Пермский филиал федерального государственного автономного образовательного учреждения высшего образования   
«Национальный исследовательский университет   
«Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Чепоков Елизар Сергеевич

**РЕАЛИЗАЦИЯ МНОГОПОЛЬЗОВАТЕЛЬСКОЙ КОМПЬЮТЕРНОЙ ИГРЫ В ЖАНРЕ РПГ**

*Курсовая работа*

студента образовательной программы бакалавриата «Программная инженерия»

по направлению подготовки *09.03.04 Программная инженерия*

Руководитель:

преподаватель кафедры информационных технологий

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

В. В. Лебедев

Пермь, 2021 год

**Аннотация**

Название: Реализация многопользовательской компьютерной игры в жанре РПГ

Автор: Чепоков Елизар Сергеевич, студент третьего курса образовательной программы «Программная инженерия».

Руководитель: Лебедев Виктор Валерьевич, преподаватель кафедры информационных технологий.

Данная курсовая работа посвящена разработке многопользовательской компьютерной игры.

Работа включает 78 страниц формата А4, из них в основной части 37 страниц.

Основная часть работы включает в себя 21 иллюстрацию и 13 таблиц.

Библиографический список состоит из 14 публикаций.

Работа включает в себя 6 приложений.

# Оглавление

[Оглавление 3](#_Toc67679728)

[Введение 4](#_Toc67679729)

[Глава 1 Анализ 6](#_Toc67679730)

[1.1 Изучение особенностей игр в жанре «RPG» 6](#_Toc67679731)

[1.2 Изучение особенностей строения игровых уровней 8](#_Toc67679732)

[1.3 Обзор и сравнение программного обеспечения для создания игр 9](#_Toc67679733)

[1.4 Обзор игр-аналогов 12](#_Toc67679734)

[1.5 Требования к разрабатываемой игре 15](#_Toc67679735)

[1.6 Выводы по главе 17](#_Toc67679736)

[Глава 2 Проектирование 18](#_Toc67679738)

[2.1 Проектирование алгоритмов 18](#_Toc67679739)

[2.1.1 Модель вариантов использования 18](#_Toc67679740)

[2.1.2 Диаграммы классов 25](#_Toc67679741)

[2.2 Проектирование графического интерфейса 26](#_Toc67679742)

[2.3 Выводы по главе 29](#_Toc67679743)

[Глава 3 Разработка и тестирование. 30](#_Toc67679746)

[3.1 Обоснование выбора средств разработки 30](#_Toc67679753)

[3.2 Описание используемых функций. 31](#_Toc67679754)

[3.3 Тестирование программного продукта 32](#_Toc67679755)

[3.4 Выводы по главе 33](#_Toc67679756)

[Заключение 34](#_Toc67679757)

[Список сокращений и условных обозначений 35](#_Toc67679758)

[Библиографический список 36](#_Toc67679759)

[ПРИЛОЖЕНИЕ А 38](#_Toc67679760)

[ПРИЛОЖЕНИЕ Б 40](#_Toc67679767)

[ПРИЛОЖЕНИЕ В 41](#_Toc67679769)

[ПРИЛОЖЕНИЕ Г 59](#_Toc67679771)

[ПРИЛОЖЕНИЕ Д 73](#_Toc67679773)

[ПРИЛОЖЕНИЕ Е 77](#_Toc67679775)

# Введение

В настоящее время игры пользуются мировой популярностью и дают человеку расслабиться и получить новые эмоции, так же во время пандемии игры стали одним из основных времяпрепровождений с друзьями. За 2020 год количество людей, которые начали играть в игры увеличилось в 1,5 раза и преодолело отметку в 3 миллиарда человек, а рыночная стоимость игровой индустрии увеличилась на 20% (до 174,9 миллиарда долларов) по сравнению с 2019 годом. Во время пандемии игры стали неотъемлемой частью досуга для 40% населения земли и количество новых пользователей будет расти с каждым годом. Особой популярностью пользуются простые многопользовательские игры, так как они не сильно нагружают мозг человека, а также позволяют проводить совместное время с друзьями, когда нет возможности встретиться в живую.

RPG (Role play game) – игры в этом жанре делают акцент на обыгрывание каждым игроком своей роли, что позволяет сильнее погрузиться в игру и представить себя в качестве персонажа игры. Неотъемлемой частью игрового процесса в данном жанре является персонализация персонажа и выбор одной или нескольких ролей в соответствии с предпочтениями игрока. Из-за чего спрос на игры в подобном жанре с каждым годом растет, ведь каждый игрок стремиться получить опыт, который не может получить в реальной жизни.[13]

Актуальность данной темы состоит в том, что игры подобного жанра пользуются огромным спросом, особенно во время и после пандемии, а также в том, что игровой рынок не может предоставить достаточного количество простых, по стилистике игр, которые бы были посты в освоении и визуальной стилистики игры. Именно поэтому разработка игры в жанре «RPG» актуальная тема.

***Объектом*** данного исследования являются компьютерные игры в жанре «RPG». ***Предметом***исследования является разработка алгоритма многопользовательской игры.

***Целью*** данной работы является разработка компьютерной игры в жанре «RPG» с наличием многопользовательской игры.

Задачи, для достижения поставленной цели:

1. Провести анализ действующих алгоритмов.
2. Проанализировать, сравнить и выбрать визуализатор для игры.
3. Изучить способы разработки многопользовательских игр.
4. Спроектировать алгоритм многопользовательской игры.
5. Спроектировать интерфейс для игры.
6. Провести опрос и проанализировать ответы по поводу интерфейса, внести изменения.
7. Реализовать код программы и предварительный интерфейс.
8. Реализовать проект с использованием алгоритма многопользовательской игры.
9. Написать пользовательскую документацию для программы.
10. Провести тестирование, для выявления недочетов и сбора критики тестировщиков.
11. Внести изменения на основании выявленных недочетов на этапе тестирования.

Работа состоит из трех глав:

1. В первой главе будет проведен анализ предметной области, исследование особенностей строения игры и действующих алгоритмов, а также обзор программного обеспечения для написания игр и обзор игр-аналогов. Результатом первой главы являются поставленные требования к написанию работы.
2. Во второй главе будет представлено проектирование будущей игры, включающее в себя проектирование интерфейса и алгоритмов. Результатом главы станет спроектированные алгоритмы и интерфейс игры.
3. В третьей главе будет показана разработка игры и ее алгоритмов и интерфейса, а также представлены результаты тестирования. Результатом главы станет готовый продукт.

Практической значимостью данной работы является расширение знаний в области разработки компьютерных игр и разработки алгоритмов для многопользовательских игр, а также создание многопользовательской игры с возможной последующей публикацией на сервисах дистрибуции игр.

По итогу после завершения данной работы будет разработана многопользовательская игра в жанре «RPG».

# Анализ

В данной главе будут рассмотрены особенности игр в жанре «RPG». Будет выбран визуализатор для игры на основании анализа используемых визуализаторов для игр, выявления их преимуществ, недостатков и особенностей. Так же будут рассмотрены и проанализированы алгоритмы многопользовательских игр, выявлены их недостатки и преимущества. Результатом первой главы являются поставленные требования к разрабатываемой системе.

## Изучение особенностей игр в жанре «RPG»

Компьютерная ролевая игра, обозначаемая аббревиатурой «RPG» - жанр компьютерных игр, базирующийся на элементах игрового процесса традиционных настольных ролевых игр. В игре предоставленного жанра игрок управляет одним или несколькими персонажами, любой из которых описан набором численных характеристик, списком способностей и умений таких как очки здоровья, показатели силы, ловкости и т. п.[9]

Из признаков данного жанра выделяют следующие:

1. Ролевая система
2. Исследование
3. Сюжет
4. Боевая система

Ролевая система – описывает способы создания и развития персонажей[9]. Для внедрения данной системы требуется реализовать необходимые элементы, из которых:

1. Управление одним или несколькими уникальными персонажами;
2. Улучшение внутриигровых значений персонажа пользователя;
3. Пользователь может создавать и улучшать своего персонажа;
4. Пользователь должен планировать развитие персонажа;
5. Основным способом взаимодействия с игровым миром является применение способностей персонажа.

