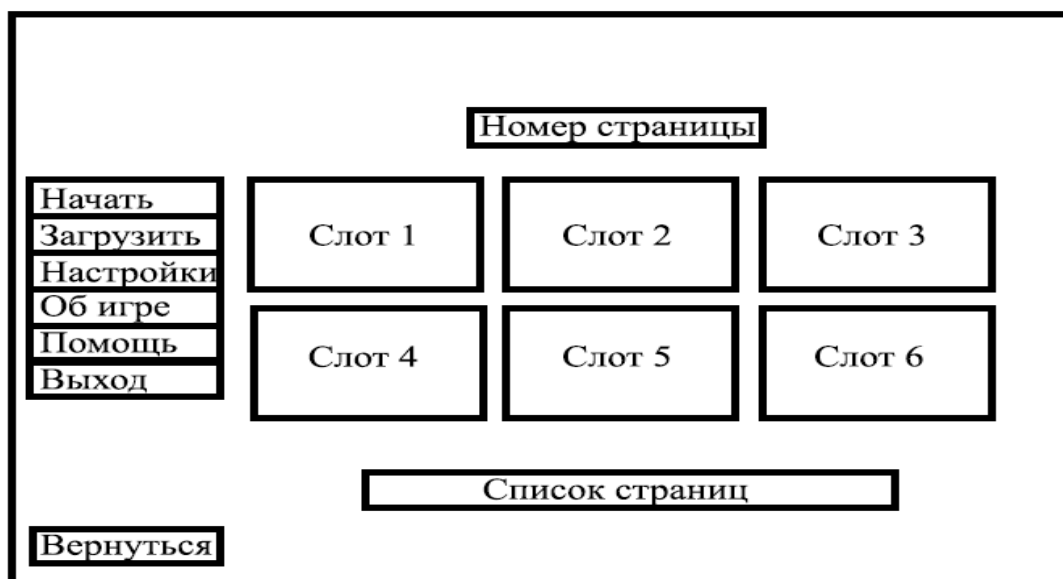


Рисунок 12 – Макет меню «Загрузить»

При нажатии на кнопку «Настройки» пользователь попадает в меню настроек приложения (рисунок 13). Игрок может изменить режим отображения окна приложения (оконный или полноэкранный), изменить скорость

появления текста на экране во время игры и скорость автоматического прочтения диалогов. Также пользователь может изменить громкость воспроизводимой музыки и звуков или полностью отключить аудио. Помимо этого, пользователь может выбрать режим пропуска диалогов – всего текста диалогов, пропуск текста после выборов или пропуск текста после переходов.

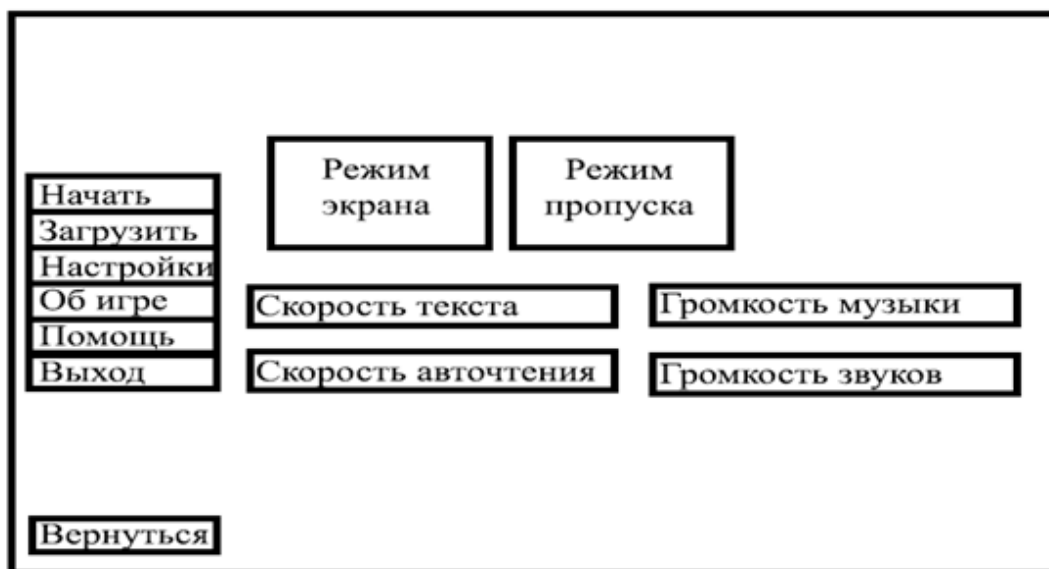


Рисунок 13 – Макет меню «Настройки»

Когда пользователь нажимает на кнопку «Об игре» он попадает в меню (рисунок 14), где содержится информация об авторе игры, лицензиях Ren'ru и ПО, с помощью которого сделано приложение.

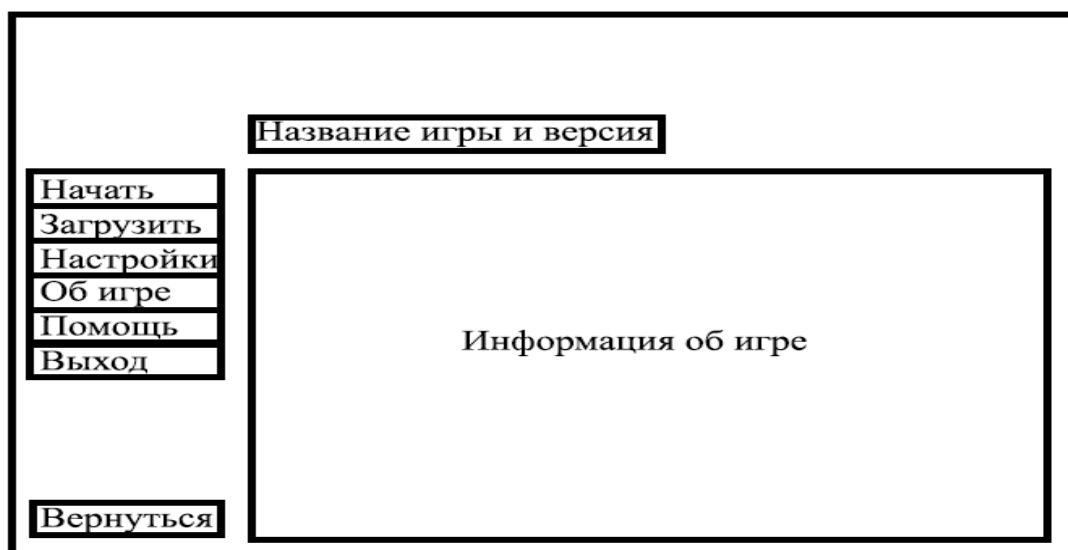


Рисунок 14 – Макет меню «Об игре»

При нажатии кнопки «Помощь» пользователь попадает в меню, где показаны все возможные комбинации клавиш для взаимодействия с приложением (рисунок 15). Данные комбинации клавиш используются в проектах Ren'ru по умолчанию, но также имеется возможность дополнения данного раздела собственными горячими клавишами при необходимости.