Исследование – описывает способы перемещения персонажа по игровому миру и взаимодействия с ним через нахождение игровых областей, предметы или объекты игрового мира. Для данной системы выделяют следующие элементы:

1. Персонаж пользователя может взаимодействовать с игровым миром, находить новые игровые области, находить предметы и источники информации;
2. Пользователь может выбрать вариант прохождения из нескольких предложенных;
3. Персонаж может воздействовать на игровой мир;
4. Игровой мир может воздействовать на персонажа;
5. Игровые области должны открываться в зависимости от развития персонажа;

Сюжет – описывает элементы повествования, игровой мир, персонажей и способы взаимодействия этих компонентов[9]. Для данной системы стоит выделить следующие элементы:

1. Персонаж пользователя может получать информацию из информационных источников игры с помощью исследования игрового мира;
2. Персонаж пользователя может выполнять задания и продвигаться по цепочке связанных событий;
3. Продвижение по сюжету требует от игрока обдумывания;
4. Сюжет зависит от решений пользователя;
5. В процессе взаимодействия пользователь может сделать выбор;

Боевая система – объясняет влияние ролевой системы, исследований и сюжета на результаты сражения. В сражениях должен присутствовать элемент случайности, а эффективность в сражении зависит от навыков персонажа.

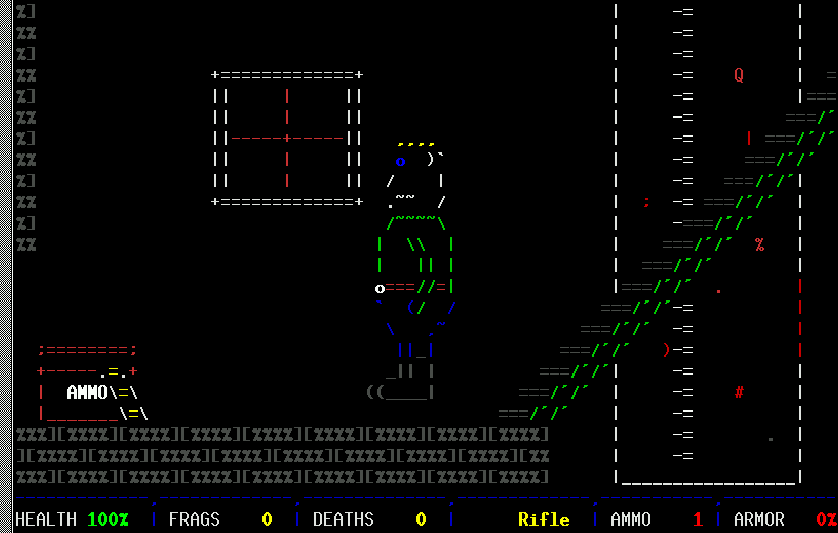
Таким образом данный жанр компьютерных игр представляет огромную вариативность в создании проектов и является важным как для пользователя, так и для разработчика[13].

Из достоинств жанра, главным достоинством является возможность создавать персонализированного персонажа пользователем, что позволяет глубже погрузиться в процесс игры и проводить в ней больше времени. Так же стоит выделить вариативность игрового пространства, которая предоставляет возможность создавать игровые уровни с разным визуальным стилем и разным построением.

Недостатком жанра «RPG» является создание игрового пространства, для размещения в нем сюжета игры, так как на разработку таких пространств уходит большое количество времени[13].

## Изучение особенностей строения игровых уровней

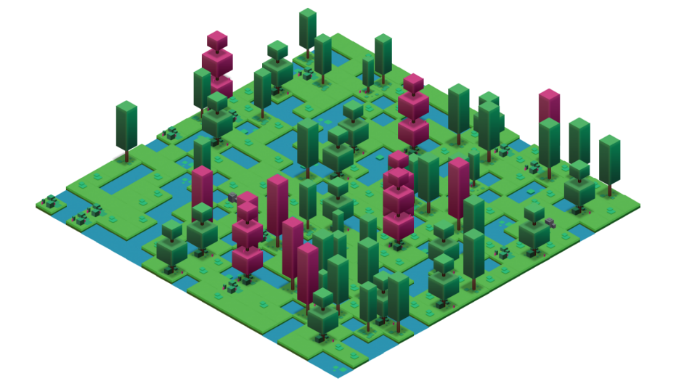
Построение карт в играх не является строгим. В некоторых играх количество уровней и обширность игрового мира может составлять тысячи или десятки тысяч локаций. Игровое пространство может организовываться по-разному, из основных видов построения выделяют:

1. Текстовая графика (см. Рисунок 1.1).

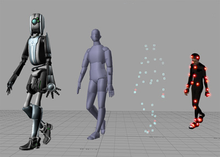
**Рисунок 1.1 – Пример текстовой графики**

1. Двумерная графика (см. Рисунок 1.2).

**Рисунок 1.2. Пример двухмерной графики**

1. Изометрическая графика (см. Рисунок 1.3).

**Рисунок 1.3. Пример изометрической графики**

1. Трехмерная графика (см. Рисунок 1.4).

**Рисунок 1.4. Пример трехмерной графики**

Для создания игрового поля различают процедурную и фиксированную генерацию. Для процедурной генерации свойственно автоматическое создание и возведение игрового пространства с использованием алгоритмов, этот способ увеличивает реиграбельность каждой новой игры, но приводит к усложнению игрового процесса, шанс выигрыша при подобной генерации значительно уменьшается, а сложность игры увеличивается. Фиксированная генерация используется чаще, так как при таком варианте игру возможно спроектировать так, чтобы гарантировать её прохождение.[10]

При наличии многопользовательской игры фиксированная генерация преобладает, так как процедурная генерация может облегчить игровой процесс для некоторых пользователей, что будет являться нечестным по отношению с остальными пользователями. Так же процедурная генерация может привести появлению ошибок в генерации игровых пространств, что может повлечь за собой застревание или выход за область игрового пространства или недоступность прохождения игрового пространства.[7]

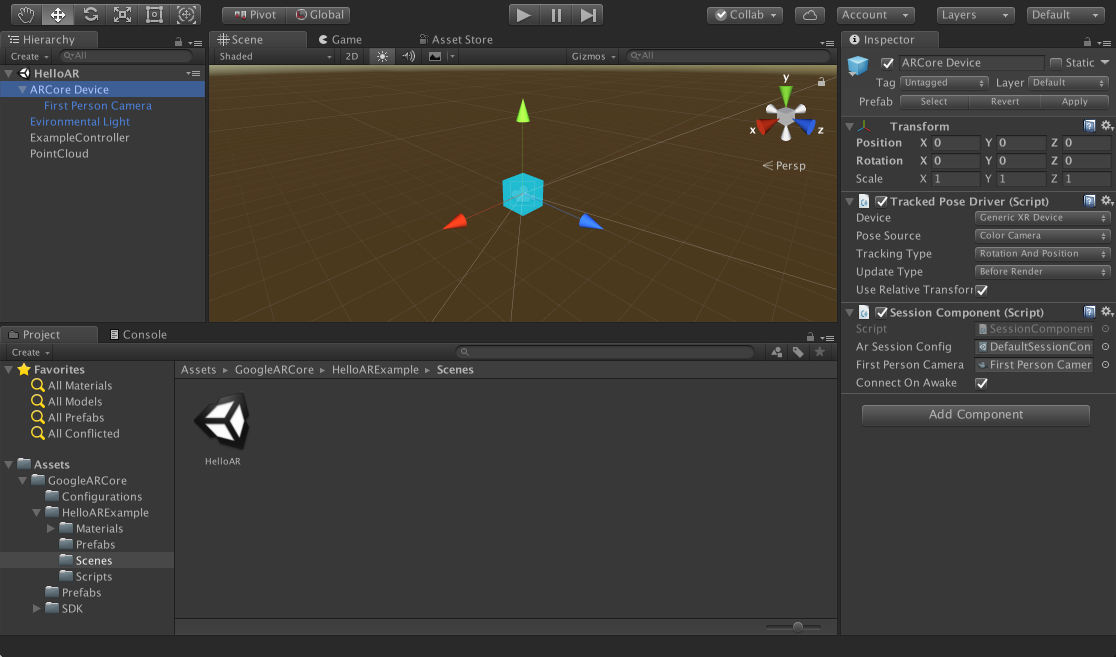
## Обзор и сравнение программного обеспечения для создания игр

Для создания и отображения игр требуется визуализатор, чтобы отобразить код программы в изображение. В качестве визуализатора для несложных и не затратных игр можно применить и обычные визуализаторы языка, к примеру, Visual Studio. Но для реализации более комплексных алгоритмов приходится сильнее углубляться в изучение языка и основы работы различных классов. Таким образом даже имея высокую квалификацию разработчик программного обеспечения имеет возможность легко допустить трудноуловимые ошибки, некоторые из которых имеют все шансы оказаться фатальными.[3]

Ввиду этого были разработаны программы для создания игр, которые не только упрощают разработку игры и избавляют от большого количества ошибок, но и привносят что-то новое, чего невозможно добиться обычными визуализаторами. Это программное обеспечение, или же иначе «game engine», в данный момент применяется практически во всех играх. Из ведущих применяемых программных обеспечений стоит отметить следующие: Unity и Unreal Engine.

**Unity** - это мультиплатформенная среда для разработки  2D и 3D игр и интерактивного контента. Первый выпуск состоялся в 2005 году и с того времени постоянно обновлялся, вводя что-то новое и исправляя старое. Основные преимущества Unity это наличие визуальной среды разработки, обширная база данных, межплатформенная поддержка и абсолютная бесплатность для начинающих разработчиков[4]. Данная технология наиболее подходит для создания игрового пространства и интерактивного интерфейса компьютерной игры. Основной язык программирования – C##, так же поддерживается и JavaScript. На рисунке 1.5 показан интерфейс программы.

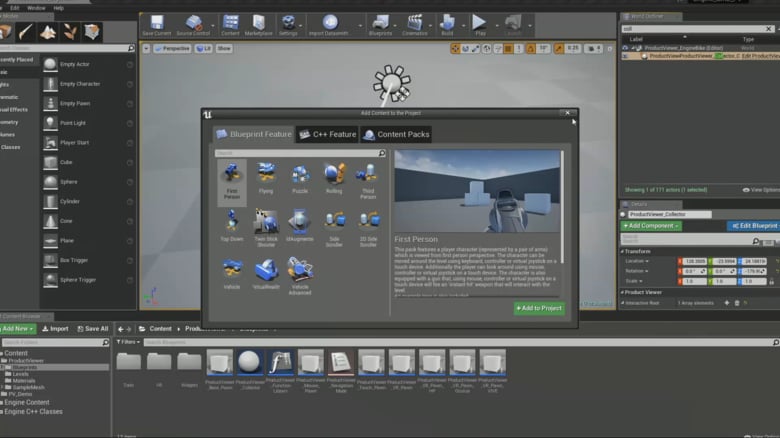
**Рисунок 1.5 – Интерфейс Unity**



**Unreal Engine** – среда для разработки компьютерных игр, разработанная в 1998 году и являющаяся собственностью компании «Epic Games». Последняя версия – Unreal Engine 4, вышедшая в 2014 году, позволяет создавать приложения на десяти операционных системах. Из преимуществ выделяют инструмент для создания уровней, интегрированный в программу и не имеющий аналогов и поддержка различных систем рендеринга. С 2015 года стала бесплатной, для всех разработчиков. Основной язык программирования – С++. В основном направлена на создание 3D игр и фильмов. Интерфейс программы представлен на рисунке 1.6.

Для того чтобы выбрать программное обеспечение для написания системы игры, нужно сравнить представленные выше программы по критериям. В таблице 1.1 выявляем основные требования к среде в которой будем создавать игру. На основе имеющихся данных можно определить какая среда для написания игр подходит лучше всего для создаваемого программного продукта.

**Рисунок 1.6 – Интерфейс Unreal Engine**



**Таблица 1.1 - Критерии сравнения среды для написания игры**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Среда | Поддерживаемые форматы | Порог вхождения | Основной язык программирования | Наличие документации | Стоимость лицензии |
| 1. Unity | 2D, 3D | Низкий | C# | да | Бесплатно |
| 2. Unreal Engine | 2D, 3D | Высокий | C++ | да | Бесплатно |

На основании сравнениия, удобными в использовании является Unreal Engine. Главными же факторами, для начинаюзего разработчика являются наличие документации и бесплатная лицензия. Таким образом лучшим вариантом для создания тестового проекта служит программное обеспечение Unity из-за большого количества документации, бесплатной лицензии и низкого порога вхождения. При модернизации проекта для публикации стоит перенести игру на язык C++, так как он является более энергоемким и ресурсоемким, из-за чего показатели быстродействия увеличатся.

## Обзор игр-аналогов

Для выявления преимуществ разрабатываемой игры необходимо провести анализ игр-аналогов и выявить их преимущества и недостатки.

Так как жанр «RPG» игр является одним из самых популярных, то в рамках данной курсовой работы будут рассмотрены более приближенные игры-аналоги. Из аналогов стоит выделить следующие:

* + 1. Enter the Gungeon;
    2. Darkest Dungeon;
    3. World of Warcraft;
    4. Genshin Impact;

Enter the Gungeon – компьютерная игра в жанрах «RPG» и «Roguelike» с видом сверху, выпущенная 5 апреля 2016. Игра написана на движке Unity. Из преимуществ стоит выделить возможность запуска игры на слабых компьютерах, а так же захватывающий игровой процесс и интерфейс игры. К минусам относятся отсутствие полноценной многопользовательской игры, отсутствие сюжета и отсутствие полноценной персонализации персонажа, так же игра является платной. Интерфейс игры представлен на рисунке 1.7.

**Рисунок 1.7 – Интерфейс Enter the Gungeon**

Darkest Dungeon – компьютерная ролевая игра с roguelike-элементами, разработанная и выпущенная независимой канадской студией Red Hook Studios 19 января 2016 года. Игра написана на движке Unity. К плюсам данной игры стоит отнести полное соответствие жанру «RPG» и возможность запускать игру в оффлайн режиме. Из минусов стоит выделить отсутствие многопользовательской игры, малая вариативность игрового процесса и отсутствие персонализации, так же игра является платной. Интерфейс игры представлен на рисунке 1.8.

**Рисунок 1.8 – Интерфейс Darkest Dungeon**

World of Warcraft – глобальная многопользовательская ролевая онлайн-игра, созданная и издаваемая компанией Blizzard Entertainment с 23 ноября 2004 года. К плюсам данной игры стоит отнести возможность многопользовательской игры и обширную персонализацию персонажа. Из минусов преобладают отсутствие возможности запуска на слабых компьютерах, а также условная бесплатность игры. Интерфейс игры представлен на рисунке 1.9.

**Рисунок 1.9 – Интерфейс World of Warkraft**

Genshin Impact – компьютерная игра в жанре Action-adventure с открытым миром и элементами RPG, разработанная китайской компанией miHoYo Limited в 2020. Игра была создана на движке Unity. Из плюсов стоит выделить возможность бесплатной игры, возможность запуска на слабых компьютерах и мобильных устройствах, а также проработанный сюжет. К минусам относятся урезанный вариант многопользовательской игры и отсутствие полноценной персонализации персонажа. Интерфейс игры представлен на рисунке 1.10.