Рисунок 15 – Макет меню «Помощь»

Вывод по четвертому разделу

Была спроектирована архитектура разрабатываемого игрового приложения, состоящая из четырех основных компонентов. Также были разработаны максимально простые и удобные макеты проектируемого игрового интерфейса в количестве 7 экземпляров.

5. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ СИСТЕМЫ

5.1. Реализация компонентов системы

Мини-игра

Мини-игра представляет собой сбор пазлов различного размера, где игроку необходимо из кусочков собрать полноценное изображение.

Мини-игра реализована в отдельном файле `Minigame` с расширением `«.gru»`, что соответствует стандартному расширению файлов на игровом движке `Ren'Py`.

`Ren'py` позволяет использовать ключевое слово `init` для инициализации языка программирования `Python` еще до того как игра будет непосредственно загружена. Таким образом, можно инициализировать различные функции, классы, стили, глобальные переменные или какие-то постоянные данные.

Для реализации мини-игры используется механика движка `Ren'py` `Drag and Drop`, позволяющая пользователю перемещать изображения по экрану. Данная механика дает возможность создания множества различных игровых систем, начиная от инвентаря и заканчивая карточными играми.

Класс `Drag` отражает объект, который может быть перемещен по экрану (параметр `Dragged`) или сброшен на другой объект (параметр `Dropped`). Помимо этого, некоторые объекты класса `Drag` могут быть статичными (не могут быть перемещены) и обладать сразу несколькими параметрами или не обладать ими вовсе.

Класс `DragGroup` содержит в себе группу объектов `Drag`, которые могут взаимодействовать друг с другом в зависимости от их параметров.

Для визуальной реализации объектов класса `Drag` используется ключевое слово `Ren'py screen`, отвечающее за вызов экрана с определенной информацией и дает возможность пользователю взаимодействовать с ней.

Исходный код функции `piece_dragged`, отвечающей за взаимодействие деталей пазла при их перемещении, представлен на рисунке 16.

```

def piece_dragged(drags, drop):

    if not drop:
        return

    p_x = drags[0].drag_name.split("-")[0]
    p_y = drags[0].drag_name.split("-")[1]
    t_x = drop.drag_name.split("-")[0]
    t_y = drop.drag_name.split("-")[1]

    a = []
    a.append(drop.drag_joined)
    a.append((drags[0], 3, 3))
    drop.drag_joined(a)

    if p_x == t_x and p_y == t_y:
        my_x = int(int(p_x)*active_area_size*x_scale_index) -
int(grip_size*x_scale_index)+puzzle_field_offset
        my_y = int(int(p_y)*active_area_size*y_scale_index) -
int(grip_size*y_scale_index)+puzzle_field_offset
        drags[0].snap(my_x,my_y, delay=0.1)
        drags[0].draggable = False
        placedlist[int(p_x),int(p_y)] = True

    for i in range(0, grid_width):
        for j in range(0, grid_height):
            if placedlist[i,j] == False:
                return

    return True
return

```

Рисунок 16 – Код функции `piece_dragged`

Стоит более подробно разобраться в том, за что отвечает данная функция. Изначально осуществляется проверка на отпускание объекта `Drag` после перемещения. Если объект был перемещен и отпущен, происходит извлечение координат того места, где деталь пазла была отпущена. Далее происходит связывание куска пазла с его новым местоположением. Если кусок пазла был отпущен в правильном положении, происходит привязка объекта и свойство подвижности `draggable` изменяется на `False`, делая объект неподвижным. В конце осуществляется проверка на нахождение всех деталей пазла на своих местах. Если все детали на месте, возвращается значение `True`.

На рисунке 17 представлен исходный код экрана `jigsaw` отвечающего за размещение объектов на экране.

```

screen jigsaw:

    add im.Scale("Minigame/_puzzle_field.png", img_width, img_height)
    pos(puzzle_field_offset, puzzle_field_offset)

    draggroup:

        for i in range(0, grid_width):
            for j in range(0, grid_height):
                $ name = "%s-%s"%(i,j)
                $ my_x =
i*int(active_area_size*x_scale_index)+puzzle_field_offset
                $ my_y =
j*int(active_area_size*y_scale_index)+puzzle_field_offset
                drag:
                    drag_name name
                    child im.Scale("Minigame/_blank_space.png",
int(active_area_size*x_scale_index), int(active_area_size*y_scale_index)
)
                    draggable False
                    xpos my_x ypos my_y

        for i in range(0, grid_width):
            for j in range(0, grid_height):
                $ name = "%s-%s-piece"%(i,j)
                drag:
                    drag_name name
                    child imagelist[i,j]
                    dragged piece_dragged

```

Рисунок 17 – Исходный код экрана jigsaw

Данный код добавляет на экран изображение поля пазла, размещает невидимые объекты типа Drag со свойством draggable имеющим значение False для отслеживания попадания на них деталей пазла, а также детали пазла, которые можно перемещать. Полный код мини-игры приведен в приложении Б.

Ресурсы

Реализованный проект имеет директорию game, в которой содержатся все ресурсы приложения, такие как: звуки, изображения, файлы сохранений, элементы интерфейса, шрифты и кэш. Файловая структура представлена на рисунке 18.

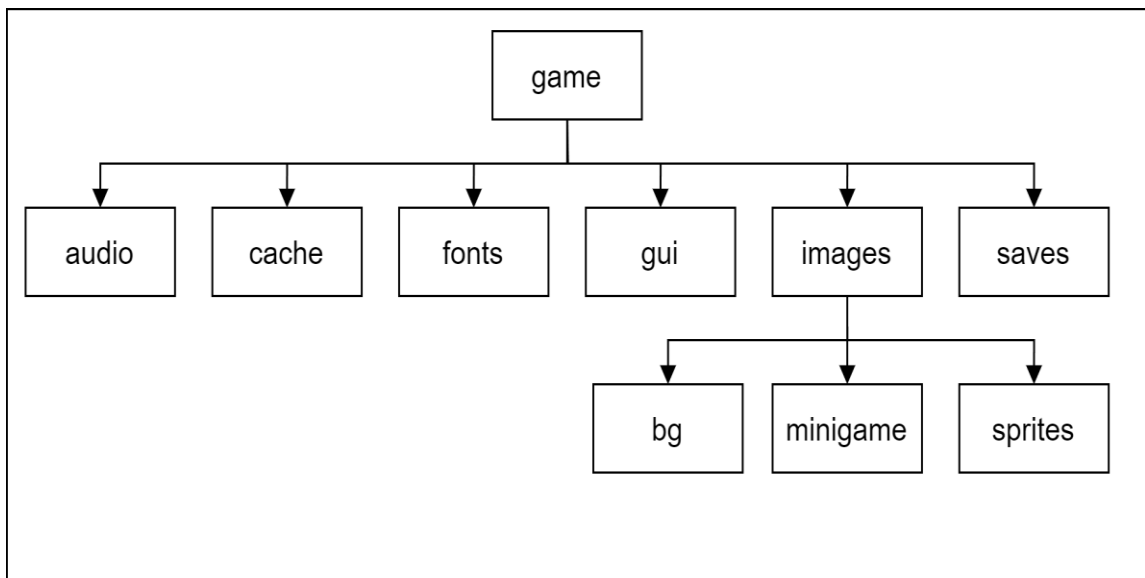


Рисунок 18 – Файловая структура директории game

Директория audio содержит аудиофайлы. В папке cache содержится кэш приложения. Директории fonts, gui и saves содержат шрифты, элементы интерфейса и сохранения соответственно. Папка images имеет три подкаталога, содержащие изображения фонов, картинки для мини-игры и спрайты персонажей соответственно. Все используемые изображения были сгенерированы с использованием нейросети StableDiffusion модель NovelAI 1.0 и в дальнейшем вручную отредактированы при помощи графического редактора Photoshop. Музыкальное сопровождение было сгенерировано с помощью сервиса Udio.

5.2. Реализация интерфейса

Интерфейс приложения реализуется в отдельном файле screens.py. Все стандартные функции взаимодействия в игре реализованы на движке Ren'py нативно, такие как: «Пропуск», «Авто», «Сохранить», «Загрузить», «Б.Сохранение» (быстрое сохранение), «Б.Загрузка» (быстрая загрузка), «Опции» и другие.

Главное меню

Главное меню было реализовано по соответствующим функциональным требованиям и приведено на рисунке 19.



Рисунок 19 – Главное меню

«Загрузить»

Меню «Загрузить» было реализовано по соответствующим функциональным требованиям и приведено на рисунке 20.

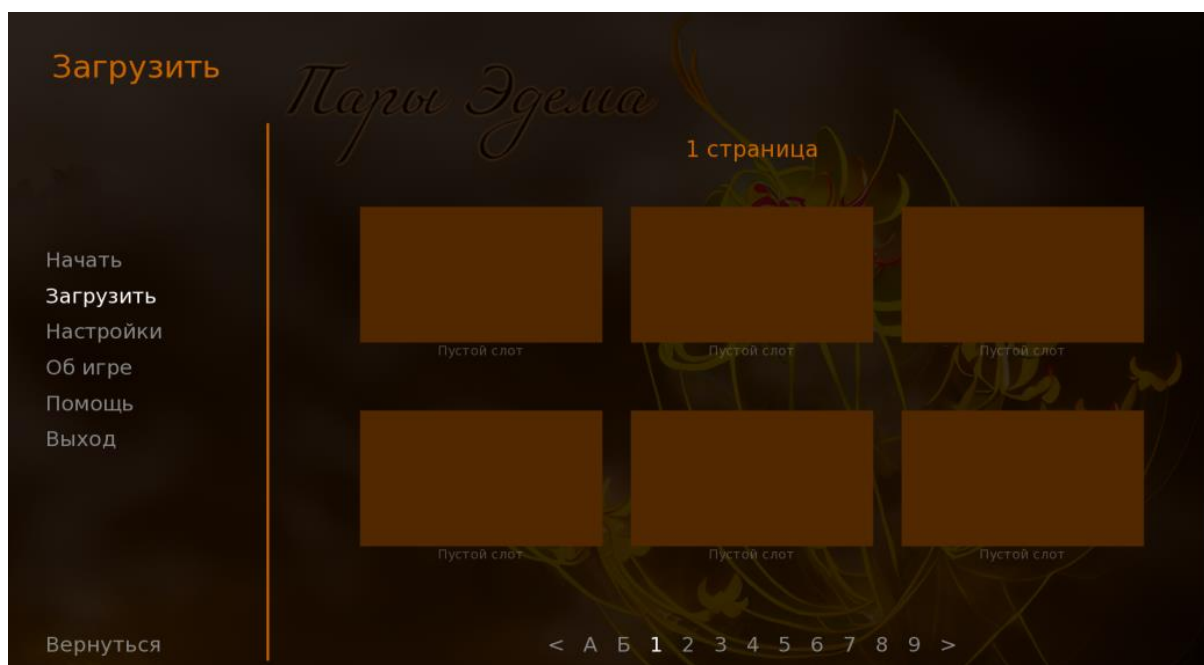


Рисунок 20 – Меню «Загрузить»

«Настройки»

Меню «Настройки» было реализовано по соответствующим функциональным требованиям и приведено на рисунке 21.