**Рисунок 1.10 – Интерфейс Genshin Impact**

Выделим основные критерии для сравнения аналогов и разрабатываемой игры (см. Таблица 1.2):

Присутствует возможность играть онлайн через интернет?

Присутствует возможность запускать игру в оффлайн режиме?

Присутствует возможность многопользовательской игры?

Присутствует возможность персонализации персонажа?

Присутствует сюжет в игре?

Присутствует возможность запуска на мобильных устройствах?

Присутствует возможность запуска на слабых компьютерах?

Присутствуют варианты игры без продвижения по сюжету (миниигры)?

Есть возможность бесплатной игры?

**Таблица 1.2 - Критерии сравнения среды для написания игры**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Среда | Enter the Gungeon | Darkest Dungeon | World of Warcraft | Genshin Impact | Разрабатываемая игра |
| Возможность играть онлайн через интернет | - | - | + | + | + |
| Возможность запускать игру в оффлайн режиме | + | + | - | - | + |
| Возможность многопользовательской игры | + | - | + | + | + |
| Возможность персонализации персонажа | + | - | + | + | + |
| Наличие сюжета в игре | - | + | + | + | + |
| Возможность запуска на мобильных устройствах | - | - | - | + | + |
| Возможность запуска на слабых компьютерах | + | + | - | - | + |
| Игры без продвижения по сюжету | - | - | + | + | + |
| Возможность бесплатной игры | - | - | + | + | + |

По полученным результатам можно сделать вывод, что разрабатываемая игра будет лучше аналогов из-за возможности запуска на мобильных устройствах, возможности бесплатной игры, возможности запуска как в онлайн, так и в оффлайн режиме, а также из-за возможности многопользовательской игры по сети.

Таким образом стоит выделить перечень требований, заимствованных у аналогов:

1. Возможность запускать игру в оффлайн и онлайн режиме;
2. Возможность многопользовательской игры;
3. Возможность персонализации персонажа;
4. Возможность запуска на мобильных устройствах и слабых компьютерах;
5. Наличие сюжета в игре;
6. Возможность бесплатной игры;
7. Наличие игр без продвижения по сюжету;

## Требования к разрабатываемой игре

Для разрабатываемой системы были определены функциональные и нефункциональные требования:

Функциональные требования:

1. Система должна загружать выбранные пользователем игровые пространства;
2. Система должна предоставлять возможность передвижения пользователя по игровому пространству;
3. Система должна взаимодействовать с пользователем и откликаться на его действия;
4. Система должна предоставлять пользователю возможность подключения к сети интернет;
5. Система должна оповещать пользователя при возникновении ошибок;
6. Система должна давать пользователю возможность персонализации персонажа;
7. Система должна предоставлять возможность выхода из системы;
8. Уровни игры должны быть спроектированы с помощью фиксированной генерации;

Нефункциональные требования:

1. Интерфейс игры должен быть простым для восприятия пользователя;
2. Архитектура системы должна позволять добавлять к нему дополнительные модули не нарушая работоспособность других модулей;
3. Система должна отвечать на запрос пользователя не более 5 секунд;
4. Функции ввода и вывода данных должны быть спроектированы таким образом, чтобы их удавалось повторно использовать на уровне объектного кода в других приложениях;
5. Архитектура игровых пространств должна позволять добавлять к нему новые игровые пространства не нарушая целостность других игровых пространств;
6. Система должна предоставлять интуитивность в использовании пользователем;
7. Система должна предоставлять безопасность данных пользователей;
8. Система должна своевременно обновляться;
9. Система должна давать пользователю возможность изменять настройки системы;
10. Система должна предоставлять возможность авторизации пользователя;
11. Интерфейс системы должен быть читабельным и понятным для восприятия;
12. Система должна давать пользователю возможность принимать решения в игровом пространстве;
13. Система должна предоставлять возможность запуска на слабых компьютерах;

## Выводы по главе

В результате анализа игрового жанра и особенностей строения игровых уровней был выбран вариант визуализации игрового процесса на основании изучения особенности строения игровых уровней и приняты решения по внедрению механик игры подходящих жанру «RPG». Были выдвинуты функциональные требования, предъявляемые к системе. Проведен краткий обзор и сравнение основных движков, а также выбран движок для реализации игры. Рассмотрены плюсы и минусы аналогов и определены преимущества разрабатываемой игры.

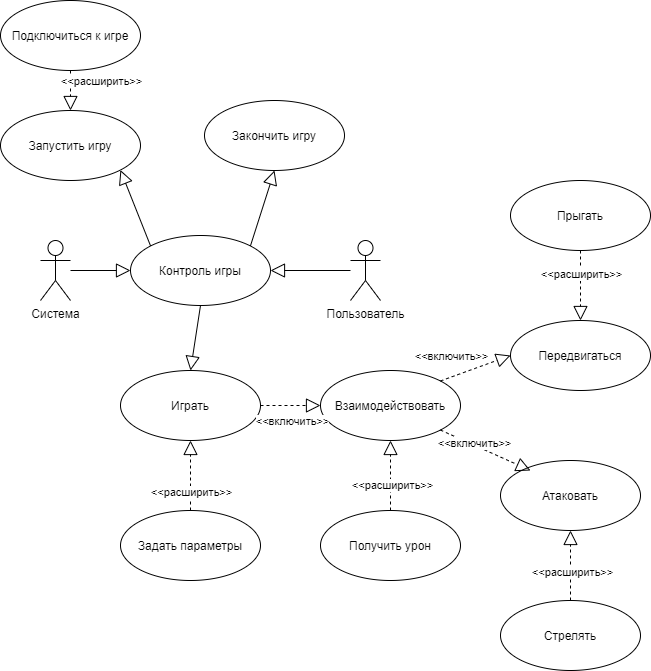


# Проектирование

В представленной главе спроектированы и рассмотрены методы и алгоритмы, которые потребуются для реализации игры и для удовлетворения требований, выявленных в ходе анализа. Результатом данной главы является спроектированная игра и её интерфейс.

## Проектирование алгоритмов

### **2.1.1 Модель вариантов использования**

Визуализация требований реализована с помощью use-case диаграммы (см. рисунок 2.1), созданная при помощи средств моделирования UML.[5]

**Рисунок 2.1 – Диаграмма прецедентов**

Документирование прецедентов представлено ниже в таблицах (см. таблицы 2.1. – 2.11).

*Таблица 2.1 – Прецедент «Контроль игры»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-1. Контроль игры. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь.  Система. | | |
| Описание | Прецедент позволяет взаимодействовать пользователю и системе в игровом процессе. | | |
| Триггер | Пользователь запускает игру. | | |
| Предварительные условия | PRE-1. Наличие подключения к сети интернет.  PRE-2. Наличие на диске файла с сохраненной конфигурацией игры. | | |
| Выходные условия | POST-1. Игра запущена.  POST-2. Пользователь отправляет команды системе. | | |
| Основной поток | 1.0 Запуск игры:   1. Выполняется прецедент «Запустить игру» (см 1.0.Е1 и 1.0.Е2) | | |
| Альтернативные потоки | - | | |
| Исключения | 1.0.Е1. Отсутствует подключение к сети интернет.   1. Система выводит сообщение о том, что отсутствует подключение к сети интернет. 2. Система возвращается к главному экрану.   1.0.Е2. Файлы игры повреждены.   1. Система выводит сообщение о том, что файлы игры повреждены. 2. Система возвращается к главному экрану. | | |
| Приоритет | Высокий | | |
| Точка расширения | UC-2. Запустить игру.  UC-3. Закончить игру.  UC-4. Играть. | | |