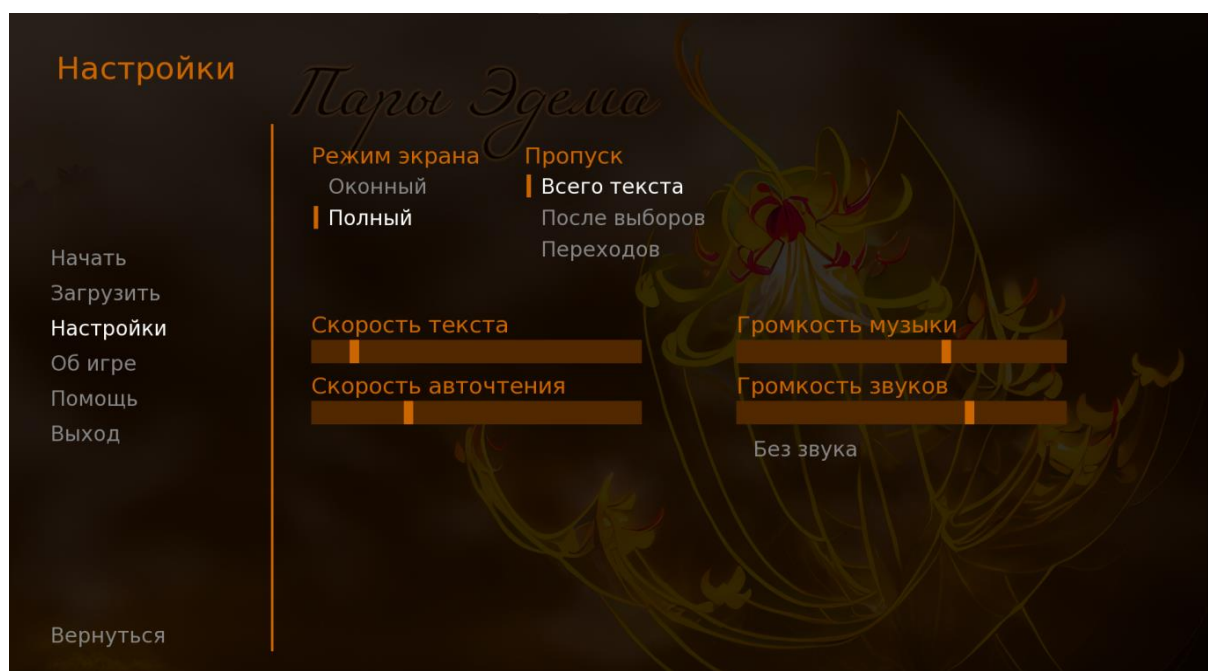


Рисунок 21 – Меню «Настройки»

«Об игре»

Меню «Об игре» было реализовано по соответствующим функциональным требованиям и приведено на рисунке 22.

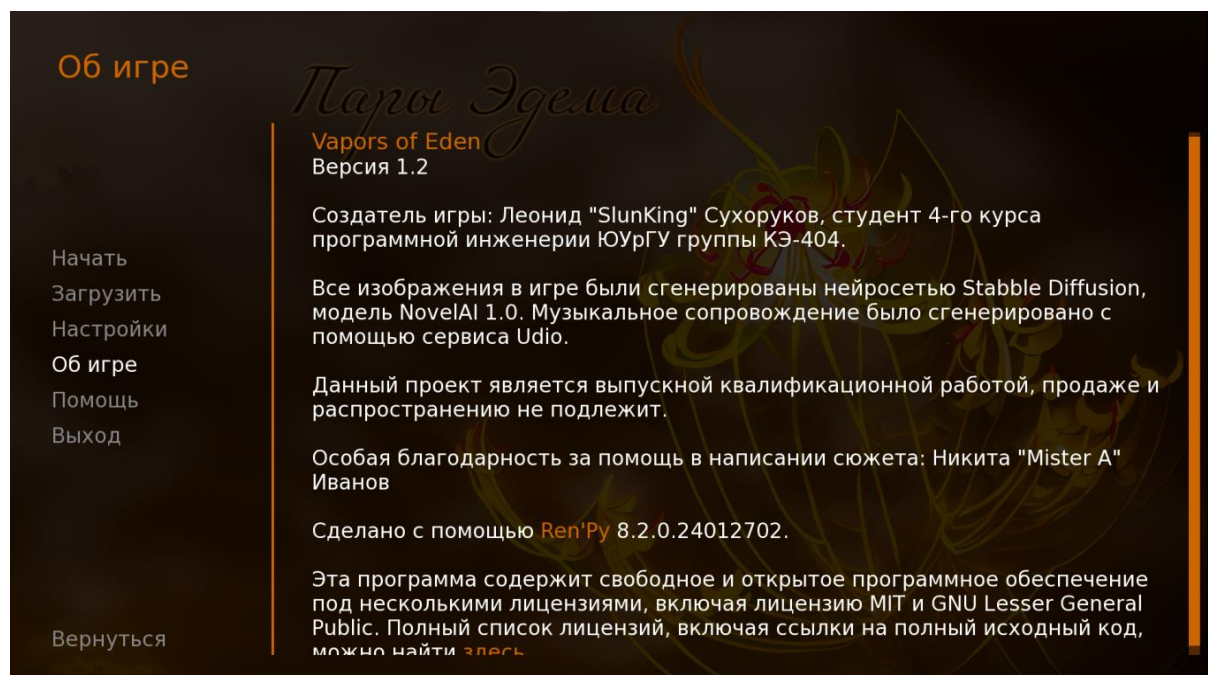


Рисунок 22 – Меню «Об игре»

«Помощь»

Меню «Помощь» было реализовано по соответствующим функциональным требованиям и приведено на рисунке 23.

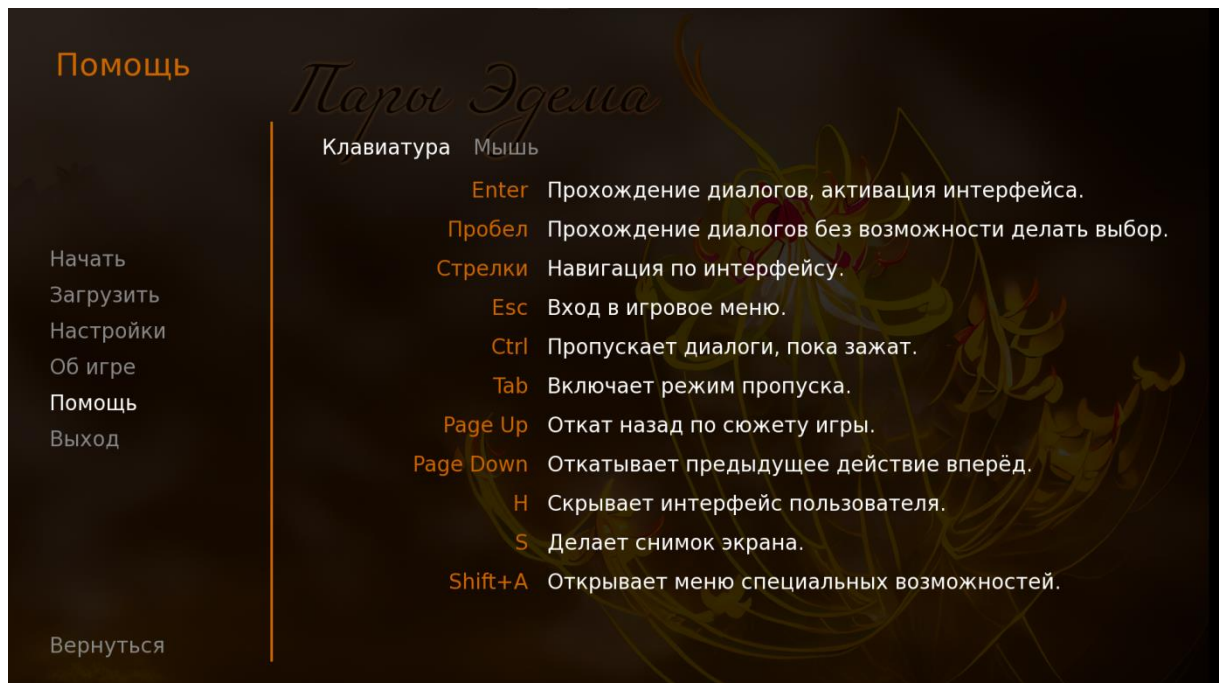


Рисунок 23 – Меню «Помощь»

Контекстное меню

Контекстное меню было реализовано по соответствующим функциональным требованиям и приведено на рисунке 24.

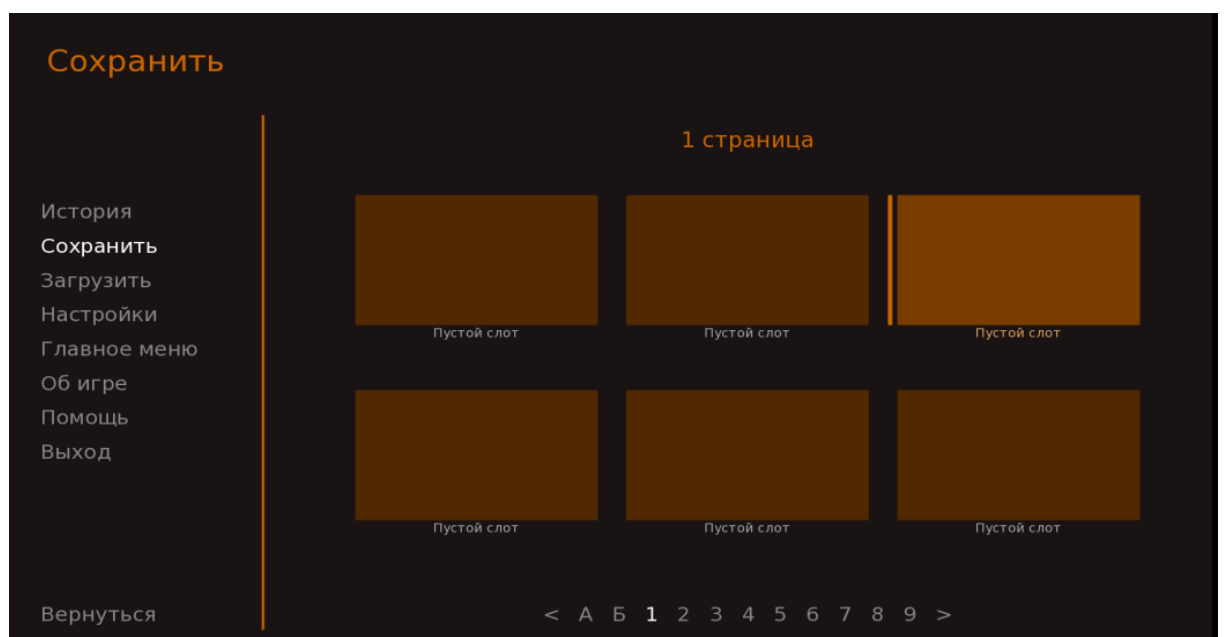


Рисунок 24 – Контекстное меню

Экран игрового процесса

Экран игрового процесса был реализован по соответствующим функциональным требованиям и приведен на рисунке 25.



Рисунок 25 – Экран игрового процесса

5.3. Тестирование системы

Для тестирования [22–23] игрового приложения был использован метод функционального тестирования, при котором проверяется реализуемость функциональных требований, то есть способность программного обеспечения решать определенные задачи, нужные пользователю в определенных условиях. Набор тестов представлен в таблице 1.

Таблица 1 – Тестирование приложения

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1	Начало игры	Пользователь нажимает кнопку «Начать в главном меню».	Начинается игровой процесс.	Да
2	Сохранение игры	1. Во время игрового процесса пользователь заходит в контекстное меню. 2. Пользователь нажимает кнопку сохранить. 3. Пользователь выбирает ячейку сохранения и нажимает на нее.	Игровой процесс сохранен, в ячейке отображается сохранение с датой и временем сохранения.	Да

Продолжение таблицы 1

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
3	Загрузка игры	1. В главном или контекстном меню пользователь нажимает кнопку загрузить. 2. Пользователь выбирает ячейку сохранения и нажимает на нее.	Происходит загрузка сохранения и начинается игровой процесс.	Да
4	Меню «Об игре»	1. Пользователь находится в главном или контекстном меню. 2. Пользователь нажимает кнопку «Об игре».	Пользователь попадает в меню «Об игре».	Да
5	Меню «Помощь»	1. Пользователь находится в главном или контекстном меню. 2. Пользователь нажимает кнопку «Помощь».	Пользователь попадает в меню «Помощь».	Да
6	Меню «Настройки»	1. Пользователь находится в главном или контекстном меню. 2. Пользователь нажимает кнопку «Настройки».	Пользователь попадает в меню «Настройки».	Да
7	Выход из игры	1. Пользователь находится в главном или контекстном меню. 2. Пользователь нажимает кнопку «Выход».	Игра запрашивает у пользователя подтверждение выхода, приложение закрывается.	Да
8	Изменение режима отображения	1. Пользователь находится в главном или контекстном меню. 2. Пользователь заходит в меню «Настройки». 3. Пользователь меняет режим отображения игры на полный экран.	Приложение открывается на полный экран.	Да
9	Изменение скорости текста	1. Пользователь находится в контекстном или главном меню. 2. Пользователь заходит в меню «Настройки». 3. Пользователь изменяет скорость появления текста на максимально быструю.	Текст в диалогах во время игрового процесса появляется на экране мгновенно.	Да
10	Чтение истории диалогов	1. Пользователь находится в контекстном меню. 2. Пользователь нажимает кнопку «История».	Пользователь попадает в меню «История», где может прочитать все предыдущие диалоги.	Да