*Таблица 2.2 – Прецедент «Запустить игру»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-2. Запустить игру. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь.  Система. | | |
| Описание | Прецедент позволяет пользователю запустить игру в онлайн или оффлайн режиме. | | |
| Триггер | Пользователь запускает игру. | | |
| Предварительные условия | PRE-1. Наличие подключения к сети интернет.  PRE-2. Наличие на диске файла с сохраненной конфигурацией игры. | | |
| Выходные условия | POST-1. Игра запущена. | | |
| Основной поток | 1.0 Запуск игры:   1. Система показывает главный экран. 2. Пользователь нажимает на кнопку «Играть онлайн». (см. 1.1) 3. Система выводит окно загрузки. 4. Система проверяет подключение к сети интернет. (см. 1.0.Е1) 5. Выполняется прецедент «Подключиться к игре». | | |
| Альтернативные потоки | 1.1 Пользователь выбрал оффлайн игру:   1. Пользователь нажимает на кнопку «Играть оффлайн». 2. Система выводит окно загрузки. 3. Система проверяет наличие файла сохранения. (см. 1.1.Е1) 4. Система восстанавливает конфигурацию и состояние из файла и загружает игровое поле. | | |
| Исключения | 1.0.Е1. Отсутствует подключение к сети интернет.   1. Система выводит сообщение о том, что отсутствует подключение к сети интернет. 2. Система возвращается к главному экрану.   1.1.Е1. Файл сохранения поврежден.   1. Система выводит сообщение о том, что файл поврежден. 2. Система возвращается к главному экрану. | | |
| Приоритет | Высокий | | |
| Точка расширения | UC-6. Подключиться к игре. | | |

*Таблица 2.3 – Прецедент «Закончить игру»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-3. Закончить игру. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь.  Система. | | |
| Описание | Прецедент позволяет выйти из системы. | | |
| Триггер | Пользователь нажимает на кнопку «Выйти». | | |
| Предварительные условия | - | | |
| Выходные условия | POST-1. Игра завершена. | | |
| Основной поток | 1.0 В игре:   1. Игрок нажимает кнопку «Выйти в меню». 2. Система запрашивает подтверждение выхода «Да\Нет». 3. Система выходит в главное меню.   2.0 В главном меню:   1. Игрок нажимает кнопку «Выйти». 2. Система запрашивает подтверждение выхода «Да\Нет». 3. Система завершает свое выполнение и выходит на рабочий стол. | | |
| Альтернативные потоки | - | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | - | | |

*Таблица 2.4 – Прецедент «Играть»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-4. Играть. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь.  Система. | | |
| Описание | Прецедент представляет игровой процесс. | | |
| Триггер | Пользователь запускает игру. | | |
| Предварительные условия | PRE-1. Выполнен прецедент «Запустить игру». | | |
| Выходные условия | POST-1. Пользователь играет в игру. | | |
| Основной поток | 1.0 Начало игры:   1. Система запускает основное игровое пространство. 2. Система позволяет пользователю выбирать следующие действия. (см 1.1, 1.2) 3. Система позволяет пользователю передвигаться и взаимодействовать с игровым пространством. 4. Пользователь передвигается к месту сражения. 5. Пользователь выбирает оружие для сражения.   2.0 Фаза выбора оружия:   1. Пользователь выбирает вид оружия (ближнее или дальнее).   3.0 Фаза сражения:   1. Пользователь атакует противника в ближнем бою. (см 3.1) 2. Пользователь уклоняется от ударов противника. 3. Пользователь выигрывает в бою (см 3.2)   4.0 Завершение боя:   1. Пользователь выигрывает сражение. 2. Система вознаграждает пользователя очками опыта и игровой валютой. 3. Система отправляет пользователя в основное игровое пространство. 4. Система позволяет пользователю выбирать следующие действия. (см 1.1, 1.2) | | |
| Альтернативные потоки | 1.1 Пользователь нажал кнопку «Выйти»:   1. Выполняется прецедент «Закончить игру»   1.2 Пользователь открыл настройки игры:   1. Выполняется прецедент «задать параметры»   3.1 Пользователь выбрал дальнее оружие:   1. Пользователь атакует противника в дальнем бою.   3.2 Пользователь проигрывает в бою:   1. Система отправляет пользователя в основное игровое пространство. | | |
| Исключения | - | | |
| Приоритет | Высокий | | |
| Точка расширения | UC-5. Задать параметры. | | |

*Таблица 2.5 – Прецедент «Задать параметры»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-5. Задать параметры. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет пользователю задать параметры игры | | |
| Триггер | Пользователь нажимает на кнопку «Параметры» или на клавишу «Esc». | | |
| Предварительные условия | PRE-1. Нажата кнопка «Параметры» или клавиша «Esc». | | |
| Выходные условия | POST-1. Параметры сохранены.  POST-2. Игра настроена в соответствии с параметрами. | | |
| Основной поток | 1.0 Изменение параметров:   1. Система открывает окно настроек игры. 2. Пользователь вносит изменения в параметры игры. | | |
| Альтернативные потоки | - | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | - | | |

*Таблица 2.6 – Прецедент «Подключиться к игре»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-6. Подключиться к игре. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь.  Система. | | |
| Описание | Прецедент подключает пользователя к игре. | | |
| Триггер | Пользователь нажимает на кнопку «Играть». | | |
| Предварительные условия | PRE-1. Выполнен прецедент «Запустить игру». | | |
| Выходные условия | POST-1. Пользователь подключился к игре. | | |
| Основной поток | 1.0 Выполняется подключение к игре:   1. Система считывает файл с конфигурацией игры. 2. Система запрашивает данные с сервера с данными о игровом пространстве. 3. Система получает данные с сервера. 4. Система загружает игровое пространство пользователя. (см. 1.1) | | |
| Альтернативные потоки | 1.1 Подключение к чужой игре:   1. Система загружает игровое пространство другого пользователя основываясь на данных с сервера. | | |
| Исключения | - | | |
| Приоритет | Высокий | | |
| Точка расширения | - | | |

*Таблица 2.7 – Прецедент «Взаимодействовать»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-7. Взаимодействовать. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет пользователю взаимодействовать с игровым миром. | | |
| Триггер | Пользователь нажимает на кнопки «W», «A», «S», «D», «Пробел» или «ЛКМ» или игровой мир взаимодействует с пользователем. | | |
| Предварительные условия | PRE-1. Запущен прецедент «Играть». | | |
| Выходные условия | POST-1. Пользователь взаимодействует с игровым миром. | | |
| Основной поток | 1.0 Передвижение персонажа:   1. Пользователь нажал кнопку «W», «A», «S», «D». 2. Система производит анимацию передвижения. 3. Система выполняет прецедент «Передвигаться».   2.0 Прыжок персонажа:   1. Пользователь нажал кнопку «Пробел». 2. Система выполняет прецедент «Прыгать».   3.0 Атака персонажа:   1. Пользователь нажал кнопку «ЛКМ». 2. Система выполняет прецедент «Атаковать».   4.0 Пользователь получает урон:   1. Система выполняет прецедент «Получить урон». 2. Система выполняет прецедент «Атаковать» 3. Система производит анимацию получения урона пользователем. (см. 4.1) | | |
| Альтернативные потоки | 4.1 Пользователь выполнил прецедент «Передвигаться»:   1. Пользователь уклоняется от получения урона. | | |
| Исключения | - | | |
| Приоритет | Высокий | | |
| Точка расширения | UC-8. Получить урон. | | |

*Таблица 2.8 – Прецедент «Получить урон»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-8. Получить урон. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет системе атаковать пользователя и получать урон пользователем. | | |
| Триггер | Система атакует пользователя. | | |
| Предварительные условия | PRE-1. Запущен прецедент «Играть».  PRE-2. Запущена фаза сражения. | | |
| Выходные условия | POST-1. Пользователь получает урон и теряет очки здоровья. | | |
| Основной поток | 1.0 Получение урона:   1. Система атакует пользователя. 2. Система проверяет уклонился ли пользователь. 3. Система производит анимацию попадания. 4. Система генерирует число урона основываясь на уровне пользователя и очках защиты пользователя. 5. Система вычитает нанесенный урон из очков здоровья пользователя. 6. Система проверяет очки здоровья пользователя (если >0). 7. Система продолжает фазу сражения прецедента «Играть». (см. 1.1) | | |
| Альтернативные потоки | 1.1 Очки здоровья меньше 0:   1. Система производит анимацию смерти персонажа. 2. Система оповещает о смерти персонажа. 3. Система загружает основное игровое пространство. | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | - | | |

*Таблица 2.9 – Прецедент «Передвигаться»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-9. Передвигаться. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет пользователю передвигаться в игровом мире. | | |
| Триггер | Пользователь нажимает на кнопки «W», «A», «S», «D». | | |
| Предварительные условия | PRE-1. Выполнен прецедент «Запустить игру».  PRE-2. Запущен прецедент «Играть». | | |
| Выходные условия | POST-1. Параметры сохранены.  POST-2. Игра настроена в соответствии с параметрами. | | |
| Основной поток | 1.0 Передвижение персонажа:   1. Пользователь нажал кнопку «W». (см. 1.1, 1.2, 1.3) 2. Система производит анимацию передвижения. 3. Система передвигает персонажа вглубь. | | |
| Альтернативные потоки | 1.1 Пользователь нажал на кнопку «A»:   1. Система производит анимацию передвижения. 2. Система передвигает персонажа в левее.   1.2 Пользователь нажал на кнопку «S»:   1. Система производит анимацию передвижения. 2. Система передвигает персонажа вблизь.   1.3 Пользователь нажал на кнопку «D»:   1. Система производит анимацию передвижения. 2. Система передвигает персонажа в правее. | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | UC-10. Прыгать. | | |

*Таблица 2.10 – Прецедент «Прыгать»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-10. Прыгать. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет пользователю прыгать в игровом пространстве. | | |
| Триггер | Пользователь нажимает на кнопку «Пробел». | | |
| Предварительные условия | PRE-1. Выполнен прецедент «Запустить игру».  PRE-2. Запущен прецедент «Играть». | | |
| Выходные условия | POST-1. Персонаж прыгнул. | | |
| Основной поток | 1.0 Прыжок персонажа:   1. Пользователь нажал кнопку «Пробел». 2. Система выполняет анимацию прыжка. | | |
| Альтернативные потоки | - | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | - | | |

*Таблица 1.11 – Прецедент «Атаковать»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-11. Атаковать. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет пользователю атаковать противников в игре. | | |
| Триггер | Пользователь нажимает на кнопку «ЛКМ». | | |
| Предварительные условия | PRE-1. Выполнен прецедент «Запустить игру».  PRE-2. Запущен прецедент «Играть».  PRE-3. Запущена фаза сражения. | | |
| Выходные условия | POST-1. Пользователь атакует. | | |
| Основной поток | 1.0 Атака персонажа:   1. Пользователь нажал кнопку «ЛКМ». 2. Система проверяет выбор оружия пользователя. 3. Система выполняет анимацию атаки ближнего боя. (см. 1.1) | | |
| Альтернативные потоки | 1.1 Атака из дальнего оружия:   1. Система выполняет прецедент «Стрелять». | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | UC-12. Стрелять. | | |

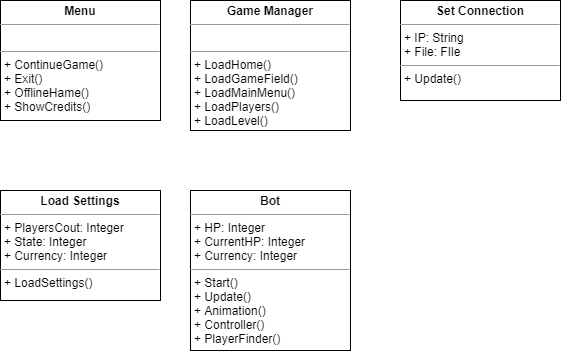
*Таблица 1.12 – Прецедент «Стрелять»*

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор и название | UC-12. Стрелять. | | |
| Автор | Чепоков Елизар | Дата | 02.02.2021 |
| Акторы | Пользователь. | | |
| Описание | Прецедент позволяет пользователю атаковать противников в игре. | | |
| Триггер | Пользователь нажимает на кнопку «ЛКМ». | | |
| Предварительные условия | PRE-1. Выполнен прецедент «Запустить игру».  PRE-2. Запущен прецедент «Играть».  PRE-3. Запущена фаза сражения. | | |
| Выходные условия | POST-1. Пользователь атакует. | | |
| Основной поток | 1.0 Атака персонажа:   1. Пользователь нажал кнопку «ЛКМ». 2. Система выполняет анимацию атаки дальнего боя. | | |
| Альтернативные потоки | - | | |
| Исключения | - | | |
| Приоритет | Средний | | |
| Точка расширения | - | | |

### **2.1.2 Диаграммы классов**

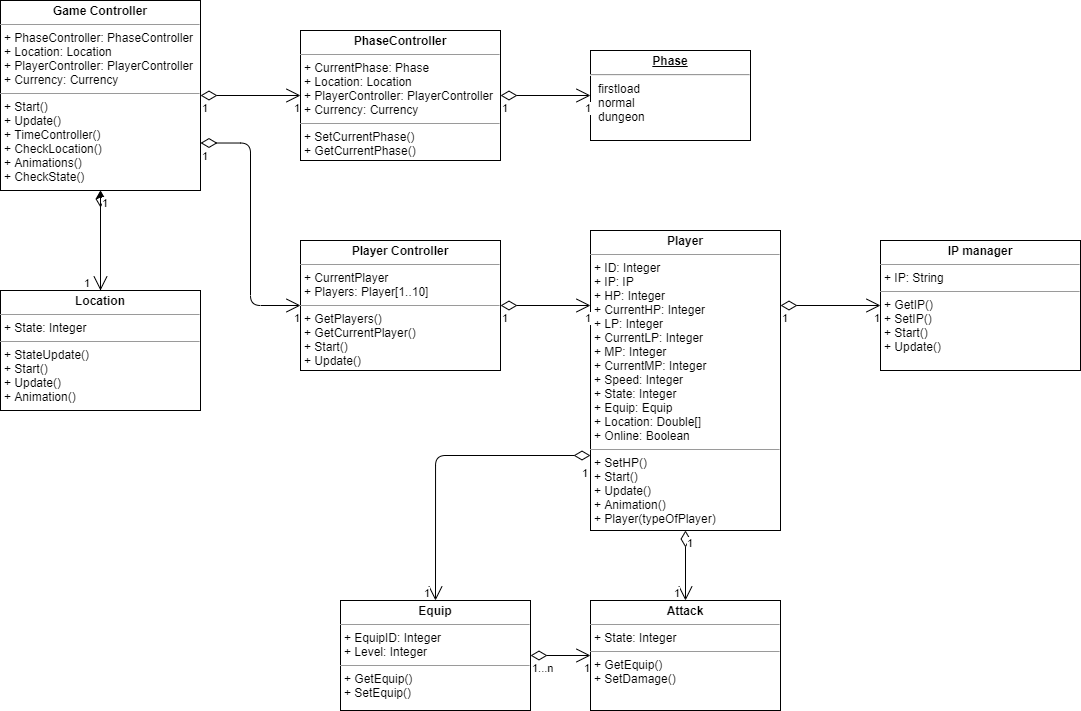
Для демонстрации игровой логики были созданы диаграммы классов. Диаграмма, изображенная на рисунке 2.2 описывает игровую логику при инициализации игры и отображение игровых объектов при загрузке игрового мира.