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
11	Пропуск диалогов	Во время игрового процесса пользователь нажимает кнопку «Пропуск» на нижней панели.	Происходит пропуск диалогов до первого выбора или экрана мини-игры.	Да
12	Авто-режим чтения	Во время игрового процесса пользователь нажимает кнопку «Авто» на нижней панели.	Включается авто-режим. Диалоги читаются автоматически.	Да
13	Выбор	1. Во время игрового процесса на экране пользователя появляется несколько вариантов ответа. 2. Пользователь выбирает один из вариантов ответа.	Если данный выбор является важным, то в зависимости от выбора игрока последующий диалог меняется.	Да
14	Мини-игра 1	Во время мини-игры пользователь перемещает пазл на нужное место.	Деталь пазла прикрепляется к месту и ее больше нельзя подвинуть.	Да
15	Мини-игра 2	Во время мини-игры пользователь пытается переместить деталь за игровой экран.	Деталь нельзя переместить за определенную рамку по краям экрана.	Да

Вывод по пятому разделу

Были реализованы все основные компоненты игрового приложения. Разработана мини-игра с пазлами и директория с ресурсами приложения. Был реализован спроектированный интерфейс, соответствующий всем функциональным требованиям. Также было проведено общее функциональное тестирование системы. Все тесты успешно пройдены.

ЗАКЛЮЧЕНИЕ

Количество людей, играющих в игры и разрабатывающих их, растет с каждым днем. Игровая индустрия становится неотъемлемой частью современной культуры, влияя на самые различные аспекты нашей жизни. Важно понимать, что разработка игр требует не только технических знаний, но и творческого подхода. Визуальные новеллы – это отличный жанр игр, позволяющий людям погружаться с головой в уникальные и захватывающие миры.

В данной выпускной квалификационной работе было разработано и реализовано приложение в жанре «Визуальная новелла» на платформе Ren'py, которое представляет собой игровое приложение для операционной системы Windows. Целью было создание игры с интересным сюжетом и визуальным стилем. В ходе выполнения работы были решены следующие задачи.

1. Выполнен анализ предметной области.
2. Изучены особенности и возможности движка Ren'Py.
3. Написан сценарий и создано визуальное оформление игры.
4. Спроектирована архитектура приложения, включая проектирование файловой системы и интерфейса.
5. Произведена программная реализация проекта.
6. Протестировано реализованное приложение.

В результате выполнения работы были решены все поставленные задачи, таким образом, цель данной работы достигнута.

ЛИТЕРАТУРА

1. DTF. [Электронный ресурс] URL: <https://dtf.ru/gamedev/648514-analiz-instrumentov-vizualnoi-novelly> (дата обращения: 14.05.2024 г.).
2. Fandom wiki. [Электронный ресурс] URL: https://typemoon.fandom.com/ru/wiki/Fate/stay_night (дата обращения: 14.05.2024 г.).
3. Fandom wiki. [Электронный ресурс] URL: https://everlasting-summer.fandom.com/ru/wiki/Everlasting_Summer_вики (дата обращения: 14.05.2024 г.).
4. Fandom wiki. [Электронный ресурс] URL: [https://steins-gate.fandom.com/ru/wiki/Steins;Gate_\(визуальная_новелла\)](https://steins-gate.fandom.com/ru/wiki/Steins;Gate_(визуальная_новелла)) (дата обращения: 14.05.2024 г.).
5. Fandom wiki. [Электронный ресурс] URL: <https://danganronpa.fandom.com/ru/wiki/Danganronpa> (дата обращения: 14.05.2024 г.).
6. Unity. [Электронный ресурс] URL: <https://unity.com/ru> (дата обращения: 14.05.2024 г.).
7. RPG Maker. [Электронный ресурс] URL: <https://www.rpgmakerweb.com/> (дата обращения: 14.05.2024 г.).
8. TyranoBuilder. [Электронный ресурс] URL: <https://tyranobuilder.com/> (дата обращения: 14.05.2024 г.).
9. Руководство по движку Ren'Py. [Электронный ресурс] URL: <https://www.renpy.org/doc/html/> (дата обращения: 14.05.2024 г.).
10. Ren'Py. [Электронный ресурс] URL: www.renpy.org (дата обращения: 14.05.2024 г.).
11. Ren'Py documentation. [Электронный ресурс] URL: <https://www.renpy.org/doc/html/> (дата обращения 14.05.2024 г.).
12. Ren'Py документация на русском языке. [Электронный ресурс] URL: <http://ru.renpyedia.shoutwiki.com> (дата обращения: 14.05.2024 г.).

13. Руководство по языку программирования Python. [Электронный ресурс] URL: <https://metanit.com/python/tutorial/> (дата обращения: 14.05.2024 г.).
14. Python documentation. [Электронный ресурс] URL: <https://www.python.org/doc> (дата обращения 14.05.2024 г.).
15. Лутц М Программирование на Python, том I / М Лутц. // СанктПетербург: 5-е издание, 2019. – 833 с.
16. Лутц М Программирование на Python, том II / М Лутц. // СанктПетербург: 5-е издание, 2020. – 722 с.
17. Андрианова Н. Яковлева С. Как создавать истории. Основы игровой сценаристики и нарративного дизайна за 12 шагов. – Москва: Бомбора, 2023. – 298 с.
18. Шелл, Д. Геймдизайн. Как создать игру, в которую будут играть все. / Д. Шелл; переводчик А. Лысенко. // Москва.: Альпина Паблишер, 2019. – 640 с.
19. Бейдер Д. Чистый Python. Тонкости программирования для профи. / Д. Бейдер; переводчик А. Логунов. // СПб.: Питер, 2022. – 288 с.
20. Зубек Р. Элементы гейм-дизайна. Как создавать игры, от которых невозможно оторваться. / Р. Зубек; переводчик О. Перфильев. // Москва.: Бомбора, 2023. – 272 с.
21. Эрик, Мэтиз Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – Москва.: Питер, 2018. – 760 с.
22. Аниче М. Эффективное тестирование программного обеспечения. / М. Аниче; переводчик А. Кисилев. // Москва.: ДМК Пресс, 2022. – 370 с.
23. Блек Р. Ключевые процессы тестирования. – Москва.: Издательство Лори, 2014. – 544 с.