**Рисунок 2.2 – Диаграмма классов при инициализации**



На рисунке 2.3 представлена диаграмма классов, описывающая игровую логику.

***Рисунок 2.3 – Диаграмма классов***

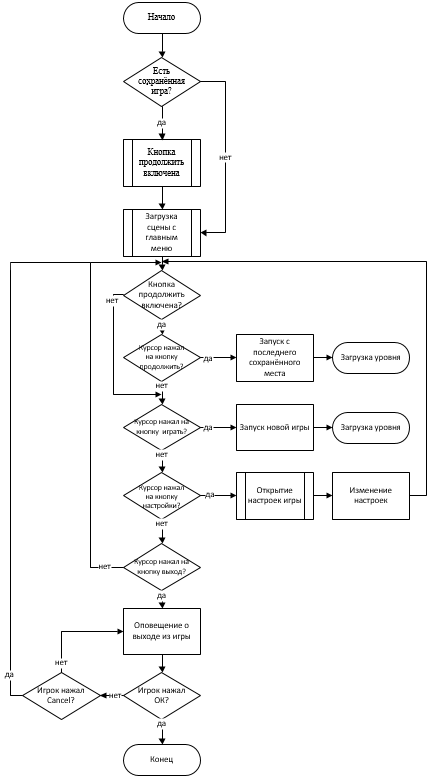


Данные классы связаны классом Game Controller отвечающим за игровой процесс и запускающим события остальных классов системы.

## Проектирование графического интерфейса

Для проектирования графического интерфейса требуется визуализировать основные алгоритмы работы интерфейса для последующей адаптации данных алгоритмов с помощью инструментов разработки Unity.

На рисунке 2.4 изображен алгоритм главного меню, которое будет запущено при запуске игры. На данном алгоритме представлен процесс выбора следующих действий пользователем.

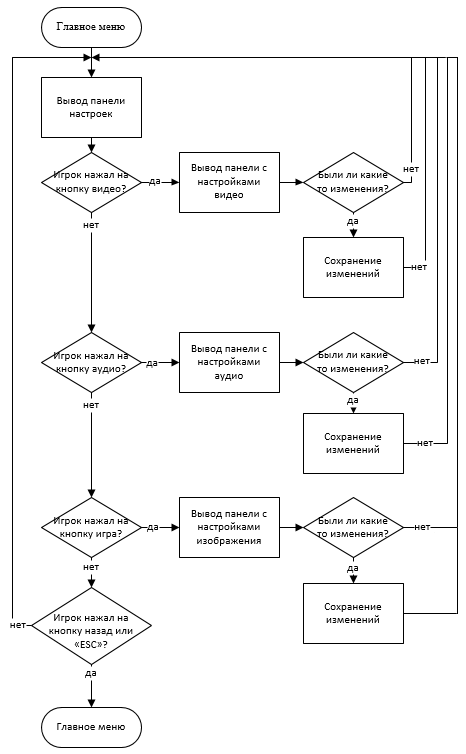


***Рисунок 2.4 – Алгоритм работы главного меню***

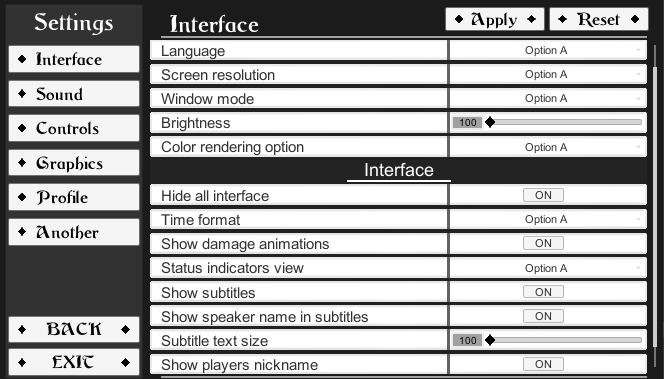
Из данного алгоритма следует что в первый запуск программы пользователю будет доступно всего три кнопки: «Играть», «Настройки» и «Выход», из-за того, что это первый запуск программы и сохранений еще нет, то кнопка «Продолжить» деактивирована и будет показана при следующих загрузках.

Кнопка «Играть» выполняет первую загрузку уровня и инициализацию всех игровых объектов. При подключении к своему миру так же генерирует ключ комнаты по которому другие игроки смогут подключиться к миру пользователя.

Кнопка «Настройки» выводит окно настроек для задания параметров конфигурации игры. В данном окне (см. Рисунок 2.6) будут располагаться варианты отображения графики, уровень прорисовки изображения, уровни громкости и изменяемые кнопки для взаимодействия пользователя с игровым пространством. На рисунке 2.5 представлен алгоритм работы окна настроек.



***Рисунок 2.5 – Алгоритм работы окна настроек***

Кнопка «Выход» позволяет пользователю выйти из приложения.

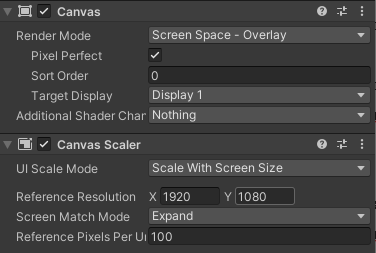
***Рисунок 2.6 – Интерфейс настроек***

Кнопка «Продолжить» позволяет загрузить игру с заранее сохраненным прогрессом.

Для реализации интерфейса игры была использована система UI игрового движка Unity. Все элементы UI расположены на полотне «Canvas», которое определяет размещение 2D объектов в игровом пространстве. За это отвечает свойство Render Mode (см. Рисунок 2.7), которое содержит следующие значения:

**Screen Space - Overlay.** Полотно масштабируется по размерам экрана и рисуется поверх всех остальных графических элементов без связи со сценой или камерой. В случае изменения размера окна, полотно будет растянуто под размеры экрана.

**Screen Space - Camera.** Полотно рисуется на плоскости перпендикулярной взгляду камеры, на некотором расстоянии от точки взгляда. Размер масштабируется относительно заданных значений, чтобы заполнять разрез видимости камеры. Интерфейс будет заслоняться любыми 3D элементами, которые находятся перед плоскостью интерфейса.

**World Space.** Полотно располагается в мировых координатах и является плоским 3D объектом.

***Рисунок 2.7 – Настройки полотна «Canvas»***

На рисунке 2.8 показан построенный макет главного меню. Полотно, на котором располагаются UI элементы имеет свойство RenderMode с установленным значением Screen Space – Overlay.

***Рисунок 2.8 – Главное меню***

## Выводы по главе

На основании результатов, полученных из первой главы, была спроектирована алгоритмическая часть архитектуры игры, представленная в виде диаграммы прецедентов и диаграммы классов. Так же был спроектирован графический интерфейс пользователя с помощью движка Unity. При помощи блок-схем описан алгоритм действий пользователя.



# Разработка и тестирование.

Данная глава обозревает реализацию и тестирование конечного продукта. В данной главе рассматриваются все основы создания программы и тестирование. Итоговым результатом третьей главы является конечный продукт.



## Обоснование выбора средств разработки

На основании сопоставления программного обеспечения во время анализа, для создания программного продукта, в качестве среды проектирования игры было выбрано программное обеспечение Unity с поддержкой Visual Studio, на основе языка программирования С#. [2]

Предоставленная система была выбрана по критериям, представленным ниже:

* 1. Большое количество документации и справочников, помогающие разобраться в написании игры.
  2. Наличие большого набора структурных типов данных.
  3. Лёгкость и удобность программного обеспечения.
  4. Лицензия программного обеспечения является бесплатной.
  5. Поддержка разработки трехмерных игр.

Одной из главных особенностей игрового движка Unity является удобный и полностью настраиваемый интерфейс, который сочетает в себе редактор сцен, игровых объектов, скриптов, анимации, а также предоставляет возможность мгновенного запуска и отладки разрабатываемого приложения или игры [6]. Все ресурсы и объекты могут перемещаться при помощи Drag-and-Drop методов.

В качестве инструмента для написания программного кода была выбрана среда Visual Studio 2019. В данной работе Visual Studio используется как лучшая альтернатива MonoDevelop, которая в игровом движке Unity представлена по умолчанию.

При будущей модернизации проекта возможен перенос всего проекта на язык C++ с использованием в качестве среды проектирования Unreal Engine, из-за большей ресурсоемкости и скорости работы языка программирования C++ и последующее сравнение скорости работы. [4]

Для реализации многопользовательского соединения было принято решение использовать встроенную библиотеку Unity «Photon», так как данная библиотека является единственной библиотекой, подходящей для многопользовательского подключения для игр, созданных на платформе Unity. [7]

## Описание используемых функций.

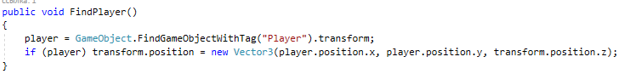
Для работы скриптов Unity существуют обязательные функции, которые обновляются на каждый кадр в зависимости от времени работы программы.

Функция «Awake» вызывается при запуске экземпляра скрипта. Данная функция применяется для инициализации переменных и состояния игры до того, как игра будет загружена. Так как эта функция вызывается всего один раз после запуска, а также в случайном порядке среди других скриптов, стоит отказаться от ее применения для передачи информации или установки связи между скриптами и использовать вместо нее функцию «Start», которая разрешает упорядочить инициализацию скриптов.

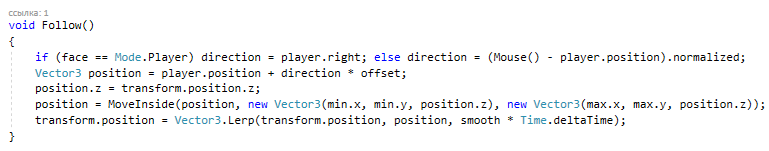
Функция «Update» вызывается перед отрисовкой кадра игры и перед прорисовкой анимации объектов, используется для анимирования и просчета кадров статичных объектов. Для не статичных объектов существует отдельная функция «FixedUpdate», которая просчитывает физику и анимацию объекта и обновляется фиксированными по времени шагами. [3]

Одним из главных скриптов в игре, является скрипт передвижения персонажа. На рисунке 3.1 представлен фрагмент кода, отвечающий за основные действия персонажа.

***Рисунок 3.1 – Скрипт персонажа***

Для противников в игре была написана функция поиска и преследования персонажа, при сближении противник начинает атаковать. Ниже представлены фрагменты кода данных функций. (см. Рисунок 3.2 и 3.3)

***Рисунок 3.3 – Скрипт поиска***

Остальные функции представлены в полном листинге программы в ПРИЛОЖЕНИИ В.

***Рисунок 3.2 – Скрипт преследования***

## Тестирование программного продукта

После написания программного кода в результате были реализованы основные функции, определенные в предыдущих главах. Этот этап является завершающим этапом создания программного продукта, который позволяет выявить проблемы и в последующем решить их при помощи отладки.

Тестирование программы проводилось с помощью группы тестировщиков и учащихся НИУ «ВШЭ-Пермь», добровольно согласившихся на участие, а также проводилось тестирование по критериям черного и белого ящиков (см. ПРИЛОЖЕНИЕ Б).

В ходе тестирования были выявлены ошибки, связанные с прохождением через элементы окружения из-за неправильного расположения данных элементов в игровом пространстве.

По окончанию разработки системы, было составлено руководство программиста в ПРИЛОЖЕНИИ Е. Листинг программы приведен в ПРИЛОЖЕНИИ Б.

## Выводы по главе

Исходя из спроектированных моделей и графического интерфейса, игровая логика, полученная во второй главе, была реализована в виде программного кода в системе разработки Unity. В данной главе приводится описание механизма взаимодействия пользователя с игровым пространством. Разработанные классы описывают основную часть функционала игры.

# Заключение

По результату проделанной работы, была разработана многопользовательская игра в жанре «RPG». Для этого были исследованы основы работы в программном обеспечении Unity и рассмотрена литература на эту тему. Были изучены особенности жанра «RPG» и выявлены основные элементы предоставленного жанра. Был проведен обзор игр-аналогов для выявления плюсов и минусов, для выдвижения требований к создаваемой игре.

При проектировании были использована библиотека C#, необходимая для, написания кода приложения. Графический интерфейс был спроектирован и разработан благодаря инструментам Unity, с помощью Visual Studio.

Программу возможно усовершенствовать, добавив дополнительное количество уровней, дорисовав текстуры и усложнив игровой процесс. Так же можно добавить поддержку разных языков.

В течении работы над программой были улучшены навыки ООП и познания в сфере разработки игр, а также усовершенствован навык проектирования программного продукта и его анализа.

Положительным итогом проведенной работы является усовершенствование навыков написания программ на объектно-ориентированном языке программирования C# и создания игр под Windows и MacOS.

# Список сокращений и условных обозначений

1. RPG – жанр компьютерных игр с упором ролевую игру с большим количеством диалогов, свободой в выборе путей решения различных задач, проработанным миром и сюжетом.
2. Очки здоровья (hp) – здоровье персонажа.
3. Движок - (graphics engine) – промежуточное программное обеспечение, программный движок, основной задачей которого является визуализация (рендеринг) двухмерной или трёхмерной компьютерной графики.
4. UML – унифицированный язык моделирования (Unified Modeling Language) система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования
5. UI – Интерфеейс пользователя, он же пользовательский интерфейс (user interface) – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы
6. ПО – программное обеспечение.

# Библиографический список

1. Арсаж, Ж. Программирование игр и головоломок/Ж. Арсаж, -М.: Книга по Требованию, 2012.
2. Бочкарев Н. А., Молотов Р. С. Подходы к трансформации объектов виртуальных пространств в среде Unity //Вестник Ульяновского государственного технического университета. – 2016. – №. 3 (75).
3. Никулин Е.А. Компьютерная графика. Модели и алгоритмы. СПб: издательство "Лань". - 708 с. (2017).
4. Официальная документация по Unity3D [Электронный ресурс] //URL:http://unity3d.com/support/documentation
5. Создание проектов и схем моделирования UML [Электронный ресурс] //URL:https://msdn.microsoft.com/ru-ru/library/dd409445.aspx?f=255&MSPPError=-2147217396
6. Торн, Алан. Искусство создания сценариев в Unity : [рус.]. — СПб : ДМК, 2016. — 362 с.
7. Хокинг, Джозеф. Unity — в действии. Мультиплатформенная разработка на C# : [рус.]. — 2. — СПб : Питер, 2016. — 336 с.
8. Blow J. Game development: Harder than you think //Queue. – 2004. – Т. 1. – №. 10.
9. Diveky, Marko; Bielikova, Maria (September 29–October 2, 2009). "Generating Educational Interactive Stories in Computer Role-Playing Games". Learning in the Synergy of Multiple Disciplines: 4th European Conference on Technology Enhanced Learning, Proceedings, Nice, France: Springer.
10. Claypool K., Claypool M. Teaching software engineering through game design //ACM SIGCSE Bulletin. – 2005. – Т. 37. – №. 3.
11. Jeff Craighead, Jenny Burke, Robin Murphy. “Using the Unity Game Engine to Develop SARGE: A Case Study”. Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008). September 2008
12. Hejlsberg A., Wiltamuth S., Golde P. The C# programming language. – 2006.
13. McNaughton, M.; Schaeffer, J.; Szafron, D.; Parker, D.; Redford J. Code Generation for AI Scripting in Computer Role-Playing Games (PDF). American Association for Artificial Intelligence (2004).
14. Stuart, Keith. 3D games enter a new generation, The Observer, London: The Guardian (19 сентября 2010).