ПРИЛОЖЕНИЯ

Приложение А. Спецификация вариантов использования

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

Таблица 1 – Спецификация ВИ «Начать новую игру»

Прецедент: Начать новую игру
ID: 1
Краткое описание: Игрок начинает новую игру.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: игрок находится в главном меню
Основной поток: 1. Игрок нажимает кнопку «Начать» 2. Начинается игровой процесс
Постусловия: 1. Игрок приступил к игровому процессу
Альтернативные потоки: Нет

Таблица 2 – Спецификация ВИ «Управлять новеллой»

Прецедент: Управлять новеллой
ID: 2
Краткое описание: Игрок приступает к игровому процессу
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: игрок приступил к игровому процессу
Основной поток: 1. Игрок пролистывает диалоги посредством нажатия на диалоговое окно. 2. Приложение возвращает игрока в главное меню по окончании пролистывания всех диалогов.
Постусловия: 1. Игрок находится в главном меню
Альтернативные потоки: I. Сюжетный выбор при пролистывании диалогов. 1. Приложение выводит несколько кнопок, из которых нужно выбрать только одну. 2. Пользователь нажимает на одну из кнопок, продолжается пролистывание диалогов. II Мини-игра при пролистывании диалогов. 1. Игрок попадает на экран мини-игры, где необходимо собрать пазл. 2. Пользователь собирает пазл или пропускает мини-игру, продолжается пролистывание диалогов.

Таблица 3 – Спецификация ВИ «Загрузить сохранение»

Прецедент: Загрузить сохранение
ID: 2
Краткое описание: Игрок попадает в меню сохранений.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: игрок находится в главном или контекстном меню
Основной поток: 1. Игрок нажимает кнопку «Загрузить». 2. Игрок попадает в меню сохранений.
Постусловия: 1. Игрок находится в меню сохранений
Альтернативные потоки: Нет

Таблица 4 – Спецификация ВИ «Выбрать сохранение»

Прецедент: Выбрать сохранение
ID: 3
Краткое описание: Игрок выбирает сохранение из списка для загрузки.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игрок находится в меню сохранений
Основной поток. 1. Игрок выбирает сохранение из списка для загрузки. 2. Происходит загрузка сохранения. 3. Начинается игровой процесс.
Постусловия: 1. Игрок приступил к игровому процессу
Альтернативные потоки: Нет

Таблица 5 – Спецификация ВИ «Настроить приложение»

Прецедент: Настроить приложение
ID: 4
Краткое описание: Игрок хочет изменить настройки отображения и поведения игры.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игрок находится в главном или контекстном меню
Основной поток. 1. Игрок нажимает кнопку «Настройки». 2. Игрок попадает в меню настроек. 3. Игрок изменяет необходимые настройки.
Постусловия: 1. Настройки сохранены 2. Игрок находится в главном или контекстном меню
Альтернативные потоки: Нет

Таблица 6 – Спецификация ВИ «Посмотреть информацию об игре»

Прецедент: Посмотреть информацию об игре
ID: 5
Краткое описание: Игрок узнает информацию об игре и ее авторах.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игрок находится в главном или контекстном меню
Основной поток. 1. Игрок нажимает кнопку «Об игре». 2. Игрок попадает в меню, где указаны люди причастные к разработке приложения, а также ПО, с помощью которого было создано приложение. 3. Игрок закрывает меню.
Постусловия: 1. Игрок находится в главном или контекстном меню
Альтернативные потоки: Нет

Таблица 7 – Спецификация ВИ «Посмотреть сочетания клавиш»

Прецедент: Посмотреть сочетания клавиш
ID: 6
Краткое описание: Игрок узнает информацию о горячих клавишах.
Главные актеры: Игрок
Второстепенные актеры: Нет
Предусловия: Игрок находится в главном или контекстном меню
Основной поток. 1. Игрок нажимает кнопку «Помощь». 2. Игрок попадает в меню, где указаны все горячие клавиши для взаимодействия с приложением. 3. Игрок закрывает меню.
Постусловия: 1. Игрок находится в главном или контекстном меню
Альтернативные потоки: Нет

Приложение Б. Листинг мини-игры

Полный код реализованной мини-игры приведен в листинге 1.

Листинг 1 – Код мини-игры по сбору пазлов

```
init python:

# Функция отвечающая за взаимодействие деталей при перемещении
def piece_dragged(drags, drop):

    if not drop:
        return

    # извлечение координат куска пазла и места его отпускания
    p_x = drags[0].drag_name.split("-")[0]
    p_y = drags[0].drag_name.split("-")[1]
    t_x = drop.drag_name.split("-")[0]
    t_y = drop.drag_name.split("-")[1]

    # связывание куска пазла с его новым местоположением
    a = []
    a.append(drop.drag_joined)
    a.append((drags[0], 3, 3))
    drop.drag_joined(a)

    #Проверка куска пазла на отпусkanie в правильном месте
    if p_x == t_x and p_y == t_y:
        my_x = int(int(p_x)*active_area_size*x_scale_index)-
int(grip_size*x_scale_index)+puzzle_field_offset_x
        my_y = int(int(p_y)*active_area_size*y_scale_index)-
int(grip_size*y_scale_index)+puzzle_field_offset_y
        drags[0].snap(my_x,my_y, delay=0.1) # смещение детали на нужное
место

        drags[0].draggable = False # отключение перемещения детали
        placedlist[int(p_x),int(p_y)] = True # присваивание детали зна-
чения True в общем списке деталей (True - расположен на верном месте)

        # Проверка расположения всех деталей на своем месте
        for i in range(0, grid_width):
            for j in range(0, grid_height):
                if placedlist[i,j] == False:
                    return
        return True
    return

screen jigsaw: # Экран мини-игра на котором размещаются элементы

    add im.Scale("minigame/_puzzle_field.png", img_width, img_height)
    pos(puzzle_field_offset_x, puzzle_field_offset_y) # Размещение поля пазла
на экране

    textbutton "Пропуск": # Кнопка пропуска
        xalign 0.9 yalign 0.1 action Return()

draggroup:

    for i in range(0, grid_width): # Размещение невидимых пустых мест
для деталей
        for j in range(0, grid_height):
            $ name = "%s-%s"%(i,j) # Для каждого пазла создается уни-
кальное имя
```

Продолжение листинга 1 приложения Б

```
        $ my_x =
i*int(active_area_size*x_scale_index)+puzzle_field_offset_x
        $ my_y =
j*int(active_area_size*y_scale_index)+puzzle_field_offset_y
    drag:
        drag_name name
        child im.Scale("minigame/_blank_space.png",
int(active_area_size*x_scale_index), int(active_area_size*y_scale_index) )
        draggable False # Пустые места неподвижны
        xpos my_x ypos my_y

    for i in range(0, grid_width): # Размещение деталей пазла
        for j in range(0, grid_height):
            $ name = "%s-%s-piece"%(i,j)
            drag:
                drag_name name
                child imagelist[i,j] # У каждой детали свой кусок изна-
                чального изображения
                dragged piece_dragged # При перемещении детали происхо-
                дит вызов функции piece_dragged
                xpos piecelist[i,j][0] ypos piecelist[i,j][1] # Опреде-
                ление позиции детали

image puzzle_background = "minigame/Minigamebg.png" # Определение фона на
экране мини-игры

label puzzle:

    python:
        # Извлекаем размер изначального изображения
        img_width, img_height = renpy.image_size(chosen_img)

        # Уменьшаем размер изображения в зависимости от соотношения сторон
        if img_width >= img_height and img_width > puzzle_field_size:
            img_scale_down_index = round( (1.00 * puzzle_field_size /
img_width), 6)
            img_to_play = im.FactorScale(chosen_img, img_scale_down_index)
            img_width = int(img_width * img_scale_down_index)
            img_height = int(img_height * img_scale_down_index)

        elif img_height >= img_width and img_height > puzzle_field_size:
            img_scale_down_index = round( (1.00 * puzzle_field_size /
img_height), 6)
            img_to_play = im.FactorScale(chosen_img, img_scale_down_index)
            img_width = int(img_width * img_scale_down_index)
            img_height = int(img_height * img_scale_down_index)

        else:
            img_to_play = chosen_img

        # Вычисление коэффициентов масштабирования по горизонтали и верти-
        кали для корректного размещения кусков пазла
        x_scale_index = round( (1.00 *
(img_width/grid_width)/active_area_size), 6)
        y_scale_index = round( (1.00 *
(img_height/grid_height)/active_area_size), 6)

        # Создание основного изображения к которому будут привязываться
        куски пазла
        mainimage =
im.Composite((int(img_width+(grip_size*2)*x_scale_index),
```

Продолжение листинга 1 приложения Б

```
int(img_height+(grip_size*2)*y_scale_index)), (int(grip_size*x_scale_index),
int(grip_size*y_scale_index)), img_to_play)

# Определение индексов кусков пазла расположенных по краям
top_row = []
for i in range (0, grid_width):
    top_row.append(i)

bottom_row = []
for i in range (0, grid_width):
    bottom_row.append(grid_width*(grid_height-1)+i)

left_column = []
for i in range (0, grid_height):
    left_column.append(grid_width*i)

right_column = []
for i in range (0, grid_height):
    right_column.append(grid_width*i + (grid_width-1) )

# jigsaw_grid содержит в себе информацию о форме каждого куска
пазла
# Образование деталей начинается с верхнего слоя слева направо до
конца изображения вниз
jigsaw_grid = []
for i in range(0,grid_height):
    for j in range (0,grid_width):
        jigsaw_grid.append([9,9,9,9]) # [Сверху, Справа, Снизу,
Слева]

# В данном цикле происходит определение формы каждого куска пазла

for i in range(0,grid_height):
    for j in range (0,grid_width):
        if grid_width*i+j not in top_row:
            if jigsaw_grid[grid_width*(i-1)+j][2] == 1:
                jigsaw_grid[grid_width*i+j][0] = 2
            else:
                jigsaw_grid[grid_width*i+j][0] = 1
        else:
            jigsaw_grid[grid_width*i+j][0] = 0

        if grid_width*i+j not in right_column:
            jigsaw_grid[grid_width*i+j][1] =
renpy.random.randint(1,2)
        else:
            jigsaw_grid[grid_width*i+j][1] = 0

        if grid_width*i+j not in bottom_row:
            jigsaw_grid[grid_width*i+j][2] =
renpy.random.randint(1,2)
        else:
            jigsaw_grid[grid_width*i+j][2] = 0

        if grid_width*i+j not in left_column:
            if jigsaw_grid[grid_width*i+j-1][1] == 1:
                jigsaw_grid[grid_width*i+j][3] = 2
            else:
                jigsaw_grid[grid_width*i+j][3] = 1
        else:
            jigsaw_grid[grid_width*i+j][3] = 0
```

Окончание листинга 1 приложения Б

```
# Словари для хранения информации о каждой детали пазла
piecelist = dict()
imagelist = dict()
placedlist = dict()

# Распределение каждой детали пазла на экране
for i in range(0,grid_width):
    for j in range (0,grid_height):
        piecelist[i,j] = [int(renpy.random.randint(0,
int(config.screen_width * 0.8 - puzzle_field_size))+puzzle_field_size /
1.5), int(renpy.random.randint(0, int(config.screen_height * 0.7)))]
        # Создание временного изображения для каждого куса пазла
        temp_img = im.Crop(mainimage,
int(i*active_area_size*x_scale_index),
int(j*active_area_size*y_scale_index),
int(puzzle_piece_size*x_scale_index), int(puzzle_piece_size*y_scale_index))

        # Создание изображения для куса пазла на основе его формы
        # Вращает изображение для формирования верхней, правой, нижней и
левой сторон детали пазла
        imagelist[i,j] = im.Composite(
            (int(puzzle_piece_size*x_scale_index),
int(puzzle_piece_size*y_scale_index)),
            (0,0), im.AlphaMask(temp_img,
im.Scale(im.Rotozoom("minigame/_00%s.png"%(jigsaw_grid[grid_width*j+i][0]),
0, 1.0), int(puzzle_piece_size*x_scale_index),
int(puzzle_piece_size*y_scale_index))),
            (0,0), im.AlphaMask(temp_img,
im.Scale(im.Rotozoom("minigame/_00%s.png"%(jigsaw_grid[grid_width*j+i][1]),
270, 1.0), int(puzzle_piece_size*x_scale_index),
int(puzzle_piece_size*y_scale_index))),
            (0,0), im.AlphaMask(temp_img,
im.Scale(im.Rotozoom("minigame/_00%s.png"%(jigsaw_grid[grid_width*j+i][2]),
180, 1.0), int(puzzle_piece_size*x_scale_index),
int(puzzle_piece_size*y_scale_index))),
            (0,0), im.AlphaMask(temp_img,
im.Scale(im.Rotozoom("minigame/_00%s.png"%(jigsaw_grid[grid_width*j+i][3]),
90, 1.0), int(puzzle_piece_size*x_scale_index),
int(puzzle_piece_size*y_scale_index)))
        )
        placedlist[i,j] = False

# Установка заднего фона, вызов экрана пазла и ссылка на конец при
сборке
show puzzle_background as puzzle_bg
call screen jigsaw
jump win

label win:
    show puzzle_background as puzzle_bg
    # Показ собранного изображения по центру
    show expression img_to_play as win_img at truecenter with dissolve

    "Пазл собран"
    hide puzzle_bg
    hide win_img
    return # Возврат в основную игру
```